Zdzislaw Bubnicki

**Modern Control Theory**

Zdzislaw Bubnicki

# Modern Control Theory

With 104 figures

Springer

**Professor Zdzislaw Bubnicki, PhD**
Wroclaw University of Technology
Institute of Information Science and Engineering
Wyb. Wyspianskiego 27
50-370 Wroclaw
Poland

*zdzislaw.bubnicki@pwr.wroc.pl*

Originally published in Polish by Polish Scientific Publishers PWN, 2002

Library of Congress Control Number: 2005925392

ISBN 10 3-540-23951-0 **Springer Berlin Heidelberg New York**
ISBN 13 978-3-540-23951-2 **Springer Berlin Heidelberg New York**



**Springer is a part of Springer Science+Business Media**
springeronline.com

Printed in Germany



Typesetting: Data conversion by the author.
Final processing by PTP-Berlin Protago-$T_E$X-Production GmbH, Germany
Cover-Design: Medionet AG, Berlin
Printed on acid-free paper 89/3141/Yu – 5 4 3 2 1 0

# Contents

# Preface

The main aim of this book is to present a unified, systematic description of basic and advanced problems, methods and algorithms of the modern control theory considered as a foundation for the design of computer control and management systems. The scope of the book differs considerably from the topics of classical traditional control theory mainly oriented to the needs of automatic control of technical devices and technological processes. Taking into account a variety of new applications, the book presents a compact and uniform description containing traditional analysis and optimization problems for control systems as well as control problems with non-probabilistic models of uncertainty, problems of learning, intelligent, knowledge-based and operation systems – important for applications in the control of manufacturing processes, in the project management and in the control of computer systems. Into the uniform framework of the book, original ideas and results based on the author's works concerning uncertain and intelligent knowledge-based control systems, applications of uncertain variables and the control of complexes of operations have been included. The material presented in the book is self-contained. Using the text does not require any earlier knowledge on the control science. The presentation requires only a basic knowledge of linear algebra, differential equations and probability theory. I hope that the book can be useful for students, researches and all readers working in the field of control and information science and engineering.

I wish to express my gratitude to Dr. D. Orski and Dr. L. Siwek, my coworkers at the Institute of Information Science and Engineering of Wroclaw University of Technology, who assisted in the preparation of the manuscript.

Z. Bubnicki

# 1 General Characteristic of Control Systems

## 1.1 Subject and Scope of Control Theory

The modern control theory is a discipline dealing with formal foundations of the **analysis and design of computer control and management systems**. Its basic scope contains problems and methods of *control algorithms* design, where the control algorithms are understood as formal prescriptions (formulas, procedures, programs) for the determination of control decisions, which may be executed by technical devices able to the information processing and decision making. The problems and methods of the control theory are common for different executors of the control algorithms. Nowadays, they are most often computer devices and systems. The computer control and management systems or wider – *decision support systems* belong now to the most important, numerous and intensively developing computer information systems. The control theory deals with the foundations, methods and decision making algorithms needed for developing computer programs in such systems.

The problems and methods of the control theory are common not only for different executors of the control algorithms but also – which is perhaps more important – for various applications. In the first period, the control theory has been developing mainly for the *automatic control* of technical processes and devices. This area of applications is of course still important and developing, and the development of the information technology has created new possibilities and – on the other hand – new problems. The full automatization of the control contains also the automatization of manipulation operations, the control of executing mechanisms, intelligent tools and robots which may be objects of the external control and should contain inner controlling devices and systems.

Taking into account the needs connected with the control of various technical processes, with the management of projects and complex plants as well as with the control and management of computer systems has led to forming foundations of modern control science dealing in a uniform and

systematic way with problems concerning the different applications mentioned here. The scope of this area significantly exceeds the framework of so called *traditional (or classical) control theory*. The needs and applications mentioned above determine also new directions and perspectives of the future development of the modern control theory.

Summarizing the above remarks one can say that the control theory (or wider – *control science*) is a basic discipline for the automatic control and robotics and one of basic disciplines for the information technology and management. It provides the methods necessary to a rational design and effective use of computer tools in the decision support systems and in particular, in the largest class of such systems, namely in control and management systems.

Additional remarks concerning the subject and the scope of the control theory will be presented in Sect. 1.2 after the description of basic terms, and in Sect. 1.5 characterizing interconnections between the control theory and other related areas.

## 1.2 Basic Terms

To characterize more precisely the term *control* let us consider the following examples:

1. Control (steering) of a vehicle movement so as to keep a required trajectory and velocity of the motion.
2. Control of an electrical furnace (the temperature control), consisting in changing the voltage put at the heater so as to stabilize the temperature at the required level in spite of the external temperature variations.
3. Stabilization of the temperature in a human body as a result of the action of inner steering organs.
4. Control of the medicine dosage in a given therapy in order to reach and keep required biomedical indexes.
5. Control of a production process (e.g. a process of material processing in a chemical reactor), consisting in proper changes of a raw material parameters with the purpose of achieving required product parameters.
6. Control of a complex manufacturing process (e.g. an assembly process) in such a way that the suitable operations are executed in a proper time.
7. Control (steering, management) of a complex production plant or an enterprise, consisting in making and executing proper decisions concerning the production size, sales, resource distributions, investments etc., with the purpose of achieving desirable economic effects.

8. Admission and congestion control in computer networks in order to keep good performance indexes concerning the service quality.

Generalizing these examples we can say that the *control* is defined as a goal-oriented action. With this action there is associated a certain *object* which is acted upon and a certain *subject* executing the action. In the further considerations the object will be called a *control plant* (CP) and the subject – a controller (C) or more precisely speaking, an executor of the control algorithm. Sometimes for the controller we use the term a *controlling system* to indicate its complexity. The interconnection of these two basic parts (the control plant and the controller) defines a *control system*. The way of interconnecting the basic parts and eventually some additional blocks determines the structure of the control system. Figure 1.1 illustrates the simplest structure of the control system in which the controller C *controls* the plant CP.

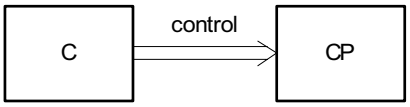


**Fig. 1.1.** Basic scheme of control system

**Remark 1.1.** Regardless different names (control, steering, management), the main idea of the control consists in decision making based on certain information, and the decisions are concerned with a certain plant. Usually, speaking about the control, we do not have in mind single one-stage decisions but a certain multistage decision process distributed in time. However, it is not an essential feature of the control and it is often difficult to state in the case when separate independent decisions are made in successive cycles with different data. □

**Remark 1.2.** The control plant and the controller are imprecise terms in this sense that the control plant does not have to mean a determined object or device. For example, the control of a material flow in an enterprise does not mean the control of the enterprise as a determined plant. On the other hand, the controller should be understood as an executor of the control algorithm, regardless its practical nature which does not have to have a technical character; in particular, it may be a human operator. □

Now we shall characterize more precisely the basic parts of the control system.

### 1.2.1 Control Plant

An object of the control (a process, a system, or a device) is called a *control plant* and treated uniformly regardless its nature and the degree of complexity. In the further considerations in this chapter we shall use the temperature control in an electrical furnace as a simple example to explain the basic ideas, having in mind that the control plants may be much more complicated and may be of various practical nature, not only technical. For example they may be different kinds of economical processes in the case of the management. In order to present a formal description we introduce variables characterizing the plant: controlled variables, controlling variables and disturbances.

By *controlled variables* we define the variables used for the determination of the control goal. In the case of the furnace it is a temperature in the furnace for which a required value is given; in the case of a production process it may be e.g. the productivity or a profit in a determined time interval. Usually, the controlled variables may be measured (or observed), and more generally – the information on their current values may be obtained by processing other information available. In the further considerations we shall use the word "to measure" just in such a generalized sense for the variables which are not directly measured. In complex plants a set of controlled variables may occur. They will be ordered and treated as components of a vector. For example, a turbogenerator in an electrical power station may have two controlled variables: the value and the frequency of the output voltage. In a certain production process, variables characterizing the product may be controlled variables.

By *controlling variables* (or control variables) we understand the variables which can be changed or put from outside and which have impact on the controlled variables. Their values are the control decisions; the control is performed by the proper choosing and changing of these values. In the furnace it is the voltage put at the electrical heater, in the turbogenerator – a turbine velocity and the current in the rotor, in the production process – the size and parameters of a raw material.

*Disturbances* are defined as the variables which except the controlling variables have impact on the controlled variables and characterize an influence of the environment on the plant. The disturbances are divided into measurable and unmeasurable where the term *measurable* means that they are measured during the control and their current values are used for the control decision making. For the furnace, it is e.g. the environment temperature, for the turbogenerator – the load, for the production process – other parameters characterizing the raw material quality, except the variables chosen as control variables.

We shall apply the following notations (Fig. 1.2)

$$u = \begin{bmatrix} u^{(1)} \\ u^{(2)} \\ \vdots \\ u^{(p)} \end{bmatrix}, \qquad y = \begin{bmatrix} y^{(1)} \\ y^{(2)} \\ \vdots \\ y^{(l)} \end{bmatrix}, \qquad z = \begin{bmatrix} z^{(1)} \\ z^{(2)} \\ \vdots \\ z^{(r)} \end{bmatrix}$$

where $u^{(i)}$ – the $i$-th controlling variable, $i = 1, 2, ..., p$; $y^{(j)}$ – the $j$-th controlled variable, $j = 1, 2, ..., l$; $z^{(m)}$ – the $m$-th disturbance, $m = 1, 2, ..., r$; $u$, $y$, $z$ denote the controlling vector (or control vector), the controlled vector and the vector of the disturbances, respectively. The vectors are written as one-column matrices.

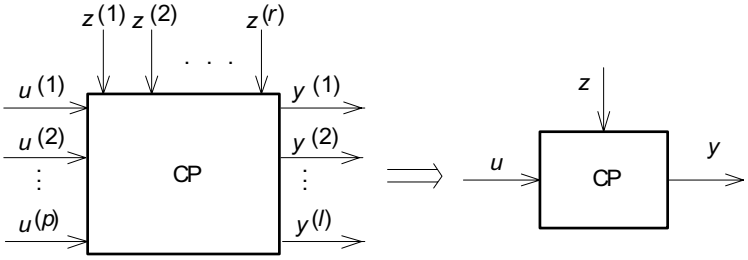


**Fig. 1.2.** Control plant

Generally, in an element (block, part) of the system we may distinguish between the input and the output variables, named shortly the *input* and the *output*. The inputs determine causes of an inner state of the plant while the outputs characterize effects of these causes (and consequently, of this state) which may be observed. In other words, there is a dependence of the output variables upon the input variables which is the "cause-effect" relation. In the control plant the controlled variables form the output and the input consists of the controlling variables and the disturbances. If the disturbances do not occur, we have the plant with the input $u$ and the output $y$. A formal description of the relationship between the variables characterizing the plant (i.e. of the "cause-effect" relation) is called a *plant model*. In simple cases it may be the function $y = \Phi(u, z)$. In more complicated cases it may be e.g. a differential equation containing functions $u(t)$, $z(t)$ and $y(t)$ describing time-varying variables. The determination of the plant model on the basis of experimental investigations is called a *plant identification*.

### 1.2.2 Controller

An executor of the control algorithm is called a *controller* C (controlling system, controlling device) and treated uniformly regardless its nature and the degree of complexity. It may be e.g. a human operator, a specialized so called *analog* device (e.g. analog electronic controller), a controlling computer, a complex controlling system consisting of cooperating computers, analog devices and human operators. The output vector of the controller is the control vector $u$ and the components of the input vector are variables whose values are introduced into C as data used to finding the control decisions. They may be values taken from the plant, i.e. $u$ and (or) $z$, or values characterizing the external information. A *control algorithm*, i.e. the dependence of $u$ upon $w$ or a way of determining control decisions based on the input data, corresponds to the model of the plant, i.e. the dependence of $y$ upon $u$ and $z$. In simple cases it is a function $u = \Psi(w)$, in more complicated cases – the relationship between the functions describing time-varying variables $w$ and $u$. Formal descriptions of the control algorithm and the plant model may be the same. However, there are essential differences concerning the interpretation of the description and its obtaining. In the case of the plant, it is a formal description of an existing real unit, which may be obtained on the basis of observation. In the case of the controller, it is a prescription of an action, which is determined by a designer and then is executed by a determined subject of this action, e.g. by the controlling computer.

In the case of a full automatization possible for the control of technical processes and devices, the controlling system, except the executor of the control algorithm as a basic part, contains additional devices necessary for the acquisition and introducing the information, and for the execution of the decisions. In the case of a computer realization, they are additional devices linking the computer and the control plant (a specific interface in the computer control system). Technical problems connected with the design and exploitation of a computer control system exceed the framework of this book and belong to *control system engineering* and *information technology*. It is worth, however, noting now that the computer control systems are *real-time systems* which means that introducing current data, finding the control decisions and bringing them out for execution should be performed in determined time intervals and if they are short (which occurs as a rule in the cases of a control of technical plants and processes, and in operating management), then the information processing and finding the current decisions should be respectively quick.

Ending the characteristic of the plant and the controller, let us add two additional remarks concerning a determined level of generalization occur-

ring here and the role of the control theory and engineering:
1. The control theory and engineering deal with methods and techniques common for the control of real plants with various practical nature. From the methodology of control algorithms determination point of view, the plants having different real nature but described by the same mathematical models are identical. To a certain degree, such a universalization concerns the executors of control algorithms as well (e.g. universal control computer). That is why, illustrating graphically the control systems, we present only blocks denoting parts or elements of the system, i.e. so called *block-schemes* as a universal illustration of real systems.
2. The basic practical effects or "utility products" of the control theory are control algorithms which are used as a basis for developing and implementing the corresponding computer programs or (nowadays, to a limited degree) for building specialized controlling devices. Methods of the control theory enable a rational control algorithmization based on a plant model and precisely formulated requirements, unlike a control based on an undetermined experience and intuition of a human operator, which may give much worse effects. The algorithmization is necessary for the automatization (the computerization) of the control but in simple cases the control algorithm may be "hand-executed" by a human operator. For that reason, from the control theory and methodology point of view, the difference between an *algorithmized* control and a control based on an imprecise experience is much more essential than the difference between automatic and hand-executed control. The function of the control computer consists in the determination of control decisions which may be executed directly by a technical device and (or) by a human operator, or may be given for the execution by a manager. Usually, in the second case the final decision is made by a manager (generally, by a decision maker) and the computer system serves as an expert system supporting the control process.

## 1.3 Classification of Control Systems

In this section we shall use the term *classification*, although in fact it will be the presentation of typical cases, not containing all possible situations.

### 1.3.1 Classification with Respect to Connection Between Plant and Controller

Taking into account a kind of the information put at the controller input and consequently, a connection between the plant and the controller – one

can consider the following cases:
1. Open-loop system without the measurement of disturbances.
2. Open-loop system with the measurement of disturbances.
3. Closed-loop system.
4. Mixed (combined) system.

These concepts, illustrated in Figs. 1.3 and 1.4, differ from each other with the kind of information (if any) introduced into the executor of the control algorithm and used to the determination of control decisions.

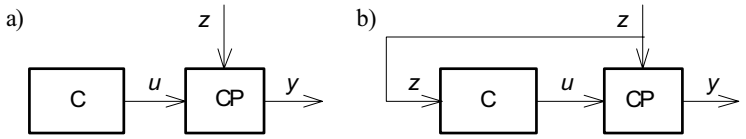


**Fig. 1.3.** Block schemes of open-loop control system: a) without measurement of disturbances, b) with measurement of disturbances

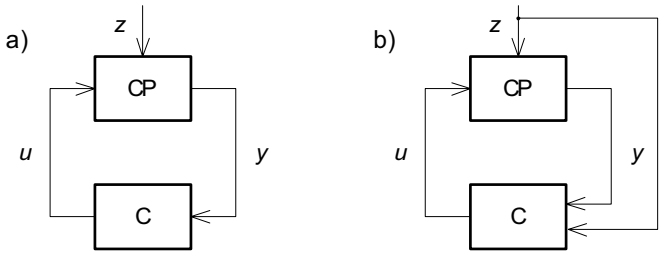


**Fig. 1.4.** Block schemes of control systems: a) closed-loop, b) mixed

The open-loop system without the measurement of disturbances has rather theoretical importance and in practice it can be applied with a very good knowledge of the plant and a lack of disturbances. In the case of the furnace mentioned in the previous sections, the open-loop system with the measurement of disturbances means the control based on the temperature measured outside the furnace, and the closed-loop system – the control based on the temperature measured inside the furnace. Generally, in system **2** the decisions are based on observations of other causes which except the control $u$ may have an impact on the effect $y$. In system **3** called also as a system with a *feed-back* – the current decisions are based on the observations of the effects of former decisions. These are two general and basic concepts of decision making, and more generally – concepts of a goal-oriented activity. Let us note that the closed-loop control systems are systems with so called *negative feed-back* which has a stabilizing character. It

means that e.g. increasing of the value $y$ will cause a change of $u$ resulting in decreasing of the value $y$. Additionally let us note that the variables occurring in a control system have a character of *signals*, i.e. variables containing and transferring information. Consequently, we can say that in the feed-back system a closed loop of the information transferring occurs.

Comparing systems **2** and **3** we can generally say that in system **2** a much more precise knowledge of the plant, i.e. of its reaction to the actions $u$ and $z$, is required. In system **3** the additional information on the plant is obtained via the observations of the control effects. Furthermore, in system **2** the control compensates the influence of the measured disturbances only, while in system **3** the influence on the observed effect $y$ of all disturbances (not only not measured but also not determined) is compensated. However, not only the advantages but also the disadvantages of the concept **3** comparing with the concept **2** should be taken into account: counteracting the changes of $z$ may be much slower than in system **2** and, if the reactions on the difference between a real and a required value $y$ are too intensive, the value of $y$ may not converge to a steady state, which means that the control system does not behave in a stabilizing way. In the example of the furnace, after a step change of the outside temperature (in practice, after a very quick change of this temperature), the control will begin with a delay, only when the effect of this change is measured by the thermometer inside the furnace. Too great and quick changes of the voltage put on the heater, depending on the difference between the current temperature inside the furnace and the required value of this temperature, may cause oscillations of this difference with an increasing amplitude. The advantages of system **2** and **3** are combined into a properly designed mixed system which in the example with the furnace requires two thermometers – inside and outside the furnace.

### 1.3.2 Classification with Respect to Control Goal

Depending on the control goal formulation, two typical cases are usually considered:

1. Control system with the required output.
2. Extremal control system.

We use the identical terms directly for the control, speaking about the control for the required output and the extremal control. In the first case the required value of $y$ is given, e.g. the required value of the inside temperature in the example with the furnace. The aim of the control is to bring $y$ to the required value and to keep the output possibly near to this value in the presence of varying disturbances. More generally – the function de-

scribing the required time variation of the output may be given. For a multi-output plant the required values or functions of time for individual outputs are given.

The second case concerns a single-output plant for which the aim of the control is to bring the output to its extremal value (i.e. to the least or the greatest from the possible values, depending on a practical sense) and to keep the output possibly near to this value in the presence of varying disturbances. For example, it can be the control of a production process for the purpose of the productivity or the profit maximization, or of the minimization of the cost under some additional requirements concerning the quality. It will be shown in Chap. 4 that the optimal control with the given output is reduced to the extremal control where a performance index evaluating the distance between the vector $y$ and the required output vector is considered as the output of the extremal control plant.

A combination of the case **1** with the case **3** from Sect. 1.3.1 forms a typical and frequently used control system, namely a closed-loop control system with the required output. Such a control is sometimes called a *regulation*. Figure 1.5 presents the simplest block scheme of the closed-loop system with the required output of the plant, containing two basic parts: the control plant CP and the controller C. The small circle symbolizes the comparison of the controlled variable $y$ with its required value $y^*$. It is an example of so called *summing junction* whose output is the algebraic sum of the inputs. The variable $\varepsilon(t) = y^* - y(t)$ is called a *control error*. The controller changes the plant input depending on the control error in order to decrease the value of $\varepsilon$ and keep it near to zero in the presence of disturbances acting on the plant. For the full automatization of the control it is necessary to apply some additional devices such as a measurement element and an executing organ changing the plant input according to the signals obtained from the controller.

In the example with the furnace, the automatic control may be as follows: the temperature $y$ is measured by an electrical thermometer, the voltage proportional to $y$ is compared with the voltage proportional to $y^*$ and the difference proportional to the control error steers an electrical motor, changing, by means of a transmission, a position of a supplying device and consequently changing the voltage put on the heater. As an effect, the speed of $u(t)$ variations is approximately proportional to the control error, so the approximate control algorithm is the following:

$$u(t) = k \int_0^t \varepsilon(t)dt.$$

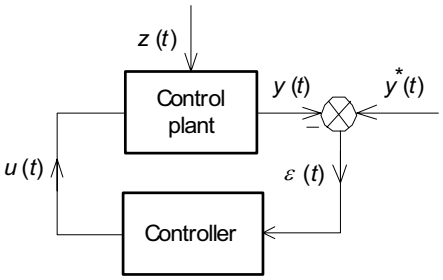


**Fig. 1.5.** Basic scheme of closed-loop control system

Depending on $y^*$, we divide the control systems into three kinds:
1. Stabilization systems.
2. Program control systems.
3. Tracking systems.

In the first case $y^* = \text{const.}$, in the second case the required value changes in time but the function $y^*(t)$ is known at the design stage, before starting the control. For example, it can be a desirable program of the temperature changes in the example with the furnace. In the third case the value of $y^*(t)$ can be known by measuring only at the moment when it occurs during the control process. For example, $y^*(t)$ may denote the position of a moving target tracked by $y(t)$.

### 1.3.3 Other Cases

Let us mention other divisions or typical cases of control systems:
1. Continuous and discrete control systems.
2. One-dimensional and multi-dimensional systems.
3. Simple and complex control systems.

**Ad 1.** In a *continuous system* the inputs of the plant can change at any time and, similarly, the observed variables can be measured at any time. Then in the system description we use the functions of time $u(t), y(t)$, etc. In a *discrete system* (or more precisely speaking – discrete in time), the changes of control decisions and observations may be carried out at certain moments $t_n$. The moments $t_n$ are usually equally spaced in time, i.e. $t_{n+1} - t_n = T = \text{const}$ where $T$ denotes a period or a length of an interval (a stage) of the control. Thus the control operations and observations are executed in determined periods or stages. In the system description we use so

called *discrete functions of time*, that is sequences $u_n, y_n$ etc. where $n$ denotes the index of a successive period. The computer control systems are of course discrete systems, i.e. the results of observations are introduced and control decisions are brought out for the execution at determined moments. If $T$ is relatively small, then the control may be approximately considered as a continuous one. The continuous control or the discrete control with a small period is possible and sensible for quickly varying processes and disturbances (in particular, in technical plants), but is impossible in the case of a project management or a control of production and economic processes where the control decisions may be made and executed e.g. once a day for an operational management or once a year for a strategic management. A continuous control algorithm determining a dependence of $u(t)$ upon $w(t)$ can be presented in a discrete form suitable for the computer implementation as a result of so called *discretization*.

**Ad 2.** In this book we generally consider multi-dimensional systems, i.e. $u$, $y$ etc. are vectors. In particular if they are scalars, that is the number of their components is equal to 1 – the system is called one-dimensional. Usually, the multi-dimensional systems in the sense defined above are called *multivariable* systems. Sometimes the term multi-dimensional is used for systems with variables depending not only on time but also e.g. on a position [76, 77]. The considerations concerning such systems exceed the framework of this book.

**Ad 3.** We speak about a *complex system* if there occurs more than one plant model and (or) more than one control algorithm. Evidently, it is not a precise definition and a system may be considered as a complex one as the result of a certain approach or a point of view. The determination of submodels of a complicated model describing one real plant and consequently – the determination of partial control algorithms corresponding to the submodels may be the result of a decomposition of a difficult problem into simpler partial problems. The complex control algorithms as an interconnection of the partial algorithms can be executed by one control computer. On the other hand – the decomposition may have a "natural" character if the real complex plant can be considered as a system composed of separate but interconnected partial plants for which separate local control computers and a coordinating computer at the higher control level are designed. Complex system problems take an important role in the analysis and design of control and management systems for complex plants, processes and projects. It is important to note that a complex computer system can be considered as such a plant.

## 1.4 Stages of Control System Design

Quite generally and roughly speaking we can list the following stages in designing of a computer control system:
1. System analysis of the control plant.
2. Plant identification.
3. Elaborating of the control algorithm.
4. Elaborating of the controlling program.
5. Designing of a system executing the controlling program.

The *system analysis* contains an initial determination of the control goal and possibly subgoals for a complex plant, a choice of the variables characterizing the plant, presented in Sect. 1.2, and in the case of a complex plant – a determination of the components (subplants) and their interconnections.

The *plant identification* [14] means an elaboration of the mathematical model of the plant by using the results of observations. It should be a model useful for the determination of the control algorithm so as to achieve the control goal. If it is not possible to obtain a sufficiently accurate model, the problem of decision making under uncertainty arises. Usually, the initial control goal should then be reformulated, that is requirements should be weaker so that they are possible to satisfy with the available knowledge on the plant and (or) on the way of the control.

The elaboration of the control algorithm is a basic task in the whole design process. The control algorithm should be adequate to the control goal and to the precisely described information on the plant, and determined with the application of suitable rational methods, that is methods which are described, investigated and developed in the framework of the control theory. The control algorithm is a basis for the elaboration of the controlling computer program and the design of computer system executing this program. In practice, the individual stages listed above are interconnected in such a sense that the realization of a determined stage requires an initial characterization of the next stages and after the realization of a determined stage a correction of the former stages may be necessary.

Not only a *control* in real-time but also a *design* of a control system can be computer supported by using special software systems called CAD (Computer Aided Design).

## 1.5 Relations Between Control Science and Related Areas in Science and Technology

After a preliminary characteristic of control problems in Sects. 1.2, 1.3 and 1.4 one can complete the remarks presented in Sect. 1.1 and present shortly relations of control theory with information science and technology, automatic control, management, knowledge engineering and systems engineering:

**1.** The control theory and engineering may be considered as part of the *information science and technology*, dealing with foundations of computer decision systems design, in particular – with elaboration of decision making algorithms which may be presented in the form of computer programs and implemented in computer systems. It may be said that in fact the control theory is a decision theory with special emphasis on real-time decision making connected with a certain plant which is a part of an information control system.

**2.** Because of universal applications regardless of a practical nature of control plants, the control theory is a part of *automatic control* and management considered as scientific disciplines and practical areas. In different practical situations there exists a great variety of specific techniques connected with the information acquisition and the execution of decisions. Nevertheless, there are common foundations of computer control systems and decision support systems for management [20] and often the terms *control*, *management* and *steering* are used with similar meaning.

**3.** The control theory may be also considered as a part of the computer science and technology because of applications for computer systems, since it deals with methods and algorithms for the control (or management) of computer systems, e.g. the control of a load distribution in a multicomputer system, the admission, congestion and traffic control in computer networks, steering a complex computational process by a computer operating system, the data base management etc. Thus we can speak about a double function of the control theory in the general information science and technology, corresponding to a double role of a computer: a computer as a tool for executing the control decisions and as a subject of such decisions.

**4.** The control theory is strongly connected with a *knowledge engineering* which deals with knowledge-based problem solving with the application of reasoning, and with related problems such as the knowledge acquisition, storing and discovering. So called *intelligent* control systems are specific

expert systems [18, 92] in which the generating of control decisions is based on a knowledge representation describing the control plant, or based directly on a knowledge about the control. For the design and realization of the control systems like these, such methods and techniques of the *artificial intelligence* as the computerization of logical operations, learning algorithms, pattern recognition, problem solving based on *fuzzy* descriptions of the knowledge and the computerization of neuron-like algorithms are applied.

**5.** The control theory is a part of a general *systems theory and engineering* which deals with methods and techniques of modelling, identification, analysis, design and control – common for various real systems, and with the application of computers for the execution of the operations listed above.

This repeated role of the control theory and engineering in the areas mentioned here rather than following from its universal character is a consequence of interconnections between these areas so that distinguishing between them is not possible and, after all, not useful. In particular, it concerns the automatic control and the information science and technology which nowadays may be treated as interconnected parts of one discipline developing on the basis of two fundamental areas: knowledge engineering and systems engineering.

## 1.6 Character, Scope and Composition of the Book

The control theory may be presented in a very formal manner, typical for so called *mathematical control theory*, or may be rather oriented to practical applications as a uniform description of problems and methods useful for control systems design. The character of this book is nearer to the latter approach. The book presents a unified, systematic description of control problems and algorithms, ordered with respect to different cases concerning the formulations and solutions of decision making (control) problems. The book consists of five informal parts organized as follows.

**Part one** containing Chaps. 1 and 2 serves as an introduction and presents general characteristic of control problems and basic formal descriptions used in the analysis and design of control systems.

**Part two** comprises three chapters (Chaps. 3, 4 and 5) on basic control problems and algorithms without uncertainty, i.e. based on complete information on the deterministic plants.

In **Part three** containing Chaps. 6, 7, 8 and 9 we present different cases

concerning problem formulations and control algorithm determinations under uncertainty, without obtaining any additional information on the plant during the control.

**Part four** containing Chaps. 10 and 11 presents two different concepts of using the information obtained in the closed-loop system: to the direct determination of control decisions and to improving of the basic decision algorithm in the adaptation and learning process.

Finally, **Part five** (Chaps. 12 and 13) is devoted to additional problems of considerable importance, concerning so called *intelligent* and *complex control systems*.

The scope and character of the book takes into account modern role and topics of the control theory described preliminarily in Chap. 1, namely the computer realization of the control algorithms and the application to the control of production and manufacturing processes, to management, and to control of computer systems. Consequently, the scope differs considerably from the topics of classical, traditional control theory mainly oriented to the needs of the automatic control of technical devices and processes. Taking into consideration a great development of the control and decision theory during last two decades on one hand, and – on the other hand – the practical needs mentioned above, has required a proper selection in this very large area. The main purpose of the book is to present a compact, unified and systematic description of traditional analysis and optimization problems for control systems as well as control problems with non-probabilistic description of uncertainty, problems of learning, intelligent, knowledge-based and operation systems – important for applications in the control of production processes, in the project management and in the control of computer systems. Such uniform framework of the modern control theory may be completed by more advanced problems and details presented in the literature. The References contain selected books devoted to control theory and related problems [1, 2, 3, 6, 60, 64, 66, 68, 69, 71, 72, 73, 78, 79, 80, 83, 84, 88, 90, 91, 94, 98, 104], books concerning the control engineering [5, 63, 65, 67, 85, 93] and papers of more special character, cited in the text. Into the uniform framework of the book, original ideas and results based on the author's works concerning uncertain and intelligent knowledge-based control systems and control of the complexes of operations have been included.

# 2 Formal Models of Control Systems

To formulate and solve control problems common for different real systems we use formal descriptions usually called *mathematical models*. Sometimes it is necessary to consider a difference between an exact mathematical description of a real system and its approximate mathematical model. In this chapter we shall present shortly basic descriptions of a variable (or signal), a control plant, a control algorithm (or a controller) and a whole control system. The descriptions of the plant presented in Sects. 2.2–2.4 may be applied to any systems (blocks, elements) with determined inputs and outputs.

## 2.1 Description of a Signal

As it has been already said, the variables in a control system (controlling variable, controlled variable etc.) contain and present some information and that is why they are often called *signals*. In general, we consider multidimensional or multivariable signals, i.e. vectors presented in the form of one-column matrices. A continuous signal

$$x(t) = \begin{bmatrix} x^{(1)}(t) \\ x^{(2)}(t) \\ \vdots \\ x^{(k)}(t) \end{bmatrix}$$

is described by functions of time $x^{(i)}(t)$ for individual components. In particular $x(t)$ for $k=1$ is a one-dimensional signal or a scalar. The term *continuous signal* does not have to mean that $x^{(i)}(t)$ are continuous functions of time, but means that the values $x^{(i)}(t)$ are determined and may change at any moment $t$. The variables $x$ are elements of the vector space $X = R^k$, that is the space of vectors with $k$ real components. If the signal is a subject of a linear transformation, it is convenient to use its *operational transform*

(or Laplace transform) $X(s) \hat{=} x(t)$, i.e. the function of a complex variable $s$, which is a result of Laplace transformation of the function $x(t)$:

$$X(s) = \int_0^\infty x(t) e^{-st} dt.$$

Of course, the function $X(s)$ is a vector as well, and its components are the operational transforms of the respective components of the vector $x$.

In discrete (more precisely speaking – discrete in time) control systems a discrete signal $x_n$ occurs. This is a sequence of the values of $x$ at successive moments (periods, intervals, stages) $n = 0, 1, \ldots$ . The discrete signal may be obtained by *sampling of the continuous signal* $x(t)$. Then $x_n = x(nT)$ where $T$ is a sampling period. If $x_n$ subjects to a linear transformation, it is convenient to use a *discrete operational transform* or $Z$-transform $X(z) \hat{=} x_n$, i.e. the function of a complex variable $z$, which is a result of so called $Z$ transformation of the function $x_n$ :

$$X(z) = \sum_{n=0}^\infty x_n z^{-n}.$$

Basic information on the operational transforms are presented in the Appendix.

## 2.2 Static Plant

A static model of the plant with $p$ inputs and $l$ outputs is a function

$$y = \Phi(u) \tag{2.1}$$

presenting the relationship between the output $y \in Y = R^l$ and the input $u \in U = R^p$ in a steady state. If the value $u$ is put at the input (generally speaking, the decision $u$ is performed) then $y$ denotes the value of the output (the response) after a transit state. In other words, $y$ depends directly on $u$ and does not depend on the history (on the previous inputs). In the example with the electrical furnace considered in Chap. 1, the function $\Phi$ may denote a relationship between the temperature $y$ and the voltage $u$ where $y$ denotes the steady temperature measured in a sufficiently long time after the moment of switching on the constant voltage $u$. Thus the function $\Phi$ describes the steady-state behaviour of the plant. Quite often $\Phi$ denotes the dependency of an effect upon a cause which has given this result, observed

in a sufficiently long time. For example, it may be a relationship between the amount and parameters of a product obtained at the end of a production cycle and the amount or parameters of a raw material fixed at the beginning of the cycle. We used to speak about an *inertia-less* or *memory-less* plant if the steady value of the output as a response for the step input settles very quickly compared to other time intervals considered in the plant. The function $\Phi$ is sometimes called a *static characteristic* of the plant.

Usually, the mathematical model $\Phi$ is a result of a simplification and approximation of a reality. If the accuracy of this approximation is sufficiently high, we may say that this is a description of the real plant, which means that the value $y$ measured at the output after putting the value $u$ at the input is equal to the value $y$ calculated from the mathematical model after substituting the same value $u$ into $\Phi$. Then we can speak about a mathematical model $\bar{y} = \bar{\Phi}(u)$ differing from the exact description $\Phi$. Such a distinction has an essential role in an identification problem. Usually, instead of saying a *plant described by a model* $\Phi$, we say shortly a *plant* $\Phi$, that is a distinction *plant – model* is replaced by a distinction *real plant – plant*. In particular, the term *static model of a real plant* is replaced by *static plant*. Similar remarks concern dynamical plants, other blocks in a system and a system as a whole.

For the linear plant the relationship (2.1) takes the form

$$y = Au + b$$

where $A \in R^{l \times p}$, i.e. $A$ is a matrix with $l$ rows and $p$ columns or is $l \times p$ matrix; $b$ is one-column matrix $l \times 1$. Changing the variables

$$\bar{y} = y - b$$

we obtain the relationship without a free term. As a rule, the variables in a control system denote increments of real variables from a fixed reference point. The location of the origin in this point means that $\bar{\Phi}(\bar{0}) = \bar{0}$ where $\bar{0}$ denotes the vector with zero components. The model (2.1) can be presented as a set of separate relationships for the individual output variables:

$$y^{(j)} = \Phi_j(u), \qquad j = 1, 2, ..., l.$$

## 2.3 Continuous Dynamical Plant

*Continuous plant* is the term we will use for plants controlled in a time-

continuous manner, that is systems where the control variables can change at any time and, similarly, the observed variables can be measured at any time. Thus a dynamic model will involve relations between the time functions describing changes of plant variables. These relationships will most often take the form of differential equations for the plants controlled continuously, or difference equations for the plants controlled discretely. Other forms of relations between the time functions characterizing a control plant may also occur.

There are three basic kinds of descriptions of the properties of a dynamic system with an input and an output (control plant in our case):
1. State vector description.
2. "Input-output" description by means of a differential or difference equation.
3. Operational form of the "input-output" description.
The last two kinds of description represent, in different ways, direct relations between the plant input and output signals.

### 2.3.1 State Vector Description

To represent relations between time-varying plant variables, we select a sufficient set of variables $x^{(1)}(t)$, $x^{(2)}(t)$, ..., $x^{(k)}(t)$ and set up a mathematical model in the form of a system of first order differential equations:

$$\left.\begin{aligned}
\dot{x}^{(1)} &= f_1(x^{(1)}, x^{(2)}, ..., x^{(k)}; u^{(1)}, u^{(2)}, ..., u^{(p)}),\\
\dot{x}^{(2)} &= f_2(x^{(1)}, x^{(2)}, ..., x^{(k)}; u^{(1)}, u^{(2)}, ..., u^{(p)}),\\
&\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\\
\dot{x}^{(k)} &= f_k(x^{(1)}, x^{(2)}, ..., x^{(k)}; u^{(1)}, u^{(2)}, ..., u^{(p)}).
\end{aligned}\right\} \tag{2.2}$$

The variables $u^{(1)}$, $u^{(2)}$, ..., $u^{(p)}$ denote input signals (control signals, in particular). Thus we consider a multi-input plant with $p$ inputs. If we are interested in the plant output variables, then the relations between the output signals $y^{(1)}$, $y^{(2)}$, ..., $y^{(l)}$ ($l$-output plant), $x^{(1)}$, $x^{(2)}$, ..., $x^{(k)}$ and $u^{(1)}$, $u^{(2)}$, ..., $u^{(p)}$, have also to be determined:

$$\left.\begin{aligned}
y^{(1)} &= \eta_1(x^{(1)}, x^{(2)}, ..., x^{(k)}; u^{(1)}, u^{(2)}, ..., u^{(p)}),\\
y^{(2)} &= \eta_2(x^{(1)}, x^{(2)}, ..., x^{(k)}; u^{(1)}, u^{(2)}, ..., u^{(p)}),\\
&\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\\
y^{(l)} &= \eta_l(x^{(1)}, x^{(2)}, ..., x^{(k)}; u^{(1)}, u^{(2)}, ..., u^{(p)}).
\end{aligned}\right\} \tag{2.3}$$

In practice, because of the inertia inherent in the plant, the signals $u^{(1)}$, $u^{(2)}$, ..., $u^{(p)}$ usually do not appear in the equation (2.3). The equations (2.2) and (2.3) can be written in a briefer form using vector notation (with $u$ already eliminated from the equation (2.3)):

$$\left.\begin{aligned} \dot{x} &= f(x,u), \\ y &= \eta(x) \end{aligned}\right\} \tag{2.4}$$

where

$$x = \begin{bmatrix} x^{(1)} \\ x^{(2)} \\ \vdots \\ x^{(k)} \end{bmatrix}, \qquad u = \begin{bmatrix} u^{(1)} \\ u^{(2)} \\ \vdots \\ u^{(p)} \end{bmatrix}, \qquad y = \begin{bmatrix} y^{(1)} \\ y^{(2)} \\ \vdots \\ y^{(l)} \end{bmatrix}.$$

The sets of functions $f_1, f_2, ..., f_k$ and $\eta_1, \eta_2, ..., \eta_l$ are now represented by $f$ and $\eta$. The function $f$ assigns a $k$-dimensional vector to an ordered pair of $k$- and $p$-dimensional vectors. The function $\eta$ assigns an $l$-dimensional vector to a $k$-dimensional one. If $u(t) = 0$ for $t \geq 0$ (or, in general, $u(t) =$ const), then the first of the equations (2.4) describes a free process

$$\dot{x} = f(x) \tag{2.5}$$

and, for a given initial condition $x_0 = x(0)$, the solution of the equation (2.5) defines the variable $x(t)$

$$x(t) = \Phi(x_0, t).$$

Under the well-known assumptions, knowledge of the function $f$ and of the value $x(t_1)$ uniquely determines $x(t_2)$ for any $t_2 > t_1$:

$$x(t_2) = \Phi[x(t_1), t_1, t_2].$$

The set $x^{(1)}, x^{(2)}, ..., x^{(k)}$ consists of as many mutually independent variables as necessary for a description of the plant dynamics in the form of a system of first-order differential equations (2.2), i.e. knowledge of the values of these variables at any time $t_1$ should be sufficient to determine their values at any subsequent instant. The variables $x^{(1)}, x^{(2)}, ..., x^{(k)}$ are called the *state variables* of the plant, vector $x$ – the *state vector*, the set $X$ of all such vectors ($x \in X$) – the *state space*, and $k$ – the *plant order*. The system of equations (2.4) is just the mathematical model described by means of the state vector.

The choice of state variables for a given plant can be done in infinitely many ways. If $x$ is a state vector of a certain plant, then the $k$-dimensional vector

$$v = g(x) \tag{2.6}$$

where $g$ is a one-to-one mapping, is also a state vector of this plant. The transformation (2.6) may, for example, be linear

$$v = Px$$

where $P$ is a non-singular matrix (i.e. $\det P \neq 0$). Substituting

$$x = g^{-1}(v)$$

into the equations (2.4), we obtain the new equations

$$\left.\begin{aligned} \dot{v} &= \bar{f}(v,u), \\ y &= \bar{\eta}(v). \end{aligned}\right\} \tag{2.7}$$

The descriptions (2.4) and (2.7) are said to be *equivalent*. Thus different choices of the state vector yield equivalent descriptions of the same plant. In particular, if $l = k$ and $\eta$ in the equation (2.4) is a one-to-one mapping, then $y$ is a state vector of the plant. We say then that the plant is *measurable*, which means that knowledge of the output $y$ at a time $t$ uniquely determines the state of the plant at this time. Since we always assume that the output signals $y$ can be measured, it is therefore implied that, in the case of a measurable plant, all the state variables can be measured at any time $t$.

In particular, for a linear plant, under the assumption that $f(\bar{0},\bar{0}) = \bar{0}$ and $\eta(\bar{0}) = \bar{0}$, the description (2.4) becomes

$$\left.\begin{aligned} \dot{x} &= Ax + Bu, \\ y &= Cx \end{aligned}\right\} \tag{2.8}$$

where $A$ is a $k \times k$ matrix, $B$ is a $k \times p$ matrix and $C$ is a $l \times k$ matrix.

In the case of a single-input and single-output plant ($p = l = 1$) we write the equations (2.8) in the form

$$\left.\begin{aligned} \dot{x} &= Ax + bu, \\ y &= c^{\mathrm{T}}x \end{aligned}\right\} \tag{2.9}$$

where $b$ and $c$ are vectors (one-column matrices). The plant with time-varying parameters is called a *non-stationary plant.* Then in the description (2.4) and in related descriptions the variable $t$ occurs:

$$\left.\begin{aligned} \dot{x} &= f(x,u,t), \\ y &= \eta(x,t). \end{aligned}\right\}$$

**Example 2.1.** Let us consider an electromechanical plant consisting of a D.C. electrical motor driving, by means of a transmission, a load containing viscous drag and inertia (Fig. 2.1).

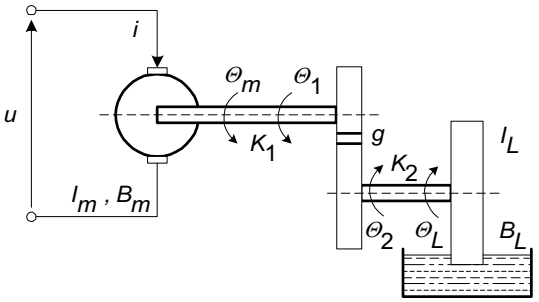


**Fig. 2.1.** Example of electromechanical plant

The dynamic properties of the system can be described by the equations:

$$u = L\,\frac{di}{dt} + r\,i + K_b\frac{d\Theta_m}{dt},$$

$$M = K_b \cdot i,$$

$$M = I_m\frac{d^2\Theta_m}{dt^2} + B_m\frac{d\Theta_m}{dt} + K_1(\Theta_m - \Theta_1),$$

$$\Theta_2 = \frac{1}{g}\,\Theta_1,$$

$$gK_1(\Theta_1 - \Theta_m) = K_2(\Theta_L - \Theta_2),$$

$$I_L\frac{d^2\Theta_L}{dt^2} + B_L\frac{d\Theta_L}{dt} + K_2(\Theta_L - \Theta_2) = 0$$

where $u$ is the supply voltage, $i$ – the current, $\Theta_m$ – the angular position of the rotor, $\Theta_1$ and $\Theta_2$ – the angular position of the gear-wheels, $\Theta_L$ – the angular position of the loading shaft, $M$ – the engine moment, $I_m$ and $I_L$ –

the moments of inertia of the rotor and load, respectively, $B_m$ and $B_L$ – the friction coefficients of the rotor and load; $K_1$ , $K_2$ , $L$, $r$, $K_b$ – the other parameters, $g$ – the transmission ratio.

On introducing five state variables:

$$x^{(1)} = i, \quad x^{(2)} = \Theta_m, \quad x^{(3)} = \dot{\Theta}_m, \quad x^{(4)} = \Theta_L, \quad x^{(5)} = \dot{\Theta}_L$$

the plant equations, after some transformation, can be reduced to the form

$$\dot{x}^{(1)} = -\frac{r}{L}x^{(1)} - \frac{K_b}{L}x^{(3)} + \frac{1}{L}u,$$

$$\dot{x}^{(2)} = x^{(3)},$$

$$\dot{x}^{(3)} = \frac{K_b}{I_m}x^{(1)} + \alpha x^{(2)} - \frac{B_m}{I_m}x^{(3)} + \beta x^{(4)},$$

$$\dot{x}^{(4)} = x^{(5)},$$

$$\dot{x}^{(5)} = \gamma x^{(2)} + \delta x^{(4)} - \frac{B_L}{I_L}x^{(5)}$$

where

$$\alpha = -\frac{K_1K_2}{I_m(g^2K_1 + K_2)}, \qquad \beta = \frac{gK_1K_2}{I_m(g^2K_1 + K_2)},$$

$$\gamma = \frac{gK_1K_2}{I_L(g^2K_1 + K_2)}, \quad \delta = -\frac{g^2K_1K_2}{I_L(g^2K_1 + K_2)}. \qquad \square$$

### 2.3.2 "Input-output" Description by Means of Differential Equation

The relationship between the input vector $u(t)$ and the output vector $y(t)$ can be described by means of a differential equation

$$F_1\left(\frac{d^m y}{dt^m}, \frac{d^{m-1} y}{dt^{m-1}}, ..., \frac{dy}{dt}, y\right) = F_2\left(\frac{d^v u}{dt^v}, \frac{d^{v-1} u}{dt^{v-1}}, ..., \frac{du}{dt}, u\right).$$

For the linear plant this equation becomes

$$\frac{d^m y}{dt^m} + A_{m-1}\frac{d^{m-1} y}{dt^{m-1}} + ... + A_1\frac{dy}{dt} + A_0 y$$

$$= B_v\frac{d^v u}{dt^v} + ... + B_1\frac{du}{dt} + B_0 u \qquad (2.10)$$

where $A_i$ ($i$ = 0, 1, ..., $m$ – 1) are $l \times l$ matrices, $B_j$ ($j$ = 0, 1, ..., $v$) are $l \times p$ matrices.

In particular, for single-input and single-output plant ($p = l$ =1)

$$y^{(m)} + a_{m-1} y^{(m-1)} + ... + a_1 \dot{y} + a_0 y = b_v u^{(v)} + ... + b_1 \dot{u} + b_0 u.$$

In a non-stationary plant at least some of the coefficients $a$ and (or) $b$ are functions of $t$.

### 2.3.3 Operational Form of "Input-output" Description

The relation between the input and the output plant signals can be described by means of an operator $\Phi$ which transforms the function $u(t)$ into the function $y(t)$:

$$y(t) = \Phi[u(t)]. \qquad (2.11)$$

For example, in the case of a one-dimensional linear plant ($p = l = 1$) with zero initial conditions, the formula (2.11) is

$$y(t) = \int_0^t k_i(t,\tau)\, u(\tau)\, d\tau \qquad (2.12)$$

where $k_i(t, \tau)$ is the weighting function (time characteristic) of the plant.

For linear plants with constant parameters, the type of models considered includes description by means of operational transmittance. Applying an operational transformation to the both sides of the equation (2.10), under zero initial conditions, we obtain

$$(Is^m + \sum_{i=0}^{m-1} A_i s^i)Y(s) = (\sum_{j=0}^{v} B_j s^j)U(s) \qquad (2.13)$$

where $I$ is the unit matrix, and $Y(s)$ and $U(s)$ denote Laplace transforms of the vectors $y(t)$ and $u(t)$, respectively. From the equation (2.13) we have

$$Y(s) = K(s)\, U(s)$$

where

$$K(s) = (Is^m + \sum_{i=0}^{m-1} A_i s^i)^{-1} \sum_{j=0}^{v} B_j s^j .$$

The matrix $K(s)$ is called a *matrix operational transmittance* (or matrix *transfer function*) of the plant. Its elements are rational functions of $s$. In the case of one-dimensional plant $K(s)$ is itself such a function, i.e.

$$K(s) = \frac{Y(s)}{U(s)}$$

where $Y(s)$ and $U(s)$ are polynomials. In real systems the degree of the numerator is not greater than the degree of the denominator. This is the condition of so called *physical existence* (or a physical realization) of the transmittance. The transmittance is related to equivalent descriptions of the plant, namely to the gain-phase (or amplitude-phase) characteristics and time characteristics (unit-step response and impulse response).

A *gain-phase characteristic* or a *frequency transmittance* is defined as $K(\mathrm{j}\omega)$ for $0 \le \omega < \infty$. The graphical representation of this function on $K(s)$ plane is sometimes called a gain-phase plot or Nyquist plot. If $u(t) = A \sin\omega t$ then in the steady state the output signal $y(t)$ is sinusoidal as well: $y(t) = B \sin(\omega t + \varphi)$ . It is easy to show that

$$|K(\mathrm{j}\omega)| = \frac{B}{A}, \qquad \arg K(\mathrm{j}\omega) = \varphi.$$

For example, the frequency transmittance $K(\mathrm{j}\omega)$ for

$$K(s) = \frac{k}{(sT_1 + 1)(sT_2 + 1)(sT_3 + 1)}$$

is illustrated in Fig. 2.2.

Let

$$u(t) = \begin{cases} 1 & \text{for } t \ge 0 \\ 0 & \text{for } t < 0. \end{cases}$$

Such a function is called a unit step and is denoted by $\mathbf{1}(t)$. The response of the plant $y(t) \triangleq k(t)$ for the unit step $u(t) = \mathbf{1}(t)$ is called a *unit-step response*. Let $u(t) = \delta(t)$. This is so called *Dirac delta*, i.e. in practice – very short and very high positive impulse in the neighbourhood of $t = 0$, for which

$$\int_{-\infty}^{\infty} \delta(t)\, dt = 1.$$

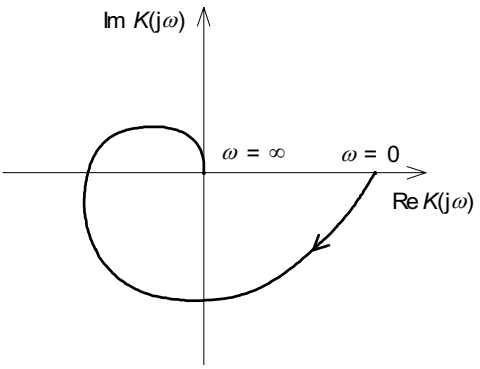


**Fig. 2.2.** Example of frequency transmittance

The response of the plant $y(t) \triangleq k_i(t)$ for the input $u(t) = \delta(t)$ is called an *impulse response*. It is easy to prove that the transmittance $K(s)$ is Laplace transform of the function $k_i(t)$ and $k_i(t) = \dot{k}(t)$. For the linear stationary plant, the relationship (2.12) takes the form

$$y(t) = \int_0^t k_i(t-\tau)\, u(\tau)\, d\tau\,.$$

For example, the plant described by the equation

$$T\,\dot{y}(t) + y(t) = k\,u(t)$$

has the transmittance

$$K(s) = \frac{k}{Ts+1},$$

the unit-step response

$$k(t) = k(1 - e^{-\frac{t}{T}})$$

and the impulse response

$$k_i(t) = \frac{k}{T}\, e^{-\frac{t}{T}}.$$

Such a plant is called a first order inert element (or an element with inertia). It is worth recalling that the descriptions presented here are used not only for plants but in general – for any dynamical elements or blocks with determined inputs and outputs. Basic elements are presented in Table 2.1.

**Table 2.1**

| Name of the element | Transmittance |
|---|---|
| Inertia-less element | $K(s) = k$ |
| First order inert element | $K(s) = \frac{k}{sT + 1}$ |
| Integrating element with first order inertia | $K(s) = \frac{k}{s(sT + 1)}$ |
| Differentiating element with first order inertia | $K(s) = \frac{sk}{sT + 1}$ |
| Oscillation element | $K(s) = \frac{k}{s^2 + 2\alpha s + \beta}$, $\beta > \alpha^2$ |

Complex blocks may be considered as systems composed of basic blocks. Figure 2.3 presents a cascade connection and a parallel connection of two blocks with the transmittance $K_1(s)$ and $K_2(s)$.

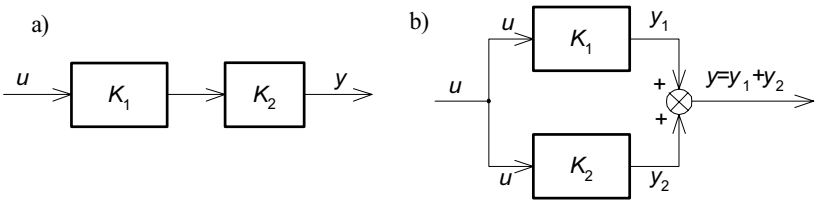


**Fig. 2.3.** a) Series connection, b) parallel connection

For multi-dimensional case, in the case of the cascade connection the number of outputs of the block $K_1$ must be equal to the number of inputs of the block $K_2$, and in the case of the parallel connection, both blocks

must have the same number of inputs and the same number of outputs. For the cascade connection

$$Y(s) = K_2(s)K_1(s)U(s).$$

For the parallel connection

$$Y(s) = [K_1(\mathrm{s}) + K_2(s)]U(s).$$

More details concerning the descriptions of linear dynamical blocks and their examples may be found in [14, 71, 76, 88].

## 2.4 Discrete Dynamical Plant

The descriptions of discrete dynamical plants are analogous to the corresponding descriptions for continuous plants presented in Sect. 2.3. The state vector description has now the form of a set of first-order difference equations, which in vector notation is written as follows:

$$\left.\begin{aligned} x_{n+1} &= f(x_n, u_n), \\ y_n &= \eta(x_n). \end{aligned}\right\}$$

The "input-output" description by means of the difference equation is now:

$$F_1(y_{n+m}, y_{n+m-1}, ..., y_n) = F_2(y_{n+v}, u_{n+v-1}, ..., u_n).$$

In particular, the linear model has the form of the linear difference equation

$$\begin{aligned} &y_{n+m} + A_{m-1}y_{n+m-1} + ... + A_1y_{n+1} + A_0y_n \\ &\quad = B_vu_{n+v} + B_{v-1}u_{n+v-1} + ... + B_1u_{n+1} + B_0u_n, \end{aligned}$$

and the operational description is as follows:

$$Y(z) = K(z)\ U(z)$$

where $K(z)$ denotes the *discrete operational transmittance*:

$$K(z) = (Iz^m + \sum_{i=1}^{m-1} A_i z_i)^{-1} (\sum_{j=1}^{v} B_j z^j).$$

The transmittance $K(z)$ is an $l \times p$ matrix whose entries $K_{ij}(z)$ are the transmittances of interconnections between the $j$-th input ant $i$-th output.

The functions $Y(z)$ and $U(z)$ denote here the *discrete operational transforms* (*Z-transforms*) of the respective discrete signals $y_n$ and $u_n$. The $K(e^{j\omega})$ for $-\pi < \omega < \pi$ is called a *discrete frequency transmittance* (*discrete gain-phase characteristic*).

We shall now present the description of a continuous plant being controlled and observed in a discrete way. Consequently, we have a discrete plant whose output $y_n$ is a result of sampling of the continuous plant output, i.e. $y_n = y(nT)$ where $T$ is the control and observation period. The input of the continuous plant $v(t)$ is formed by a sequence of decisions $u_n$ determined by a discrete controller and treated as the input of the discrete plant. It is a typical situation in the case of a computer control of the continuous plant. In the simplest case one assumes that $v(t) = u_n$ for $nT \le t < (n+1)T$. Such a signal for one-dimensional input can be presented as an effect of putting a sequence of Dirac impulses $u_n\delta(t-nT)$ at so called *zero-order hold* $E_O$ (see Fig. 2.4) with the transmittance

$$K_E(s) = \frac{1 - e^{-sT}}{s}$$

where $e^{-sT}$ denotes a delay equal to $T$. It may be shown that the transmittance of the discrete plant with the input $u_n$ and the output $y_n$ is equal to the $Z$-transform of the function $k_i(nT)$ where $k_i(t)$ denotes the impulse response of the element with transmittance $K_E(s)K_O(s)$, and $K_O(s)$ denotes the transmittance of the continuous plant. It is easy to note that

$$k_i(t) = \bar{k}_i(t) - \bar{k}_i(t-T)\,\mathbf{1}(t-T) \tag{2.14}$$

where $\bar{k}_i(t)$ is the impulse response of the element with the transmittance $\frac{1}{s}K_O(s)$.

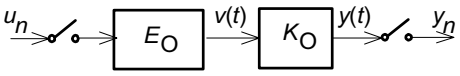


**Fig. 2.4.** Discrete plant with zero-order hold

**Example 2.2.** One should determine the transmittance of the discrete plant for the continuous plant described by the transmittance

$$K_{\mathrm{O}}(s) = \frac{k}{sT_0 + 1}.$$

It is easy to find

$$\bar{k}_i(t) = kt - kT_0(1 - e^{-\frac{t}{T_0}}).$$

After substituting $\bar{k}_i(t)$ into (2.14), putting $t = nT$ and applying $Z$-transformation, we obtain

$$K(z) = k\frac{[T - T_0(1-D)]z + T_0 - D(T_0 + T)}{z^2 - (1+D)z + D}$$

where $D = \exp(-\frac{T}{T_0})$. □

## 2.5 Control Algorithm

In Chap. 1 we have introduced a term controlling device or *controller* as an executor of the control algorithm. Hence the control algorithm may be considered as a description (mathematical model) of the controller. Since the descriptions of the plant presented in the previous sections can be applied to any elements or parts with a determined input and output, then they can be used as the basic forms of a control algorithm in these cases when it may be presented in an analytical form, i.e. as a mathematical formula. We shall often use the term controller in the place of a control algorithm (and vice versa) as well as the term plant in the place of a model of the plant. Let us denote by $w$ the input variable of the control algorithm (see similar notation concerning the controller in Sect. 1.2). Then a static control algorithm has the form

$$u = \Psi(w)$$

and a continuous dynamical control algorithm presented by means of the state vector $x_{\mathrm{R}}(t)$ is described by a set of equations

$$\dot{x}_{\mathrm{R}}(t) = f_{\mathrm{R}}[x_{\mathrm{R}}(t), w(t)],$$

$$u(t) = \eta_{\mathrm{R}}[x_{\mathrm{R}}(t)].$$

Similarly, one may speak about the descriptions by means of a differential

equation or descriptions in an operational form – analogous to those presented for a plant. For example, in technical control systems one often uses the one-dimensional controller described by the equation

$$\dot{u}(t) = k_1\dot{\varepsilon}(t) + k_2\varepsilon(t) + k_3\ddot{\varepsilon}(t)$$

or

$$u(t) = k_1\varepsilon(t) + k_2\int_0^t \varepsilon(t)dt + k_3\dot{\varepsilon}(t)$$

where $\varepsilon(t)$ denotes the control error. This is so called PID controller, or proportional-integrating-differentiating controller. Its transmittance

$$K_{\mathrm{R}}(s) = \frac{U(s)}{E(s)} = k_1 + \frac{k_2}{s} + k_3 s$$

where $E(s)$ is the Laplace transform of the function $\varepsilon(t)$.

Similarly, the forms of a discrete dynamical control algorithm are such as the descriptions of a discrete plant presented in Sect. 2.4. This is a control algorithm with a memory, i.e. in order to determine the decision $u_n$ it is necessary to remember the former decisions and the values $w$. The description of the algorithm is used as a basis for elaborating the corresponding program for the computer realization of the algorithm in a computer control system. One may say that it is an initial description of the control program. The block scheme of the algorithm written in the form

$$x_{\mathrm{R},n+1} = f_{\mathrm{R}}(x_{\mathrm{R},n}, w_n) \tag{2.15}$$

$$u_n = \eta_{\mathrm{R}}(x_{\mathrm{R}n}) \tag{2.16}$$

is presented in Fig. 2.5. The controlling computer determines the decisions $u_n$ in a real-time, in successive periods (intervals) which should be sufficiently long to enable the computer to calculate the decision $u_n$ in one interval. It determines the requirements concerning the execution time of the control program. The description of the control algorithm in the form of the difference equation is as follows:

$$u_{n+m} + a_{m-1}u_{n+m-1} + \ldots + a_1u_{n+1} + a_0u_n = b_{m-1}w_{n+m-1} + \ldots + b_1w_{n+1} + b_0w_n.$$

It is more convenient to present it in the form

$$u_n = -a_{m-1}\, u_{n-1} - \ldots - a_1\, u_{n-m+1} - a_0\, u_{n-m} + b_{m-1}\, w_{n-1} + \ldots + b_1\, w_{n-m-1} - b_0\, w_{n-m}.$$

This is the direct prescription for finding $u_n$ by using the former values $u$ and $w$, and the coefficients $a$ and $b$ placed in the data base. The number $m$ determines the required length of the memory.

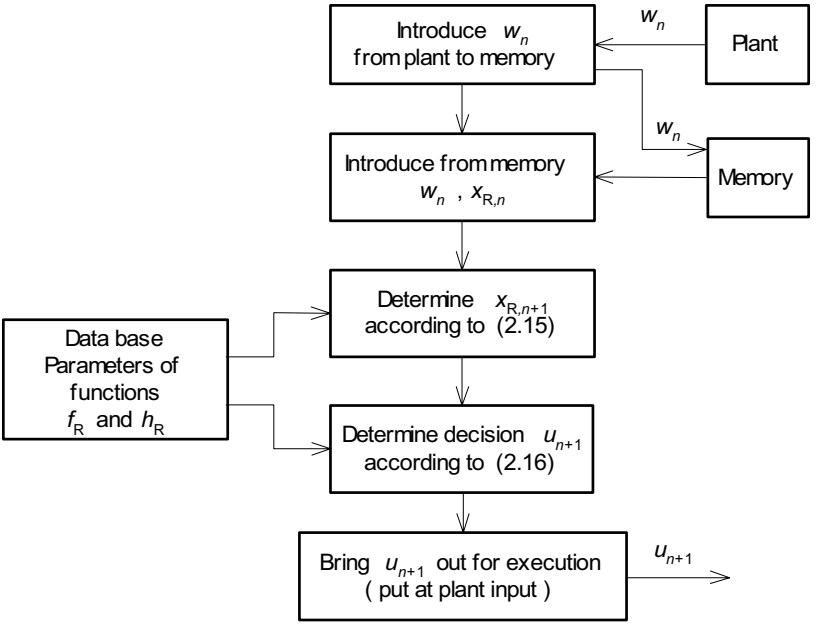


**Fig. 2.5.** Basic block scheme of control algorithm

## 2.6 Introduction to Control System Analysis

The description of the control system consists of formal models of the parts in the system and the description of the structure, that is interconnections between the parts. For example, the description of the closed-loop discrete control system by means of the state vector is the following

$$\left.\begin{aligned} x_{O,n+1} &= f_O(x_{O,n}, u_n, z_n), \\ y_n &= \eta_O(x_{O,n}), \end{aligned}\right\} \tag{2.17}$$

$$\left.\begin{aligned} x_{\mathrm{R},n+1} &= f_{\mathrm{R}}(x_{\mathrm{R},n}, y_n, y_n^*), \\ u_n &= \eta_{\mathrm{R}}(x_{\mathrm{R},n}) \end{aligned}\right\} \tag{2.18}$$

where $x_{\mathrm{O},n}$ is the state vector of the plant, $x_{\mathrm{R},n}$ is the state vector of the controller, $z_n$ is the vector of external disturbances acting on the plant, $y_n^*$ is the varying required value. Thus, in the system two basic parts are determined: the plant (2.17) and the controller (2.18). If it is the control error $\varepsilon_n = y_n^* - y_n$ which is put at the input of the plant, then the first equation in (2.18) takes the form

$$x_{\mathrm{R},n+1} = f_{\mathrm{R}}(x_{\mathrm{R},n}\,,\ \varepsilon_n).$$

The control system under consideration may be treated as one dynamical block whose state is $c_n^{\mathrm{T}} = [x_{\mathrm{O},n}^{\mathrm{T}}, x_{\mathrm{R},n}^{\mathrm{T}}]$ and whose inputs are the disturbances $z_n$ and $y_n^*$. This block is described by the equations

$$\left.\begin{aligned} c_{n+1} &= f(c_n, z_n, y_n^*), \\ y_n &= \eta_{\mathrm{O}}(x_{\mathrm{O},n}) \end{aligned}\right\} \tag{2.19}$$

if we consider $y_n$ as the output of the system as a whole. The first equation in (2.19) may be obtained via the elimination of $u_n$ and $y_n$ from the sets (2.17) and (2.18), i.e. by substituting $u_n = \eta_{\mathrm{R}}(x_{\mathrm{R},n})$ into the first equation in (2.17) and $y_n = \eta_{\mathrm{O}}(x_{\mathrm{O},n})$ into the first equation in (2.18). The description of the continuous control system by means of the state vector is analogous to that presented above for the discrete system.

The description of the control system forms a basis for its analysis. We may consider a qualitative analysis consisting in the investigation of some properties of the system (e.g. stability analysis) or a quantitative analysis consisting in the determination of a response of the system for a determined input and the determination of a performance (quality) index. For example, the analysis of the dynamical control system may consist in finding the transit response $y_n$ for the given initial state $c_0$ and the given functions $z_n$ and $y_n^*$ for $n \geq 0$. It requires to solve the set of the difference equations (2.17), (2.18). Usually, the analysis task is considered as the first stage of a design which in a parametric case consists in choosing the values of control algorithm parameters. The results of the analysis such as dependencies of the investigated property, the transient response or the per-

formance index upon the control algorithm parameters are needed to the proper choice of the values of these parameters.

In the case of non-linear and non-stationary (time-varying) systems, solving the analysis problem in an analytical way may be very difficult or impossible. Then we apply numerical methods or computer simulations, e.g. for the investigation of the influence of the control algorithm parameters on the performance index.

### 2.6.1 Continuous System

The operational description of the linear continuous closed-loop control system (Fig. 2.6) is the following:

$$\left.\begin{aligned} Y(s) &= K_{\mathrm{O}}(s)U(s) + \overline{K}(s)Z(s), \\ U(s) &= K_{\mathrm{R}}(s)E(s), \\ E(s) &= Y^{*}(s) - Y(s). \end{aligned}\right\} \tag{2.20}$$

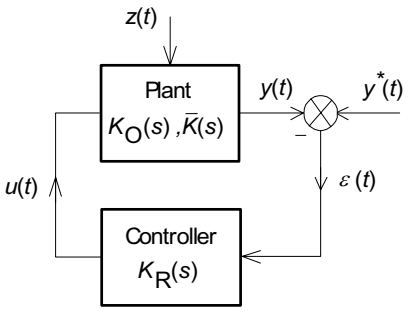


**Fig. 2.6.** Block scheme of control system under consideration

The plant is described by two transmittances: $K_{\mathrm{O}}(s)$ determining the influence of $u$ on $y$, and $\overline{K}(s)$ describing the influence of $z$ on $y$. From the set of equations (2.20) we obtain

$$Y(s) = [I + K_{\mathrm{O}}(s)K_{\mathrm{R}}(s)]^{-1}[K_{\mathrm{O}}(s)K_{\mathrm{R}}(s)Y^{*}(s) + \overline{K}(s)Z(s)], \tag{2.21}$$

$$E(s) = [I + K_{\mathrm{O}}(s)K_{\mathrm{R}}(s)]^{-1}[Y^{*}(s) - \overline{K}(s)Z(s)] \tag{2.22}$$

where $I$ is the unit matrix. The product $K_{\mathrm{O}}(s)K_{\mathrm{R}}(s) \triangleq K(s)$ is called a *transmittance of the open-loop control system*, and

$$[I+K(s)]^{-1}K(s) \triangleq K_Z(s)$$

is called a *transmittance of the closed-loop control system*. For the given functions $y^*(t)$ and $z(t)$, under assumption of zero initial conditions, one can find (or read from a table) the Laplace transforms $Y^*(s)$ and $Z(s)$, determine $Y(s)$ according to the formula (2.21) or $E(s)$ according to the formula (2.22), and by applying the inverse Laplace transform to determine the transient response $y(t)$ or $\varepsilon(t)$. Most often we investigate the transient response for the step inputs, i.e. for $y(t) = \bar{y}\,\mathbf{1}(t)$ or $z(t) = \bar{z}\,\mathbf{1}(t)$ where $\bar{y}$ and $\bar{z}$ denote constants, under the assumption that $\varepsilon(t) = 0$ for $t \le 0$. For example, in the temperature control system considered in Chap. 1 we can determine the function $\varepsilon(t)$ for $t > 0$, under the assumption that till the moment $t = 0$ the system was in the equilibrium state ($\varepsilon = 0$) and at the moment $t = 0$ a step change of the required value occurred.

In general, we have considered the multivariable system with the matrix transmittances and the vectors $y(t)$, $\varepsilon(t)$, $z(t)$. In one-dimensional case, i.e. for the plant with the single input $u$, the single disturbance $z$ and the single output $y$, the formulas (2.21) (2.22) take the simpler forms

$$Y(s) = \frac{K(s)Y^*(s) + \overline{K}(s)Z(s)}{1+K(s)}, \tag{2.23}$$

$$E(s) = \frac{Y^*(s) - \overline{K}(s)Z(s)}{1+K(s)}. \tag{2.24}$$

Let us denote by $L(s)$ and $M(s)$ polynomials in numerator and denominator of $K(s)$, respectively. From the form of the inverse transform of a function rational with respect to $s$ it follows that for a step change of $z$ and (or) $y^*$, the control error is a sum of components having the form $A_i e^{s_i t}$ or $A_i t^r e^{s_i t}$, and eventually a constant component, where $s_i$ are the roots of the equation

$$L(s) + M(s) = 0.$$

This is so called *characteristic equation of the closed-loop system*. If this equation has complex roots with imaginary parts differing from zero, then the sum of the components $A_i e^{s_i t}$ corresponding to the pair of conjugate roots is reduced to one component having the form $B_j e^{\sigma_j t} \sin(\omega_j t + \varphi_j)$

where $\sigma_j = \mathrm{Re}\, s_i$ . Thus, if the all roots satisfy the condition $\mathrm{Re}\, s_i < 0$ then $\varepsilon(t)$ converges to a constant (in particular, to zero) for $t \to \infty$. If the all roots are real then in the function $\varepsilon(t)$ the oscillation components will not occur.

### 2.6.2 Discrete System

Now in the system description *Z*-transforms and discrete transmittances occur. The formulas for the system are the same as (2.20) – (2.24) in which one should put $z$ in the place of $s$, having in mind that e.g. $E(z)$ denotes now the *Z*-transform of the function $\varepsilon_n$ , $K_\mathrm{R}(z)$ denotes the *discrete* transmittance of the controller etc. In particular, the formula for the control error $E(z)$ in one-dimensional system is now written as follows:

$$E(z) = \frac{Y^*(z) - \overline{K}(z)Z(z)}{1 + K(z)}. \tag{2.25}$$

In order to determine $\varepsilon_n$ for the given $y_n^*$ and $z_n$, one should find (or read from a table) the *Z*-transforms $Y^*(z)$ and $Z(z)$, determine $E(z)$ according to the formula (2.25) and by applying the inverse *Z*-transform determine $\varepsilon_n$. From the form of the inverse transform of a function rational with respect to $z$ it follows that for a step change of the disturbance $z$ and (or) $y^*$, the control error is a sum of components having the form $A_i z_i^n$ or $A_i n^r z_i^n$ where $z_i$ are the roots of the characteristic equation

$$L(z) + M(z) = 0,$$

$L(z)$ and $M(z)$ denote polynomials in numerator and denominator of the transmittance $K(z) = K_\mathrm{O}(z)K_R(z)$, respectively. It is easy to note that if the all roots satisfy the condition $|z_i| < 1$ then $\varepsilon_n$ converges to a constant (in particular, to zero) for $n \to \infty$.

We shall return to the analysis of control systems in Chaps. 5 and 10, when a parametric optimization and a stability will be discussed.

**Example 2.3.** Let us consider the one-dimensional closed-loop control system with the first-order plant and the controller I, i.e.

$$K_\mathrm{O}(s) = \frac{k_\mathrm{O}}{1 + sT}, \qquad K_\mathrm{R}(s) = \frac{k_\mathrm{R}}{s}.$$

Let us determine the transient response $\varepsilon(t)$ after the step change of the

required value $y^*(t) = \mathbf{1}(t)$. According to the formula (2.24) for $z(t) = 0$ we have

$$E(s) = \frac{1}{s(1 + \frac{k}{s(1+sT)})} = \frac{1+sT}{T(s^2 + \frac{s}{T} + \frac{k}{T})} \tag{2.26}$$

where $k = k_O k_R$. If the system parameters satisfy the condition $4kT < 1$ then the characteristic equation of the closed-loop system

$$s^2 + \frac{s}{T} + \frac{k}{T} = 0$$

has two real negative roots

$$s_{1,2} = \frac{-1 \pm \sqrt{1 - 4kT}}{2T}$$

and the formula (2.26) may be written in the form

$$E(s) = \frac{1}{T}(\frac{A_1}{s - s_1} + \frac{A_2}{s - s_2})$$

where

$$A_1 = \frac{1 + Ts_1}{s_1 - s_2}, \qquad A_2 = \frac{1 + Ts_2}{s_2 - s_1}.$$

After the inverse transformation we obtain

$$\varepsilon(t) = \frac{1}{T}(A_1 e^{s_1 t} + A_2 e^{s_2 t}).$$

Under the condition $4kT < 1$, the control error $\varepsilon(t)$ converges aperiodically (without oscillations) to zero for $t \to \infty$. If $4kT > 1$ then $\varepsilon(t)$ has a sinusoidal form with the amplitude exponentially decreasing to zero for $t \to \infty$. □

**Example 2.4.** Let us consider the discrete closed-loop control system with the following transmittances of the plant and the controller:

$$K_O(z) = \frac{k_O(z+b)}{z-a}, \qquad K_R(z) = \frac{k_R}{z-1}, \qquad k = k_O k_R = 1.$$

Let us determine the transient response $\varepsilon_n$ after the step change of the re-

quired value $y_n^* = \mathbf{1}(n)$. According to the formula (2.25), under the assumption that there are no disturbances acting on the plant

$$E(z) = \frac{z}{(z-1)(1+\dfrac{z+b}{(z-a)(z-1)})} = \frac{z(z-a)}{(z-a)(z-1)+z+b}. \qquad (2.27)$$

If the system parameters satisfy the condition $a^2 > 4(a+b)$ then the characteristic equation of the closed-loop system

$$z^2 - az + a + b = 0$$

has two real roots

$$z_{1,2} = \frac{a \pm \sqrt{a^2 - 4(a+b)}}{2}$$

and the formula (2.27) may be presented in the form

$$E(z) = \frac{A_1 z}{z - z_1} + \frac{A_2 z}{z - z_2}$$

where

$$A_1 = \frac{z_1 - a}{z_1 - z_2}, \qquad A_2 = \frac{z_2 - a}{z_2 - z_1}.$$

After the inverse transformation we obtain

$$\varepsilon_n = A_1 z_1^n + A_2 z_2^n.$$

If $|z_{1,2}| < 1$ then the control error converges to zero for $n \to \infty$. □

# 3 Control for the Given State (the Given Output)

Chapters 3, 4 and 5 form the **second part** of the book (see remarks in Sect. 1.6) in which deterministic control problems and algorithms are considered. It means that we consider a deterministic control plant (i.e., the values of the output are determined by the values of the input), and the description of the plant is precisely known. The exact meaning of these terms will be additionally explained in Sect. 6.1 where two kinds of an uncertainty will be considered: uncertainty concerning the plant (i.e., the plant is nondeterministic) and uncertainty of an expert giving the description of the plant. When the second uncertainty does not occur, we often say about the *control with full information of the plant*. For dynamical plants, this information contains not only the knowledge of the plant description but also the initial state and the function describing time-varying disturbances from the initial to the final moments of the control, if such disturbances exist.

This chapter is devoted to a *basic control problem* (a basic decision problem), i.e., the determination of the control for which we obtain the given required output value for a static plant or the given value of the state for a dynamical plant. Such a control may be executed in an open-loop or a closed-loop system. For the dynamical plant the execution in a closed-loop system may require the application of so called *observer* which determines the values of the current states of the plant using the results of the output measurements.

## 3.1 Control of a Static Plant

Let us consider a static plant described by a function

$$y = \Phi(u, z) \tag{3.1}$$

where $u \in U$ is the input vector (or the control vector) with $p$ components, $y \in Y$ is the output vector with $l$ components and $z \in Z$ is a vector of external disturbances with $r$ components. For this plant the following problems may be formulated:

**Analysis problem:** For the given function $\Phi$ and the values $u$ and $z$

one should determine the value $y$.

**Decision making (control) problem:** For the given function $\Phi$, the value z and the required value $y^*$ one should determine such a decision $u$ that its execution (putting at the input) gives the required output value $y^*$.

For the determination of the control decision one should solve the equation (3.1) with respect to $u$, with $y = y^*$. Under the assumption of existence and uniqueness of the solution we obtain the control algorithm in the form of the function

$$u = \Psi(z). \tag{3.2}$$

This algorithm is executed in the open-loop control system (Fig. 3.1). If the solution does not exist, the plant is called *uncontrollable*. If the solution is not unique, we obtain a set of possible decisions for the given $z$. For every decision from this set, the requirement $y = y^*$ will be satisfied.

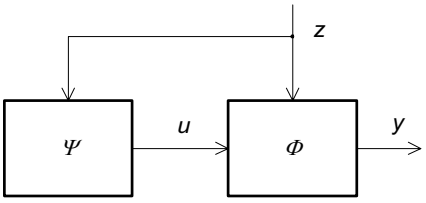


**Fig. 3.1.** Open-loop control system

In particular, for the linear plant

$$y = Au + Bz, \tag{3.3}$$

under the assumption that $p = l$ and $A$ is a nonsingular matrix (i.e., the determinant $\det A \neq 0$), the control algorithm is the following

$$u = A^{-1}(y^* - Bz) \tag{3.4}$$

where $A^{-1}$ denotes an inverse matrix. The control computer should then execute the following operations:

1. Multiplication of the matrix $B$ by the vector $z$.
2. Subtraction of the result of the operation 1 from $y^*$.
3. Inverting of the matrix $A$.
4. Multiplication of the matrix $A^{-1}$ by the result of the operation 2.

Obtaining the solution of the equation (3.1) for $y = y^*$ may be difficult for a nonlinear plant. Then a computational algorithm determining a se-

quence of approximate solutions may be applied. The basic algorithm of the successive approximation has the following form:

$$u_{m+1} = u_m + K\,[\,y^* - \Phi(u_m, z)] \tag{3.5}$$

where $u_m$ denotes the $m$-th approximation and $K$ is a coefficient matrix which should be chosen in such a way as to assure the convergence of the sequence $u_m$ to the solution (3.2). It is also necessary to determine the stop of the procedure, i.e., to determine the final approximation; e.g. if the distance between $u_{m+1}$ and $u_m$ is less than the given number, the value $a_{m+1}$ is assumed as a decision which is put at the input of the plant. If $z$ is varying in time then the formulas (3.1) and (3.2) take the form

$$y_n = \Phi(u_n, z_n), \qquad u_n = \Psi(z_n),$$

respectively, where $u_n$, $y_n$, $z_n$ denote the values in the $n$-th moment of the control. Then the algorithm (3.5) has the form

$$u_{n,m+1} = u_{n,m} + K\,[\,y^* - \Phi(u_{n,m}, z_n)]$$

where $u_{n,m}$ denotes the $m$-th approximation in the $n$-th period (the $n$-th interval) of the control. The approximation process till the stop should not exceed the control interval, then the convergence of the process must be sufficiently fast.

If $z$ is constant then the algorithm (3.5) may be executed in the closed-loop control system (Fig. 3.2). It means that instead of putting the successive approximation into the formula (3.1) and calculating the value $\Phi(u_m, z)$, one puts $u_n$ at the input of the plant and measures the output $y_n$. The value $u_m$ is now the $m$-th approximation of the solution (3.2) and, on the other hand, the control decision in the $m$-th decision interval. For the unification of the notations for discrete-time control system, the index $n$ is used instead of $m$, as is denoted in Fig. 3.2. According to the formula (3.5), the control algorithm in the closed-loop system is the following

$$u_{n+1} = u_n + K \cdot \varepsilon_n \tag{3.6}$$

where $\varepsilon_n = y^* - y_n$.

If the model $\Phi$ describes precisely the plant (and this is assumed in our considerations in this chapter), then the value $y$ calculated from the model and the value measured at the output are identical. Consequently, the sequence $u_m$ in the algorithm (3.5) is exactly the same as the sequence $u_n$ in the algorithm (3.6). An advantage of the control in the closed-loop sys-

tem consists in avoiding possible computational difficulties connected with the determination of the value $y$ from the model. Essential advantages arise in the case of control based on an incomplete knowledge of the plant. That is why in Chap. 10 we shall return to the concept presented here and to the convergence problem.

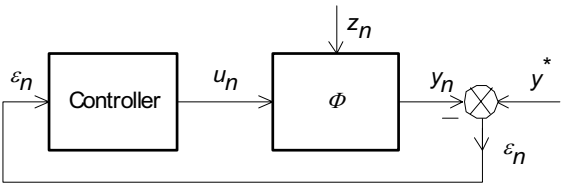


**Fig. 3.2.** Closed-loop control system

## 3.2 Control of a Dynamical Plant. Controllability

The problem analogous to that presented in Sect. 3.1, for the dynamical plant is much more complicated and consists in the determination of a control $u(t)$ in the continuous case or $u_n$ in the discrete case, which remove the plant from the initial state to the given final state in a finite time interval. The essential difficulty for the dynamical plant is caused by the fact that the output value in a determined fixed moment depends not only on the nearest input value but also on the former inputs, and the requirement does not concern the output value but the state of the plant. As a rule, it is an equilibrium state, i.e. when it is reached and the control is finished, the output does not change in the next moments. Further considerations concerning the control of the dynamical plants in this chapter will be limited to discrete-time plants, and particular exact results and control algorithms – to the linear plants. In [76, 80] one may found details concerning the problem considered in this chapter for the dynamical plants and, in particular, properties called controllability and observability which we shall introduce here.

Let us consider the discrete plant

$$x_{n+1} = f(x_n, u_n), \tag{3.7}$$

where $x_n$ is the state vector and $u_n$ is the input (the control) vector.

**Decision making (control) problem:** For the given function $f$, the initial state $x_0$ and the final state $x^*$ one should determine the sequence of the

control decisions $u_0, u_1, \ldots, u_{N-1}$ such that $x_N = x^*$ and $N < \infty$.

With the existence of the solution the property called *controllability* of the plant is related.

**Definition 3.1.** The plant (3.7) is called *controllable* for a pair $(x_0, x^*)$ if there exists a solution of the problem under consideration, i.e., there exists a control which removes the plant from the state $x_0$ to the state $x^*$ in a finite time interval. The plant is called *fully controllable* if it is controllable for every pair $(x_0, x^*)$. □

The general approach to the problem solution is based on the set of equations containing the equations (3.7) for $n = 0, 1, \ldots, N-1$ and the equation $x_N = x^*$. This set should be solved with respect to $u_0, u_1, \ldots, u_{N-1}$. The number $N$ must be such that the solution exists if it is possible. The formulation of the controllability condition and the determination of the solution for a nonlinear plant may be very difficult. The precise analytical solution may be given for linear plants.

Let us consider a linear plant with constant parameters and a single input

$$x_{n+1} = Ax_n + bu_n. \tag{3.8}$$

Assume that $A$ is a nonsingular matrix, i.e. $\det A \neq 0$. Besides, let us assume that $x^* = \overline{0}$ (a vector with zero components). The set of equations used to determine the values $u_0, u_1, \ldots, u_{N-1}$ has now the form

$$\left.\begin{aligned} x_{n+1} &= Ax_n + bu_n \quad \text{for} \quad n = 0, 1, \ldots, N-2, \\ \overline{0} &= Ax_{N-1} + bu_{N-1}. \end{aligned}\right\} \tag{3.9}$$

For $N = k$ we have the set of $k$ linear equations with $k$ unknowns. By the successive substitutions and elimination of the variables $x_1, x_2, \ldots, x_{k-1}$ from the set (3.9) we obtain

$$\left.\begin{aligned} x_1 &= Ax_0 + bu_0, \\ x_2 &= Ax_1 + bu_1 = A^2x_0 + Abu_0 + bu_1, \\ &\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\ x_k &= \overline{0} = A^kx_0 + A^{k-1}bu_0 + A^{k-2}bu_1 + \ldots + Abu_{k-2} + bu_{k-1}. \end{aligned}\right\} \tag{3.10}$$

The set of equations (3.10) may be rewritten in a vector-matrix notation

$$M\,\bar{u}_{0,k} = -A^k x_0 \tag{3.11}$$

where

$$M = [\,A^{k-1}b \quad A^{k-2}b \;\ldots\; Ab \quad b], \qquad \bar{u}_{0,k}^{\mathrm{T}} = [\,u_0 \quad u_1 \;\ldots\; u_{k-1}].$$

In this notation $A^{k-1}b$, ..., $Ab$, $b$ denote columns of the matrix $M$, T denotes the transposition of the matrix and $\bar{u}_{0,k}$ is a column-vector with components $u_0$, $u_1$,..., $u_{k-1}$. From the equation (3.11) one obtains

$$\bar{u}_{0,k} = -M^{-1}A^k x_0 \tag{3.12}$$

under the assumption $\det M \neq 0$. In such a way it has been proved that $\det M \neq 0$ is a sufficient controllability condition, i.e., if this condition is satisfied then there exists a control which in $k$ periods (then, in a finite time) removes the plant from the state $x_0$ to the state $x^*$. It may be shown that if such a control does not exist for $N = k$ then it does not exist for $N > k$, i.e., it may be proved that $\det M \neq 0$ is also a necessary controllability condition in the case under consideration. Of course, we are speaking about a full controllability because the controllability condition does not depend on $(x_0, x^*)$.

The above results may be summarized in the form of the following theorem:

**Theorem 3.1** (*controllability condition*). The plant (3.8) in which $\det A \neq 0$ is fully controllable if and only if

$$\det [\,A^{k-1}b \quad A^{k-2}b \;\ldots\; Ab \quad b] \neq 0. \tag{3.13}$$

□

The formula (3.12) presents the *control algorithm in the open-loop system*. It shows how to determine the proper sequence of control decisions $\bar{u}_{0,k}$ for the given initial state and the plant parameters $A$, $b$. Let us complete the considerations with three important remarks:

**Remark 3.1.** It may be proved that for the multi-input plant

$$x_{n+1} = Ax_n + Bu_n,$$

under the assumption $\det A \neq 0$, the necessary and sufficient condition of the full controllability is as follows

$$r([A^{k-1}B \quad A^{k-2}B \quad \ldots \quad AB \quad B]) = k \tag{3.14}$$

where $A^{k-i}B$ $(i = 1, 2, ..., k)$ denote submatrices, and $r$ denotes a rank of the matrix, i.e. the number of linearly independent rows (columns). The condition (3.14) presents a generalization of the condition (3.13) which may be written in the form $r(M) = k$. □

**Remark 3.2.** Let us note that the control may be shorter than $k$ periods. If $N < k$ then in the solution (3.12) $u_N = u_{N+1} = ... = u_{k-1} = 0$. If $x_0$ is such that there exists $u_0$ for which $A^k x_0 = -bu_0$ then $N = 1$ (see (3.8)). Generally, if $x_0$ is such that there exists a sequence $u_0, ..., u_{N-1}$, for which

$$A^k x_0 = -A^{N-1}bu_0 - A^{N-2}bu_1 - ... - bu_0 ,$$

then the control contains $N$ intervals. It may be said that $x_0$ may be taken to $x^* = \bar{0}$ during $N$ periods $(N < k)$ if $A^k x_0$ belongs to $N$-dimensional subspace of the vector space $X$ generated by the basis $A^{N-1}b, A^{N-2}b, ..., b$. □

**Remark 3.3.** Consider the plant with external disturbances $z_n$. Then $x_{n+1} = f(x_n, u_n, z_n)$ and for the linear plant with one-dimensional disturbance

$$x_{n+1} = Ax_n + bu_n + cz_n .$$

Then in the equation presented in the description of the general approach, and in particular in the set (3.9) and in the equation analogous to (3.10) the sequence $z_0, z_1, ..., z_{N-1}$ appears. *A priori* knowledge of this sequence before the determination of the decision sequence $\bar{u}_{0,N-1}$ belongs to the full information on the plant under consideration. □

## 3.3 Control of a Measurable Plant in the Closed-loop System

The control presented by the formula (3.12) may be obtained in the closed-loop system for a measurable plant. In this case $y_n = x_n$, i.e. in the successive moments the state of the plant is measured. Let us treat the current state in $n$-th moment as an initial state for the next part of the process. Us-

ing (3.12), for the given $x_n$ put in place of $x_0$ one can determine the current decision $u_n$ considered in place of $u_0$, i.e. the first component of the vector $\bar{u}_{0,k}$. As a result one obtains

$$u_n = - w_1 x_n \tag{3.15}$$

where $w_1$ denotes the first row of the matrix $M^{-1}A^k$. The formula (3.15) presents the *control algorithm in the closed-loop system* and shows how to determine the current decision $u_n$ using the result of the measurement $x_n$. In the closed-loop control system (Fig. 3.3) the control error $\varepsilon_n = x^* - x_n$.

The assumption $x^* = \bar{0}$, introduced in Sect. 3.2 means that new variables have been introduced and the state $x_n$ is the difference between $x^*$ and the original state $\bar{x}_n$. In other words, the origin in the state space $X$ has been located in the point $x^*$. In the case under consideration we then apply linear static controller with constant parameters $u_n = w_1 \varepsilon_n$, which assures the finite time of the control, i.e., which takes the components of the control error to zero values in a finite time interval. The value of the control decision $u_n$ is a linear combination of the components of the state vector $x_n$, and the components of the vector $-w_1$ are the coefficients in this combination.

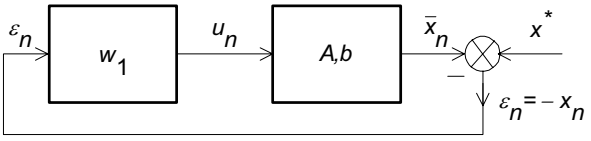


**Fig. 3.3.** Closed-loop control system for the plant under consideration

Let us note that the concept of the control in the closed-loop system presented here is in some sense analogous to the concept for the static plant presented in Sect. 3.1. If the values of $A$ and $b$ in the model are the same as in a real plant (as it has been assumed here) then the sequence of the values $u_n$ determined in real time according to the algorithm (3.15) is the same as the sequence (3.12) for $n = 0, 1, ..., k–1$. In the open-loop system, the whole sequence of the decisions $u_0, u_1, ..., u_{k–1}$ should be determined and put into memory before starting the control process. In the closed-loop system it is sufficient to determine the current decisions $u_n$ in real time,

using the measurements of the result of former decisions, i.e. the state $x_n$. The control algorithm in the closed-loop system is much simpler than that in the open-loop system. After determination of the values $w_1$ by a designer of the system before starting the control – finding the decisions in successive moments is reduced to calculating the linear combination $w_1 x_n$ in which the values of the components of the vector $x_n$ are transferred from the plant. One should however take into account that the current decision $u_n$ must be calculated relatively quickly, at the beginning of the $n$-th interval (period) of the control process. Introducing the values from the plant, determining the current decisions and executing them (putting at the input of the plant) in successive control intervals means a *real time control*. Essential advantages of the control in closed-loop systems arise in the case of control based on an incomplete knowledge of the plant, i.e., when the values of $A, b$ accepted for the calculations by a designer differ from the values in the real plant. We shall return to this problem in the **fourth part** of the book.

**Example 3.1.** Let us check the controllability condition and determine the control algorithm in the closed-loop system for the second-order plant described by equations

$$\left.\begin{aligned} x_{n+1}^{(1)} &= 2x_n^{(1)} - x_n^{(2)} + u_n, \\ x_{n+1}^{(2)} &= x_n^{(1)} + x_n^{(2)} - u_n. \end{aligned}\right\}$$

For our plant the matrices $A$ and $b$ are then as follows:

$$A = \begin{bmatrix} 2 & -1 \\ 1 & 1 \end{bmatrix}, \qquad b = \begin{bmatrix} 1 \\ -1 \end{bmatrix}.$$

Hence,

$$Ab = \begin{bmatrix} 2 & -1 \\ 1 & 1 \end{bmatrix}\begin{bmatrix} 1 \\ -1 \end{bmatrix} = \begin{bmatrix} 3 \\ 0 \end{bmatrix}, \qquad M = [\,Ab \;\; b] = \begin{bmatrix} 3 & 1 \\ 0 & -1 \end{bmatrix}.$$

Since $\det M = -3 \neq 0$, the plant is fully controllable. Next, one should calculate

$$M^{-1} = \begin{bmatrix} \frac{1}{3} & \frac{1}{3} \\ 0 & -1 \end{bmatrix}, \qquad M^{-1}A^2 = \begin{bmatrix} 2 & -1 \\ -3 & 0 \end{bmatrix}.$$

According to (3.15)

$$u_n = -[2 \ -1]\begin{bmatrix} x_n^{(1)} \\ x_n^{(2)} \end{bmatrix} = -2x_n^{(1)} + x_n^{(2)}. \qquad \square$$

## 3.4 Observability

Let us assume that the plant is not measurable, i.e., the result of the measurement $y_n$ at the output does not determine the state in the $n$-th moment. Then we have a problem of so called **observation of the plant**, which consists in the determination of $x_n$ using the results of the measurements of the output $y_n$ in a finite time interval up to the $n$-th moment. It is also an important problem for the plant without a control, described by equations

$$\left.\begin{aligned} x_{n+1} &= f(x_n), \\ y_n &= \eta(x_n) \end{aligned}\right\} \tag{3.16}$$

where $y_n$ is the output vector.

**Observation problem for the plant without a control:** For the given functions $f$, $\eta$ and the sequence $y_n, y_{n-1}, ..., y_{n-(N-1)}$, where $N < \infty$, one should determine the value $x_n$. The sequence $y_n, y_{n-1}, ..., y_{n-(N-1)}$, i.e. the sequence of outputs successively measured during $N$ intervals of observations is called an *observation sequence*. The relationship

$$x_n = G(y_n, y_{n-1}, ..., y_{n-(N-1)}) \tag{3.17}$$

is called an *observation algorithm* and a system (a unit) executing this algorithm is called an *observer*. With the existence of the solution of the observation problem the property called *observability* of the plant is related.

**Definition 3.2.** The state $x_n$ of the plant (3.16) is called *observable* if it may be determined using the finite observation sequence. The plant is fully observable if its every state is observable. $\square$

The general approach to the problem solution is based on the set of equations

$$\left.\begin{aligned} x_i &= f(x_{i-1}), \\ y_i &= \eta(x_i), \end{aligned}\right\}$$

for $i = n, n-1, ..., n-(N-2)$ and the equation

$$y_{n-(N-1)} = \eta(x_{n-(N-1)}).$$

This set should be solved with respect to $x_n$ by the elimination of the variables $x_{n-1}, ..., x_{n-(N-1)}$. The number $N$ must be such that the solution exists if it is possible. Consider now the plant with the control decisions $u_n$ described by

$$\left.\begin{aligned} x_{n+1} &= f(x_n, u_n), \\ y_n &= \eta(x_n). \end{aligned}\right\} \tag{3.18}$$

In this case it is easy to note that in the set of equations described above and obtained by successive substitutions $n, n-1, ..., n-(N-1)$ in (3.18), the sequence of the control decisions appears.

**Observation problem for the controlled plant:** For the given functions $f$, $\eta$, the sequence of control decisions $u_{n-1}, ..., u_{n-(N-1)}$ and the observation sequence $y_n, ..., y_{n-(N-1)}$ where $N < \infty$, one should determine the value $x_n$.

The general form of the observation algorithm is then the following:

$$x_n = G(u_{n-1}, ..., u_{n-(N-1)}; y_n, ..., y_{n-(N-1)}), \tag{3.19}$$

and the control sequence occurs also in the definition of observability. Consequently, the existence of the solution of the observation problem may depend on the control sequence. For a single-input and single-output plant ($p = l = 1$) the control and observation sequences may be presented in the form of vectors

$$\overline{u}^{\mathrm{T}}_{n-1,N-1} = [u_{n-1} \;\; u_{n-2} \;\; ... \;\; u_{n-(N-1)}], \qquad \overline{y}^{\mathrm{T}}_{n,N} = [y_n \;\; y_{n-1} \;\; ... \;\; y_{n-(N-1)}],$$

and the observation algorithm (3.19) has the form

$$x_n = G(\overline{u}_{n-1,N-1}, \;\; \overline{y}_{n,N}).$$

The block scheme of the observation system is presented in Fig. 3.4.

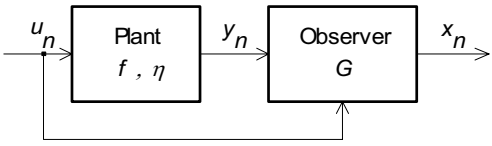


**Fig. 3.4.** Observation system

The formulation of the observability condition and the determination of the observation algorithm for a nonlinear plant may be very difficult. The precise analytical solution may be given for linear plants. Consider a linear single-input plant with constant parameters

$$x_{n+1} = Ax_n, \qquad y_n = c^{\mathrm{T}} x_n \tag{3.20}$$

and assume $\det A \neq 0$. The set of equations used to determine $x_n$ has now the following form:

$$x_n = Ax_{n-1}, \qquad y_n = c^{\mathrm{T}} x_n,$$

$$x_{n-1} = Ax_{n-2}, \qquad y_{n-1} = c^{\mathrm{T}} x_{n-1},$$

$$\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots$$

$$x_{n-(N-2)} = Ax_{n-(N-1)}, \qquad y_{n-(N-2)} = c^{\mathrm{T}} x_{n-(N-2)},$$

$$y_{n-(N-1)} = c^{\mathrm{T}} x_{n-(N-1)}.$$

Assume $N = k$ (the number of components of $x_n$). Applying the successive substitutions we obtain

$$y_n = c^{\mathrm{T}} x_n,$$

$$y_{n-1} = c^{\mathrm{T}} x_{n-1} = c^{\mathrm{T}} A^{-1} x_n,$$

$$y_{n-2} = c^{\mathrm{T}} x_{n-2} = c^{\mathrm{T}} A^{-2} x_n,$$

$$\ldots\ldots\ldots\ldots\ldots\ldots\ldots$$

$$y_{n-(k-1)} = c^{\mathrm{T}} x_{n-(k-1)} = c^{\mathrm{T}} A^{-(k-1)} x_n.$$

The above set of equations with the unknown $x_n$ may be rewritten in a vector-matrix notation as follows:

$$\bar{y}_{n,k} = \widetilde{M}\, x_n = \overline{M}\, A^{1-k} x_n \tag{3.21}$$

where

$$\widetilde{M} = \begin{bmatrix} c^{\mathrm{T}} \\ c^{\mathrm{T}} A^{-1} \\ \vdots \\ c^{\mathrm{T}} A^{-(k-2)} \\ c^{\mathrm{T}} A^{-(k-1)} \end{bmatrix}, \qquad \overline{M} = \begin{bmatrix} c^{\mathrm{T}} A^{k-1} \\ c^{\mathrm{T}} A^{k-2} \\ \vdots \\ c^{\mathrm{T}} A \\ c^{\mathrm{T}} \end{bmatrix}. \tag{3.22}$$

In this notation the rows of the matrices $\widetilde{M}$ and $\overline{M}$ have been presented. From the equation (3.21) we obtain

$$x_n = A^{k-1} \overline{M}^{-1} \bar{y}_{n,k} \tag{3.23}$$

under the assumption $\det \overline{M} \neq 0$. In such a way it has been proved that $\det \overline{M} \neq 0$ is a sufficient condition of the full observability. It follows from the fact that if this condition is satisfied then any state $x_n$ may be determined by using the observation sequence containing $k$ intervals, i.e., a finite observation sequence. It may be shown that if the solution of our problem does not exist for $N = k$ then it does not exist for $N > k$, which means that $\det \overline{M} \neq 0$ is also a necessary condition of the full observability. The above results may be summarized in the form of the following theorem:

**Theorem 3.2** (*observability condition*): The plant (3.20) in which $\det A \neq 0$ is fully observable if and only if $\det \overline{M} \neq 0$ where the matrix $\overline{M}$ is determined in (3.22). □

Since $\overline{M} = \widetilde{M} A^{k-1}$ and $\det A \neq 0$, it is easy to note that the condition $\det \overline{M} \neq 0$ may be replaced by the equivalent condition $\det \widetilde{M} \neq 0$. In the case under consideration the formula (3.23) presents the *observation algorithm* which shows what operations should be performed as to determine $x_n$ using the observation sequence. If the observation sequence with $N < k$ is sufficient to the determination of $x_n$ then in the matrix $\widetilde{M}$ there are zero columns for $N+1, ..., k$.

The considerations for the controlled single-input, single-output plant

$$x_{n+1} = Ax_n + bu_n, \qquad y_n = c^{\mathrm{T}} x_n \tag{3.24}$$

are analogous but more complicated. Now the set of equations from which $x_n$ should be determined has the following form:

$$
\begin{aligned}
&y_n = c^{\mathrm{T}} x_n ,\\
&y_{n-1} = c^{\mathrm{T}} x_{n-1} = c^{\mathrm{T}} A^{-1}(x_n - b u_{n-1}) = c^{\mathrm{T}} A^{-1} x_n - c^{\mathrm{T}} A^{-1} b u_{n-1} ,\\
&y_{n-2} = c^{\mathrm{T}} x_{n-2} = c^{\mathrm{T}} A^{-2} x_n - c^{\mathrm{T}} A^{-2} b u_{n-1} - c^{\mathrm{T}} A^{-1} b u_{n-2} ,\\
&\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\\
&y_{n-(k-1)} = c^{\mathrm{T}} x_{n-(k-1)}\\
&\qquad = c^{\mathrm{T}} A^{-(k-1)} x_n - c^{\mathrm{T}} A^{-(k-1)} b u_{n-1} - \ldots - c^{\mathrm{T}} A^{-1} b u_{n-(k-1)} .
\end{aligned}
$$

The above set of equations with the unknown $x_n$ may be written in the form

$$\bar{y}_{n,k} = \tilde{M} x_n - D \bar{u}_{n-1,k-1} \tag{3.25}$$

where $D$ is the following matrix with $k$ rows and $(k-1)$ columns:

$$D = \begin{bmatrix} 0 & 0 & 0 & \ldots & 0 \\ c^{\mathrm{T}} A^{-1} b & 0 & 0 & \ldots & 0 \\ c^{\mathrm{T}} A^{-2} b & c^{\mathrm{T}} A^{-1} b & 0 & \ldots & 0 \\ \ldots\ldots & \ldots\ldots & \ldots\ldots & \ldots\ldots & \ldots\ldots \\ c^{\mathrm{T}} A^{-(k-1)} b & c^{\mathrm{T}} A^{-(k-2)} b & c^{\mathrm{T}} A^{-(k-3)} b & \ldots & c^{\mathrm{T}} A^{-1} b \end{bmatrix} . \tag{3.26}$$

By solving the equation (3.25) we obtain the *observation algorithm*

$$x_n = \tilde{M}^{-1}(\bar{y}_{n,k} + D \bar{u}_{n-1,k-1}). \tag{3.27}$$

Note that the observability condition is now the same as for the plant without control, i.e., $\det \tilde{M} \neq 0$ or $\det \overline{M} \neq 0$. It may be shown that for the multi-output plant with $y_n = C x_n$, the condition of the full observability is as follows:

$$r\left( \begin{bmatrix} CA^{k-1} \\ CA^{k-2} \\ \vdots \\ C \end{bmatrix} \right) = k . \tag{3.28}$$

It is a generalization of the condition $\det \overline{M} \neq 0$ for single-output plant, which may be written in the form $r(\overline{M}) = k$.

## 3.5 Control with an Observer in the Closed-loop System

The observer may be used for the control of an unmeasurable plant in the closed-loop system. If

$$u_n = \Psi(x_n) \tag{3.29}$$

denotes the control algorithm for the measurable plant then putting the observation algorithm (3.19) into (3.29) one obtains the control algorithm $\overline{\Psi}$ in the closed-loop system in which the output $y_n$ is measured:

$$\begin{aligned} u_n &= \Psi[G(u_{n-1}, ..., u_{n-(N-1)}; y_n, ..., y_{n-(N-1)})] \\ &\triangleq \overline{\Psi}\ (u_{n-1}, ..., u_{n-(N-1)}; y_n, ..., y_{n-(N-1)}). \end{aligned} \tag{3.30}$$

Comparing with the control algorithm for the measurable plant let us note that the control algorithm with the measurement of the output (which is not a state) contains a memory and describes the determination of the current decision $u_n$ not only on the basis of $y_n$, but also the results of the former measurements $y_{n-1}$,..., $y_{n-(N-1)}$ and the former decisions $u_{n-1}$, ..., $u_{n-(N-1)}$ taken from the memory. The control is performed in the closed-loop system containing the observer (Fig. 3.5).

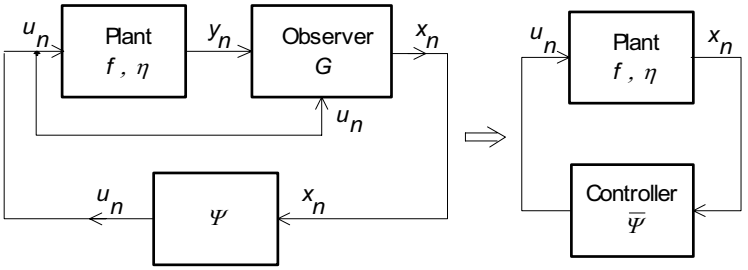


**Fig. 3.5.** Closed-loop control system with the observer

It is worth noting that the control can start after the determination of the first state, i.e., after measuring $y_0$, $y_1$, ..., $y_{N-1}$ and determining $x_{N-1}$. In practical situations taking the state $x_0$ to $x^*$ may be not a unique task and may be repeated after a disturbance consisting in the change of the given required state $x^*$. Let us remind that $x_n$ is a difference between a real state initially formulated and the state $x^*$ (see Fig. 3.3). For a single-output plant as the state variables $x_n$ we often accept successive values of

the output $y_n, y_{n-1}, ..., y_{n-(k-1)}$ or successive values of the control error $\varepsilon_n$, $\varepsilon_{n-1}, ..., \varepsilon_{n-(k-1)}$ if the required value $y^*$ differs from zero. In such a case there is no additional observation problem (the dependence of $x_n$ on the sequence $y_n$ follows directly from the definition) and the algorithm (3.29) is reduced directly to the algorithm with a memory for the plant with the measured output.

Let us note that the composition of the algorithms $\Psi$ and $G$ leads to one *resulting* control algorithm $\overline{\Psi}$ as it is denoted in the right part of Fig. 3.5. In the computer implementation it is however worth keeping two separate parts $G$ and $\Psi$, i.e., to design the control program in the form of two co-operating parts: subprogram of the observation and subprogram of the control based on $x_n$. It can make easier computer simulations of the system and changes of the program parameters according to changes of the plant parameters in an adaptive system.

For the linear plant (3.24) considered above, the control algorithm $\overline{\Psi}$ presented generally by (3.30) has a precise specific form which may be obtained by substituting the observation algorithm (3.27) into the control algorithm (3.15). As a result we obtain the following control algorithm in the closed-loop system:

$$u_n = - w_1 \tilde{M}^{-1} ( \bar{y}_{n,k} + D \bar{u}_{n-1,k-1} ) \tag{3.31}$$

where $w_1$ denotes the first row of the matrix $M^{-1}A^k$,

$$M = [A^{k-1}b \quad A^{k-2}b \ \ldots \ Ab \quad b],$$

and the matrices $\tilde{M}$, $D$ are defined by the formulas (3.22), (3.26), respectively. After some transformations the formula (3.31) may be reduced to the form

$$u_n = - a_{k-2}\, u_{n-1} - \ldots - a_0\, u_{n-(k-1)} + b_{k-1}\, y_n + \ldots + b_0\, y_{n-(k-1)}\,, \tag{3.32}$$

or

$$u_{n+k-1} + a_{k-2}\, u_{n+k-2} + \ldots + a_0\, u_n = b_{k-1}\, y_{n+k-1} + \ldots + b_0\, y_n\,.$$

This relationship presents the control algorithm in the form of a difference equation. It may be also presented as an operational transmittance

$$K_{\mathrm{R}}(z) = \frac{U(z)}{Y(z)} = \frac{b_{k-1}z^{k-1} + \ldots + b_1 z + b_0}{z^{k-1} + a_{k-2}z^{k-2} + \ldots + a_1 z + a_0}\,. \tag{3.33}$$

This is a *transmittance of the controller* assuring the finite time of the control process. After $k$ intervals (in a particular case it may be a smaller number) the control error is reduced to zero and kept at this level up to the appearing of a disturbance which requires a new control.

Taking into account the control problem under consideration let us pay attention to two different applications of a computer, related with a control system:

1. A computer as a tool aiding the design of the system.
2. A computer as an executor of the control in real time.

In the first case, the computer is applied at the stage of the design and determines the values of the coefficients in the control algorithm (3.32) using the data $A$, $b$, $c$ introduced at the input. Algorithm of the design and consequently the *design program* consists of the following operations:

a. Determination of the matrices $\tilde{M}^{-1}$ and $D$.

b. Determination of the matrix $\tilde{M}^{-1}D$.

c. Determination of $w_1$, i.e. the first row of the matrix $\tilde{M}^{-1}A^k$.

d. Determination of the vector of coefficients $[b_{k-1} \ \ldots \ b_0] \stackrel{\Delta}{=} \bar{b}$ :

$$\bar{b} = -w_1 \tilde{M}^{-1}. \tag{3.34}$$

e. Determination of the vector of coefficients $[a_{k-2} \ \ldots \ a_0] \stackrel{\Delta}{=} \bar{a}$ :

$$\bar{a} = w_1 \tilde{M}^{-1} D = -\bar{b} D. \tag{3.35}$$

In the second case, the computer executes the control according to the algorithm (3.32), i.e. according to the *control program* implemented, using the data $\bar{a}$ and $\bar{b}$ introduced to the data base for the control plant, and the given values $u$ and $y$ introduced currently in successive intervals from the memory and from the plant. A block scheme of the control algorithm, i.e. the procedure of the determination of the decision $u_n$ in the $n$-th step is presented in Fig. 3.6. In a similar way as for the design, the computer may play only a role of a tool aiding or supporting the control. Then introducing the current results of the observations $y_n$ and transferring the decisions for the execution (e.g. in a management process) is performed by a human operator. For technical plants in which $y_n$ is a result of a measurement and $u_n$ is put at the input of the plant by special executing devices, full automation is possible, i.e., the values $y_n$ are transferred directly to the control computer and the values $u_n$ are delivered directly to

the executing devices.

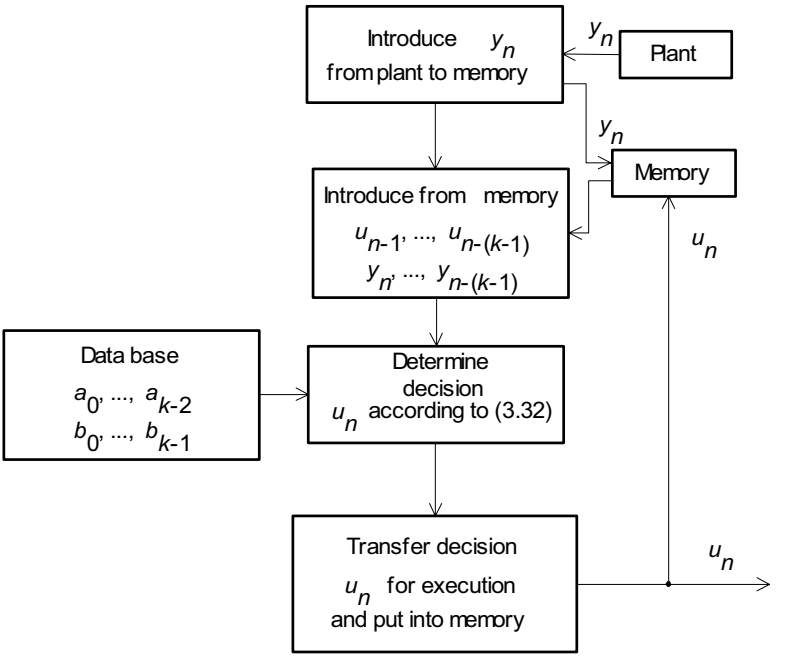


**Fig. 3.6.** Block scheme of the control algorithm in the case under consideration

**Example 3.2.** Let us check the observability condition and determine the control algorithm in the closed-loop system for the plant considered in Example 3.1, in which

$$y_n = x_n^{(1)} + 2x_n^{(2)},$$

i.e., $c^{\mathrm{T}} = [1 \;\; 2]$. After substituting the numerical data we have

$$\widetilde{M} = \begin{bmatrix} c^{\mathrm{T}} \\ c^{\mathrm{T}} A^{-1} \end{bmatrix} = \begin{bmatrix} 1 & 2 \\ -\frac{1}{3} & \frac{5}{3} \end{bmatrix}, \qquad D = \begin{bmatrix} 0 \\ c^{\mathrm{T}} A^{-1} b \end{bmatrix} = \begin{bmatrix} 0 \\ -2 \end{bmatrix}.$$

Since $\det \widetilde{M} = \frac{7}{3} \neq 0$, then the plant is fully observable. Then we calculate

$$\widetilde{M}^{-1} = \begin{bmatrix} \frac{5}{7} & -\frac{6}{7} \\ \frac{1}{7} & \frac{3}{7} \end{bmatrix}$$

and use the row $w_1 = [2 \ -1]$ determined in Example 3.1. After substituting the numerical data into (3.34) and (3.35) we obtain

$$\overline{b} = \begin{bmatrix} -\frac{9}{7} & \frac{15}{7} \end{bmatrix}, \qquad \overline{a} = \frac{30}{7} .$$

Consequently, the control algorithm (3.32) and the transmittance of the controller (3.33) are as follows:

$$u_n = -\frac{30}{7} u_{n-1} - \frac{9}{7} y_n + \frac{15}{7} y_{n-1} ,$$

$$K_R(z) = \frac{-9z + 15}{7z + 30}$$ □

## 3.6 Structural Approach

For the plant

$$x_{n+1} = Ax_n + bu_n , \tag{3.36}$$

let us introduce one-to-one linear mapping

$$v_n = Px_n , \qquad \det P \neq 0 , \tag{3.37}$$

which reduces the equation (3.36) to the form

$$v_{n+1} = \overline{A}\, v_n + \overline{b}\, u_n \tag{3.38}$$

where $\overline{A} = PAP^{-1}$, $\overline{b} = Pb$. Such an operation means introducing new state variables $v_n$ in place of $x_n$. The descriptions of the plant (3.36) and (3.38) are equivalent. Consider the following form of the equation (3.38) containing zeros in the respective places of the matrices $\overline{A}$ and $\overline{b}$ :

$$\begin{bmatrix} v_{n+1}^{\mathrm{I}} \\ v_{n+1}^{\mathrm{II}} \end{bmatrix} = \begin{bmatrix} A_{11} & A_{12} \\ \overline{0} & A_{22} \end{bmatrix} \begin{bmatrix} v_n^{\mathrm{I}} \\ v_n^{\mathrm{II}} \end{bmatrix} + \begin{bmatrix} b^{\mathrm{I}} \\ \overline{0} \end{bmatrix} u_n \tag{3.39}$$

where $v_n^{\mathrm{I}}$ is a subvector of the vector $v_n$ with $k_1$ components, $v_n^{\mathrm{II}}$ is a subvector of the vector $v_n$ with $k_2$ components ($k_1 + k_2 = k$), $A_{11}$, $A_{12}$, $A_{21}$, $A_{22}$ are submatrices of the matrix $\overline{A}$, $A_{21} = \overline{0}$ (all the entries are equal to zero), $b^{\mathrm{I}}$ with $k_1$ components and $b^{\mathrm{II}}$ with $k_2$ components are subvectors of the vector $\overline{b}$, the matrix $A_{11}$ has $k_1$ rows and $k_1$ columns, the matrix $A_{12}$ has $k_1$ rows and $k_2$ columns, the matrix $A_{21} = \overline{0}$ has $k_2$ rows and $k_1$ columns, the matrix $A_{22}$ has $k_2$ rows and $k_2$ columns. From the equation (3.39) it follows

$$v_{n+1}^{\mathrm{I}} = A_{11} v_n^{\mathrm{I}} + A_{12} v_n^{\mathrm{II}} + b^{\mathrm{I}} u_n ,$$
$$v_{n+1}^{\mathrm{II}} = A_{22} v_n^{\mathrm{II}} .$$

It is easy to see that the control $u_n$ has neither direct nor indirect influence on the changes of the state vector $v^{\mathrm{II}}$. It follows from the fact that $v_{n+1}^{\mathrm{II}}$ does not depend on $v_n^{\mathrm{I}}$. The following theorem may be proved:

**Theorem 3.3.** The plant (3.36) is fully controllable if and only if one-to-one mapping $P$ for which $\overline{A}$ and $\overline{b}$ have the form such as in equation (3.38) does not exist. □

The decomposition of the state vector $v_n$ into two subvectors means the decomposition of the plant into two parts: part I with state vector $v_n^{\mathrm{I}}$ and part II with state vector $v_n^{\mathrm{II}}$. Assume that the first part is controllable, i.e., the pair $A_{11}$, $b^{\mathrm{I}}$ satisfies the controllability condition (3.13) in which $A$ and $b$ are replaced by $A_{11}$ and $b^{\mathrm{I}}$, respectively (it may be shown that the existence of the second input in the form $A_{11} v_n^{\mathrm{II}}$ does not change the controllability condition in the part under consideration). Hence, the plant (3.38) is decomposed into controllable part and uncontrollable part.

Analogous considerations may be related to the observability. Let us assume that the mapping (3.37) reduces the plant equation

$$x_{n+1} = Ax_n\,, \qquad y_n = c^{\mathrm{T}} x_n \tag{3.40}$$

to the form

$$\begin{bmatrix} v_{n+1}^{\mathrm{I}} \\ v_{n+1}^{\mathrm{II}} \end{bmatrix} = \begin{bmatrix} A_{11} & \bar{0} \\ A_{21} & A_{22} \end{bmatrix} \begin{bmatrix} v_n^{\mathrm{I}} \\ v_n^{\mathrm{II}} \end{bmatrix}, \qquad y_n = [c^{\mathrm{I}} \;\; \bar{0}\,] \begin{bmatrix} v_n^{\mathrm{I}} \\ v_n^{\mathrm{II}} \end{bmatrix} \tag{3.41}$$

where $c^{\mathrm{I}}$ is the first part of the row $c^{\mathrm{T}}$ and $A_{12} = \bar{0}$. Then $y$ neither directly nor indirectly depends on $v^{\mathrm{II}}$. It follows from the fact that $v_{n+1}^{\mathrm{I}}$ does not depend on $v_n^{\mathrm{II}}$.

**Theorem 3.4.** The plant (3.40) is fully observable if and only if one-to-one mapping $P$ reducing the equation (3.40) to the form (3.41) does not exist. □

If the pair $A_{11}$, $c^{\mathrm{I}}$ satisfies the observability condition for part I then the decomposition of the state vector $v_n$ into two subvectros presented here means the decomposition of the plant into two parts: the observable part with state vector $v_n^{\mathrm{I}}$ and the unobservable part with state vector $v_n^{\mathrm{II}}$. The considerations concerning the existence of the mapping $P$ for non-controllability and non-observability may be presented together and generalized for a multi-input and multi-output plant. In a general case, there may exist such a nonsingular mapping $P$ that the plant equation

$$x_{n+1} = Ax_n + Bu_n\,, \qquad y_n = Cx_n$$

is reduced to the form

$$\begin{bmatrix} v_{n+1}^{\mathrm{I}} \\ v_{n+1}^{\mathrm{II}} \\ v_{n+1}^{\mathrm{III}} \\ v_{n+1}^{\mathrm{IV}} \end{bmatrix} = \begin{bmatrix} A_{11} & \bar{0} & A_{12} & \bar{0} \\ A_{21} & A_{22} & A_{23} & A_{24} \\ \bar{0} & \bar{0} & A_{33} & \bar{0} \\ \bar{0} & \bar{0} & A_{43} & A_{44} \end{bmatrix} \begin{bmatrix} v_n^{\mathrm{I}} \\ v_n^{\mathrm{II}} \\ v_n^{\mathrm{III}} \\ v_n^{\mathrm{IV}} \end{bmatrix} + \begin{bmatrix} B_1 \\ B_2 \\ \bar{0} \\ \bar{0} \end{bmatrix} u_n\,, \tag{3.42}$$

$$y_n = [C^{\mathrm{I}} \;\; \overline{0} \;\; C^{\mathrm{III}} \;\; \overline{0}] \begin{bmatrix} v_n^{\mathrm{I}} \\ v_n^{\mathrm{II}} \\ v_n^{\mathrm{III}} \\ v_n^{\mathrm{IV}} \end{bmatrix}. \tag{3.43}$$

Matrices $B_1$, and $B_2$ are submatrices of matrix $B$; $A_{11}$, $A_{22}$, $A_{33}$ and $A_{44}$ are quadratic matrices and the other matrices are rectangular with respective numbers of rows and columns. This means that in the plant under consideration it is possible to distinguish four parts (Fig.3.7): controllable and observable with the state $v^{\mathrm{I}}$, controllable and unobservable with the state $v^{\mathrm{II}}$, uncontrollable and observable with the state $v^{\mathrm{III}}$, uncontrollable and unobservable with the state $v^{\mathrm{IV}}$. With the help of $u_n$ it is possible to influence only the parts I and II, and only the parts I and III may be observed by measuring $y_n$.

## 3.7 Additional Remarks

The analogous considerations for continuous plants are more complicated. It may be shown that for the plant

$$\dot{x} = Ax + Bu, \qquad y = Cx$$

the controllability and observability conditions are the same as for the respective discrete plant, i.e., they are the conditions (3.14) and (3.28), respectively. The control algorithms in the closed-loop system analogous to (3.15) and (3.31) are time-varying (non-stationary). It means that in a continuous linear stationary system (i.e. the system with constant parameters), unlike a discrete system, the finite control time is not possible.

Let us note that the decomposition of the plant into four interconnected parts presented above has been obtained in a quite formal way by applying the mapping $P$ to the equation describing the plant. It does not have to mean that inside the plant illustrated in Fig. 3.7 there are real separate four parts, e.g. four interconnected technical devices. The structural description of the plant presented in Sect. 3.6 is not of constructive

importance because it is difficult to check if a given plant is controllable and observable by using the conditions in Theorems 3.3 and 3.4.

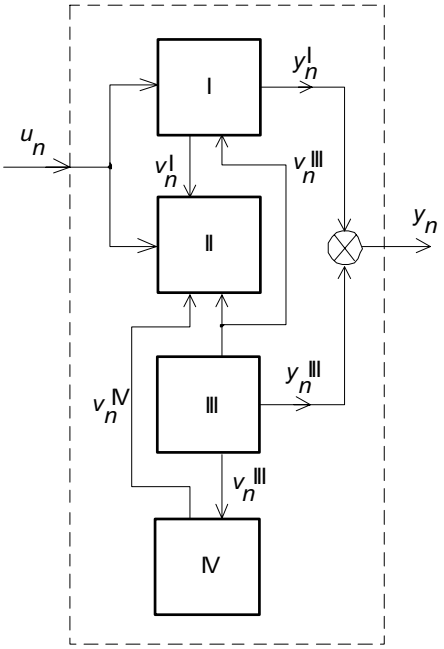


**Fig. 3.7.** Structure of the plant under consideration

The structural approach, however, is of a certain methodological and auxiliary importance. In particular, using the concepts of controllability and observability as well as the decomposition described in Sect. 3.6, it is easy to show that, in general, the different descriptions of dynamical plants presented in Chap. 2 are not equivalent. The description using a state vector is the most precise and fully representing the dynamical properties of the plant (in general – the system with an input and output). It contains the whole plant, i.e. four parts presented above. The *input-output* description in the form of a differential or difference equation comprises the observable parts. For a linear plant with constant parameters it may be shown that the description in the form of a transmittance comprises the controllable and observable part only. The three descriptions are then equivalent if the plant is fully controllable and observable.

It is worth noting that the non-controllability and non-observability conditions (reduced to the statements that the determinants of the respective matrices are equal to zero) are very strong or crisp in this sense

that they may be not satisfied as a result of very small changes of the plant parameters. In other words, these conditions are very *sensitive* to the changes of the plant parameters. That is why non-controllability and non-observability are not likely to occur in practice, except the situation when a practical plant consists of real four interconnected parts presented in Fig. 3.7, i.e., in a real plant the selected entries of the matrices $A$, $B$ and $C$ are precisely equal to zero as it was presented in the description (3.42), (3.43).

In all considerations concerning the dynamical plant it was assumed that there are no external disturbances $z_n$. If the disturbances occur then one should use the plant equation

$$x_{n+1} = f(x_n, u_n, z_n).$$

Hence, in the set of equations considered above the sequence $z_0$, $z_1$, ..., $z_{N-1}$ will appear. Consequently, to determine the decision $u_n$ it is necessary to know the whole sequence of the disturbances, i.e., not only $z_0$, ..., $z_n$, but also $z_{n+1}$, ..., $z_{N-1}$. Then, the full information on plant assumed in our consideration for the whole **second part** of the book contains now *a priori* (i.e. before starting the control) knowledge of the values of disturbances which will occur in future. Usually, if the sequence $z_n$ is *a priori* known, $z_n$ denotes time-varying parameter of the plant with a known description. In practical considerations we use the term *external disturbance* if in the $n$-th moment we may know only $z_0$, $z_1$, ..., $z_n$ (if the disturbances are measured and stored in the memory), but we do not know the future values of $z$. That is why in this chapter concerning the full knowledge of the plant, the plants without disturbances have been considered.

Summarizing, let us note that in this chapter precise **control algorithms** (3.4), (3.6), (3.12), (3.15), (3.31) and (3.32) and **observation algorithms** (3.23), (3.27) have been presented. They may be used as a basis for the development of programs for **computer aided design** and for **real-time computer control** (direct digital control) in control systems considered in this chapter.

# 4 Optimal Control with Complete Information on the Plant

This is the second chapter of the second part of the book, devoted to the determination of the control in the case of a complete information on the deterministic plant. In this chapter we shall consider problems and algorithms of the control which is the best in a determined sense. This sense is formulated by so called *performance* (*quality*) *index*. It is worth reminding (see the corresponding remarks in Chap. 1) that the control problem is understood here in a wide sense as a decision making problem. In the case of discrete plants, it will be the optimization problem for the multistage decision making process which has many practical applications not only to the control of technical systems and processes, but also to the management (e.g. to the optimization of investment processes [20]).

## 4.1 Control of a Static Plant

Let us consider a static (memory-less) plant with the input (control) vector $u$, the output vector $y$ and the vector of disturbances $z$ – described in Sect. 3.1. Let us introduce the performance index $\varphi(y, y^*)$ evaluating the value $y$ with respect to the given required value $y^*$. For the fixed $y^*$, the function $\varphi$ assigns real non-negative numbers to the values $y \in Y$, and $\varphi(y^*, y^*) = 0$. Usually, $\varphi(y, y^*) = \| y - y^* \|$ or $\varphi(y, y^*) = \| y - y^* \|^2$ where $\| \cdot \|$ denotes the norm of a vector. Then $\| y - y^* \|$ denotes the distance between the points $y$ and $y^*$ in the space $Y$. The most frequently used forms of the performance indexes are as follows:

$$\varphi_1(y, y^*) = (y - y^*)^{\mathrm{T}}(y - y^*) = \sum_{i=1}^{l} (y^{(i)} - y^{(i)*})^2 , \tag{4.1}$$

$$\varphi_2(y, y^*) = \sum_{i=1}^{l} |y^{(i)} - y^{(i)*}|, \tag{4.2}$$

$$\varphi_3(y, y^*) = \max_i |y^{(i)} - y^{(i)*}| \tag{4.3}$$

where $\sqrt{\varphi_1}$ is an Euclidean norm, and $\varphi_2, \varphi_3$ are so called modular norms. The performance index may be interpreted as a loss caused by the fact that $y$ differs from $y^*$. More complicated forms of performance indexes with weight coefficients for the different components of the vector $y - y^*$ may be used. The next extension of the forms (4.1), (4.2), (4.3) may be obtained by putting $Py$ in the place of $y$ into (4.1), (4.2), (4.3) where $P$ is a nonsingular matrix, i.e. by applying a nonsingular mapping of the vector $y$. Then

$$\varphi_1(y, y^*) = (y - y^*)^{\mathrm{T}} R (y - y^*) \tag{4.4}$$

where $R = P^{\mathrm{T}}P$ is a symmetric, positive definite matrix (i.e. all eigenvalues of this matrix are positive), and $\varphi_2, \varphi_3$ are the respective norms of the vector $P(y - y^*)$. For the determined performance index, the optimal control problem consists in the determination of the decision $u = \Psi(z)$ minimizing the value of $\varphi$. If there are no constraints concerning the choice of $u$, then the problem is reduced to the determination of $u = \Psi(z)$ for which $y = y^*$, i.e. to the problem considered in Sect. 3.1. The formulation and solution of the optimization problem make sense if the solution of the equation $y = y^*$ with respect to $u$ for the fixed $z$ does not exist or the solution does not satisfy the constraints. The formulation of the performance index may be extended to the function $\varphi(y, u)$ containing the evaluation not only of $y$ but also $u$. Usually it is the sum of components evaluating separately $y$ and $u$:

$$\varphi(y, u) = \varphi_y(y) + \varphi_u(u). \tag{4.5}$$

In particular, one may use so called *quadratic performance index* in the form

$$\varphi(y, u) = y^{\mathrm{T}} R_y y + u^{\mathrm{T}} R_u u \tag{4.6}$$

where $R_y$ and $R_u$ are symmetric, positive definite matrices. The absence

of $y^*$ in the first component may denote that the quality is formulated directly for $y$ without using $y^*$ or that $y^* = \overline{0}$, i.e. in the place of $y - y^*$ in the formula (4.4) the new variable denoted by $y$ has been introduced. Usually, the component $\varphi_u(u)$ is interpreted as a *cost of the control* or the cost of resources (energy, raw materials etc.) used for the control; the interpretation of the first component is different in different practical applications. For example, the components of the vector $y$ may denote amounts of some components in a product and $y^*$ – the desirable values of these components. Then $(y - y^*)^{\mathrm{T}} R_y (y - y^*)$ denotes the evaluation of the product quality. In other practical situations $\varphi_y$ may denote the productivity of the production process which we want to maximize taking into account the amount of a raw material $\varphi_u$ used in the production process. For unification we usually assume that the optimization consists in minimization of the performance index. Then the negative value of the productivity evaluation $\varphi_y$ should be put into the sum (4.5).

Different kinds of constraints may appear in the optimization problem. Most frequently, they concern the control decisions $u$ and are formalized by a set of admissible values $D_u \subset U$. For example, $D_u$ is a set of all values $u$ for which $\varphi_u(u)$ is not greater than the given positive number $\alpha$, i.e.

$$D_u = \{u \in U: \quad \varphi_u(u) \le \alpha\}. \tag{4.7}$$

Of course, in this case $\varphi_u$ is not a component of the performance index (4.5). For example, the problem may consist in the determination of $u$ maximizing the productivity $\varphi_y(y)$, satisfying the constraints concerning the resources in the form $\varphi_u(u) \le \alpha$. Frequently, the set $D_u$ has the following form

$$D_u = \{u \in U: \bigwedge_{i \in \overline{1,p}} (|u^{(i)}| \le M_i)\} \tag{4.8}$$

which means that for each $i = 1, 2, ..., p$, i.e. $i \in \overline{1,p}$ the condition $|u^{(i)}| \le M_i$ must be satisfied, i.e. the control decisions $u^{(i)}$ must lie within the determined limits.

The general formulation of the decision problem for the static plant (i.e. the problem of the static optimization) is as follows:

**Decision making (control) problem:** For the given function $\Phi$, i.e. the model of the plant (3.1), the given function $\varphi(y,u)$ and the value $z$ one should determine the decision $u$ satisfying the given constraint and minimizing the function

$$\varphi[\Phi(u,z),u] \stackrel{\Delta}{=} \overline{\Phi}(u,z). \tag{4.9}$$

The result of the minimization depends on $z$ and, in the case of the unique solution, the optimal decision is a function of $z$, i.e.

$$u = \arg\min_{u} \overline{\Phi}(u,z) \stackrel{\Delta}{=} \Psi(z). \tag{4.10}$$

The function (4.10) presents the control algorithm in an open-loop system (see Fig. 3.1). It may be said that $\overline{\Phi}$ is a model of a static optimization plant, i.e. the plant with the control input $u$ and one-dimensional output $\overline{y} = \overline{\Phi}(u,z)$. In this formulation, the goal of the control does not consist in reaching by $\overline{y}$ a given desirable value but in reaching the minimal value from the set of all possible $\overline{y}$. Sometimes in this case we speak about an extremal control and a plant of the extremal control, which has been mentioned in Chap. 1. Let us assume that the function $\overline{\Phi}$ in the formula (4.9) is differentiable with respect to the particular components of the vector $u$, there are no constraints concerning $u$, and the value $u = \Psi(z)$ minimizing the performance index is a unique point for which

$$\operatorname{grad}_{u} \overline{\Phi}(u,z) \stackrel{\Delta}{=} \tilde{\Phi}(u,z) \stackrel{\Delta}{=} w = \overline{0} \tag{4.11}$$

where $w$ denotes the gradient of the function $\overline{\Phi}$ with respect to $u$, i.e. the vector with components

$$w^{(i)} = \frac{\partial \overline{\Phi}(u,z)}{\partial u^{(i)}}, \qquad i = 1, 2, ..., p.$$

Under these assumptions, the determination of the value $u$ minimizing the output $\overline{y}$ of the optimization plant is reduced to the determination of the decision $u$ for the given desirable output $w^* = \overline{0}$ for a *substitutional* plant with the output $w$ and the model $\overline{\Phi}$, i.e. to the problem considered in Sect. 3.1. The equation (4.11) from which the control algorithm $u = \Psi(z)$ may be determined, is analogous to the equation (3.1) for $y = y^*$. The algorithms analogous to (3.5) and (3.6) have now the forms

$$u_{m+1} = u_m - K\widetilde{\Phi}(u_m, z)\,, \tag{4.12}$$

$$u_{n+1} = u_n - Kw_n. \tag{4.13}$$

For the execution of the algorithm (4.13) in a closed-loop system, it is necessary to "measure" (more precisely speaking – to obtain in a proper way adequate to a practical situation) the variables $w_n$ at the output of the substitutional plant.

There exist different algorithms of the static optimization, i.e. algorithms for the determination of successive approximations of the value $u$ minimizing a multivariable function, subject to constraints. We shall come back to the static optimization problem in a closed-loop system in Chap. 10 where the control under uncertainty will be considered.

## 4.2 Problems of Optimal Control for Dynamical Plants

### 4.2.1 Discrete Plant

The formulation of the optimal control problem for dynamical plants contains the determination of the plant model, the initial state, the performance index and the constraints. Let us consider a discrete plant described by the equation

$$x_{n+1} = f(x_n, u_n). \tag{4.14}$$

Usually, the evaluation (estimation) of the control quality concerns the state of the plant, and more precisely speaking, the state trajectory $x_1$, $x_2$, ..., $x_N$ where $N$ is a *horizon* of the control, i.e. the time of the control in which the quality is estimated and in which the sequence of the control decisions $u_0$, $u_1$, ..., $u_{N-1}$ should be determined. Usually, the performance index has the additive form

$$Q_N = \sum_{n=1}^{N} \varphi(x_n) \tag{4.15}$$

where $\varphi(x_n)$ denotes the local evaluation of the single state $x_n$. The form of the function $\varphi$ is the same as in Sect. 4.1 for the evaluation of the output, e.g. $\varphi(x_n) = \| x_n - x^* \|$ where $x^*$ is the given state. Then the control

problem consists in the determination of the sequence $u_0, u_1, ..., u_{N-1}$ minimizing $Q_N$ for the given initial state $x_0$. In a similar way as for the static plant, the estimation of the quality may contain also the cost of the control. Then, in the $n$-th moment (the $n$-th period of the control) the function $\varphi(x_n, u_{n-1})$ evaluates the state $x_n$ and the control $u_{n-1}$ which gives the state $x_n$. Usually, this is a sum of two components evaluating separately $x_n$ and $u_{n-1}$:

$$Q_N = \sum_{n=1}^{N} \varphi(x_n, u_{n-1}) = \sum_{n=1}^{N} \varphi_x(x_n) + \sum_{n=1}^{N} \varphi_u(u_{n-1}) . \tag{4.16}$$

This form is analogous to that in (4.5), and the practical interpretation is similar. In the case of a quadratic performance index, $\varphi_x$ and $\varphi_y$ are such as in the formula (4.6) and the second component of the sum (4.16) may be interpreted e.g. as an energy used for the control. The constraints may concern the state and the control as well. In the first case, the typical constraint has the form $x_N = x^*$, which means that reaching the state $x^*$ at the end of the control is required. It is a control problem with the fixed end of the trajectory. If such a requirement is not stated, we speak about the free end of the trajectory. Two kinds of the constraints in the form of inequalities may concern the whole state trajectory:

$$1. \bigwedge_{n \in \overline{1,N}} (x_n \in D_{x,n}) , \tag{4.17}$$

$$2. \sum_{n=1}^{N} \varphi_x(x_n) \le \alpha_x \tag{4.18}$$

where $D_{x,n}$ denotes a set of admissible values, in general – different for different moments $n$, and $\alpha_x$ denotes a given positive number. Of course, in the case (4.18) the component evaluating $x_n$ will not appear in the formula (4.16). The constraints concerning the choice of the control decisions may have the similar forms:

$$1. \bigwedge_{n \in \overline{0,N-1}} (u_n \in D_{u,n}) , \tag{4.19}$$

$$2.\ \sum_{n=1}^{N} \varphi_u(u_{n-1}) \le \alpha_u. \tag{4.20}$$

In the case (4.20) the component evaluating the control does not appear in the formula (4.16). As one can see, the forms of the performance index and the constraints should be matched in a proper way, as to keep the practical sense of the problem. The optimization with two components presented in the right hand side of the formula (4.16) may be called a *two-criteria* problem. In the formulation (4.16) we use a *global criterion* estimating the state and the control as well. Other problem formulations may consist in minimization of the first component in (4.16) and take into account the constraint in the form of the inequality (4.20), concerning the second component or in minimization of the second component only with the constraint (4.18) concerning the first component. For example, from the set of all controls such that the state trajectory satisfies the constraint (4.18) one should choose the control with the minimal energy.

Let us consider an important special case when in the formula (4.18) the function $\varphi(x_n)$ is equal to 1 for all $n$. Then $Q_N = N$ which means that the control horizon is not given in advance, but we want to obtain the control time as small as possible. Of course, such a problem makes sense if an additional requirement is taken into account. Most often, it is a fixed end of the trajectory, i.e. $x_N = x^*$. Then one should determine the *time-optimal control*, i.e. the control which removes the plant from the state $x_0$ to the state $x^*$ in minimum time. It is worth noting that such a control occurred in Chap. 3, in the part concerning a dynamical plant. Usually, however, there are constraints for the choice of $u_n$ which may be not satisfied by the control determined in Chap. 3.

Summarizing the above remarks one can formulate general optimal control problem with the additive form of the performance index, containing a variety of special cases.

**Optimal control problem**

**Data:** a description of the plant, i.e. the function $f$, the initial state $x_0$, the required final state $x^*$ (for the problem with the fixed end of the trajectory), the performance index, i.e. $\varphi$ in the formula (4.16), and constraints if there are any.

**One should determine:** the sequence of optimal decisions $u_0^*$, $u_1^*$, ..., $u_{N-1}^*$ satisfying the constraints and minimizing the performance index, i.e.

$$(u_0^*, \ldots, u_{N-1}^*) = \arg \min_{u_0, \ldots, u_{N-1}} \sum_{n=1}^{N} \varphi(x_n, u_{n-1}).$$

Not using the word *control* we may say that the problem under consideration consists in the determination of the optimal sequence of decisions in a multistage decision process, or shortly, in the **optimization of multistage decision process**. If the sequence $u_0, \ldots, u_{N-1}$ denotes the execution program for a multistage project to be planned (e.g. a business plan), then this program must not be determined *from step to step*, but one should take into account the estimation of the effect for the whole execution time and in this way to take into account relations between the successive stages: the state $x_n$ is not only a result of the decision $u_{n-1}$ but also the initial state for the next stage.

Sometimes forms of performance indexes different than the additive form are used, e.g.

$$Q_N = \varphi(x_N) \qquad \text{or} \qquad Q_N = \max_{n \in \overline{1,N}} \varphi(x_n).$$

In the first case we evaluate the final effect only, having in mind that it depends on the whole sequence $u_0, \ldots, u_{N-1}$. In this case we speak about a *terminal control*.

### 4.2.2 Continuous Plant

In an analogous way one can formulate the optimal control problem for a continuous plant

$$\dot{x} = f(x, u) \tag{4.21}$$

where $x(t)$ and $u(t)$ are functions of $t$. Now, instead of an additive performance index we use an integral performance index in the form

$$Q_T = \int_0^T \varphi(x,u)\,dt = \int_0^T \varphi_x(x)\,dt + \int_0^T \varphi_u(u)\,dt \tag{4.22}$$

where $T$ denotes a control horizon. In the case of a time-optimal control $\varphi(x, u) \equiv 1$, $Q_T = T$, $x(T) = x^*$. For the continuous plant the control from $t = 0$ to $t = T$ should be determined. In other words, one should determine

a function $u(t)$ in the interval $[0,T]$, which we shall denote by $u(t)\big|_0^T$ to differ it from $u(t)$ denoting the value $u$ in the moment $t$.

**Optimal control problem**

**Data:** a description of the plant, i.e. the function $f$ in (4.21), the initial state $x_0$, the required final state $x^*$ (for the problem with the fixed end of the trajectory) , the performance index, i.e. $\varphi$ in the formula (4.22), and constraints if there are any.

**One should determine:** the function $u^*(t)\big|_0^T$ satisfying the constraints and minimizing the performance index, i.e.

$$u^*(t)\big|_0^T = \arg \min_{u(t)\big|_0^T} \int_0^T \varphi(x,u)\,dt\,.$$

Let us note that the performance index (4.22) together with the plant equation (4.21) and the initial state $x_0$ formulate a *functional*, i.e. a mapping which assigns numbers $Q_T$ to the functions $u(t)\big|_0^T$.

The determination of the optimal control may be then formulated as the minimization of a functional, i.e. as a *variational* problem. However, the direct application of methods based on a classical variational analysis may be very difficult because of the following reasons:

1. The functional to be minimized with respect to $u(t)\big|_0^T$ is not given directly but in an indirect form by the relationship between $x(t)$ and $u(t)$ in the form the differential equation (4.21).
2. Possible constraints in the problem should be taken into account.

Similar difficulties occur in the determination of the optimal control for a discrete plant. Now, the determination of the discrete function $u_n \triangleq u(n)$ in the interval $[0, N-1]$ is reduced to the determination of the sequence $u_0$, ... , $u_{N-1}$ Consequently, optimal control problem is reduced to the known problem of a multivariable function minimization. However, the size of the problem may be very large, e.g. for $N = 20$ and the plant with four inputs one should minimize a function with 80 variables. That is why classical mathematical methods are rather not convenient to solve the optimal control problem, and special approaches to handle this problem have been elaborated. Two of them are the most popular:

1. A method based on a *principle of optimality*.
2. A method based on a *maximum principle*.

## 4.3 Principle of Optimality and Dynamic Programming

Let us consider two problems of the optimal control for the plant (4.21) and the local performance index $\varphi(x,u)$:

1. The first problem is the same as in Sect. 4.2, i.e. consists in the determination of $u(t)\big|_0^T$ minimizing the performance index (4.22) for the plant (4.21) and the initial state $x_0$. Let us denote the result by $u^{\mathrm{I}}(t)\big|_0^T$ and the trajectory obtained as a result of this control (i.e. the optimal trajectory) by $x^{\mathrm{I}}(t)\big|_0^T$.

2. The second problem consists in the determination of the control starting at a moment $\bar{t}$ between 0 and $T$, for the initial condition $x(\bar{t}) = x^{\mathrm{I}}(\bar{t})$, which means that the initial state lies on the optimal trajectory obtained as a solution of the first problem. Hence, one should determine $u(t)\big|_{\bar{t}}^T$ minimizing

$$Q_{T-\bar{t}} = \int_{\bar{t}}^{T} \varphi(x,u)dt$$

for the initial state $x(\bar{t}) = x^{\mathrm{I}}(\bar{t})$ and constraints the same as in the first problem. Denote the results by

$$u^{\mathrm{II}}(t)\big|_{\bar{t}}^T \qquad \text{and} \qquad x^{\mathrm{II}}(t)\big|_{\bar{t}}^T .$$

Under some general assumptions usually satisfied in practice, the following property of the optimal control, called the principle of optimality, may be proved:

**Principle of optimality:** For $t \in [\bar{t}, T]$

$$u^{\mathrm{I}}(t)\big|_{\bar{t}}^T = u^{\mathrm{II}}(t)\big|_{\bar{t}}^T \qquad \text{and} \qquad x^{\mathrm{I}}(t)\big|_{\bar{t}}^T = x^{\mathrm{II}}(t)\big|_{\bar{t}}^T .$$

It means that if the control optimal with respect to the initial state $x_0$ is in-

terrupted in the moment $\bar{t}$ and starting from this moment the control optimal with respect to the initial state $x^{\mathrm{I}}(\bar{t})$ is applied, then it will be a control the same as the control $x^{\mathrm{I}}(t)$ for $t \geq \bar{t}$ without the interruption. Then one may continue the interrupted control knowing the current state $x^{\mathrm{I}}(t)$ only and not knowing the earlier control. The application of the principle of optimality to a discrete plant leads to a recursive procedure called *dynamic programming*. In this procedure the calculations start from the end and are executed *step by step* up to $n = 1$. Let us consider the plant (4.15), the performance index (4.16) and the problem without constraints, with a free end of the trajectory. Let us put (4.14) into $\varphi$ in (4.16) and introduce auxiliary notations:

$$\varphi(x_n, u_{n-1}) = \varphi[f(x_{n-1}, u_{n-1}), u_{n-1}] \triangleq g(x_{n-1}, u_{n-1}),$$

$$Q_N = \sum_{n=1}^{N} g(x_{n-1}, u_{n-1}) = \sum_{n=0}^{N-1} g(x_n, u_n), \tag{4.23}$$

$$V_{N-n}(x_n) = \min_{u_n, u_{n+1}, \ldots, u_{N-1}} \sum_{i=n}^{N-1} g(x_i, u_i).$$

For $n = N-1$ we solve the optimal control problem for the initial state $x_{N-1}$, i.e. we minimize the last component of the sum (4.23) with respect to $u_{N-1}$, with $x_{N-1}$ treated as a parameter:

$$V_1(x_{N-1}) = \min_{u_{N-1}} g(x_{N-1}, u_{N-1}).$$

As a result we obtain the relationship $u^*_{N-1} \triangleq \Psi_{N-1}(x_{N-1})$, i.e. the dependence of the last optimal decision upon the current state, denoted by $\Psi_{N-1}$. For two stages from the end we minimize two last components of the sum (4.23) with respect to $u_{N-2}$, taking into account the result of the former minimization and the plant equation (4.14):

$$V_2(x_{N-2}) = \min_{u_{N-2}, u_{N-1}} \{ g(x_{N-2}, u_{N-2}) + g(x_{N-1}, u_{N-1}) \}$$

$$= \min_{u_{N-2}} \{ g(x_{N-2}, u_{N-2}) + V_1(x_{N-1}) \}$$

$$= \min_{u_{N-2}} \{ g(x_{N-2}, u_{N-2}) + V_1[f(x_{N-2}, u_{N-2})] \}.$$

As a result we obtain the relationship $u^*_{N-2} = \Psi_{N-2}(x_{N-2})$, in general different than $\Psi_{N-1}$ because the form of the function to be minimized may differ from the form in the former step. For three stages from the end

$$V_3(x_{N-3}) = \min_{u_{N-3}} \{ g(x_{N-3}, u_{N-3}) + V_2[f(x_{N-3}, u_{N-3})] \},$$

$$u^*_{N-3} = \Psi_{N-3}(x_{N-3}).$$

The *algorithm* for the determination of the optimal control may be presented as the following *recursive procedure*:

$$V_{N-n}(x_n) = \min_{u_n} \{ g(x_n, u_n) + V_{N-n-1}[f(x_n, u_n)] \}, \tag{4.24}$$

$$n = N-1, N-2, \ldots, 0; \qquad V_0 = 0.$$

As the result we obtain the relationships

$$u^*_n = \Psi_n(x_n). \tag{4.25}$$

To determine the numerical values of the control decisions $u_0, u_1, \ldots, u_{N-1}$, it is necessary to execute the second part of calculations, *from the beginning to the end*, using the given value $x_0$ and determining *step by step* the values $x_n$ from the plant equation (4.14):

$$u_0 = \Psi_0(x_0), \qquad x_1 = f(x_0, u_0),$$
$$u_1 = \Psi_1(x_1), \qquad x_2 = f(x_1, u_1),$$
$$\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots$$
$$u_{N-1} = \Psi_{N-1}(x_{N-1}).$$

In this way, a designer can find the *control program* $u_0, u_1, \ldots, u_{N-1}$ which can be executed in the future in an open-loop control system.

The identical decision sequence may be obtained in real time, in a closed-loop system for a measurable plant (which means that the current values $x_n$ can be measured and introduced into the controller). Then the relationship $u_n = \Psi_n(x_n)$ will denote a control algorithm or a formula for the determination of current control decisions using the results of measurement $x_n$ (Fig. 4.1). As in the case considered in Sect. 3.3, the decisions

$u_n$ determined be a designer before starting the control will be the same as the decisions determined currently by a controller, if the plant equation (4.14) describes precisely the real plant, which is assumed in this part of our considerations. Then the value $x_{n+1}$ obtained by substituting the value $u_n$ into the plant equation in the second part of calculations is the same as the value $x_{n+1}$ measured at the output of the plant after putting the decision $u_n$ at its input.

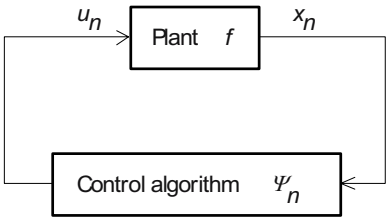


**Fig. 4.1.** Closed-loop control system with measurable plant

In an analogous way the dynamic programming may be applied to the terminal control problem which consists in the determination of the decision sequence $u_0, u_1, \ldots, u_{N-1}$ minimizing the performance index

$$Q_N = \varphi(x_N).$$

Let us introduce auxiliary notations:

$$\varphi(x_n) = \varphi[f(x_{n-1}, u_{n-1})] \stackrel{\Delta}{=} g(x_{n-1}, u_{n-1}),$$

$$V_{N-n}(x_n) = \min_{u_n, u_{n+1}, \ldots, u_{N-1}} g(x_{N-1}, u_{N-1}).$$

For $n = N-1$ we solve the optimal control problem for the initial state $x_{N-1}$, i.e. we minimize $g(x_{N-1}, u_{N-1})$ with respect to $u_{N-1}$, with $x_{N-1}$ treated as a parameter:

$$V_1(x_{N-1}) = \min_{u_{N-1}} g(x_{N-1}, u_{N-1}).$$

As a result we obtain the relationship between the last optimal decision and the current state $u^*_{N-1} \stackrel{\Delta}{=} \Psi_{N-1}(x_{N-1})$. For two stages from the end we minimize $Q_N$ with respect to $u_{N-2}$, taking into account the result of the

former minimization and the plant equation (4.14):

$$V_2(x_{N-2}) = \min_{u_{N-2},\,u_{N-1}} g(x_{N-1}, u_{N-1}) = \min_{u_{N-2}} V_1(x_{N-1})$$

$$= \min_{u_{N-2}} V_1[f(x_{N-2}, u_{N-2})].$$

As a result we obtain the relationship $u^*_{N-2} = \Psi_{N-2}(x_{N-2})$, in general different than $\Psi_{N-1}$ because the form of the function to be minimized may differ from the form in the former step. For three stages from the end

$$V_3(x_{N-3}) = \min_{u_{N-3}} V_2[f(x_{N-3}, u_{N-3})],$$

$$u^*_{N-3} = \Psi_{N-3}(x_{N-3}).$$

The *algorithm* for the determination of the optimal control may be presented as the following *recursive procedure*:

$$V_{N-n}(x_n) = \min_{u_n} V_{N-n-1}[f(x_n, u_n)], \qquad n = N-1, N-2, ..., 0,$$

$$V_0 = g(x_{N-1}, u_{N-1}).$$

As the result we obtain the relationships

$$u^*_n = \Psi_n(x_n),$$

i.e. the *control algorithm in a closed-loop system* (Fig. 4.1). To determine the control program $u_0, u_1, ..., u_{N-1}$ for the control in an open-loop system, it is necessary to execute the second part of calculations, *from the beginning to the end* in a similar way as in the former case with the additive performance index.

**Remark 4.1.** The algorithm (4.25) has been obtained for the problem with a free end of the trajectory and under the assumption that there are no constraints. □

**Remark 4.2.** Let us note that the relationship $u_n = \Psi_n(x_n)$ should be determined in an analytical form presented by formulas in both cases: for the determination of the control program in the open-loop system and for the execution of the current control in the closed-loop system. It follows from the assumption that $x_n$ may be any point from the whole $k$-dimensional real number vector space (in general, from a continuous subspace). In a particular case, if the set of possible states is a finite set, the relationship

$u_n = \Psi_n(x_n)$ may be determined in a form of a table giving the values $u_n$ for all possible values $x_n$. □

The application of dynamic programming to more complicated cases may be found in a reach literature on the optimal control, in particular in the book [3] written by R. Bellman, the author of the dynamic programming.

For the reasons explained at the end of Chap. 3 it has been assumed that there are no external disturbances $z_n$ during the control process. For the plant described by the equation

$$x_{n+1} = f(u_n, z_n),$$

to determine the decisions $u_n$ according to the recursive procedure of the dynamic programming, it would be necessary to know the sequence of disturbances $z_0, ..., z_n, ..., z_{N-1}$. Finally, it is worth noting that (like in Chap. 3) the control problem for the dynamical discrete plant may be considered more generally as a problem of multistage decision making. The dynamic programming is a basic method of the optimization of multistage decisions or – as it is sometimes called – the determination of an optimal decision strategy.

## 4.4 Bellman Equation

The principle of optimality may be also applied to a continuous plant. Let us consider the plant (4.21) and the performance index (4.22) with the local criterion $\varphi(x,u)$. Introduce the following notation:

$$V_{T-t}[x(t)] \triangleq V[x(t), t] = \min_{u(t)|_t^T} \int_t^T \varphi(x,u)dt$$

$$= \min_{u(t)|_t^T} \left[ \int_t^{t+\Delta t} \varphi(x,u)dt + \int_{t+\Delta t}^T \varphi(x,u)dt \right].$$

If the increment $\Delta t$ is small, it may be approximately assumed that the function $\varphi(x,u)$ is constant in the interval $[t, t+\Delta t]$. Then, taking into account the equality

$$\min_{u(t)|_{t+\Delta t}^{T}} \int_{t+\Delta t}^{T} \varphi(x,u)dt = V[x(t+\Delta t), t+\Delta t],$$

we obtain

$$V[x(t), t] = \min_{u(t)} \{\varphi[x(t), u(t)] + V[x(t+\Delta t), t+\Delta t]\} + O_1(\Delta t) \tag{4.26}$$

where $O_1(\Delta t)$ denotes an error, i.e. the difference between the accurate and approximate values. It may be proved that for $\Delta t \to 0$ the error $O_1(\Delta t)$ converges to zero faster than $\Delta t$, i.e.

$$\lim_{\Delta t \to 0} \frac{O_1(\Delta t)}{\Delta t} = 0. \tag{4.27}$$

Under the assumption that the function $x(t)$ is differentiable with respect to $t$ and the function $V(x, t)$ is differentiable with respect to $x$ and $t$, these functions may be expanded in Taylor series. If only the first terms of these series are taken into account, we obtain the following approximate relationship

$$V[x(t+\Delta t), t+\Delta t] \approx V[x(t) + \dot{x}(t)\,\Delta t), t+\Delta t]$$

$$\approx V[x(t), t] + \operatorname*{grad}_{x(t)}^{\mathrm{T}} V[x(t), t]\,\dot{x}(t)\,\Delta t + \frac{\partial V[x(t),t]}{\partial t}\Delta t \tag{4.28}$$

where T denotes the transposition of the gradient. Substituting $f(x, u)$ from the plant equation in the place of $\dot{x}$ and putting (4.28) into (4.26) yields

$$V[x, t] = \min_{u} \{\varphi(x, u) + V(x, t) + \operatorname*{grad}_{x}^{\mathrm{T}} V(x, t)\, f(x, u)\Delta t$$

$$+ \frac{\partial V(x,t)}{\partial t}\Delta t\} + O_2(\Delta t) = V(x, t) + \frac{\partial V(x,t)}{\partial t}\Delta t$$
$$+ \min_{u} \{\varphi(x, u) + \operatorname*{grad}_{x}^{\mathrm{T}} V(x, t)\, f(x, u)\Delta t\} + O_2(\Delta t) \tag{4.29}$$

where $O_2(\Delta t)$ is the error representing all approximations. It may be shown that the property (4.27) is also satisfied by $O_2(\Delta t)$. After dividing of both sides of the equation (4.29) by $\Delta t$ and taking into account the property

(4.27) for $O_2$, one obtains for $\Delta t \to 0$ the following equation, called **Bellman equation**:

$$-\frac{\partial V(x,t)}{\partial t} = \min_u \{\varphi(x, u) + \operatorname{grad}^{\mathrm{T}}_x V(x, t) f(x, u)\}. \tag{4.30}$$

This equation formulates the necessary condition for the optimal control which means that the optimal control $u(t)$ and the respective optimal trajectory $x(t)$ satisfy the equation (4.30). More precisely speaking, if $u^*(t)$ is an optimal control and $x^*(t)$ is an optimal trajectory then for every $t \in [0, T]$

$$u^* = \arg \min_u \{\varphi(x^*, u) + \operatorname{grad}^{\mathrm{T}}_x V(x,t)\big|_{x=x^*} \cdot f(x^*, u)\} \tag{4.31}$$

and

$$-\frac{\partial V(x^*,t)}{\partial t} = \varphi(x^*, u^*) + \operatorname{grad}^{\mathrm{T}}_x V(x,t)\big|_{x=x^*} \cdot f(x^*, u^*). \tag{4.32}$$

If it is *a priori* known that the optimal control exists and is unique then it may be determined by using the equation (4.30). Let us note that this equation does not determine directly the solution $u(t)$, because as a result of minimization of the right hand side of the equation (4.30) with respect to $u$ we may obtain the formula presenting the dependence of $u$ upon $x$ and $V$. The dependence of $u$ upon $x$ might be accepted as a control algorithm in a closed-loop system, but the dependence of $u$ upon $V$ introduces a significant complication in our considerations because $V(x,t)$ may be determined only after the determination of the optimal control $u(t)$ and the corresponding trajectory $x(t)$. That is why effective determination of the optimal control from Bellman equation is in general very difficult and it is possible to obtain analytical results in simple cases only.

Under the assumption that the functions $f$ and $\varphi$ are differentiable, the general procedure of the determination of the control algorithm in the closed-loop system is the following:

1. One should solve with respect to $u$ the equation obtained as a result of the assumption that the gradient of the function in the right hand side of (4.30) is equal to zero vector, i.e. solve the equation

$$\operatorname*{grad}_u \varphi(x, u) + J_u[f(x, u)] \cdot \operatorname*{grad}_x V(x, t) = \bar{0} \tag{4.33}$$

where

$$J_u[f(x,u)] = \begin{bmatrix} \dfrac{\partial f^{(1)}(x,u)}{\partial u^{(1)}} & \dfrac{\partial f^{(2)}(x,u)}{\partial u^{(1)}} & \cdots & \dfrac{\partial f^{(k)}(x,u)}{\partial u^{(1)}} \\ \dfrac{\partial f^{(1)}(x,u)}{\partial u^{(2)}} & \dfrac{\partial f^{(2)}(x,u)}{\partial u^{(2)}} & \cdots & \dfrac{\partial f^{(k)}(x,u)}{\partial u^{(2)}} \\ \cdots & \cdots & \cdots & \cdots \\ \dfrac{\partial f^{(1)}(x,u)}{\partial u^{(p)}} & \dfrac{\partial f^{(2)}(x,u)}{\partial u^{(p)}} & \cdots & \dfrac{\partial f^{(k)}(x,u)}{\partial u^{(p)}} \end{bmatrix}. \tag{4.34}$$

The matrix (4.34) is called a Jacobian matrix of the set of functions $f^{(1)}$, $f^{(2)}$, ..., $f^{(k)}$. Solving the equation (4.33) with respect to $u$ one obtains the dependence of $u$ upon $x$ and $V$.

2. Substituting this relationship into the equation (4.30) without $\min_u$ one obtains the differential equation with the unknown function $V(x,t)$.

3. Putting the solution of this equation into the relationship obtained in the point 1, one obtains the relationship $u = \Psi(x)$.

It is worth noting that if $T = \infty$ then for the stationary plant (4.21) $V$ does not depend directly on $t$. Then $V(x, t) = V(x)$ and the left hand side of the equation (4.30) is equal to zero.

Let us illustrate the above procedure for the linear single-input plant

$$\dot{x} = Ax + bu \tag{4.35}$$

with the quadratic performance index

$$Q_T = \int_0^\infty (x^{\mathrm{T}} Rx + \frac{1}{2}u^2)dt \tag{4.36}$$

where $R$ is a symmetric, positive definite matrix (see (4.6)). Assume that there are no constraints concerning the choice of $u$. Then we have

$$\varphi(x, u) = x^{\mathrm{T}}Rx + \frac{1}{2}u^2, \qquad f(x, u) = Ax + bu, \tag{4.37}$$

$$\operatorname*{grad}_{u} \varphi(x, u) = \frac{\partial \varphi(x,u)}{\partial u} = u\,, \quad J_u[f(x, u)] = b^{\mathrm{T}}. \tag{4.38}$$

Substituting (4.38) into (4.33) yields

$$u + b^{\mathrm{T}} \cdot \operatorname*{grad}_{x} V(x) = 0.$$

Thus,

$$u = -\, b^{\mathrm{T}} \operatorname*{grad}_{x} V(x). \tag{4.39}$$

Putting the formulas (4.37) and (4.39) into the equation (4.30) we obtain the equation with the unknown function $V(x)$. Let us prove that this equation is satisfied by the function

$$V(x) = x^{\mathrm{T}} K x$$

where $K$ is a properly chosen symmetric matrix. Then we have

$$\operatorname*{grad}_{x} V(x) = 2Kx \tag{4.40}$$

and the equation (4.30) takes the form

$$0 = x^{\mathrm{T}} R x + 2(b^{\mathrm{T}} K x)^2 + 2x^{\mathrm{T}} K A x - 4x^{\mathrm{T}} K b \cdot b^{\mathrm{T}} K x$$

or

$$0 = x^{\mathrm{T}} R x + 2x^{\mathrm{T}} K b \cdot b^{\mathrm{T}} K x + x^{\mathrm{T}} (KA + A^{\mathrm{T}} K) x - 4x^{\mathrm{T}} K b \cdot b^{\mathrm{T}} K x.$$

After proper transformation we obtain

$$0 = x^{\mathrm{T}} (R - 2Kb \cdot b^{\mathrm{T}} K + KA + A^{\mathrm{T}} K) x. \tag{4.41}$$

The quadratic form in the right hand side of the equation (4.41) is equal to zero for every $x$ if all entries in the matrix of this form are equal to zero. Consequently, the matrix $K$ must satisfy the equation

$$R + KA + A^{\mathrm{T}} K = 2Kb \cdot b^{\mathrm{T}} K. \tag{4.42}$$

Finally, according to (4.39) and (4.40), the control algorithm is as follows

$$u = -\, 2b^{\mathrm{T}} K x \tag{4.43}$$

where $K$ is the solution of the matrix equation (4.42). The determination of the solution of this equation may be difficult for $k > 2$ and may require the application of proper numerical methods. Let us note that the optimal con-

trol algorithm (4.43) is linear and stationary.

**Example 4.1.** Determine the algorithm of the optimal controller in the closed-loop control system, for the integrating plant described by a transmittance

$$\frac{Y(s)}{U(s)} = \frac{k}{s(sT+1)} \tag{4.44}$$

and the performance index

$$Q_\infty = \int_0^\infty \{\varepsilon^2(t) + [\dot{\varepsilon}(t)]^2 + \frac{1}{2}u^2(t)\}dt$$

where $\varepsilon(t) = y^* - y(t)$ denotes the control error. Using (4.44) we obtain the description of the plant in the form of the differential equation

$$T\ddot{y} + \dot{y} = ku\,,$$

and after introducing the state variables $x^{(1)} = \varepsilon,\ x^{(2)} = \dot{\varepsilon}$ we have the following description

$$\begin{bmatrix} \dot{x}^{(1)} \\ \dot{x}^{(2)} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & -\frac{1}{T} \end{bmatrix} \begin{bmatrix} x^{(1)} \\ x^{(2)} \end{bmatrix} - \begin{bmatrix} 0 \\ \frac{k}{T} \end{bmatrix} u.$$

Then we have

$$A = \begin{bmatrix} 0 & 1 \\ 0 & -\frac{1}{T} \end{bmatrix}, \qquad b = -\begin{bmatrix} 0 \\ \frac{k}{T} \end{bmatrix}, \qquad R = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}. \tag{4.45}$$

Denote the entries of the matrix $K$ to be found by $\alpha, \beta, \gamma$:

$$K = \begin{bmatrix} \alpha & \beta \\ \beta & \gamma \end{bmatrix}. \tag{4.46}$$

Putting (4.45) and (4.46) in the equation (4.42) gives after some transformations

$$\alpha = \frac{\sqrt{1 + 2k^2 + 2\sqrt{2}kT}}{k\sqrt{2}}, \qquad \beta = \frac{T}{k\sqrt{2}},$$

$$\gamma = T\frac{-1+\sqrt{1+2k^2+2\sqrt{2}kT}}{2k^2}.$$

According to (4.43)

$$u = 2\begin{bmatrix} 0 & \frac{k}{T} \end{bmatrix}\begin{bmatrix} \alpha & \beta \\ \beta & \gamma \end{bmatrix}\begin{bmatrix} \varepsilon \\ \dot{\varepsilon} \end{bmatrix} = \frac{2k}{T}(\beta\varepsilon + \gamma\dot{\varepsilon}).$$

Then, as an optimal controller we obtain PD controller with the transmittance

$$K_{\mathrm{R}}(s) = \frac{U(s)}{E(s)} = \frac{2k}{T}(\beta + \gamma s).$$

The transmittance $K_{\mathrm{R}}$ does not depend on $\alpha$. For example, for numerical data $k = 2\sqrt{2}$ and $T = 8$ one obtains $\beta = 2$, $\gamma = 4$ and $K_{\mathrm{R}}(s) = \sqrt{2}(1+2s)$. □

## 4.5 Maximum Principle

In the determination of the optimal control using Bellman equation, the main difficulty is connected with finding the function $V(x, t)$ satisfying the differential equation (4.32). It may be easier and more convenient to use another differential equation with the unknown function $\underset{x}{\mathrm{grad}}\, V(x, t)$. Let us introduce the notation

$$\varphi(x, u) + \underset{x}{\mathrm{grad}}^{\mathrm{T}}\, V(x, t)\, f(x, u) \triangleq \overline{H}(x,u,t) \tag{4.47}$$

and determine the gradient of the function $\overline{H}$ with respect to $x$

$$\begin{aligned} \underset{x}{\mathrm{grad}}\, \overline{H}(x,u,t) &= \underset{x}{\mathrm{grad}}\, \varphi(x, u) + J_{xx}[V(x, t)]\, f(x, u) \\ &\quad + J_x[f(x, u)] \cdot \underset{x}{\mathrm{grad}}\, V(x, t) \end{aligned} \tag{4.48}$$

where $J_x$ is Jacobian matrix of the vector function $f(x,u)$ with respect to $x$ (see (4.34)), and $J_{xx}$ denotes the matrix of second order derivatives of the scalar function $V(x,t)$, i.e.

$$J_{xx}[V(x,t)] = \left[\frac{\partial^2 V(x,t)}{\partial x^{(i)} \partial x^{(j)}}\right]_{\substack{i=1,\ldots,k \\ j=1,\ldots,k}}.$$

If $u^*(t)$ denotes an optimal control and $x^*(t)$ is a respective optimal trajectory then in every moment $t \in [0, T]$ the function $\overline{H}(x, u^*, t)$ takes on its maximal value for $x = x^*$. Then

$$\operatorname*{grad}_x \overline{H}(x, u^*, t)\Big|_{x=x^*} = \overline{0}. \tag{4.49}$$

Using (4.48) and (4.49), and substituting $f(x,u)$ according to the plant equation (4.21), we obtain

$$J_{xx}[V(x,t)]\Big|_{x=x^*} \cdot \dot{x}^*(t) = -\operatorname*{grad}_x \varphi(x,u^*)\Big|_{x=x^*}$$

$$- J_x[f(x,u^*)]\Big|_{x=x^*} \cdot \operatorname*{grad}_x V(x,t)\Big|_{x=x^*}.$$

Let us note that

$$J_{xx}[V(x,t)]\,\dot{x}(t) = \frac{d}{dt}\left[\operatorname*{grad}_x V(x,t)\right].$$

Consequently, we obtain the following differential equation

$$\frac{d}{dt}\left[\operatorname*{grad}_x V(x,t)\Big|_{x=x^*}\right] = -\operatorname*{grad}_x \varphi(x,u^*)\Big|_{x=x^*}$$

$$- J_x[f(x,u^*)]\Big|_{x=x^*} \cdot \operatorname*{grad}_x V(x,t)\Big|_{x=x^*}. \tag{4.50}$$

The necessary condition for the optimal control, analogous to (4.31) and (4.32), may be presented by two relationships:

$$u^* = \arg \min_u \{\varphi(x^*, u) + \operatorname*{grad}_x{}^{\mathrm{T}} V(x,t)\Big|_{x=x^*} \cdot f(x^*, u)\} \tag{4.51}$$

and (4.50). Let us rewrite this condition introducing the notations

$$-\operatorname*{grad}_x V(x,t) \triangleq \psi(t), \qquad -\overline{H}(x,u,t) \triangleq H(x,u,\psi).$$

Then we have

$$H(x,u,\psi) = -\varphi(x,u) + \psi^{\mathrm{T}}(t) f(x,u), \tag{4.52}$$

$$\operatorname*{grad}_x H(x,u,\psi) = -\operatorname*{grad}_x \varphi(x,u) + J_x[f(x,u)]\psi(t), \tag{4.53}$$

and the relationships (4.51), (4.50) are reduced to

$$u^* = \arg\max_u H(x^*, u, \psi^*), \tag{4.54}$$

$$\dot{\psi}^* = -\operatorname*{grad}_x H(x^*, u^*, \psi^*) = \operatorname*{grad}_x \varphi(x,u^*)\Big|_{x=x^*} - J_x[f(x,u^*)]\Big|_{x=x^*} \cdot \psi^*(t). \tag{4.55}$$

The formulas (4.54) and (4.55) present the basic formulation of the **maximum principle**. The function $H(x, u, \psi)$ is called a Hamiltonian, and $\psi(t)$ is called a vector of conjugate variables. The maximum principle may be then formulated as follows:

*If* $u^*(t)$ *is an optimal control and* $x^*(t)$ *is an optimal trajectory then for every* $t \in [0, T]$ *the control* $u^*$ *maximizes Hamiltonian in which* $\psi^*(t)$ *is the function satisfying the differential equation* (4.55).

It is worth noting that the necessary condition for the optimal control presented in the form (4.54) and (4.55) is analogous to that presented in the form (4.31) and (4.32) in Sect. 4.4. The maximum principle was formulated by L. Pontryagin independently of the principle of optimality. The notations used here, in particular – Hamiltonian $H$, were used earlier in optimization problems in theoretical mechanics.

The general procedure of the determination of the control algorithm based on the maximum principle is analogous to that presented for Bellman equation in Sect. 4.4:

1. One should find $u^*(x, \psi)$ maximizing Hamiltonian. On the assumption that the functions $\varphi$ and $f$ are differentiable with respect to $u$, it is reduced to solving the equation

$$\operatorname*{grad}_{u} H(x, u, \psi) = -\operatorname*{grad}_{u} \varphi(x, u) + J_u[f(x, u)]\,\psi = \bar{0}\,. \tag{4.56}$$

with respect to $u$.

2. Substituting $u^*(x, \psi)$ into the equation

$$\dot{\psi} = \operatorname*{grad}_{x} \varphi(x, u) - J_x[f(x, u)]\cdot\psi \tag{4.57}$$

and into the plant equation (4.21), one obtains the set of differential equations with unknown functions $x(t)$ and $\psi(t)$.

3. Putting the solution of this set of equations to the relationship $u^*(x, \psi)$ one obtains the optimal control $u^*(t)$.

In fact, not only using Bellman equation but also using the maximum principle we may have great difficulties in the determination of the optimal control and, except simple cases in which the analytical solution is possible, proper computational procedures are required to determine successive approximations of the optimal solution. Let us note that for finding a particular solution of the set of equations mentioned in part 2 of the procedure, it is necessary to assume proper values $\psi(0)$ and $\psi(T)$. Usually, it is more convenient to find the control algorithm in a closed-loop system (as in Sect. 4.4) because it is not necessary to determine the function $\psi(t)$ in an explicit form. It is sufficient to determine the dependence of $\psi$ upon $x$ from the set of equations for $x(t)$ and $\psi(t)$, and to put this relationship into $u^*(x, \psi)$. In the literature (e.g. [2, 84]) one may find additional necessary conditions for the optimal control (except the conditions presented in the basic formulation of the maximum principle given above), which may be helpful in finding the effective solution.

Let us illustrate the application of the maximum principle using as a typical case a linear single-input plant described by

$$\dot{x} = Ax + bu.$$

For the given values $x_0$ and $x^*$ one should find the control which for every $t$ satisfies the constraint $|u(t)| \le M$, and removing the plant from the state $x_0$ to the required state $x^*$ in minimum time. Now $\varphi(x, u) \equiv 1$, and the maximization of Hamiltonian (4.52) is reduced to the maximization of the function

$$\psi^{\mathrm{T}} f(x, u) = \psi^{\mathrm{T}} Ax + (\psi^{\mathrm{T}} b)u.$$

Taking into account the constraint concerning $u$, one obtains

$$u^* = M \operatorname{sign}(\psi^{\mathrm{T}} b). \tag{4.58}$$

Hence, the optimal control is piecewise constant, equal to $+M$ in the time intervals in which $\psi^{\mathrm{T}}(t)b > 0$, and equal to $-M$ in other time intervals. In the moments when $\psi^{\mathrm{T}}(t)b = 0$, called *switching moments*, $u^*(t)$ changes its value from $+M$ to $-M$ or inversely. For the unique determination of $u^*(t)$ it is necessary to know the initial value $u^*(0)$ and the switching moments. For this purpose one should solve the equation (4.57) which in this case takes a form

$$\dot{\psi}(t) = -A^{\mathrm{T}} \psi(t). \tag{4.59}$$

It is a linear differential equation and for the given initial condition $\psi(0)$ it is possible to obtain its unique solution $\psi^*(t)$, and after substituting it to (4.58), to obtain the final result $u^*(t)$. A difficulty consists here in such a choice of $\psi(0)$ that at the end of the control the given value $x^*$ is achieved. In Example 4.2 it will be shown how to avoid this difficulty, determining the algorithm $u = \Psi(x)$ in a closed-loop system. Let us note that if all eigenvalues of the matrix $A$ are real then the solution of the equation (4.59) has an aperiodic form (without oscillations) and the number of sign changes of the function $\psi^{\mathrm{T}}(t)b$ in the interval $t \in [0, \infty)$ is not grater than $k-1$ where $k$ is an order of the plant. Hence, in this case the number of switches during the optimal control is smaller than $k$. This property is sometimes called a *switching rule*.

**Example 4.2.** Determine a time-optimal controller in a closed-loop system for the plant described by a transmittance

$$\frac{Y(s)}{U(s)} = \frac{k}{s^2}. \tag{4.60}$$

For example, $y(t)$ may denote the position of a moving object. This object should be removed in minimum time to the final position $y^*$ in which the object should be stopped. In other words, one should control the object so as to in minimum time achieve $\varepsilon = 0$ and $\dot{\varepsilon} = 0$ where $\varepsilon = y^* - y$ is a control error. If the executive device is an electrical motor the velocity of which is proportional to the input voltage $u(t)$ and which changes the position $y(t)$, then the control plant with the input $u$ and the output $y$ is an

integrating element. The great inertia of the motor and/or the moving object may be approximately taken into account by assuming the transmittance of a second order integrating element (4.60). Using (4.60) we obtain the plant description in the form $\ddot{y} = ku$ or

$$\dot{x}^{(1)} = x^{(2)}, \qquad \dot{x}^{(2)} = -ku \tag{4.61}$$

where $x^{(1)} = \varepsilon,\ x^{(2)} = \dot{\varepsilon}$. For the practical reason, the constraint $|u(t)| \le M$ should be taken into consideration. According to (4.58), the optimal control takes the value $+M$ or $-M$ in different time intervals. If $u = +M$ then from the equations (4.61) we obtain

$$x^{(2)}(t) = -kut + x_0^{(2)},$$

$$x^{(1)}(t) = -\frac{kut^2}{2} + x_0^{(2)} t + x_0^{(1)}.$$

This is a parametric equation describing the state trajectory for the given $x_0^{(1)}$, $x_0^{(2)}$. Determining $t$ from the first equation and substituting it into the second equation we obtain the trajectory equation in the form

$$x^{(1)} = -\frac{1}{2kM}(x^{(2)})^2 + W(x_0^{(1)}, x_0^{(2)}) \tag{4.62}$$

where $W$ denotes a term depending on the initial state. For $u = -M$

$$x^{(1)} = \frac{1}{2kM}(x^{(2)})^2 - W(x_0^{(1)}, x_0^{(2)}). \tag{4.63}$$

In Fig. 4.2 the families of graphs (4.62) and (4.63) for different initial conditions are presented. The state trajectory for $x^{(1)} = \varepsilon$ and $x^{(2)} = \dot{\varepsilon}$ is called a *phase trajectory*. That is why we say that Fig. 4.2 presents a *phase portrait* of the plant in two cases: for $u(t) = \text{const.} = +M$ and $u(t) = \text{const.} = -M$. The arrows indicate the direction for the increasing time, i.e. for the increasing $t$ the point $[x^{(1)}(t), x^{(2)}(t)]$ moves in the direction indicated by the arrowhead. Through every point on the plane two trajectories run: for $u = +M$ and $u = -M$.

For example, if $P$ denotes the initial state then one should switch $u(t) = +M$ on, change $u$ from $+M$ to $-M$ in the moment when the point $Q$ is reached, and switch the control off ($u = 0$) when the point $(0,0)$ is reached. As a result we obtain the path with minimum time, because, according to the switching rule, one switch at most may occur ($k = 2$).

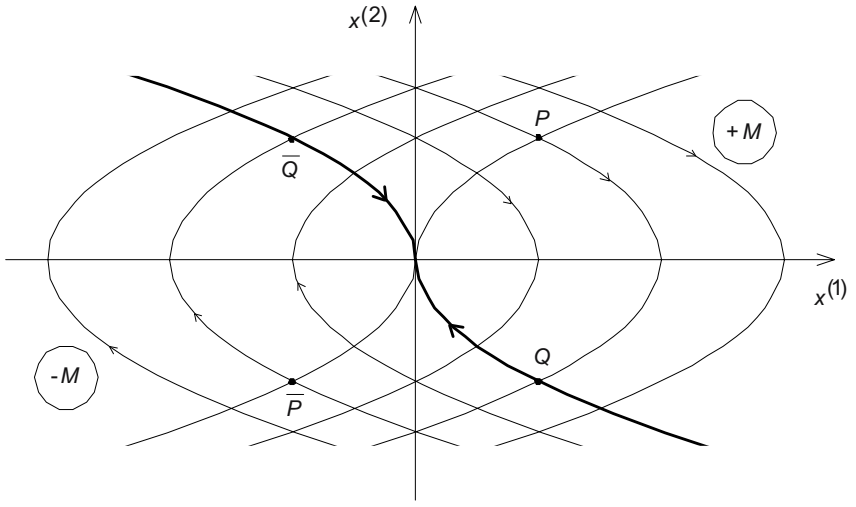


**Fig. 4.2.** Phase trajectories in Example 4.2

Similarly, for the initial point $\overline{P}$ the optimal control is as follows: $u(t) = -M$ until the moment when $\overline{Q}$ is reached, $u(t) = +M$ until the moment when the point $(0,0)$ is reached, and switching off. The bold line in Fig. 4.2 is called a *switching curve*. It divides the state plane (the phase plane) into two parts marked in the figure by $+M$ and $-M$ in circles. If the point $x(t)$ lies in the part $+M$ then $u(t) = +M$ and vice versa. The sign of $u$ has to be changed after reaching the switching curve. Such a control algorithm may be presented by the formula $u = \Psi(x)$. Let us note that the equation describing the switching curve is the following

$$x^{(1)} = -\frac{1}{2kM} |x^{(2)}| x^{(2)}.$$

Hence,

$$u = M \operatorname{sign} (x^{(1)} + \frac{1}{2kM} |x^{(2)}| x^{(2)}) = M \operatorname{sign} (\varepsilon + v) \tag{4.64}$$

where

$$v = \frac{1}{2kM} |\dot{\varepsilon}| \dot{\varepsilon}. \tag{4.65}$$

The relationship (4.64) presents the algorithm of the optimal controller. It consists of three elements: a differentiating element, a nonlinear static element described by (4.65) and a switching element described by $u = M$ sign $w$ where $w = \varepsilon + v$ (Fig. 4.3). For the purpose of a computer implementation it is necessary to present this algorithm in a discrete form. If $T$ is a control period in a discrete control then $y_n = y(nT)$, $\varepsilon_n = \varepsilon(nT)$ and $\dot{\varepsilon}$ is approximately replaced by corresponding increment:

$$\dot{\varepsilon}(t) = \frac{\varepsilon_n - \varepsilon_{n-1}}{T}.$$

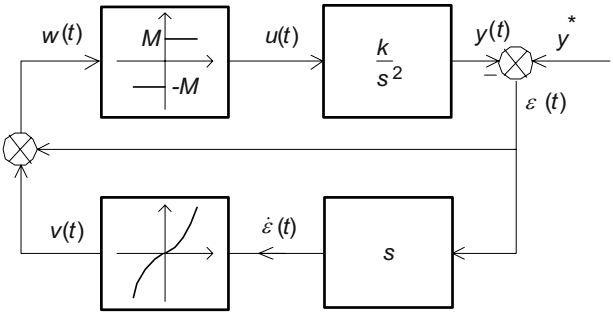


**Fig. 4.3.** Block scheme of the control system under consideration

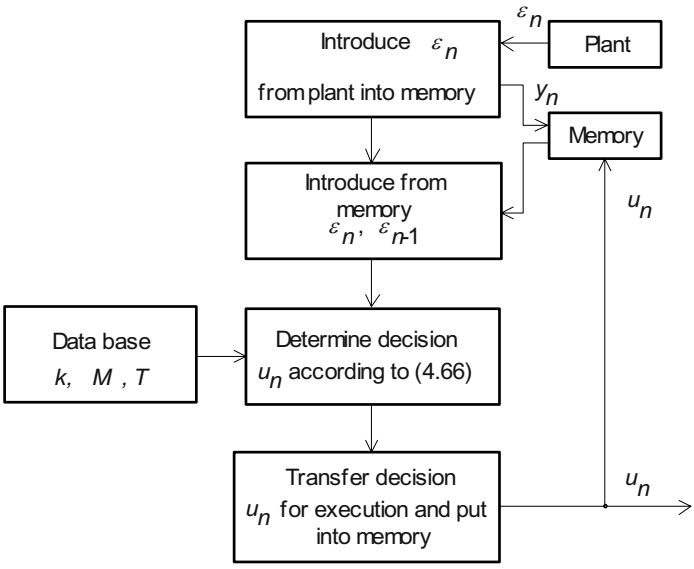


**Fig. 4.4.** Block scheme of the control algorithm in the case under consideration

Consequently, according to (4.64), the discrete control algorithm is the following:

$$u_n = M \operatorname{sign}\left(\varepsilon_n + \frac{1}{2kMT^2}\,|\varepsilon_n - \varepsilon_{n-1}|\,(\varepsilon_n - \varepsilon_{n-1})\right). \tag{4.66}$$

The block scheme of the control algorithm (the program of real-time control), i.e. the procedure of the determination of the decision $u_n$ in the $n$-th period is presented in Fig. 4.4. The remarks concerning functions of the control computer are the same as in Sect. 3.5, referring to Fig. 3.6. □

## 4.6 Linear-quadratic Problem

Optimal control problem for the linear plant and quadratic performance index is shortly called a *linear-quadratic problem*. In Sect. 4.4, the application of Bellman equation to such a problem with the single-input plant (4.35) and the quadratic performance index has been considered. Now, we shall apply maximum principle to a more general problem, with the multi-input plant

$$\dot{x} = Ax + Bu$$

and the performance index

$$Q_T = \int_0^\infty (x^{\mathrm{T}} R_x x + u^{\mathrm{T}} R_u u)\,dt$$

where $R_x$ and $R_u$ are symmetric, positive definite matrices. The equation (4.56) takes now the form

$$-2R_u\,u + B^{\mathrm{T}}\psi = \overline{0}\,.$$

Hence,

$$u^* = \frac{1}{2} R_u^{-1} B^{\mathrm{T}} \psi\,. \tag{4.67}$$

This is a result of the first point of the procedure described in Sect. 4.5. In this case, the set of differential equations for $\psi(t)$ and $x(t)$ mentioned in the second point of this procedure is as follows:

$$\left.\begin{aligned} \dot{\psi} &= 2R_x x - A^{\mathrm{T}}\psi, \\ \dot{x} &= Ax + Bu. \end{aligned}\right\}$$

Putting (4.67) yields

$$\left.\begin{aligned} \dot{\psi} &= 2R_x x - A^{\mathrm{T}}\psi, \\ \dot{x} &= Ax + \frac{1}{2}BR_u^{-1}B^{\mathrm{T}}\psi. \end{aligned}\right\}. \tag{4.68}$$

To obtain the control algorithm in a closed-loop system, one should determine the dependence of $\psi$ upon $x$. Let us remind that in Sect. 4.4 for a single-input plant we assumed $V(x) = x^{\mathrm{T}}Kx$, i.e.

$$-\operatorname*{grad}_x V(x) = \psi = -2Kx \tag{4.69}$$

where $K$ is a symmetric matrix. Now we shall prove that for the properly chosen matrix $K$, the relationship $\psi = -2Kx$ satisfies the set of equations (4.68). Putting (4.69) into (4.68) and multiplying both sides of the second equation by $K$, we obtain

$$\left.\begin{aligned} -K\dot{x} &= R_x x + A^{\mathrm{T}}Kx, \\ K\dot{x} &= KAx - KBR_u^{-1}B^{\mathrm{T}}Kx. \end{aligned}\right\} \tag{4.70}$$

Hence,

$$-(R_x + A^{\mathrm{T}}K)\,x = (KA - KBR_u^{-1}B^{\mathrm{T}}K)\,x. \tag{4.71}$$

Since the equality (4.71) must be satisfied for every $x$, the matrix $K$ must satisfy the equation

$$R_x + KA + A^{\mathrm{T}}K = KBR_u^{-1}B^{\mathrm{T}}K\,. \tag{4.72}$$

Finally, according to (4.67) the control algorithm is as follows:

$$u = -Mx \tag{4.73}$$

where

$$M = R_u^{-1}B^{\mathrm{T}}K\,, \tag{4.74}$$

and $K$ is a solution of the matrix equation (4.72). It is easy to note that for a single-input plant the equation (4.72) and the algorithm (4.73) are reduced to the equation (4.42) and the algorithm (4.43), respectively. It is worth noting that the results (4.72) and (4.73) for a multi-input plant might be also obtained by applying the function $V(x)$ and Bellman equation. The

matrix $M$ should by determined by a designer before the execution of the control. The *design algorithm* consists of the following operations:
1. Determination of the matrix $K$, i.e. solution of the equation (4.72).
2. Determination of the matrix $M$ according to the formula (4.74).

For $k > 2$, finding the solution of the equation (4.72), i.e. determining the values of the entries of $K$ may be difficult. Since $K$ is a symmetric matrix, the matrix equation (4.72) denotes a set of $\frac{k(k+1)}{2}$ nonlinear scalar equations. Solving such a set of equations requires proper numerical (computational) methods. After a determination of the numerical values of the entries of $M$ we obtain a concrete *control algorithm* for the real-time control, i.e. the relationship (4.73).

Let us consider a linear-quadratic problem for the linear non-stationary plant

$$\dot{x} = A(t)x + B(t)u$$

and the performance index

$$Q_T = \int_0^T [x^{\mathrm{T}} R_x(t) x + u^{\mathrm{T}} R_u(t) u]dt .$$

Now we assume $\psi(t) = -2K(t)x(t)$ and put $-2K(t)\dot{x} - 2\dot{K}(t)x$ in the place of $\dot{\psi}$ in the set of equations (4.68). Consequently, the equations analogous to (4.70), (4.71) and (4.72) are the following:

$$\left.\begin{aligned} -K\dot{x} &= R_x x + A^{\mathrm{T}} K x + K\dot{x}, \\ K\dot{x} &= KAx - KBR_u^{-1} B^{\mathrm{T}} K x, \end{aligned}\right\}$$

$$-(R_x + A^{\mathrm{T}}K + \dot{K})\, x = (KA - KBR_u^{-1}B^{\mathrm{T}}K)\, x,$$

$$\dot{K} = -KA - A^{\mathrm{T}}K + KBR_u^{-1}B^{\mathrm{T}}K - R_x , \qquad (4.75)$$

respectively. In general, the control algorithm is then non-stationary

$$u^*(t) = -R_u^{-1}(t)B^{\mathrm{T}}(t)K(t)$$

where $K(t)$ denotes the solution of the matrix differential equation (4.75) which is called Riccati matrix equation. It may be shown that $\psi(T) = 0$ for every $x(T)$. Then, to obtain a particular solution $K(t)$ one should assume the condition $K(T) = \overline{0}$ (zero matrix). If the matrices $A$, $B$, $R_x$ and $R_u$ do not depend on $t$ and $T = \infty$ then $K(t) = const.$ and the differential equation (4.75) is reduced to the algebraic equation (4.72).

# 5 Parametric Optimization

This is the third and the last chapter of the **second part** of the book, devoted to deterministic control problems. In comparison with the problems presented in the previous chapter, now we shall consider the optimization problem with restricted possibilities of decision making by a designer. We shall assume that the form of the control algorithm has been given in advance and the designer should determine the best values of the parameters in this form.

## 5.1 General Idea of Parametric Optimization

Quite often a designer accepts a determined form of the control algorithm with unknown parameters and the problem consists in finding the values of these parameters optimizing the control quality, i.e. minimizing the performance index $Q$. Thus the choice of the optimal control algorithm is restricted to the choice from a class of algorithms determined by the accepted form. The problem of finding the optimal values of parameters in a given form of the control algorithm is called a *parametric optimization*. For a deterministic plant and full information on this plant, one should assume a determined form of the control algorithm if the absolutely optimal algorithm (without a restriction mentioned above) is too difficult to find or to perform. In the case of uncertainties caused by a non-deterministic behaviour of the plant or an incomplete information on the plant, the parameters in the assumed form of the control algorithm may be changed in an adaptation process described in Chap. 11.

Let us denote by $a$ a vector of parameters in the control algorithm, which should be found by minimization of $Q$. For example, in the linear control algorithm considered in Sect. 5.2, the components of the vector $a$ may be all entries of matrices in the description of this algorithm, or only some of them if the rest are fixed and their values are not to be chosen by a designer. The problem of the static optimization for the control system may be considered as a problem of the optimal control for a static plant, presented in Sect. 4.1. The control system to be optimized may be treated

as a static plant where $Q$ is an output $\bar{y}$ and the vector of the parameters $a$ is an input. Of course, it is a discrete control with a long control interval, sufficient for the estimation of $Q$. To find the optimal value $a^*$ it is necessary to determine the function $Q = \Phi(a)$ and then to minimize this function with respect to $a$, taking into account constraints concerning $a$ if there are any. To obtain the function $Q = \Phi(a)$ one should determine functions of time describing the control process, i.e. functions used in the formula defining $Q$, e.g. $\varepsilon(t)$ or $\varepsilon_n$ if the integral or additive performance index evaluates the control error. This is a problem of the control system analysis mentioned in Sect. 2.6.

In Sects. 5.2, 5.3 and 5.4, the control analysis and the parametric optimization for selected cases of linear system will be considered. It will be shown that for linear stationary systems and quadratic performance indexes it is possible to determine $Q$ using operational transform of time functions included into the formula for $Q$ (Laplace transform or $Z$ transform in a continuous or discrete case, respectively).

For nonlinear systems, in very simple special cases only it is possible to obtain an analytical solution of differential or difference equations describing the control process and consequently, to obtain a formula for $Q$. Usually, for the fixed value $a$, only the approximate value of $Q$ may be calculated by applying respective numerical methods. Then $a^*$ is determined by using one of successive approximation methods mentioned in Sect. 4.1, i.e. the successive approximations $a_m$ of the exact result $a^*$ are determined in a way analogous to that presented for $u_m$ in Sect. 4.1. For example, the algorithm analogous to (4.12) has the form

$$a_{m+1} = a_m - Kw_m$$

where $w_m$ denotes an approximate value of the gradient of $Q$ with respect to $a$, in $m$-th step of calculations.

In the formulation and solution of the parametric optimization problem the given assumed form of the control algorithm is used. In practice, this form may be accepted as a result of a designer's experience or an experience of a human operator controlling real plants. We shall return to this problem in the **third part** of the book, in the considerations concerning the control in uncertain systems. Let us note that different given *a priori* forms of the control algorithm with the numerical values of parameters in these forms may be compared by using the performance index $Q$ which may be calculated or obtained as a result of simulations for the known control plant. For the **given control plant** and two assumed forms of the control

algorithm, this form is better for which the minimum value of $Q$ (i.e. the value $Q$ for $a = a^*$) is smaller. In Sect. 5.5 we shall present several frequently used forms of the control algorithm (or forms of a controller) in a closed-loop system: a linear controller (in particular, PID controller) and three nonlinear controllers (including so called fuzzy controller). Let us note that the comparison of these controllers based on $Q$ requires the knowledge of the plant necessary to determine the value $Q$. So, a general statement that e.g. a fuzzy controller is better than a linear PID controller (or on the contrary) has no sense, even when the controllers with their optimal parameters $a^*$ are compared. The result of comparison depends not only on the controller but also on a form of the plant equation and values of its parameters.

The parametric optimization is applied to different cases of the control systems with different forms of the performance index. This is not always the integral form considered above, especially with the limits of integration 0, ∞, which requires a convergence to zero of a function to be integrated. For example, in the case of two-position controller which will be presented in Sect. 5.5, it is easy to note that $\varepsilon(t)$ is a periodic function for $t$ greater than a certain number. Then as a performance index $Q$ we can use an integral of $\varepsilon^2(t)$ or $|\varepsilon(t)|$ for the time interval equal to the period of $\varepsilon(t)$. The amplitude of $\varepsilon(t)$ may also be used as a performance index in this case.

## 5.2 Continuous Linear Control System

Let us consider the parametric optimization problem for the closed-loop control system with a linear plant

$$\dot{x} = A_O x + B_O u, \tag{5.1}$$

$$y = C_O x, \tag{5.2}$$

in which the following linear dynamical control algorithm (linear controller) has been applied:

$$\dot{v} = A_R v - B_R y, \tag{5.3}$$

$$u = C_R v \tag{5.4}$$

where $x$ and $v$ denote the state of the plant and the state of the controller, respectively. In fact, the control error $\varepsilon(t) = y^* - y(t)$ is put at the input of the controller. To simplify the notation we assume $y^* = \overline{0}$ (Fig. 5.1).

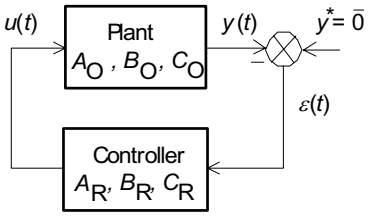


**Fig. 5.1.** Control system under consideration

Substituting (5.2) into (5.3) and (5.4) into (5.1) one obtains the description of the control system as a whole with the state vector $c$

$$\dot{c} = Ac \tag{5.5}$$

where

$$c = \begin{bmatrix} x \\ v \end{bmatrix}, \qquad A = \begin{bmatrix} A_{\mathrm{O}} & B_{\mathrm{O}} C_{\mathrm{R}} \\ -B_{\mathrm{R}} C_{\mathrm{O}} & A_{\mathrm{R}} \end{bmatrix}.$$

**Parametric optimization problem with quadratic performance index**

**Data:** $A_{\mathrm{O}}$, $B_{\mathrm{O}}$, $C_{\mathrm{O}}$ and a symmetric positive definite matrix $R$.

**One should determine:** controller parameters $A_{\mathrm{R}}$, $B_{\mathrm{R}}$, $C_{\mathrm{R}}$ minimizing

$$Q = \int_0^{\infty} (c^{\mathrm{T}} R c) dt \,. \tag{5.6}$$

Hence, one should determine $Q$ as a function of $A_{\mathrm{R}}$, $B_{\mathrm{R}}$, $C_{\mathrm{R}}$, i.e. $Q(A_{\mathrm{R}}, B_{\mathrm{R}}, C_{\mathrm{R}})$, and then minimize this function with respect to $A_{\mathrm{R}}$, $B_{\mathrm{R}}$, $C_{\mathrm{R}}$. A necessary condition for the existence of the integral (5.6) is $\lim c(t) = \bar{0}$ for $t \to \infty$. It is easy to prove that for the linear system under consideration this is also a sufficient condition. The value $Q$ for the data $A_{\mathrm{O}}$, $B_{\mathrm{O}}$, $C_{\mathrm{O}}$, $A_{\mathrm{R}}$, $B_{\mathrm{R}}$, $C_{\mathrm{R}}$ and the given initial state $c_0$ may be determined in two ways: by determining $c(t)$ from the equation (5.5) and then determining $Q$ for the function $c(t)$ obtained, or directly from the definition of $Q$. In the second way there is no need to solve the equation (5.5). Let us apply the second way. We shall prove that

$$Q = c_0^{\mathrm{T}} Q_c c_0 \tag{5.7}$$

where $Q_c$ is a properly chosen symmetric positive definite matrix. If the dependency of $Q$ upon the initial state is described by the formula (5.7)

then, for the fixed moment $t \geq 0$ treated as an initial moment, one may write

$$Q(t) = \int_t^\infty c^{\mathrm{T}}(t) Rc(t) dt = c^{\mathrm{T}}(t) Q_c c(t). \tag{5.8}$$

Differentiating both sides of the equality (5.8) with respect to $t$ yields

$$-c^{\mathrm{T}}(t) Rc(t) = \dot{c}^{\mathrm{T}}(t)\, Q_c c(t) + c^{\mathrm{T}}(t) Q_c \dot{c}(t)$$

and after substituting into (5.5)

$$-c^{\mathrm{T}} Rc = c^{\mathrm{T}}(A^{\mathrm{T}} Q_c + Q_c A)\, c. \tag{5.9}$$

The equality (5.9) is satisfied for any $c$ if and only if

$$-R = A^{\mathrm{T}} Q_c + Q_c A. \tag{5.10}$$

Hence, the performance index $Q$ may be determined according to the formula (5.7) where $Q_c$ is a solution of the matrix equation (5.10). This is a set of equations which are linear with respect to entries of the matrix $Q$ and for $\det A \neq 0$ it is easy to obtain its solution. The problem is simpler for the performance index

$$Q = \int_0^\infty (x^{\mathrm{T}} R_x x + v^{\mathrm{T}} R_v v) dt\,.$$

Then

$$R = \begin{bmatrix} R_x & \bar{0} \\ \bar{0} & R_v \end{bmatrix}.$$

In particular, if the performance index evaluates $y(t)$ and $u(t)$, i.e.

$$Q = \int_0^\infty (y^{\mathrm{T}} R_y y + u^{\mathrm{T}} R_u u) dt \tag{5.11}$$

then, according to (5.2) and (5.4)

$$R_x = C_{\mathrm{O}}^{\mathrm{T}} R_y C_{\mathrm{O}}\,, \qquad R_u = C_{\mathrm{R}}^{\mathrm{T}} R_v C_{\mathrm{R}}\,.$$

For a measurable plant ($y = x$) and the static control algorithm $u = -Mx$, the equation (5.5) is reduced to

$$\dot{x} = Ax, \qquad A = A_{\mathrm{O}} - B_{\mathrm{O}}M.$$

It may be proved that in this case, the result of the parametric optimization is the same as in Sect. 4.6, i.e. the matrix $M$ minimizing $Q$ should be determined according to the formula (4.74) where $K$ is a solution of the matrix equation (4.72). We often consider and estimate the control process under the assumption that until the moment $t = 0$ the control system was in an equilibrium state (for the description (5.1) – (5.4) it means that the initial state $c_0 = \overline{0}$) and in the moment $t = 0$ the required output value has been changed from $\overline{0}$ to a value $y^*$ constant during the time of the control. Then in (5.3) we introduce $\varepsilon = y^* - y$ instead of $y$ and the description of the control system is as follows:

$$\dot{c} = Ac + \begin{bmatrix} \overline{0} \\ B_{\mathrm{R}} \end{bmatrix} y^*. \tag{5.12}$$

Under the assumption $\det A \neq 0$ we introduce a new variable

$$\overline{c} \triangleq c + A^{-1}\begin{bmatrix} \overline{0} \\ B_{\mathrm{R}} \end{bmatrix} y^*.$$

Consequently, (5.12) takes the form $\dot{\overline{c}} = A\overline{c}$. The formulation and solution of the parametric optimization problem are then such as for (5.5) with the initial state

$$\overline{c}_0 = c_0 + A^{-1}\begin{bmatrix} \overline{0} \\ B_{\mathrm{R}} \end{bmatrix} y^* = A^{-1}\begin{bmatrix} \overline{0} \\ B_{\mathrm{R}} \end{bmatrix} y^*.$$

Now $\overline{c}$ instead of $c$ occurs in the index (5.6), and the index (5.11) has the form

$$Q = \int_0^\infty (\varepsilon^{\mathrm{T}} R_\varepsilon \varepsilon + u^{\mathrm{T}} R_u u)dt\,. \tag{5.13}$$

Hence, if $\det A \neq 0$ then for the system (5.5) one may assume $y^* = \overline{0}$ without a loss of generality. If the control system as a whole with the input $y^*(t)$ and the output $y(t)$ is controllable and observable, or (which is usually satisfied) we are interested in controllable and observable part of the system only (i.e. we want to know the change of the output $y(t)$ caused by

the disturbance $y^*(t)$), then it is sufficient to use the description with the help of transmittances

$$Y(s) = K_O(s)U(s),$$

$$U(s) = K_R(s)E(s), \qquad E(s) = Y^*(s) - Y(s)$$

where $K_R$ is an assumed form of the controller transmittance with parameters to be determined by a designer. Then

$$E(s) = [K_O(s)K_R(s) + I]^{-1}Y^*(s)$$

(see (2.22)).

Applying an inverse transformation for $Y^*(s) = \frac{y^*}{s}$ we obtain the control error $\varepsilon(t)$ in the situation considered (zero initial conditions and a step change of the required output value). If in the index (5.13) $R_u = \overline{0}$, i.e. only the function $\varepsilon(t)$ is evaluated, then one should determine $Q$ for the obtained function $\varepsilon(t)$ and minimize $Q$ with respect to the parameters in the assumed form of the controller transmittance $K_R$. To take into account the second component in (5.13) it is necessary to determine $u(t)$ as an inverse transform of the function $K_R(s)E(s)$. It is also possible to determine $Q$ directly from the functions $E(s)$ and $U(s)$, without finding the functions $\varepsilon(t)$ and $u(t)$. According to *Parseval's Theorem*, if $\varepsilon(t) = u(t) = \overline{0}$ for $t < 0$ then

$$Q = \frac{1}{2\pi}\int_{-\infty}^{\infty}[E^T(j\omega)R_\varepsilon E(-j\omega) + U^T(j\omega)R_u U(-j\omega)]d\omega. \qquad (5.14)$$

In particular, for one-dimensional plant ($p = l = 1$) and the performance index evaluating $\varepsilon(t)$ only, we have

$$Q = \int_0^{\infty}\varepsilon^2(t)dt = \frac{1}{2\pi}\int_{-\infty}^{\infty}|E(j\omega)|^2\, d\omega. \qquad (5.15)$$

Let us be reminded that the components of the vectors $E(s)$ and $U(s)$ are rational functions. For such functions, the formulas for $Q$ in the case of small (sufficient in practice) degrees of polynomials in numerators and denominators of the particular components have been determined. For example, for

$$E(s) = \frac{b_2 s^2 + b_1 s + b_0}{s^3 + a_2 s^2 + a_1 s + a_0}$$

the formula (5.15) is as follows:

$$Q = \int_0^\infty \varepsilon^2 dt = \frac{b_2^2 a_0 a_1 + (b_1^2 - 2b_0 b_2) a_0 + b_0^2 a_2}{2a_0(a_1 a_2 - a_0)}. \tag{5.16}$$

If except the disturbance $y(t) = \mathbf{1}(t)y^*$, the step disturbance $z(t) = \mathbf{1}(t)z^*$ acts on the plant, then we determine $E(s)$ from the formula (2.24) for $Y^*(s) = \frac{y^*}{s}$ and $Z(s) = \frac{z^*}{s}$. Next, the index $Q$ and the optimal controller parameters should be determined in the way described above.

Finally, let us note that, except the one-dimensional case, a practical utility of the parametric optimization is rather limited. According to the formulas (5.7) and (5.14), the optimal parameters of the controller in a closed-loop system (in other words, parameters of the optimal controller) in general depend on $c_0$ (for an unmeasurable plant) or on $y^*$ and $z^*$. Consequently, they should be changed according to successive disturbances with different values $c_0$, or $y^*$ and $z^*$.

**Example 5.1.** For the second order plant

$$K_O(s) = \frac{k_O}{(sT_1 + 1)(sT_2 + 1)}, \qquad T_{1,2} > 0,$$

let us assume the integrating controller

$$K_R(s) = \frac{k_R}{s}$$

and find the optimal value $k_R$ for the requirement $y^*(t) = 0$, the disturbance acting on the plant $z(t) = \mathbf{1}(t)z^*$ and the performance index (5.15). The function $E(s)$ is as follows:

$$E(s) = \frac{-z^* K_O}{s(1 + K_O K_R)} = \frac{-z^* k_O}{T_1 T_2 s^3 + (T_1 + T_2)s^2 + s + k}$$

where $k = k_O k_R$. Then $b_1 = b_2 = 0$,

$$b_0 = -\frac{z^* k_O}{T_1 T_2}, \quad a_2 = \frac{T_1 + T_2}{T_1 T_2}, \quad a_1 = \frac{1}{T_1 T_2}, \quad a_0 = \frac{k}{T_1 T_2}.$$

After substituting of these variables into (5.16) and some transformations we obtain

$$Q = \frac{(z^* k_O)^2 (T_1 + T_2)}{2k(T_1 + T_2 - kT_1 T_2)}.$$

It is easy to note that

$$k_{opt} = \arg \min_k Q = \arg \max_k k(T_1 + T_2 - kT_1 T_2) = \frac{T_1 + T_2}{2T_1 T_2}.$$

The optimal controller parameter $k_R = \frac{k_{opt}}{k_O}$. The condition assuring $Q < \infty$, i.e. $\varepsilon(t) \to 0$ for $t \to \infty$ is the following:

$$0 < k < \frac{T_1 + T_2}{T_1 T_2}.$$

Assume that we have the first order plant, e.g. $T_2 = 0$. Then

$$Q = \frac{(z^* k_O)^2}{2k}$$

and the optimization problem without a constraint for $k$ has no solution, i.e. $Q$ may be arbitrary small if $k_R$ is respectively great. For the constraint $k_R \le \bar{k}_R$ (which indirectly means a constraint concerning $\dot{u}(t)$), the optimal parameter of the controller $k_R = \bar{k}_R$. □

## 5.3 Discrete Linear Control System

Considerations for the discrete plant are analogous to those for the continuous plant presented in Sect. 5.2. The relationships corresponding to the equations (5.1) – (5.5) are now as follows:

$$x_{n+1} = A_O x_n + B_O u_n,$$

$$y_n = C_O x_n,$$

$$v_{n+1} = A_R v_n - B_R y_n,$$

$$u_n = C_R v_n,$$

$$c_{n+1} = A c_n$$

where

$$c_n = \begin{bmatrix} x_n \\ v_n \end{bmatrix}, \qquad A = \begin{bmatrix} A_O & B_O C_R \\ -B_R C_O & A_R \end{bmatrix}.$$

**Parametric optimization problem with quadratic performance index**
**Data:** $A_O, B_O, C_O, R$.
**One should determine:** controller parameters $A_R, B_R, C_R$ minimizing

$$Q = \sum_{n=0}^{\infty} c_n^T R c_n \ .$$

Assume

$$Q = c_0^T Q_c c_0 \tag{5.17}$$

where $Q_c$ is a properly chosen symmetric positive definite matrix. The relationships analogous to (5.8) – (5.10) are as follows:

$$Q_i = \sum_{n=i}^{\infty} c_n^T R c_n = c_i^T Q_c c_i$$

$$Q_{i+1} - Q_i = -c_i^T R c_i = c_{i+1}^T Q_c c_{i+1} - c_i^T Q_c c_i \,,$$

and after substituting $c_{i+1} = A c_i$

$$-c_i^T R c_i = c_i^T (A^T Q_c A - Q_c) c_i$$

and finally

$$-R = A^T Q_c A - Q_c \,. \tag{5.18}$$

Thus, we determine $Q$ from the formula (5.17) in which $Q$ is a solution of

the matrix equation (5.18).

If the description with the help of transmittances is used then for the step disturbance $y_n^* = \mathbf{1}(n)y^*$ we find

$$E(z) = [K_O(z)K_R(z) + I]^{-1} \frac{y^* z}{z-1}$$

and

$$U(z) = K_R(z)E(z).$$

In a way analogous to that in the continuous case, the quadratic performance index

$$Q = \sum_{n=0}^{\infty} (\varepsilon_n^T R_\varepsilon \varepsilon_n + u_n^T R_u u_n) \tag{5.19}$$

may be determined directly from the functions $E(z)$ and $U(z)$, without determining the functions $\varepsilon_n$ and $u_n$. It may be proved that

$$Q = \frac{1}{2\pi} \int_{-\pi}^{\pi} [E^T(e^{j\omega})R_\varepsilon E(e^{-j\omega}) + U^T(e^{j\omega})R_u U(e^{-j\omega})]d\omega . \tag{5.20}$$

This is a discrete version of Parseval's Theorem (5.14). In particular, for one-dimensional plant and the performance index evaluating $\varepsilon_n$ only

$$Q = \sum_{n=0}^{\infty} \varepsilon_n^2 = \frac{1}{2\pi} \int_{-\pi}^{\pi} |E(e^{j\omega})|^2 \, d\omega . \tag{5.21}$$

The integral (5.21) has been determined for small degrees of polynomials in numerator and denominator of $E(z)$, in a similar way as the integral in (5.15). However, the respective formulas are now much more complicated than in the continuous case. They may be found in books devoted to the theory of linear discrete or sampled-data control systems (e.g. [102]).

## 5.4 System with the Measurement of Disturbances

In Chaps. 3 and 4 we noted that if during the control process some disturbances act on the plant then for the determining of a proper control in an open-loop system or a control algorithm in a closed-loop system, the knowledge of the functions describing these disturbances is required by a

designer before starting the control. In the case of parametric optimization considered in this chapter, we can determine optimal parameters in the control algorithm when the disturbances are measured. It means that for the assumed form of the control algorithm we determine the parameters which are optimal for the current knowledge of the disturbances, i.e. are optimal in the situation when to determine the control decision in a fixed moment we use the knowledge of the disturbances until this moment only and not the knowledge of their future values. The control may be performed in an open-loop system with the measurements of disturbances (Fig. 5.2) or in a combined system in which the disturbance $z$ as well as the control error are put at the input of the controller. In the first case $Y(s)$ should be determined from the equations

$$Y(s) = K_O(s)U(s) + K_Z(s)Z(s), \tag{5.22}$$

$$U(s) = K_S(s)Z(s). \tag{5.23}$$

Then

$$Y(s) = [K_O(s)K_S(s) + K_Z(s)]Z(s). \tag{5.24}$$

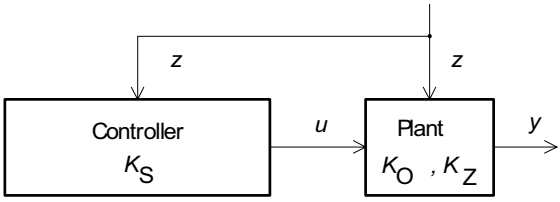


**Fig. 5.2.** Block scheme of open-loop control system

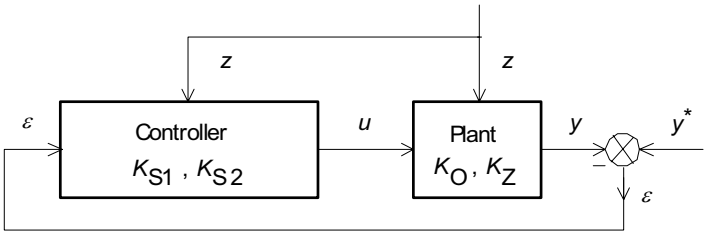


**Fig. 5.3.** Block scheme of combined control system

For the step disturbance $z(t) = \mathbf{1}(t)z^*$, i.e. $Z(s) = \dfrac{z^*}{s}$ we find $U(s)$ and $Y(s)$ according to (5.23) and (5.24), respectively and then we determine the per-

formance index as a function of parameters of the transmittance $K_S$ for the given matrices $R_y$ and $R_u$:

$$Q = \frac{1}{2\pi} \int_{-\infty}^{\infty} [Y^T(j\omega) R_y Y(-j\omega) + U^T(j\omega) R_u U(-j\omega)] d\omega .$$

This is the performance index (5.13) under the assumption $y^* = 0$. The optimal values of the parameters of $K_S$ are obtained as a result of minimization of $Q$ with respect to these parameters. Let us note that if

$$K_S(s) = -K_O(s)^{-1} \cdot K_Z(s)$$

then $y(t) \equiv 0$. Consequently, for $R_u = \overline{0}$, the performance index $Q = 0$. Unfortunately, the transmittance of the controller $K_S(s)$ determined in such a way may be proved to be physically unrealizable. Then it is reasonable to propose a physically realizable transmittance with undetermined parameters, and to choose optimal values of its parameters in a way shown above. In a particular case when the matrices $K_O$, $K_Z$, $K_S$ are reduced to scalar, one-dimensional transmittances (i.e. the plant has two single inputs $u$ and $z$, and one output $y$), the transmittance assuring $y(t) = 0$ for $t \geq 0$, i.e. the full compensation of the influence of $z$ on $y$, has the following form:

$$K_S(s) = \frac{L_Z(s) M_O(s)}{L_O(s) M_Z(s)}$$

where $L_Z(s)$, $M_Z(s)$, $L_O(s)$ and $M_O(s)$ denote polynomials in numerator and denominator of the transmittances $K_Z(s)$ and $K_O(s)$, respectively. The transmittance $K_S$ is physically realizable, i.e. the full compensation of the influence of $z$ on $y$ is possible, if the sum of the degrees of polynomials $L_Z$ and $M_O$ is smaller than the respective sum for polynomials $L_O$ and $M_Z$.

The combined system is described by the set of equations (5.22) and

$$U(s) = K_{S1}(s)Z(s) + K_{S2}(s)E(s), \qquad E(s) = Y^*(s) - Y(s). \tag{5.25}$$

Then, after some transformations we obtain

$$E(s) = [K_O(s)K_{S2}(s) + I]^{-1} \{Y^*(s) - [K_O(s)K_{S1}(s) + K_Z(s)]Z(s)\}. \tag{5.26}$$

For the step disturbances $Y^*(s) = \frac{y^*}{s}$, $Z(s) = \frac{z^*}{s}$, we determine $E(s)$ according to the formula (5.26) and $U(s)$ according to the formula (5.25), and next we find the performance index (5.14) which is minimized with respect to parameters of the functions $K_{S1}$ and $K_{S2}$.

The considerations and solutions for a discrete system are analogous to those for a continuous system. The equations for the discrete transforms and transmittances are the same as (5.22) – (5.26) with the variable $z$ in the place of $s$, and the performance index (5.19) is calculated by using the formula (5.20). In the case of an open-loop system $y_n \equiv 0$ for

$$K_S(z) = -K_O(z)^{-1} K_Z(z).$$

The full compensation may be obtained if the discrete transmittance determined in such a way is physically realizable. In particular, for one-dimensional case

$$K_S(z) = \frac{L_Z(z) M_O(z)}{L_O(z) M_Z(z)}$$

where $L_Z$, $M_Z$, $L_O$ and $M_O$ denote polynomials in numerator and denominator of the transmittances $K_Z(z)$ and $K_O(z)$, respectively. The full compensation is possible if the sum of the degrees of polynomials $L_Z$ and $M_O$ is not greater than the respective sum for polynomials $L_O$ and $M_Z$. Let us note that in general the considerations are concerned with multi-dimensional system, i.e. in the equation presented above for the continuous as well as for the discrete plant appear vectors and matrices. The problem is considerably simplified for one-dimensional system, i.e. for a system in which $z(t)$, $u(t)$ and $y(t)$ are single variables (scalars). It is also worth noting that in general we do not minimize $Q$ with respect to all parameters of the control algorithm, but only with respect to parameters which may be chosen by a designer, with fixed values of other parameters.

## 5.5 Typical Forms of Control Algorithms in Closed-loop Systems

We shall present shortly several typical forms of control algorithms often used in closed-loop systems, with undetermined parameters. For an assumed form, the optimal values of the parameters may be determined by a

designer solving the parametric optimization problem with the complete knowledge of the plant, or may be changed in an adaptation process which will be considered in Chap. 11. For simplicity, let us assume that it is possible to put the state vector $x(t)$ at the input of the controller and for single-input and single-output plant this is a vector with the components

$$x^{(1)}(t) = \varepsilon(t), \quad x^{(2)}(t) = \dot{\varepsilon}(t), \quad \ldots, \quad x^{(k)}(t) = \varepsilon^{(k-1)}(t), \tag{5.27}$$

or

$$x(t) = \begin{bmatrix} \varepsilon(t) \\ \dot{\varepsilon}(t) \\ \cdot \\ \cdot \\ \cdot \\ \varepsilon^{(k-1)}(t) \end{bmatrix}$$

where $\varepsilon^{(k-1)}(t)$ denotes the $(k-1)$-th derivative of the function $\varepsilon(t)$. The descriptions presented below concern single-output controllers.

### 5.5.1 Linear Controller

In the linear controller $u(t)$ is a linear combination of components of the vector $x$

$$u = a_1 x^{(1)} + a_2 x^{(2)} + \ldots + a_k x^{(k)} = a^{\mathrm{T}} x \tag{5.28}$$

where $a^{\mathrm{T}} = [a_1 \;\; a_2 \;\; \ldots \;\; a_k]$. In the case (5.27)

$$u = a_1 \varepsilon + a_2 \dot{\varepsilon} + \ldots + a_k \varepsilon^{(k-1)}. \tag{5.29}$$

The parameters $a_1, a_2, \ldots, a_k$ take a role of weight coefficients. For $k=2$ it is PD controller. For $k > 2$ instead of the form (5.29), PID controller, i.e. the form

$$u(t) = a_1 \varepsilon(t) + a_2 \dot{\varepsilon}(t) + a_3 \int_0^t \varepsilon(t)dt$$

is often used.

### 5.5.2 Two-position Controller

Now

$$u = M \operatorname{sign} a^{\mathrm{T}} x,$$

and more generally

$$u = M \operatorname{sign}(a^{\mathrm{T}} x + b). \tag{5.30}$$

The parameter $b$ may be called a *threshold*. The decision $u$ may take two values only: $u = +M$ if the value $a^{\mathrm{T}} x$ is greater than the threshold $-b$, and $u = -M$ otherwise. The controller (5.30) is a special case of a two-position controller

$$u = M \operatorname{sign} f(x)$$

where the function $f(x)$ has a fixed form. Such a controller occurred in Example 4.2 where

$$u = M \operatorname{sign} (\varepsilon + \frac{1}{2kM} |\dot{\varepsilon}| \dot{\varepsilon}).$$

In the simplest case $u = M \operatorname{sign} \varepsilon$, which means that the control decision depends on the sign of the control error only, but does not depend on its value.

### 5.5.3 Neuron-like Controller

The form of the controller (5.30) may be generalized to the form

$$u = f(a^{\mathrm{T}} x + b) \tag{5.31}$$

where $f$ is a fixed function of the variable $a^{\mathrm{T}} x + b$. It is useful and reasonable to apply a complex algorithm, which is a system consisting of the elements (5.31). In such a way one may improve possibilities of *fitting* the controller to the plant because the number of parameters which may be chosen is greater than in the simple case (5.31). These are parameters $a$ and $b$, i.e. the weights and the thresholds in all elements being parts of the controller. Usually the structure of the controller has a multi-layer (or multi-level) form, i.e. the elements are composed into layers, the inputs of the first layer elements are the inputs of the controller (components of the vector $x$), the inputs of successive layers are the outputs of former layers, and the outputs of the last layer are the outputs of the whole controller (components of the vector $u$). A structure of the complex controller with

three layers and three state variables is illustrated in Fig. 5.4. For example, in the second layer there are three elements; $f_{ij}$ denote the function $f$ in the $j$-th element of the $i$-th layer.

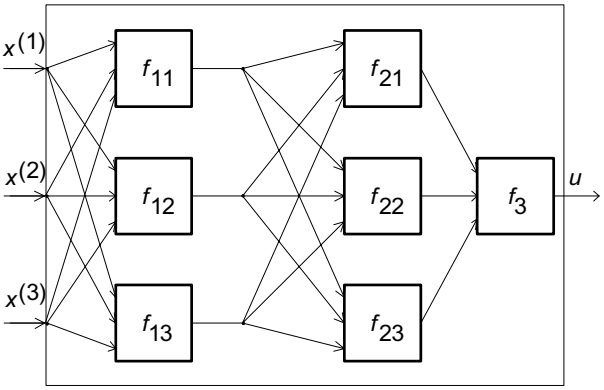


**Fig. 5.4.** Structure of controller under consideration

### 5.5.4 Fuzzy Controller

In this case a control algorithm is determined by a given set of functions:

$$\left.\begin{array}{l} \mu_{xi}(x^{(i)}) \ , \quad i = 1,2,...,k, \\ \mu_{uj}(u) \ , \quad j = 1,2,...,l. \end{array}\right\} \tag{5.32}$$

Each function in this set takes non-negative values and its maximum value is equal to 1. The control algorithm is defined in the following way:

$$u = \Psi(x) = \frac{\int_{-\infty}^{\infty} u\mu(u;x)du}{\int_{-\infty}^{\infty} \mu(u;x)du} \tag{5.33}$$

where

$$\mu(u;x) = \max_j [\mu_j(u;x)], \tag{5.34}$$

$$\mu_j(u;x) = \min\{\mu_{uj}(u), \mu_x(x)\}, \qquad j = 1, 2, ..., l, \tag{5.35}$$

$$\mu_x(x) = \min\{\mu_{x1}(x^{(1)}), \mu_{x2}(x^{(2)}), ..., \mu_{xk}(x^{(k)})\}. \tag{5.36}$$

Minimum in the formulas (5.35) and (5.36) denotes the least number from the set in brackets. The control algorithm performed by the fuzzy controller can be presented in the form of the following procedure:

1. Introducing the values $x^{(1)}, x^{(2)}, ..., x^{(k)}$ from the plant.
2. Finding the number $\mu_x(x)$ according to (5.36).
3. Determining the function $\mu_j(u; x)$ according to (5.35) for $j = 1, 2, ..., l$.
4. Determining the function $\mu(u; x)$ according to (5.34).
5. Finding the value of $u$ according to the formula (5.33).

In this way the relationship $u = \Psi(x)$ is determined. The block scheme of the control algorithm (or a structure of the fuzzy controller) is presented in Fig. 5.5. The left hand side blocks represent the given functions uniquely determining the relationship $u = \Psi(x)$. The right hand side blocks denote the procedure defined by the respective numbers.

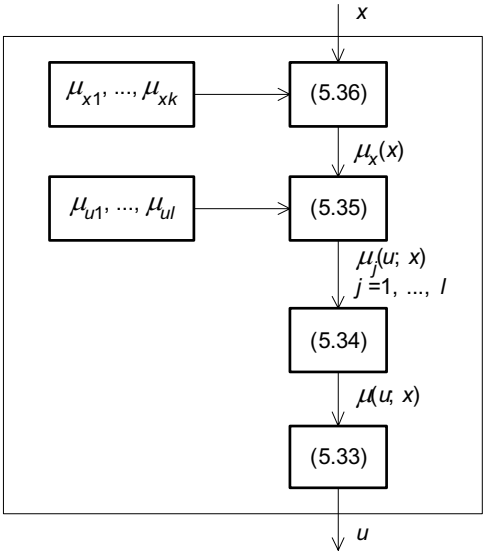


**Fig. 5.5.** Block scheme of control algorithm in fuzzy controller

The controller parameters are here the parameters of the functions $\mu$ in the set (5.32). The integrals in the formula (5.33) may be calculated approximately, by using sums instead of integrals:

$$u \approx \frac{\sum_{m=-\infty}^{\infty} u_m \, \mu(u_m; x)}{\sum_{m=-\infty}^{\infty} \mu(u_m)}$$

where $u_m = m \cdot \Delta_u$ and $\Delta_u$ is sufficiently small.

The simplest versions of *neuron-like controller* and *fuzzy controller* have been described in this section. The names of these controllers and their interpretations will be presented in Chap. 9 for the fuzzy case and in Chap. 12 where applications of so called neural networks in control systems will be described. In Chap. 9 we shall see that, in general, the set of functions $\mu_{xi}(x^{(i)})$ is different for different $j$. Consequently, the function $\mu_x(x)$ determined according to the formula (5.36) also depends on $j$.

The forms of control algorithms $u = \Psi(x)$ listed in 5.5.1, 5.5.2, 5.5.3 and 5.5.4 are identical for continuous and discrete systems, because we considered memory-less controllers. It is a relationship between $u$ and $x$ for every $t$ in a continuous case or for every $n$ in a discrete case. In the discrete case, the state variables analogous to (5.27) are as follows

$$x_n^{(1)} = \varepsilon_n, \quad x_n^{(2)} = \varepsilon_{n-1}, \ldots, \quad x_n^{(k)} = \varepsilon_{n-(k-1)} \tag{5.37}$$

or

$$x_n = \begin{bmatrix} \varepsilon_n \\ \varepsilon_{n-1} \\ \cdot \\ \cdot \\ \cdot \\ \varepsilon_{n-(k-1)} \end{bmatrix}.$$

It is worth noting that in the case (5.27) and in the case (5.37) as well, the state variables of an observable part of the plant are defined.

# 6 Application of Relational Description of Uncertainty

The **second part** of the book containing Chaps. 3, 4 and 5 has been devoted to control problems with full information of a deterministic plant. Now we start considerations concerning the control under uncertainty or – as it is often called – the control in uncertain systems. The problems of decision making and control under uncertainty are very frequent in real situations and that is why methods and algorithms concerning uncertain systems are very important from the practical point of view. There exists a great variety of definitions and formal models of uncertainties and uncertain systems [52, 81, 82, 86, 96]. Analysis and decision making problems are formulated in a way adequate to the applied description of the uncertainty, i.e. so that the problem has a practical sense accepted by a user and may be solved by using the assumed model of the uncertainty.

The **third part** of the book contains Chaps. 6, 7, 8 and 9 in which we present different cases of the problem formulations and the determinations of control algorithms based on descriptions of uncertainty given in advance, without using any additional information obtained during the control process. In this chapter, the plants and consequently, the control algorithms and the uncertain control systems will be described by relations which are not reduced to functions.

## 6.1 Uncertainty and Relational Knowledge Representation

As it was mentioned in the introduction to Chap. 3, there are two basic reasons of the uncertainty in the decision making:
1. The plant with a fixed input and output (a state) is non-deterministic.
2. There is no full information of the plant.

Ad 1. Let us consider a static plant with the input vector $u \in U$ and the output vector $y \in Y$. We say that the plant with the input $u$ and the output $y$ acts (or behaves) in deterministic way, or shortly, the plant is *deterministic* if the value $u$ determines (i.e. uniquely defines) the value $y$. It means that in the same conditions the decision $u$ gives always the same effect $y$,

or that the plant is described by a function $y = \Phi(u)$.

For example, let $y$ denote the amount of a product in one cycle of a production process and $u$ denote the amount of a resource (e.g. the amount of a raw material), and $y = ku$. It means that if the value $u$ will be the same in different cycles then in each cycle we obtain the same amount of the product $y = ku$. If the parameter $k$ varies in successive cycles and $u_n$, $y_n$, $k_n$ denote the values in the $n$-th cycle then $y_n = k_n u_n$, which means that the amount of the product $y_n$ is uniquely determined by the amount of the raw material $u_n$ and the value of the coefficient $k_n$, or – when the sequence $k_n$ is determined in advance – by the index of a cycle. The coefficient $k_n$ may be treated as a second input, i.e. a disturbance $z_n$. In the case when the disturbances occur in the description of the plant, we can say that the plant with the fixed input $(u, z)$ and the output $y$ is deterministic if the values $(u, z)$ uniquely determine the value $y$, i.e. the plant is described by a function $y = \Phi(u, z)$. Such plants have been considered in Sects. 3.1 and 4.1. As it has been shown above, a non-deterministic plant with the input $u$ and the output $y$ may be proved to be a deterministic one if other inputs which also have an influence on $y$ are taken into account. In practice to introduce or even to call them may be impossible. Consequently, for the fixed $u$ only a set of possible outputs may be given. In the example considered, the amount of the product may depend not only on the amount of the raw material but also on many other variables and for the fixed $u$, only the set of possible values of $y$ may be given, e.g. in the form of the inequality $c_1 u \le y \le c_2 u$ with the given values $c$, which means that $c_1 \le k \le c_2$. In different production cycles one may obtain different values $y_n$ for the same value $u_n$. Hence, the different pairs $(u_n, y_n)$ are possible in our plant. A set of examples of such points is illustrated in Fig. 6.1 where the shaded domain is a set of all possible points. Of course, the figure concerns a general plant of this kind, in which negative values $u$ and $y$ are possible, i.e. the plant described by the inequalities

$$c_1 u \le y \le c_2 u$$

for $u \ge 0$ and

$$c_2 u \le y \le c_1 u$$

for $u \le 0$, under the assumption that $c_1, c_2 > 0$.

Ad 2. The plant is deterministic but the function $y = \Phi(u)$ is unknown or is not completely known. If in the known form of the function $\Phi$ some parameters are unknown, we speak about *parametric uncertainty*.

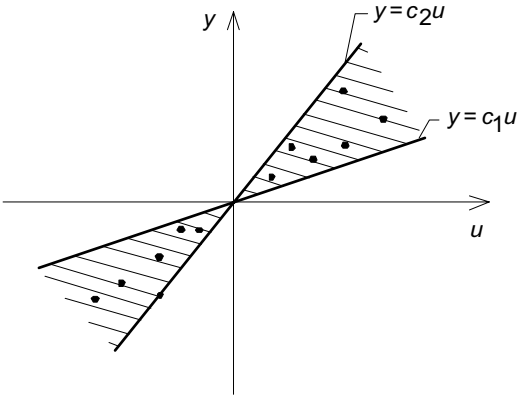


**Fig. 6.1.** Illustration of the relationship between $u$ and $y$ in the example under consideration

Using terms *known*, *unknown*, *uncertain* etc. we must determine a subject they are concerned with (who knows or does not know?, who is not certain or rather not sure?). It is convenient to distinguish in our considerations three subjects: an expert as a source of the knowledge, a designer and an executor of the decision algorithm (controlling device, controlling computer). The uncertainty caused by an incomplete knowledge of the plant concerns the expert who formulates the knowledge, and consequently is transferred to the designer which uses this knowledge to design a decision algorithm.

Both reasons of the uncertainty (points 1. and 2. listed at the beginning of this section) concern the designer: the designer's uncertainty may be caused by the non-deterministic behaviour of the plant (an objective uncertainty as a consequence of the non-deterministic plant) or by an incomplete information on the plant given by an expert (a subjective uncertainty or the expert's uncertainty). In the first case, the sets of possible values $y$ which may occur in the plant for the fixed $u$ are exactly defined by the expert. In the example considered above it means that the values $c_1$ and $c_2$ are known.

In the case of the second kind of uncertainty $y = ku$. The expert, not knowing exactly the value $k$, may give its estimation in the form of the inequality $c_1 \le k \le c_2$. Formally, the designer's uncertainty is the same as in the first case, i.e. the designer knows the set of possible values $y$: $c_1 u \le y \le c_2 u$ for $u \ge 0$. However, the interpretation is now different: possible points $(u_n, y_n)$ lie on the line $y = ku$ located between the lines $y = c_1 u$ and $y = c_2 u$ in Fig. 6.1.

Both kinds of uncertainty may occur together. It means that for the fixed $u$, different values $y$ may occur in the plant and the expert does not know exactly the sets of possible values $y$, e.g. does not know the values $c_1$ and $c_2$ introduced in our example when the first kind of uncertainty was considered. In both cases of uncertainty described above we shall shortly speak about an *uncertain plant*, remembering that in fact, an uncertainty is not necessary to be a feature of a plant but it may be an expert's uncertainty. In a similar sense we speak generally about an *uncertain algorithm* (uncertain decision maker, uncertain controller) and an *uncertain system*. These names are used for different formal descriptions of an uncertainty, not only for the relational description considered in this chapter.

Let us denote by

$$D_y(u) \subseteq Y$$

the set of all possible values $y$ for the fixed $u \in U$. In the example considered above

$$D_y(u) = \{ y:\ c_1 u \le y \le c_2 u \}$$

independently from the different interpretations of this set. The formulation of the sets $D_y(u)$ for all values $u$ which may occur in the plant means the determination of the set of all possible pairs $(u, y)$ which may appear. This is a subset of Cartesian product $U \times Y$, i.e. the set of all pairs $(u, y)$ such that $u \in U$ and $y \in Y$. Such a subset is called a *relation*

$$u \rho y \triangleq R(u, y) \subseteq U \times Y. \tag{6.1}$$

In a special case for the deterministic plant, this is the function $y = \Phi(u)$, i.e.

$$R(u, y) = \{(u, y) \in U \times Y:\ (u \in D_u) \wedge y = \Phi(u)\}$$

where $D_u$ denotes the set in which the function is defined (in particular $D_u = U$). For simplicity, the plant described by the relation (6.1) we shall call a *relational plant*, remembering that in fact the relational description does not have to be a feature of the plant but is a form of the uncertainty description. In the further considerations we shall assume that the relation describing the plant is not reduced to a function, i.e. the plant is uncertain. Usually, the relation (6.1) is defined by a property $\varphi(u, y)$ concerning $u$ and $y$, which for fixed values of the variables $u$ and $y$ is a proposition in two-valued logic. Such a property is called a *predicate*. The relation $R$ denotes a set of all pairs $(u, y)$ for which this property is satisfied, i.e.

$$R(u, y) = \{(u, y) \in U \times Y:\ \mathrm{w}[\varphi(u, y)] = 1\} \stackrel{\Delta}{=} \{(u, y) \in U \times Y:\ \varphi(u, y)\}$$

where $\mathrm{w}[\varphi(u, y)] \in \{0,1\}$ is a logical value (0 or 1 means that a sentence is false or true, respectively). Usually the property $\varphi(u, y)$ is directly called a relation and instead saying that $(u, y)$ belongs to the relation $R$, we say that it satisfies this relation. In our example $\varphi(u, y) =$ "$c_1 u \le y \le c_2 u$". The relation $R(u, y)$ has often a form of a set of equalities and (or) inequalities concerning the vectors $u$ and $y$. Below, four examples of the description of a relational plant are given:

1. $p = 2$, $l = 3$ (two inputs, three outputs)

$$u^{(1)} + 2u^{(2)} - y^{(1)} + 5y^{(2)} + y^{(3)} = 0,$$

$$3u^{(1)} - u^{(2)} + y^{(1)} - 2y^{(2)} + y^{(3)} = 4.$$

2. $p = 2$, $l = 2$

$$u^{(1)} + (y^{(2)})^2 = 4,$$

$$u^{(1)} + u^{(2)}u^{(1)} + y^{(1)} \le 0,$$

$$u^{(2)} + y^{(2)} \ge 1.$$

3. $p = l = 1$

$$(u^{(1)})^2 + (u^{(2)})^2 = 4.$$

4. $p = l = 1$

$$(u^{(1)})^2 + (u^{(2)})^2 \le 4.$$

It is easy to show that none of the above relations is a function.

If the disturbances $z \in Z$ act on the plant then they appear in the description of the plant by a relation

$$R(u, y, z) \subseteq U \times Y \times Z \tag{6.2}$$

or the relation $R(u, y; z) \subseteq U \times Y$ for the fixed $z$. In many cases an expert presents the knowledge on the plant in the form of a set of relations

$$R_i(u, w, y, z), \quad i = 1, 2, ..., k \tag{6.3}$$

where $w \in W$ denotes the vector of additional auxiliary variables appearing in the knowledge description. The set (6.3) may be reduced to one relation by eliminating the variable $w$:

$$R(u, y, z) = \{(u, y, z) \in U \times Y: \bigvee_{w \in W} [(u, w, y, z) \in \bigcap_{i=1}^{k} R_i(u, w, y, z)]\}. \quad (6.4)$$

It is then the set of all triplets $(u, y, z)$ for which there exists $w$ such that $(u, w, y, z)$ satisfies all relations $R_i$. The formal description of the knowledge of the plant differing from a traditional model (for the static plant it is a functional model $y = \Phi(u)$) is sometimes called a *knowledge representation* of the plant. More generally, we speak about the knowledge representation as a description of the knowledge given by an expert and concerning a determined part of a reality, a domain, a system, a way of acting etc. In the computer implementation, the knowledge representation is called a *knowledge base* which may be treated as a generalization of a traditional data base. The term knowledge representation is defined and understood in different ways (not always precisely). That is why, independently of the names used, it is so important to formalize precisely terms occurring in concrete considerations and to formalize concrete problems based on these terms. In the case considered in this chapter the knowledge representation is the relation (6.1) or (6.2), and more generally – the set of relations (6.3). This is a relational knowledge representation of the static plant under consideration [24, 52]. In the next sections analysis and decision making problems based on the relational knowledge representation will be described.

## 6.2 Analysis Problem

Before the description of a decision problem which is a basic problem for a designer, it is useful to present an analysis problem. For the plant described by the model in a form of a function $y = \Phi(u)$, the analysis problem consists in determination of the value $y = y^*$ for the given value $u = u^*$. In the case of the relational knowledge representation, when the relation is not reduced to the function, only a set of possible values $y$ may be found. The information concerning $u$ may also be imprecise and consist in giving a set $D_u \subset U$ such that $u \in D_u$. Consequently, the formulation of the analysis problem for the plant without disturbances, adequate to the considered model of uncertainty, is the following:

**Analysis problem:** For the given relation $R(u, y)$ and the set $D_u \subset U$ one should determine the smallest set $D_y \subset Y$ for which the implication

$$u \in D_u \rightarrow y \in D_y$$

is satisfied.

The term "the smallest set" means that we are interested in the information concerning $y$ as precise as possible. In other words, the set $D_y$ should be such that if $y$ does not belong to this set then in the set $D_u$ there is no $u$ such that $(u, y) \in R(u, y)$. The analysis problem for the single-input and single-output plant is illustrated in Fig. 6.2 where the shaded domain denotes $R(u, y)$ and the interval $D_y$ denotes the problem solution for the given interval $D_u$.

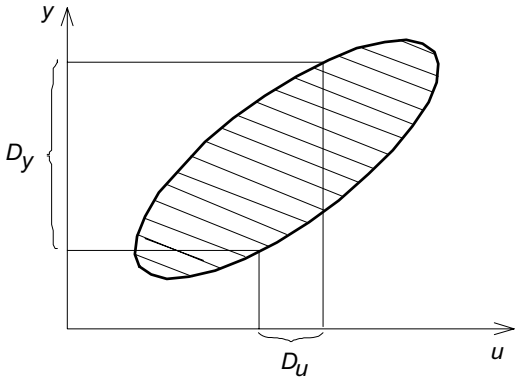


**Fig. 6.2.** Illustration of analysis problem

The general form of the analysis problem solution is the following:

$$D_y = \{ y \in Y: \bigvee_{u \in D_u} [(u, y) \in R(u, y)] \}. \tag{6.5}$$

This is then the set of all values $y$ for which in the set $D_u$ there exists $u$ such that $(u, y)$ belongs to $R$. In particular, for the known value $u$, i.e. for $D_u = \{u\}$ (a singleton)

$$D_y(u) = \{ y \in Y: (u, y) \in R(u, y) \} \tag{6.6}$$

where $D_y(u)$ denotes the solution of the problem for the given value $u$. Finding the solution for a concrete form of the relation $R(u, y)$ may be very difficult and may require special computational methods adequate for given forms of the knowledge representations. For example, the methods of solving the set of equalities and inequalities if the relation $R(u, y)$ de-

scribing the plant is defined by such a set. We shall return to this problem in Chap. 12 in which a universal analysis algorithm for the case where the relations (6.3) are presented in a form of logical operations will be described. It is worth noting that the analysis problem under consideration is a generalization of the problem presented at the beginning of this section for a functional plant described by a function $y = \Phi(u)$. The properties $u \in D_u$ and $y \in D_y$ may be called *input* and *output properties*, respectively. In the analysis problem for the functional plant these properties have the form $u = u^*$ and $y = y^*$. In general, the analysis problem consists then in the determination of the output property (exactly speaking, the strongest output property with the smallest set $D_y$) for the given input property.

If there are external disturbances $z$ acting on the plant, and as a result of an observation it is known that $z \in D_z$ then the analysis problem is formulated as follows:

For the given relation $R(u, y, z)$ and the given sets $D_u$ and $D_z$ one should determine the smallest set $D_y$ for which the implication

$$(u \in D_u) \wedge (z \in D_z) \rightarrow y \in D_y$$

or

$$(u, z) \in D_u \times D_z \rightarrow y \in D_y$$

is satisfied.

According to the formula (6.5) we obtain

$$D_y = \{ y \in Y: \bigvee_{u \in D_u} \bigvee_{z \in D_z} [(u, y, z) \in R(u, y, z)] \}. \tag{6.7}$$

Hence, $D_y$ is a set of all values $y$ for which in the set $D_u$ there exists such $u$ and in the set $D_z$ there exists such $z$ that $(u, y, z)$ belongs to $R$. For the fixed value $z$

$$D_y(z) = \{ y \in Y: \bigvee_{u \in D_u} [(u, y, z) \in R(u, y, z)] \}. \tag{6.8}$$

**Example 6.1.** Consider the plant with two inputs $u^{(1)}$ and $u^{(2)}$ described by the inequality

$$c_1 u^{(1)} + d_1 u^{(2)} \leq y \leq c_2 u^{(1)} + d_2 u^{(2)} \tag{6.9}$$

and the set $D_u$ determined by the inequalities

$$au^{(1)} + bu^{(2)} \le \alpha, \tag{6.10}$$

$$u^{(1)} \ge u^{(1)}_{\min}, \qquad u^{(2)} \ge u^{(2)}_{\min}. \tag{6.11}$$

For example, $y$ may denote the amount of a product obtained in a certain production process, and $u^{(1)}$ and $u^{(2)}$ – the amounts of two kinds of raw materials, the inequality (6.10) – the constraints concerning the cost of the both raw materials, and the inequalities (6.11) – additional constraints caused by technical conditions of the process. The parameters $c_1$, $c_2$, $d_1$, $d_2$, $a$, $b$, $\alpha$, $u^{(1)}_{\min}$, $u^{(2)}_{\min}$ have the given positive values and $c_1 < c_2$, $d_1 < d_2$. One should determine the set of all possible values $y$ under the assumption that the values $u^{(1)}$ and $u^{(2)}$ satisfy the inequalities (6.10) and (6.11).

It is easy to note that

$$C_1 u^{(1)}_{\min} + d_1 u^{(2)}_{\min} \le y \le y_{\max} \tag{6.12}$$

where

$$y_{\max} = \max_{u^{(1)}, u^{(2)}} (c_2 u^{(1)} + d_2 u^{(2)})$$

with the constraints (6.10) and (6.11). The determination of $y_{\max}$ is a simple problem of so called *linear programming* which is easy to solve with the help of a graphical illustration. In Fig. 6.3, the shaded domain denotes the set $D_u$ and P denotes a straight line with the equation $c_2u^{(1)} + d_2u^{(2)} = y$ for the fixed value $y$. Then the point $(u^{(1)}, u^{(2)})$ maximizing $c_2u^{(1)} + d_2u^{(2)}$ lies in one of the vertexes $W_1$, $W_2$, depending on the inclination of the line P:

1. If

$$\frac{c_2}{d_2} < \frac{a}{b}$$

then the point maximizing $c_2u^{(1)} + d_2u^{(2)}$ lies in the vertex $W_1$, i.e.

$$u^{(1)} = u^{(1)}_{\min}, \qquad u^{(2)} = \frac{\alpha - au^{(1)}_{\min}}{b},$$

$$y_{\max} = c_2 u_{\min}^{(1)} + \frac{d_2}{b}(\alpha - a u_{\min}^{(1)}). \tag{6.13}$$

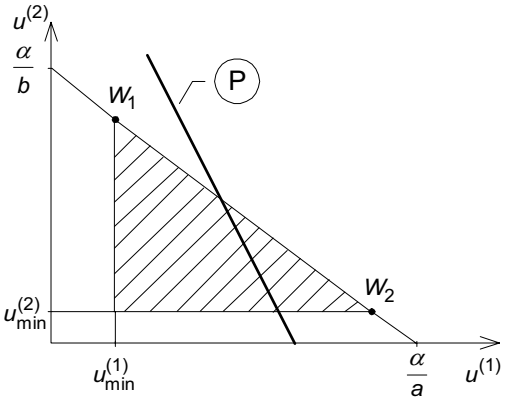


**Fig. 6.3.** Illustration of example

2. If

$$\frac{c_2}{d_2} > \frac{a}{b}$$

then the point maximizing $c_2 u^{(1)} + d_2 u^{(2)}$ lies in the vertex $W_2$, i.e.

$$u^{(2)} = u_{\min}^{(2)}, \qquad u^{(1)} = \frac{\alpha - b u_{\min}^{(2)}}{a},$$

$$y_{\max} = \frac{c_2}{a}(\alpha - b u_{\min}^{(2)}) + d_2 u_{\min}^{(2)}. \tag{6.14}$$

3. If

$$\frac{c_2}{d_2} = \frac{a}{b}$$

then for any point $(u^{(1)}, u^{(2)})$ lying in the line P between $W_1$ and $W_2$, the variable $y$ takes the maximum value determined by the formula (6.13) or (6.14) (the results obtained from these formulas are identical). For example, for numerical data $c_1 = 1$, $c_2 = 2$, $d_1 = 2$, $d_2 = 4$, $a = 1$, $b = 4$, $\alpha = 3$, $u_{\min}^{(1)} = 1$, $u_{\min}^{(2)} = 0.5$ we obtain

$$\frac{c_2}{d_2} = \frac{1}{2}, \qquad \frac{a}{b} = \frac{1}{4} < \frac{c_2}{d_2}.$$

Substituting the data into the formula (6.14) gives $y_{\max} = 4$ and $y_{\min} = c_1 u_{\min}^{(1)} + d_1 u_{\min}^{(2)} = 2$, according to the formula (6.12). The set $D_y$ is then determined by the inequality $2 \le y \le 4$. □

## 6.3 Decision Making Problem

It is an inverse problem to the analysis problem formulated in Sect. 6.2, and for the plant described by the function $y = \Phi(u)$, it consists in determining such a decision $u = u^*$ that the respective output $y = \Phi(u^*)$ is equal to the given required value $y^*$. This problem for the functional plant has been considered in Sect. 3.1. In the relational plant it is not possible to satisfy the requirement $y = y^*$ but it has a sense to formulate the requirement in the form $y \in D_y$ for the fixed set $D_y$, and to find a decision $u$ for which this requirement is satisfied. Solving the problem consists in determining the set $D_u$ of all possible (or acceptable) decisions, i.e. in determining all values $u$ for which the property $y \in D_y$ will be fulfilled.

**Decision making (control) problem:** For the given relation $R(u, y)$ and the set $D_y \subset Y$ determining a user's requirement one should find the largest set $D_u \subset U$ such that the implication

$$u \in D_u \rightarrow y \in D_y \tag{6.15}$$

is satisfied.

The general form of the problem solution is as follows:

$$D_u = \{u \in U:\ D_y(u) \subseteq D_y\} \tag{6.16}$$

where $D_y(u)$ is defined by the formula (6.6). Then, $D_u$ is the set of all such values $u$ for which the set of possible values $y$ belongs to the given set $D_y$. A remark on difficulties connected with the determination of a final solution for concrete forms of $R(u, y)$ and $D_y$, and on a universal algorithm in the case of logical operations is now analogous to that for the analysis problem in Sect. 6.2. Similarly as in Sect. 6.2 it is worth noting

that the decision problem for the relational plant may be considered as a generalization of the respective problem for the functional plant where the input property $u \in D_u$ and the output property $y \in D_y$ are reduced to the forms $u = u^*$ and $y = y^*$, respectively.

The solution of the decision problem under consideration may not exist, i.e. $D_u$ may be an empty set. Such a case is illustrated in Fig. 6.4: For the given interval $D_y$, the interval $D_u$ for which the implication (6.15) could be satisfied does not exist. It means that the requirement is too strong, i.e. that the interval $D_y$ is too small. The requirement may be satisfied for the greater interval $D_y$ (see Fig. 6.2). If $D_u = \varnothing$ (empty set), we can say that the plant $R(u, y)$ *is non-controllable* for the requirement $y \in D_y$. For example, let $D_y = [y_1, y_2]$ in the example illustrated by Fig. 6.1, i.e. the property $y_1 \le y \le y_2$ is required by a user. It is easy to note that the solution for $y_1 > 0$ is as follows:

$$D_u = \left[ \frac{y_1}{c_1}, \frac{y_2}{c_2} \right]$$

and the controllability condition has the form

$$\frac{y_1}{c_1} \le \frac{y_2}{c_2}.$$

If external disturbances $z$ act on the plant and as a result of measurement it is known that $z \in D_z$ then the decision problem is formulated as follows:

For the given relation $R(u, y, z)$ and the given sets $D_z$ and $D_y$ one should find the largest set $D_u$ for which the implication

$$(u \in D_u) \wedge (z \in D_z) \rightarrow y \in D_y$$

is satisfied.

The general form of the decision problem solution is now the following:

$$D_u = \{u \in U: \bigwedge_{z \in D_z} [D_y(u, z) \subseteq D_y]\} \tag{6.17}$$

where

$$D_y(u, z) = \{ y \in Y: (u, y, z) \in R(u, y, z)\}. \tag{6.18}$$

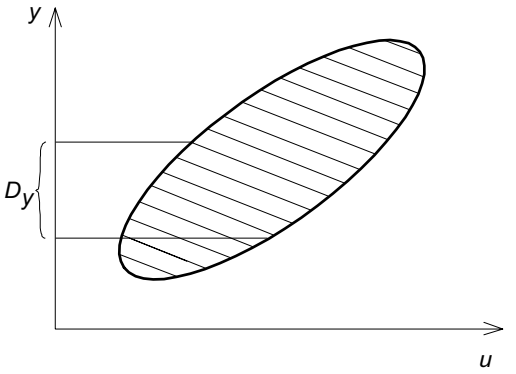


**Fig. 6.4.** Illustration of the case when solution does not exist

It is then the set of all decisions $u$ such that for every $z$ belonging to $D_z$ the set of all possible values $y$ belongs to the given set $D_y$. One may say that the solution $D_u$ is *robust* with respect to $z$, which means that it is not sensitive to $z$, i.e. it gives a satisfying solution $y$ for every value of the disturbance $z$ from the fixed $D_z$. For the fixed $z$, the set of possible decisions is defined by the formula (6.16) in which $D_y(u)$ should be determined according to the formula (6.6) for the given relation $R(u, y, z)$, i.e. the relation $R(u, y; z) \subset U \times Y$ with the parameter $z$. Consequently

$$D_u(z) = \{ u \in U:\ D_y(u, z) \subseteq D_y \} \triangleq \overline{R}(z, u) \tag{6.19}$$

where $D_y(u, z)$ is defined by the formula (6.18). The formula (6.19) defines a relation between $z$ and $u$ which has been denoted by $\overline{R}(z, u)$. This relation may be considered as a description of a *relational decision* (*control*) *algorithm* in the open-loop system (Fig. 6.5) or the relational representation of a knowledge on the control (i.e. the knowledge representation of the controller in the open-loop system). For the functional plant the system in Fig. 6.5 is reduced to the system presented in Fig. 3.1. It is interesting and important to note that for an *uncertain* plant one obtains an *uncertain* control algorithm determined by using a knowledge representation of the plant. In the case of the relational description of uncertainty considered in this chapter, it is the *relational* plant and the corresponding *relational* control algorithm.

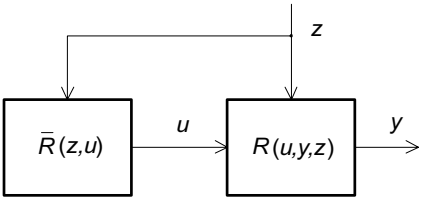


**Fig. 6.5.** Open-loop control system with relational plant and relational control algorithm

**Example 6.2.** Consider the plant with two scalar inputs $u$, $z$ and the single output $y$, described by the inequality

$$cu + z \le y \le 2cu + z, \qquad c > 0. \tag{6.20}$$

Determine the set $D_u$ for the given sets $D_z = [z_1, z_2]$ and $D_y = [y_1, y_2]$. In other words, we want to obtain the set of all control decisions satisfying the requirement $y_1 \le y \le y_2$ for every $z$ from the interval $[z_1, z_2]$. The set (6.18) is now defined directly by the inequality (6.20). According to (6.17) the set $D_u$ is then defined by the inequality

$$\frac{y_1 - z_1}{c} \le u \le \frac{y_2 - z_2}{2c}.$$

The solution exists if $2(y_1 - z_1) \le (y_2 - z_2)$. For the given value $z$ the set $D_u(z)$ is determined by the inequality

$$\frac{y_1 - z}{c} \le u \le \frac{y_2 - z}{2c}.$$

It is $\bar{R}(z,u)$ or the relational control algorithm in the open-loop system. □

## 6.4 Dynamical Relational Plant

The considerations are analogous to those for the static plant but respective notations and calculations may now be much more complicated [22, 29, 52]. That is why the considerations in this section are limited to the simplest form of the relational knowledge representation and the simplest version of analysis and decision problems formulated for a discrete plant described with the help of a state vector. The deterministic dynamical plant is described by the equations

$$\left.\begin{aligned} x_{n+1} &= f(x_n, u_n), \\ y_n &= \eta(x_n) \end{aligned}\right\} \tag{6.21}$$

where $x_n \in X$ is the state vector, $u_n \in U$ is the control vector and $y_n \in Y$ is the output vector. As in the case of the static plant, in the description of the relational dynamical plant the functions $f$ and $\eta$ are replaced by the relations

$$\left.\begin{aligned} &R_{\mathrm{I}}(u_n, x_n, x_{n+1}) \subseteq U \times X \times X, \\ &R_{\mathrm{II}}(x_n, y_n) \subseteq X \times Y. \end{aligned}\right\} \tag{6.22}$$

The relations $R_{\mathrm{I}}$ and $R_{\mathrm{II}}$ form the *relational knowledge representation of the dynamical plant*. The relation $R_{\mathrm{I}}$ describes a relationship between the state vectors $x_n$, $x_{n+1}$ and the input $u_n$, the relation $R_{\mathrm{II}}$ describes a relationship between the state vector $x_n$ and the output $y_n$. The description (6.21) as well as (6.22) concerns the stationary plant (the plant with constant parameters). For the non-stationary plant the functions $f$ and $\eta$ in the functional case and the relations $R_{\mathrm{I}}$ and $R_{\mathrm{II}}$ in the relational case depend on $n$. As in the static case, a typical form of the relations is presented by a set of equalities and (or) inequalities concerning the components of the respective vectors. The relations (6.22) have often the form

$$\left.\begin{aligned} &f_1(u_n, x_n) \le x_{n+1} \le f_2(x_n, u_n), \\ &\eta_1(x_n) \le y_n \le \eta_2(x_n), \end{aligned}\right\} \tag{6.23}$$

following from the uncertain information on the plant (6.21). The *differential inequalities* (6.23) denote the set of inequalities for the respective components of the vectors.

Let $D_{un}$, $D_{xn}$ and $D_{yn}$ denote sets of the vectors $u_n$, $x_n$, $y_n$, respectively, i.e.

$$u_n \in D_{un} \subseteq U, \qquad x_n \in D_{xn} \subseteq X, \qquad y_n \in D_{yn} \subseteq Y.$$

**Analysis problem:** For the given relations (6.22), the set $D_{x0}$ and the sequence of sets $D_{un}$ ($n = 0, 1, ...$) one should determine the sequence of the smallest sets $D_{yn} \subset Y$ ($n = 1, 2, ...$) for which the implication

$$(u_0 \in D_{u0}) \wedge (u_1 \in D_{u1}) \wedge ... \wedge (u_{n-1} \in D_{u,n-1}) \rightarrow y_n \in D_{yn}$$

is satisfied.

This is a generalization for a relational case of the analysis problem for the plant (6.21) consisting in the determination of the sequence $y_n$ for the given sequence $u_n$, the known functions $f$ and $\eta$, and the initial condition $x_0$. For the fixed $n$, the plant under consideration may be treated as a cascade connection of two static relational plants (Fig. 6.6). The analysis problem for the dynamical plant is then reduced to the analysis problem for the relational plants $R_{\mathrm{I}}$ and $R_{\mathrm{II}}$, i.e. to the problem described in Sect. 6.2 for the static plant. As a result, according to the formula (6.5) applied successively to $R_{\mathrm{I}}$ and $R_{\mathrm{II}}$, we obtain the following **recursive procedure** for $n = 1, 2, \ldots$ .

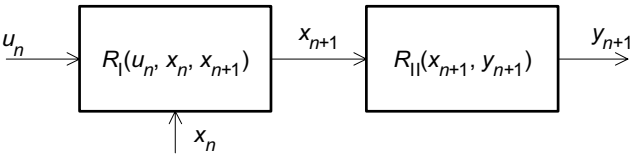


**Fig. 6.6.** Relational dynamical plant

1. For the given $D_{un}$ and $D_{xn}$ determined in the former step, we determine $D_{x,n+1}$ using $R_{\mathrm{I}}(u_n, x_n, x_{n+1})$:

$$D_{x,n+1} = \{x_{n+1} \in X: \bigvee_{u_n \in D_{un}} \bigvee_{x_n \in D_{xn}} [(u_n, x_n, x_{n+1}) \in R_{\mathrm{I}}(u_n, x_n, x_{n+1})]\}. \quad (6.24)$$

2. For the obtained $D_{x,n+1}$ we determine $D_{y,n+1}$ using $R_{\mathrm{II}}(x_{n+1}, y_{n+1})$:

$$D_{y,n+1} = \{y_{n+1} \in Y: \bigvee_{x_{n+1} \in D_{x,n+1}} [(x_{n+1}, y_{n+1}) \in R_{\mathrm{II}}(x_{n+1}, y_{n+1})]\}. \quad (6.25)$$

For $n = 0$ in the formula (6.24) we use the given set $D_{x0}$.

In the case of the plant with disturbances, the analysis problem and the solving procedure are analogous, similar to (6.7) and (6.8) in the case of the static plant.

**Decision making (control) problem:** For the given relations (6.22), the set $D_{x0}$ and the sequence of sets $D_{yn}$ ($n = 1, 2, \ldots, N$) defining a user's requirement concerning $y_1, y_2, \ldots, y_N$ one should determine the sequence of sets $D_{un}$ ($n = 0, 1, \ldots, N–1$) for which the implication

$$(u_0 \in D_{u0}) \wedge (u_1 \in D_{u1}) \wedge \ldots \wedge (u_{N-1} \in D_{u,N-1}) \\ \rightarrow (y_1 \in D_{y1}) \wedge (y_2 \in D_{y2}) \wedge \ldots \wedge (y_N \in D_{yN})$$

is satisfied.

This is one of possible formulations of a control problem for the relational dynamical plant. If the requirement concerned the state $x_N$, i.e. had a form $x_N \in D_{xN}$ for the given set $D_{xN}$ then the problem under consideration would be a generalization of the problem described in Chap. 3 and consisting in the determination of the control $u_0, u_1, \ldots, u_{N-1}$ removing the plant from the state $x_0$ to the given state $x_N$ in a finite time. To determine the sequence $D_{un}$ we may apply a decomposition consisting in the determination of $D_{un}$ *step by step* starting from the end, in a similar way as in the dynamic programming procedure presented in Chap. 4. Let us note that in the formulation of the control problem we do not use the words "the sequence of largest sets". Now the set of all possible controls denotes the set of all sequences $u_0, u_1, \ldots, u_{N-1}$ for which the requirements are satisfied. If we decide not to determine all such sequences then we may obtain the solution of the control problem by applying the following recursive procedure starting from $n = 0$:

1. For the given $D_{y,n+1}$, using $R_{\mathrm{II}}$ we determine the largest set $D_{x,n+1}$ for which the implication

$$x_{n+1} \in D_{x,n+1} \rightarrow y_{n+1} \in D_{y,n+1}$$

is satisfied. It is a decision problem for the plant $R_{\mathrm{II}}$. Using (6.16) we obtain

$$D_{x,n+1} = \{x_{n+1} \in X:\ D_{y,n+1}(x_{n+1}) \subseteq D_{y,n+1}\} \tag{6.26}$$

where

$$D_{y,n+1}(x_{n+1}) = \{y_{n+1} \in Y:\ (x_{n+1}, y_{n+1}) \in R_{\mathrm{II}}(x_{n+1}, y_{n+1})\}.$$

2. For $D_{x,n+1}$ just determined and $D_{xn}$ found in the former step, using $R_{\mathrm{I}}$ we determine the largest set $D_{un}$ for which the implication

$$(u_n \in D_{un}) \wedge (x_n \in D_{xn}) \rightarrow x_{n+1} \in D_{x,n+1}$$

is satisfied.

According to (6.17) we have

$$D_{un} = \{u_n \in U: \bigwedge_{x_n \in D_{xn}} [D_{x,n+1}(u_n, x_n) \subseteq D_{x,n+1}]\} \tag{6.27}$$

where

$$D_{x,n+1}(u_n, x_n) = \{x_{n+1} \in X: \ (u_n, x_n, x_{n+1}) \in R_I(u_n, x_n, x_{n+1})\}.$$

**Example 6.3.** Consider a simple case of the first order plant described by the inequalities

$$a_1 x_n + b_1 u_n \le x_{n+1} \le a_2 x_n + b_2 u_n,$$

$$c_1 x_{n+1} \le y_{n+1} \le c_2 x_{n+1}.$$

Assume that $x_{01} \le x_0 \le x_{02}$ , $b_1, b_2, c_1, c_2 > 0$. The requirement concerning $y_n$ is as follows:

$$\bigwedge_{n \ge 1} (y^{(1)} \le y_n \le y^{(2)}) \tag{6.28}$$

which means that for every $n$ the plant output is required to belong to the constant interval $[y^{(1)}, y^{(2)}]$. For the given $x_{01}, x_{02}, y^{(1)}, y^{(2)}$ and the coefficients $a_1, a_2, b_1, b_2, c_1, c_2$, one should find the respective sequence of the sets of possible control decisions. For $n = 0$, according to the formula (6.26) the set $D_{x1}$ is defined by the inequalities

$$c_2 x_1 \le y^{(2)}, \qquad c_1 x_1 \ge y^{(1)}.$$

Then

$$D_{x1} = \left[\frac{y^{(1)}}{c_1}, \frac{y^{(2)}}{c_2}\right].$$

According to (6.27) for $u_0$ one obtains the following inequalities:

$$a_2 x_{02} + b_2 u_0 \le \frac{y^{(2)}}{c_2},$$

$$a_1 x_{01} + b_1 u_0 \ge \frac{y^{(1)}}{c_1}.$$

Hence,

$$D_{u0} = \left[ \frac{y^{(1)}}{b_1 c_1} - \frac{a_1 x_{01}}{b_1}, \frac{y^{(2)}}{b_2 c_2} - \frac{a_2 x_{02}}{b_2} \right].$$

For $n \geq 1$ the set $D_{x,n+1} = D_{x1}$ and according to (6.27) the set $D_{un}$ is defined by the inequalities

$$a_2 \frac{y^{(2)}}{c_2} + b_2 u_n \leq \frac{y^{(2)}}{c_2},$$

$$a_1 \frac{y^{(1)}}{c_1} + b_1 u_n \geq \frac{y^{(1)}}{c_1}.$$

After some transformations one obtains

$$D_{un} = \left[ \frac{y^{(1)}(1-a_1)}{b_1 c_1}, \frac{y^{(2)}(1-a_2)}{b_2 c_2} \right].$$

Consequently, if

$$\frac{y^{(1)}}{b_1 c_1} - \frac{a_1 x_{01}}{b_1} \leq u_0 \leq \frac{y^{(2)}}{b_2 c_2} - \frac{a_2 x_{02}}{b_2}$$

and for every $n > 0$

$$\frac{y^{(1)}(1-a_1)}{b_1 c_1} \leq u_n \leq \frac{y^{(2)}(1-a_2)}{b_2 c_2}$$

then the requirement (6.28) will be satisfied. The conditions for the existence of the solution are the following:

$$\frac{y^{(1)}}{b_1 c_1} - \frac{a_1 x_{01}}{b_1} \leq \frac{y^{(2)}}{b_2 c_2} - \frac{a_2 x_{02}}{b_2}, \tag{6.29}$$

$$\frac{y^{(1)}}{c_1} \leq \frac{y^{(2)}}{c_2}, \tag{6.30}$$

$$\frac{y^{(1)}(1-a_1)}{b_1 c_1} \leq \frac{y^{(2)}(1-a_2)}{b_2 c_2}. \tag{6.31}$$

If $y^{(1)} > 0$ and $a_2 < 1$ then the conditions (6.30) and (6.31) are reduced to the inequality

$$\frac{y^{(2)}}{y^{(1)}} \geq \max(\alpha, \beta)$$

where max denotes a greater number in the pair $(\alpha, \beta)$,

$$\alpha = \frac{b_2 c_2}{b_1 c_1} \frac{1 - a_1}{1 - a_2}, \qquad \beta = \frac{c_2}{c_1}.$$

Hence, the sets $D_{un}$ are not empty if the requirement concerning $y_n$ is not too strong, i.e. the ratio of $y^{(1)}$ to $y^{(2)}$ is sufficiently great. It should be noted that the inequalities (6.29), (6.30) and (6.31) present the conditions for the existence of the solution obtained by applying the method presented above. The obtained solution may not contain all the sequences $u_n$ for which the requirement is fulfilled. Then, if the conditions (6.29), (6.30) and (6.31) are not satisfied, a sequence satisfying the requirement (6.28) may exist. □

## 6.5 Determinization

Replacing an uncertain description by its deterministic representation will be called a *determinization*. In our case it means replacing the relational description by the deterministic description in the form of a function. The determinization may concern the plant and the control algorithm as well. A frequently used way of the determinization consists in using a mean value of the output of an uncertain system and formulating a dependence of this value upon the input. After finding the set of possible control decisions (6.17) one must choose from this set and put at the input of the plant a concrete decision. It may be a mean value defined as follows:

$$\tilde{u} = \frac{\int_{D_u} u du}{\int_{D_u} du}.$$

Then in the computer control system (Fig. 6.7) we may distinguish a block denoting the generation of the set $D_u$ based on the knowledge representa-

tion of the plant $R(u, y, z)$ and the result of the disturbance observation in the form of the set $D_z$, and a block denoting the determinization, determining one concrete decision $u$. For the fixed $z$ one may apply the determinization of the relational control algorithm, i.e. the mean value for the set $D_u(z)$, defined by the formula (6.19)

$$\tilde{u}(z) = \frac{\int_{D_u(z)} u du}{\int_{D_u(z)} du} \triangleq \tilde{\Psi}(z) . \tag{6.32}$$

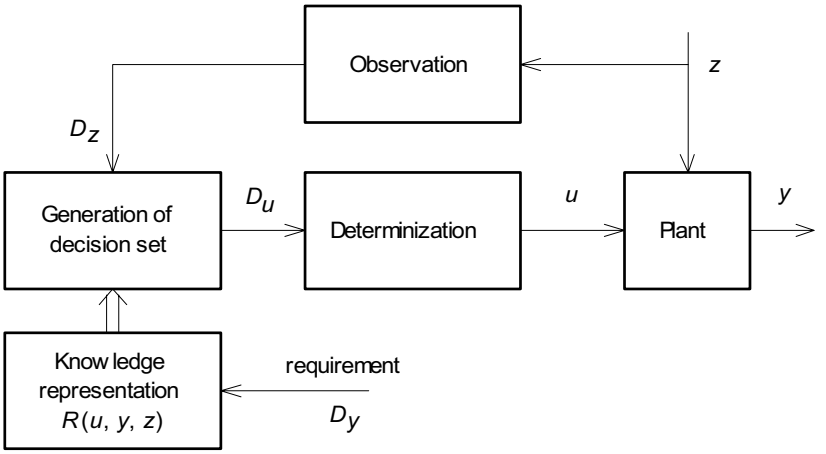


**Fig. 6.7.** Structure of knowledge-based control system in the case under consideration

In such a way, the deterministic control algorithm $\tilde{\Psi}(z)$ is obtained. In the case with the given required value $y^*$, the determinization of the relational plant or the determinization of the relational control algorithm may be applied. Let us present successively these two concepts. In the first case we determine

$$\tilde{y} = \frac{\int_{D_y(u,z)} y dy}{\int_{D_y(u,z)} dy} \triangleq \Phi(u, z) \tag{6.33}$$

where the set $D_y(u, z)$ is defined by the formula (6.18). The relational plant is then replaced by the deterministic plant described by the function $\Phi$, for which it is possible to formulate the decision problem so as in Sect. 3.1, i.e. to find $u$ for which $y = y^*$. Then from the equation $\Phi(u, z) = y^*$ the deterministic control algorithm $u = \Psi(z)$ is obtained under the assumption that for the given $z$ this equation has a unique solution with respect to $u$.

In the second case we consider the relation $R(u, y^*, z)$, i.e. the set of all pairs $(u, z)$ at the input of the plant for which $y = y^*$ may occur at the output, or the set of all possible inputs $(u, z)$ when $y = y^*$. Let us introduce the notation

$$R(u, y^*, z) \triangleq R_d(z, u). \tag{6.34}$$

It is in this case a description of the knowledge on the control or a relational control algorithm determined for the given knowledge representation of the plant and the given value $y^*$. Of course, $R_d(u, z)$ differs from the relation $\overline{R}(z, u)$ introduced in the formula (6.19). The determinization of the relational algorithm $R_d$ leads to the deterministic algorithm

$$u_d(z) = \frac{\int_{D_{ud}(z)} u du}{\int_{D_{ud}(z)} du} \triangleq \Psi_d(z) \tag{6.35}$$

where

$$D_{ud}(z) = \{u \in U: (u, z) \in R_d(z, u)\}.$$

Thus, for the relational plant one can determine two deterministic algorithms in a closed-loop system: the algorithm $\Psi(z)$ obtained as a result of the determinization of the plant (Fig. 6.8) and the algorithm $\Psi_d(z)$ obtained as a result of the determinization of the relational control algorithm determined by using the relational knowledge representation of the plant (Fig. 6.9). In the first case the determinization (i.e. the liquidation of the uncertainty) occurs just at the level of the plant, and in the second case the uncertainty in the plant is transferred to the control algorithm. The similar two concepts will be considered for other descriptions of uncertainty in the next chapters. The comparison of these two frequently used ideas is so important that it is useful to present it in the form of a theorem.

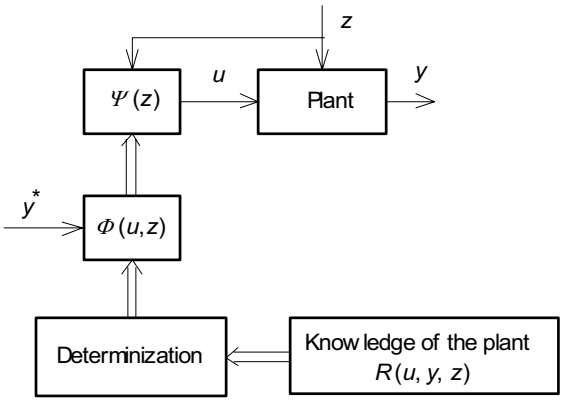


**Fig. 6.8.** Decision making via determinization of knowledge of the plant

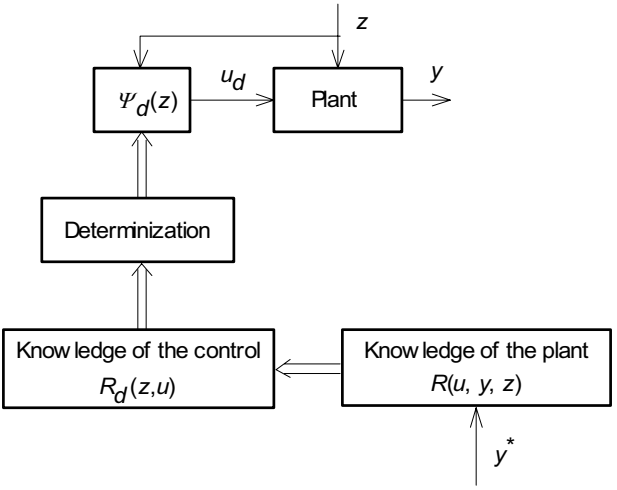


**Fig. 6.9.** Decision making via determinization of knowledge of the control

**Theorem 6.1.** In general $\Psi(z) \neq \Psi_d(z)$. □

The theorem may be proved by an example showing that in a particular case $\Psi(z) \neq \Psi_d(z)$ (see Example 6.4). The theorem means that the control decisions determined in the both cases for the same $z$ may be different. In practice, they usually are different except special cases. In other words, for $u_d(z)$ the mean value of $y$ in general differs from the required value $y^*$.

Relatively simple problems and ways of decision making considered in this chapter for uncertain plants with a relational description illustrate two

general concepts concerning the determinization and two respective ways of decision making under uncertainty:

1. Two concepts of the determinization:
a. *Determinization of the knowledge of the plant*.
b. *Determinization of the knowledge of the decision making*, based on the knowledge of the plant, i.e. obtained by using the knowledge of the plant.
2. Two ways of obtaining the knowledge of the decision making:
a. The knowledge of the decision making is determined by using *the knowledge of the plant given by an expert* (*a descriptive approach*).
b. The knowledge of the decision making *is given directly by an expert* (*a prescriptive approach*).

The knowledge representation concerning a fixed determined part of a reality (which we called a plant) is *a description* of this existing reality, so it has a descriptive (or declarative) character and presents a knowledge about WHAT THE PLANT IS. The knowledge representation concerning the decision making (or a decision maker) is a kind of *a prescription* or instruction, so it has a prescriptive (or imperative) character and presents a knowledge about HOW TO ACT. In the case of the control, the prescriptive approach means that the expert's experience and knowledge concerning the plant is not formulated directly in the form of a description of the plant but indirectly in the form of a prescription describing how to control this plant. Such an approach is widely used in so called fuzzy controllers mentioned in Sect. 5.5 and presented more precisely in Chap. 9. It is important to have in mind that the effect of the control with the algorithm obtained by using the knowledge of the control given directly by an expert depends on the plant and can be estimated by a performance index for the given description of the plant, as it was shown in Chap. 5 for the given form of the control algorithm. In general, this effect is worse than the effect of the control according to the algorithm obtained by using the known description of the plant. In the problem under consideration, the effects of the both approaches will be the same if the knowledge representation given by an expert is identical with $\overline{R}(z,u)$ in the formula (6.19) in the case considered in Sect. 6.3 or with $R_d(z, u)$ in the case considered in this section.

Finally, let us summarize the decision making problems and their solutions for the plant $R(u, y, z)$ with the fixed $z$:

1. For the requirement $y \in D_y$ we obtain the relational control algorithm $D_u(z)$ or $\overline{R}(z,u)$ in the formula (6.19). As a result of the determinization of the relational algorithm according to (6.32) we have the deterministic

control algorithm $u = \widetilde{\Psi}(z)$.

2. For the requirement formulated with the help of the given desirable value $y^*$ we obtain:

a. the deterministic control algorithm $u = \Psi(z)$ as a result of the determinization of the plant according to (6.33)

b. the deterministic control algorithm $u_d = \Psi_d(z)$ as a result of the determinization of the relational control algorithm according to (6.35).

**Example 6.4.** As a result of the determinization of the plant presented in Example 6.2, according to (6.33) we obtain

$$\widetilde{y} = \frac{3}{2}cu + z = \Phi(u, z).$$

Consequently, for the given value $y^*$ from the equation $\Phi(u, z) = y^*$ we obtain the control algorithm

$$u = \Psi(z) = \frac{2(y^* - z)}{3c}.$$

Substituting $y^*$ into (6.20) yields the representation of the knowledge of the decision making $R_d(z, u)$ in the form of the inequality

$$\frac{y^* - z}{2c} \le u \le \frac{y^* - z}{c}$$

and after the determinization

$$u_d = \Psi_d(z) = \frac{3(y^* - z)}{4c} \ne \Psi(z).$$ □

# 7 Application of Probabilistic Descriptions of Uncertainty

In this chapter we shall assume that unknown parameters are values of random variables, which means that they have been randomly chosen from some sets. As a result, the control will satisfy requirements formulated with the help of mean values, i.e. the determined requirements will be fulfilled in the average. The plant with a probabilistic description of uncertainty will be called shortly a *probabilistic plant* like a relational plant with the relational description of uncertainty considered in the previous chapter. In Sects. 7.1 and 7.2 analysis and decision problems which are called here **basic problems** will be presented. Section 7.3 is devoted to the control based on current information on the unknown parameters, being obtained during the control process in an open-loop system, and Sect. 7.4 concerns the case without the knowledge of probability distributions. Sections 7.5–7.8 are concerned with a dynamical plant. In Sect. 7.5 the basic problem for the discrete dynamical plant will be presented, and in Sects.7.6–7.8 special problems for the linear control systems very important from a practical point of view will be described. In Sect. 7.9 we shall consider a plant with a second order uncertainty, i.e. a relational plant with random parameters.

## 7.1 Basic Problems for Static Plant and Parametric Uncertainty

Let us consider the static plant

$$y = \Phi(u, c) \tag{7.1}$$

where $u \in U$ is the input vector, $y \in Y$ is the output vector and $c \in C$ is the vector of unknown parameters. It is then the case of a functional plant with a parametric uncertainty. An expert can give only the set $D_c \subset C$ of possible values of the unknown vector parameters $c$, or can give also some preferences and additional evaluations for the particular values in this set. We

shall now assume that these evaluations are presented in the form of a probability distribution and that this distribution is known. For example, let $D_c = \{c_1, c_2, c_3\}$ and for each value the probability of its appearing in the plant is given:

$$p_i = P(c = c_i), \qquad i = 1, 2, 3.$$

Such information means that the unknown parameter $c$ has been randomly chosen (drawn) from the set of the values of $c$, this set contains $m$ elements, $m_i$ of them having the value $c_i$, and then $p_i = \frac{m_i}{m}$. For example, if the set contains 500 elements with the value $c_1$, 200 elements with the value $c_2$ and 300 elements with the value $c_3$ then

$$p_1 = P(c = c_1) = 0.5, \quad p_2 = P(c = c_2) = 0.2, \quad p_3 = P(c = c_3) = 0.3.$$

These elements may be e.g. resistors with the resistance $c$ of a certain production series for which the values $p_1, p_2, p_3$ are obtained as results of statistical investigations. We assume that the element with an unknown resistance *built* in the plant has been chosen randomly from this series. Let us note that the probabilities $p_i$ given by an expert are not a **subjective** characteristic of his uncertainty but are an **objective** characteristic of the set from which the value $c$ has been randomly chosen, known by the expert.

Formally, the assumption about the random choosing of $c$ from a determined set means the assumption that $c$ is a value of a *random variable* $\underline{c}$, i.e. that there exists a probability distribution. In further considerations we shall assume that this is a continuous random variable for which there exists a *probability density*

$$f_c(c) = \frac{dF_c(c)}{dc}$$

where $F_c(c) = P(\underline{c} \le c)$ is a *distribution function*. For the given probability density $f_c(c)$

$$P(\underline{c} \in D_c) = \int_{D_c} f_c(c)dc$$

where $D_c \subseteq C$ is a subset of the vector space $C$ and

$$\mathrm{E}(\underline{c}) = \int_C c\, f_c(c)dc \tag{7.2}$$

where $\mathrm{E}(\underline{c})$ denotes an *expected value* of the variable $\underline{c}$. More generally,

$$\underset{\underline{c}}{\mathrm{E}}[\Phi(\underline{c})] = \int_C \Phi(c) f_c(c)\, dc\,.$$

The density $f_c(c)$ characterizes a set (a population) from which the values $c$ are randomly chosen. If $c_1, c_2, ..., c_n$ denotes a sequence of the results of independent samplings (or so called simple random sample) then for $n \to \infty$ the arithmetic mean of these values converges in a probabilistic sense to the expected value $\mathrm{E}(\underline{c})$. For the sequence of random variables $\underline{c}_n$ the probabilistic convergence can be understood in different ways. The definitions for one-dimensional case ($C = R^1$) are the following:

1. The sequence $\underline{c}_n$ is convergent to a number $a$ *stochastically* (or *in probability*) if, for any $\varepsilon > 0$

$$\lim_{n\to\infty} P(|\underline{c}_n - a| > \varepsilon) = 0\,.$$

2. The sequence $\underline{c}_n$ converges to $a$ *in the r-th mean*, if

$$\lim_{n\to\infty} \mathrm{E}(|\underline{c}_n - a|^r) = 0\,.$$

We can also consider a convergence *with probability one*, which means that the probability of the convergence to $a$ is equal to 1, or that the sequence is *almost always* convergent to $a$.

In order to formulate the decision problem for the plant (7.1), let us introduce the performance index $\varphi(y, y^*)$ described in Sect. 4.1.

**Decision making (control) problem:** For the given value $y^*$, the functions $\Phi$ and $\varphi$, and the probability density $f_c(c)$ one should find the decision $u^*$ minimizing the performance index

$$Q = \underset{\underline{c}}{\mathrm{E}}[\varphi(\underline{y}, y^*)] = \int_C \varphi[\Phi(u,c), y^*] f_c(c) dc \overset{\Delta}{=} \overline{\Phi}(u)\,. \tag{7.3}$$

This is a *probabilistic optimization problem* or the probabilistic version of the problem considered in Sect. 4.1. The knowledge of the probability distribution (the probability density in our case) means that we have the problem with a *full probabilistic information*. The procedure of determining $u^*$ consists of two operations:

1. Determination of the integral (7.3).
2. Minimization of the function $\overline{\Phi}(u)$ subject to possible constraints concerning the decision $u$.

Except simple cases, the both operations require the application of respective computational method to obtain an approximate value of the integral (7.3) with a particular numerical value $u$, and to find the successive approximations of the value $u^*$ minimizing the function $\overline{\Phi}(u)$. The respective computer program consists then of two cooperating subprograms. In one step of the subprogram determining the successive approximations $u_m$ of the value $u^*$, the full subprogram of calculating the integral (7.3) for $u = u_m$ should be executed.

For the plant

$$y_n = \Phi(u_n, z_n, c) \tag{7.4}$$

with the disturbance $z_n$ which can be measured, the value $u_n = \Psi(z_n)$ minimizing

$$Q = \int_C \varphi[\Phi(u_n, z_n, c), y^*] f_c(c) dc \tag{7.5}$$

is determined in each interval of the control, and the program calculating $u_n$ described above is the program for the computer controlling the plant in an open-loop system. A simplified block scheme of the control algorithm (or real-time control program), i.e. the algorithm finding the decision $u_n$ in the $n$-th interval is presented in Fig. 7.1

The determination of $u_{n,m+1}$ using $u_{nm}$ is performed according to a proper recursive procedure. The condition of the stop may be e.g.:

$$||u_{nm} - u_{n,m+1}|| \le \alpha$$

where $\alpha$ is a given number. As it was already said, in simple cases it is possible to obtain an analytical solution. Let us consider a linear-quadratic problem for the plant with $p$ inputs and the single output

$$y = c^{\mathrm{T}} u,$$

and quadratic performance index $\varphi(y, y^*) = (y - y^*)^2$. Then we have

$$Q = \mathrm{E}(\underline{c}^{\mathrm{T}} u - y^*)^2 = u^{\mathrm{T}} \mathrm{E}(\underline{c}\,\underline{c}^{\mathrm{T}}) u - 2 y^* \mathrm{E}(\underline{c}^{\mathrm{T}}) u + (y^*)^2$$

where the operations E concern the particular entries of the matrices. The value $u^*$ can be determined from the equation

$$\operatorname*{grad}_{u} Q = 2\mathrm{E}(\underline{c}\,\underline{c}^{\mathrm{T}})u - 2y^{*}\mathrm{E}(\underline{c}) = \overline{0}$$

where $\overline{0}$ denotes the vector with zero components.

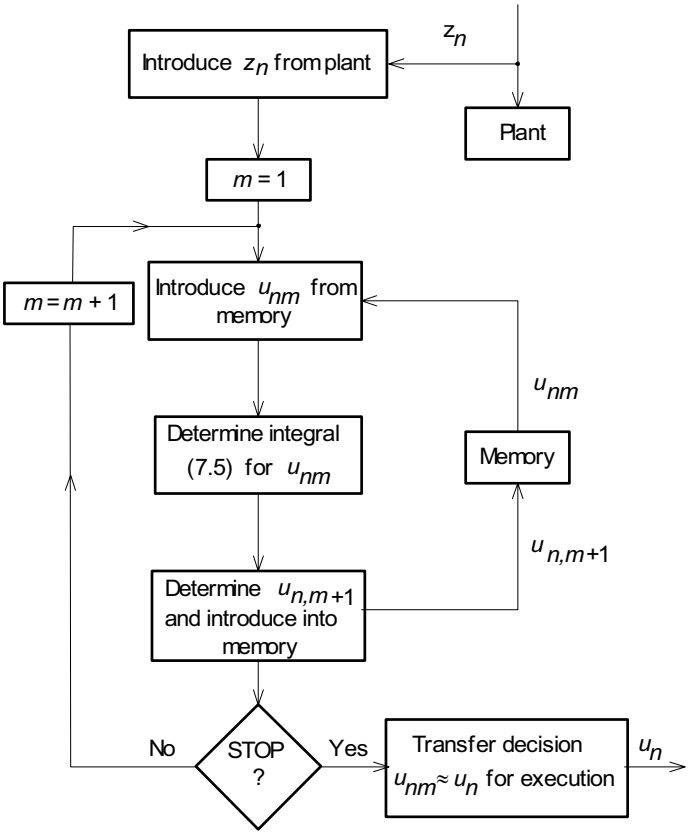


**Fig. 7.1.** Block scheme of control algorithm in the case under consideration

Consequently,

$$u^{*} = [\mathrm{E}(\underline{c}\,\underline{c}^{\mathrm{T}})]^{-1}\mathrm{E}(\underline{c})y^{*} \tag{7.6}$$

where $\mathrm{E}(\underline{c}\,\underline{c}^{\mathrm{T}})$ is the matrix of the second order moments of the components of the vector $\underline{c}$, i.e. a symmetric matrix containing in the principal diagonal the second order moments $\mathrm{E}[(\underline{c}^{(i)})^2]$, $i \in \overline{1,p}$, and the mixed moments $\mathrm{E}[\underline{c}^{(i)}\underline{c}^{(j)}]$, $i \in \overline{1,p}$, $j \in \overline{1,p}$ as other entries of the matrix. As it

is seen, in the case considered the knowledge of the probability density $f_c(c)$ is not needed; it is sufficient to know only the moments occurring in the formula (7.6). In particular, for one-dimensional plant $y = cu$

$$u^* = \frac{y^* \mathrm{E}(\underline{c})}{\mathrm{E}(\underline{c}^2)} = \frac{y^* \tilde{c}}{(\tilde{c})^2 + \sigma_c^2} \tag{7.7}$$

where $\tilde{c}$ denotes the expected value and $\sigma_c^2$ is the variance of the variable $\underline{c}$, i.e. $\sigma_c^2 = \mathrm{E}[(\underline{c} - \tilde{c})]^2$.

For the probabilistic optimization problem not only the interpretation of the probabilistic assumption explained above but also the proper interpretation of the result $u^*$ is very important. Nothing can be said about the quality of the decision $u^*$ applied in one individual plant but for a sufficiently large set of the plants (7.1) with different values $c$ chosen randomly from the set of values mentioned above, the value $u^*$ is the best *in the average*. It means that if in each plant from the considered set of the plants the decision $u^*$ is applied then the arithmetic mean of the values $\varphi$ for all the plants from this set will be the smallest. This statement is true under two conditions: the random choosing of the value $c$ does not change the distribution $f_c(c)$ and the number of the plants is sufficiently great as to accept the arithmetic mean as an approximation of the expected value.

The probabilistic optimization problem is similar for the plant $y_n = \Phi(u_n, z_n)$ in which the disturbance $z_n$ is not measured but one may assume that for each $n$ the vector $z_n \in Z$ is the value of a random variable $\underline{z}_n$ described by the probability distribution $f_z(z)$. It means that the values $z_n$ are randomly chosen from the same population characterized by the density $f_z(z)$. In other words, these are the randomly chosen values of the variable $\underline{z}$ with the distribution $f_z(z)$. The disturbance $z_n$ may be also denoted by $c_n$ and called a time-varying parameter of the plant. So we can say that randomly changing disturbances act on the plant or that this is the plant with randomly changing parameter. The sequence $\underline{z}_n$ or the function which to the moments $n$ assigns the respective random variables $\underline{z}_n$ is called a *discrete stochastic process*. That is why in the case under consideration we often speak about the plant with a stochastic disturbance.

The decision making (control) problem consists now in the determina-

tion of the decision $u^*$ minimizing

$$Q = \underset{\underline{z}}{\mathrm{E}}(\varphi(\underline{y}, y^*)] = \int_Z \varphi[\Phi(u,z), y^*] f_z(z)dz \,. \tag{7.8}$$

From both the formal and computational point of view this is the problem identical with the minimization of the performance index (7.3) for the plant with the constant random parameter. However, the interpretation of the result of the probabilistic optimization is now different: the value $u^*$ is the **optimal in the average with respect to time** and not **in the average with respect to a set**. It means that if in the plant the decision $u^*$ constant in the successive moments is applied then the arithmetic mean of the values $\varphi$ for a large number of the moments (i.e. in a sufficiently large time interval) will be minimal. However, nothing can be said about the quality of the decision $u^*$ in one particular moment.

Finally, let us pay attention to two other possibilities of the decision problem statement for the plant (7.1) or (7.4), i.e.

$$y = \Phi(u, z, c): \tag{7.9}$$

a) One should find $u^*$ such that $\mathrm{E}(\underline{y}) = y^*$.

b) One should find

$$u_b^* = \arg\max_u f_y(y^*; u)$$

where $f_y(y; u)$ is the probability density of the variable $\underline{y}$ for the fixed value $u$.

In the version a), for the decision $u^*$ the expected value of the output is equal to the required value $y^*$, and in the version b) the decision $u_b^*$ maximizes the value of the probability density for $y = y^*$. If the distribution $f_y(y)$ is symmetric, the value $y$ maximizing $f_y(y)$ is equal to $\mathrm{E}(\underline{y})$. For the plant (7.9), by solving the equation

$$\mathrm{E}(\underline{y}; u, z) = \int_C \Phi(u,z,c) f_c(c)dc \triangleq \overline{\Phi}(u,z) = y^* \tag{7.10}$$

with respect to $u$, we obtain the control algorithm in an open-loop system $u = \Psi(z)$. In the second problem formulation, the control algorithm can be

obtained as a result of maximization of the probability density $f_y(y^*; u, z)$ with respect to $u$:

$$u_b = \arg\max_u \Phi_b(u, z) \triangleq \Psi_b(z) \tag{7.11}$$

where $\Phi_b(u, z) = f_y(y^*; u, z)$ and the density $f_y$ can be obtained by using the known function $\Phi$ and the density $f_c(c)$ and applying a known way of the determination of the distribution of a random variable which is a function of another random variable. The functions $\overline{\Phi}$ and $\Phi_b$ are the results of two ways of the plant determinization, the functions $\Psi$ and $\Psi_b$ are the decision algorithms found by using the knowledge representation of the plant, i.e. are based on the knowledge of the plant

$$\text{KP} = <\Phi, f_c> \tag{7.12}$$

in the version a) and b), respectively. In this case the *knowledge of the plant* (KP) contains the function $\Phi$ and the probability distribution $f_c$. The solution of the equation

$$\Phi(u, z, c) = y^* \tag{7.13}$$

with respect to $u$ yields the relationship

$$u = \Phi_d(z, c) \tag{7.14}$$

which together with $f_c$ may be treated as a *knowledge of the decision making* (KD)

$$\text{KD} = <\Phi_d, f_c>. \tag{7.15}$$

The relationship (7.14) together with the additional characteristic of the parameter $c$ in the form of $f_c$ may be also called a *probabilistic control algorithm* (or a probabilistic decision algorithm) in an open-loop system, based on the knowledge of the plant. The determinization of this algorithm leads to two different versions of the deterministic control algorithm, corresponding to the versions a) and b) or (7.10) and (7.11) in the case of the plant determinization:

a) $$u_d = \text{E}(\underline{u}; z) = \int_C \Phi_d(z, c) f_c(c) dc \triangleq \Psi_d(z), \tag{7.16}$$

b) $$u_{bd} = \arg\max_{u} f_u(u; z) \triangleq \Psi_{bd}(z) \tag{7.17}$$

where $f_u(u; z)$ is the probability density of the variable $\underline{u}$ which can be obtained on the basis of KD.

**Theorem 7.1.** In general, for the plant described by KP (7.12) and for KD (7.15),

$$\Psi(z) \neq \Psi_d(z), \qquad \Psi_b(z) \neq \Psi_{bd}(z). \qquad \square$$

The theorem can be proved by an example showing that in a particular case the inequalities presented above are satisfied (see Example 7.1). The theorem means that the control decisions $u$ determined via the different ways of the determinization may be different. It is worth paying attention to the analogy between the relational plant considered in Chap. 6 and the plant described in the probabilistic way. The knowledge of the plant $R(u, y, z)$ corresponds now to KP (7.12), the knowledge of the decision making $R_d(z, u)$ corresponds to KD (7.15), two concepts of the determinization are analogous as well, in particular Theorem 7.1 is analogous to Theorem 6.1.

**Example 7.1.** Let $u, y, z \in R^1$ (one-dimensional variables) and

$$y = cu + z.$$

Let us find the deterministic decision algorithm via the determinization of the plant.

In the version a), according to (7.10)

$$\mathrm{E}(\underline{y}; z) = u\mathrm{E}(\underline{c}) + z = y^*$$

and

$$u = \Psi(z) = (y^* - z)[\mathrm{E}(\underline{c})]^{-1}.$$

In the version b), according to (7.11)

$$u_b = \Psi_b(z) = \arg\max_{u} f_c\left(\frac{y^* - z}{u}\right)\frac{1}{|u|}.$$

From the equation $cu + z = y^*$ we obtain the probabilistic decision algorithm $\Phi_d$:

$$u = \Phi_d(z,c) = \frac{y^* - z}{c}.$$

Applying the determinization of this algorithm gives two versions of the deterministic decision algorithm.

In the version a), according to (7.16)

$$u_d = \mathrm{E}(\underline{u}; z) = \Psi_d(z) = (y^* - z)\mathrm{E}(\underline{c}^{-1}) \neq \Psi(z).$$

In the version b), according to (7.17)

$$u_{bd} = \Psi_{bd}(\mathrm{z}) = \arg\max_u f_c\left(\frac{y^* - z}{u}\right)\frac{|y^* - z|}{u^2} \neq \Psi_b(z). \qquad \square$$

## 7.2 Basic Problems for Static Plant and Non-parametric Uncertainty

Now we shall consider an uncertainty referring to the description of the plant as a whole and not referring to the unknown constant or time-varying parameters in the known description, as it was considered in Sect. 7.1. For the plant described by the relation $R(u, y)$ presented in Sect. 6.1 we can speak about an **additional characteristic of the uncertainty** which may consist in giving some preferences or additional evaluations for different points $(u, y)$ in the set $R(u, y)$ of all possible pairs $(u, y)$ which may appear in this plant. More generally, we may accept a possibility of appearing the pairs $(u, y)$ not belonging to $R$, which means that the fact $(u, y) \in R(u, y)$ is not certain. Assume that $(u, y)$ are values of the pair of continuous random variables $(\underline{u}, \underline{y})$. The property $(u, y) \in R$ is then a random fact and its truth is a random event characterized by the probabilities

$$p_1 = P[(\underline{u}, \underline{y}) \in R(u, y)], \qquad p_2 = P[(\underline{u}, \underline{y}) \notin R(u, y)] = 1 - p_1. \qquad (7.18)$$

Then, the simplest description of the uncertain plant consists in giving the relation $R(u, y)$ and the probabilities (7.18). The more precise or more exact description consists in giving a joint probability density $f(u, y)$ defined in the whole set $U \times Y$. The probability density $f(u, y)$ is an additional characteristic of the uncertainty referring to $R$, under the assumption that the points $(u, y)$ may belong to $R$ only, i.e. $f(u, y) \equiv 0$ for $(u, y) \in U \times Y - R$ (a complement of the set $R$). If the pairs not belonging to $R$ may occur as

well, then the joint probability density $f(u, y)$ characterizes the pairs $(u, y)$ in $R$ and in its complement as well. Consequently, the determination of the relation $R$ is no more needed. Now the density $f(u, y)$ is the knowledge representation of the plant. This is a product of the density $f_u(u)$ and the conditional density $f_y(y|u)$, i.e. the density of the variable $\underline{y}$ for the fixed value $u$:

$$f(u, y) = f_u(u) f_y(y|u) \tag{7.19}$$

where

$$f_u(u) = \int_Y f(u, y)dy\,, \qquad f_y(y \mid u) = \frac{f(u, y)}{\int_Y f(u, y)dy}\,. \tag{7.20}$$

More precisely speaking, the acting of the plant itself is described only by the density $f_y(y|u)$ characterizing a dispersedness of possible outputs $y$ for the fixed value $u$, and $f_u(u)$ characterizes a dispersedness of possible inputs of the plant. Knowing the distribution $f_y(y|u)$ it is possible to formulate and solve the problem of the determination of $u^*$ minimizing the performance index

$$Q = \underset{\underline{y}}{\mathrm{E}}[\varphi(\underline{y}, y^*)] = \int_Y \varphi(y, y^*) f_y(y \mid u)du\,.$$

This problem is analogous to the problem of the minimization of the index (7.3) in the parametric case. For the plant with the disturbance $z$ with the known density $f(u|y; z)$ for the fixed $z$, one can formulate the optimization problem analogous to the minimization (7.5) and consisting in the determination of $u = \Psi(z)$ minimizing the performance index

$$Q = \underset{\underline{y}}{\mathrm{E}}[\varphi(\underline{y}, y^*)] = \int_Y \varphi(y, y^*) f_y(y \mid u; z)dy\,.$$

Let us consider now two versions of the decision problem with the given required value $y^*$, analogous to the versions a) and b) in Sect. 7.1.

**Decision making problem** for the given $f(u, y; z)$, $z$ and $y^*$:

a) One should find $u$ for which the expected value of the output is equal to the required value, i.e. $\mathrm{E}(\underline{y} \mid u; z) \overset{\Delta}{=} \tilde{y}$ is equal to $y^*$ or, according to (7.20)

$$\tilde{y} = \frac{\int_Y y f(u, y; z)\, dy}{\int_Y f(u, y; z)\, dy} \triangleq \Phi(u, z) = y^* . \tag{7.21}$$

Solving the equation (7.21) with respect to $u$ we obtain the decision algorithm $u = \Psi(z)$.

b) One should determine

$$u_b = \arg\max_u \Phi_b(u, z) \triangleq \Psi_b(z) \tag{7.22}$$

where

$$\Phi_b(u, z) = f_y(y^* \mid u; z) = \frac{f(u, y^*; z)}{\int_Y f(u, y; z)\, dy} .$$

The functions $\Phi$ and $\Phi_b$ are the results of two versions concerning the plant determinization, and $\Psi$ and $\Psi_b$ are the decision algorithms based on the knowledge of the plant KP = $f(u, y; z)$ in the version a) and b), respectively. Putting $y^*$ into the function $f(u, y; z)$ we obtain the function $f(u, y^*; z) \triangleq f_{ud}(u; z)$ which may be treated as the knowledge of the decision making KD or the probabilistic decision algorithm in our case. The determinization of this algorithm gives two versions of the deterministic algorithm, corresponding to the versions a) and b) of the plant determinization:

a)
$$u_d = \mathrm{E}(\underline{u} \mid y^*; z) = \frac{\int_U u f_{ud}(u; z)\, du}{\int_U f_{ud}(u; z)\, du} \triangleq \Psi_d(z), \tag{7.23}$$

b)
$$u_{bd} = \arg\max_u \Phi_{bd}(u, z) \triangleq \Psi_{bd}(z) \tag{7.24}$$

where

$$\Phi_{bd}(u, z) = f_u(u \mid y^*; z) = \frac{f_{ud}(u; z)}{\int_U f_{ud}(u; z)\, du} .$$

The formulas (7.21), (7.22), (7.23) and (7.24) are analogous to the respective formulas (7.10), (7.11), (7.16), (7.17) for the parametric uncertainty. It is worth noting that as KP and KD it is sufficient to accept $f_y(y \mid u; z)$ and $f_u(u \mid y; z)$, respectively. In the formulas (7.22), (7.23) and (7.24) these probability densities should be determined as marginal densities from the joint densities

$$\text{KP} = <f(u, y; z)>, \qquad \text{KD} = <f_{ud}(u; z)>.$$

**Theorem 7.2.** In general, for the plant described by KP = $f(u, y; z)$ and for KD = $f_{ud}(u; z)$

$$\Psi(z) \neq \Psi_d(z), \qquad \Psi_b(z) \neq \Psi_{bd}(z)\ . \qquad \square$$

It may be shown by using the result in Example 7.1. Let the density $f(u, y; z)$ be such that

$$f_y(y \mid u; z) = f_c(\frac{y-z}{u})\frac{1}{|u|}, \qquad f_u(u \mid y; z) = f_c(\frac{y-z}{u})\frac{|y-z|}{u^2}$$

where $f_c(c)$ is the density such as in Example 7.1. Then, according to the result in this example

$$\Psi(z) \neq \Psi_d(z) \quad \text{and} \quad \Psi_b(z) \neq \Psi_{bd}(z).$$

Let us pay attention to another decision problem which consists in the determination of a probabilistic decision algorithm in the form of $f_u(u;z)$ for the given required distribution $f_y(y)$ and the given $f_u(u|y;z)$ or $f_y(y|u;z)$ characterizing the plant.

**Decision making problem** with the given $z$ and $f_y(y)$:

1. One should find $f_u(u;z)$ for the given $f_u(u|y;z)$.
2. Under the assumption that $z$ is a value of a random variable, one should determine $f_u(u;z) = f_u(u|z)$ for the given $f_z(z)$ and $f_y(y|u;z)$.

In the first case

$$f_u(u;z) = \int_Y f_u(u \mid y;z) f_y(y) dy\,, \tag{7.25}$$

under the assumption that the equality (7.25) cannot be satisfied for another $f_y(y)$, i.e. that the integral equation

$$\int_Y f_u(u \mid y; z) f_y(y) dy = \int_Y f_u(u \mid y; z) \bar{f}_y(y) dy$$

with the unknown function $\bar{f}_y(y)$ has only one solution $\bar{f}_y(y) = f_y(y)$.

In the second case $f_u(u;z) = f_u(u|z)$ is determined by the integral equation

$$f_y(y) = \int_{U} \int_{Z} f_z(z) f_u(u \mid z) f_y(y \mid u; z) \, du \, dz \,. \tag{7.26}$$

This problem is analogous to that presented in Sect. 6.3 for the relational plant and the fixed $z$. The relation $R(u, y; z)$ corresponds to the description of the plant in the form of the probability density $f_y(y|u;z)$ or $f_u(u|y;z)$ characterizing a dispersedness of $y$ for the fixed $u$ or on the contrary. This is the more precise information on the plant than in the relation case, and makes it possible to replace the requirement $y \in D_y$ for the given $D_y$ by the more precise requirement in the form of the given distribution $f_y(y)$ characterizing a dispersedness of the values which may occur at the output of the plant. In this way, for any set $D_y$ we determine the probability that $\underline{y} \in D_y$. Consequently, the result having the form of the set of decisions $D_u(z)$ or the relational decision algorithm (6.19) is replaced by the result in the form $f_u(u;z)$ or by the probabilistic decision algorithm which for the measured value $z$ gives the probability distribution for the decision $u$. A concrete, particular decision may be chosen randomly from the set $U$, according to the probability distribution $f_{ud}(u;z)$. For the realization of such a concept, the application of a random numbers generator is needed. One may also perform a determinization of the probabilistic algorithm and determine

$$u = \mathrm{E}(\underline{u}\,; z) \triangleq \Psi(z) \,.$$

The application of the algorithm $\Psi(z)$ does not, however, assure the satisfaction of the requirement in the form of the probability distribution $f_y(y)$. In the prescriptive approach, KD described by an expert in the form of $f_{ud}(u;z)$ will give the same result as the descriptive approach, i.e. will assure the satisfaction of the requirement in the form of $f_y(y)$, if $f_{ud}(u;z)$ is equal to the density $f_u(u;z)$ presented by the formula (7.25) in the first case, or if it satisfies the equation (7.26) in the second case.

## 7.3 Control of Static Plant Using Results of Observations

Let us consider a static plant

$$y = \Phi(u,c)$$

with the unknown parameter $c$ and the known probability distribution $f_c(c)$. Assume that it is possible to increase the initial information on the unknown parameter in the form $f_c(c)$ as a result of current observations during the control process and consequently, it is possible to improve successively the control decisions. Speaking about the observation we have in mind the measuring of the parameter $c$ in the presence of random disturbances (random noises).

Let

$$w_n = h(c, z_n) \tag{7.27}$$

denote the result of the $n$-th measurement, dependent on the value of the measured parameter $c$ (Fig. 7.2), $h$ is a function determining this dependence, $z_n$ denotes the disturbance, and

$$\overline{w}_n = (w_1, w_2, ..., w_n)$$

is a sequence of the results of measurements till the moment $n$, used to the determination of the control decision according to the control algorithm $\Psi_n$, i.e.

$$u_n = \Psi_n(\overline{w}_n)$$

which should be properly designed.

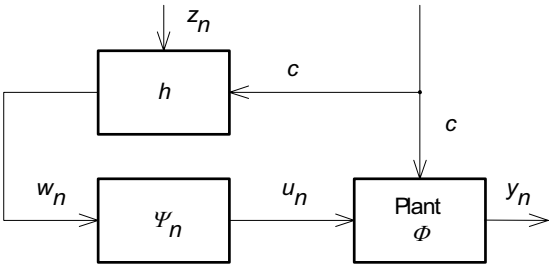


**Fig. 7.2.** Block scheme of the control system under consideration

We assume that $z_n$ is varying in a random way or, more exactly speak-

ing, that for every $n$ the vector $z_n$ is a value of the random variable $\underline{z}_n$ with the probability density $f_z(z)$, the same for different $n$, and that the variables $\underline{z}_n$ are stochastically independent for different $n$. The latter assumption means that for every $n$ the joint density of the pair $(\underline{z}_n, \underline{z}_{n+1})$ is equal to the product of marginal densities for $\underline{z}_n$ and $\underline{z}_{n+1}$, i.e. is equal to $f_z(z_n) f_z(z_{n+1})$. We shall present two ways of the determination of the algorithm $\Psi_n$: indirect and direct approach.

### 7.3.1 Indirect Approach

The idea consists in a decomposition of the problem under consideration into two easier problems:
1. Determination of the control for the known parameter $c$.
2. Estimation of the unknown parameter using the result of the observation.
The first problem has been considered in Chaps. 3 and 4. It consists in finding the value $u$ for which $y = y^*$ or, more generally, the value $u$ which minimizes the given performance index $\varphi(y, y^*)$. For the given parameter $c$, this value depends on $c$. Consequently, the result of this problem is the determination of the dependence of $u$ upon $c$, which we shall denote by $H$, i.e. $u = H(c)$. The second problem consists in the determination of the estimator $c_n$ for the parameter $c$ on the basis of $\overline{w}_n$

$$c_n = G_n(\overline{w}_n)$$

where $G_n$ denotes the *estimation algorithm*. Substituting the estimate $c_n$ into $H$ in the place of $c$ gives the control algorithm $\Psi_n$:

$$u_n = H[G_n(\overline{w}_n)] \stackrel{\Delta}{=} \Psi_n(\overline{w}_n). \tag{7.28}$$

If the estimator is consistent (i.e. $\underline{c}_n$ converges in probabilistic sense to $c$ for $n \to \infty$) and $H(c)$ is a continuous function, then $\underline{u}_n$ converges to $H(c)$, i.e. to the decision which we would determine for the known parameter $c$. Then, the control algorithm (Fig. 7.3) consists of two blocks $G_n$ and $H$, and consequently, the program in the controlling computer contains two parts: the part determining the estimation of the unknown parameter and the part finding the decision $u$ for the known parameter. The composition of these two subalgorithms leads to one control algorithm (control pro-

gram) determining $u_n$ on the basis of $\overline{w}_n$. Usually we try to present the estimation algorithm in a recursive form, i.e. in the form of a formula showing how to calculate $c_{n+1}$ on the basis of $c_n$ and $w_n$. Then it is necessary to keep only $c_n$ in the computer memory to calculate the next estimation.

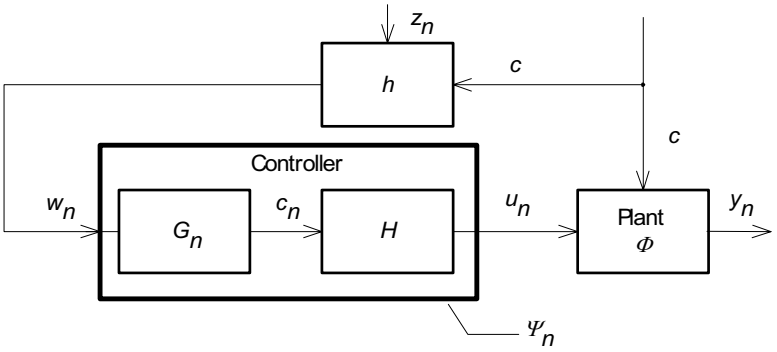


**Fig. 7.3.** Control system with two blocks of control algorithm

A universal method for the determination of the estimator *G* under the assumption that *c* is a value of a random variable $\underline{c}$ and with the known distribution $f_c(c)$ is a *minimum risk method.* To evaluate the quality of the estimation, let as introduce so called *loss function* $L(c, c_n)$ whose value is equal to zero if and only if $c = c_n$, and for $c \neq c_n$ is positive and evaluates the distance between $c$ and $c_n$. Most often $L(c, c_n) = \|c - c_n\|$ or $L(c, c_n) = \|c - c_n\|^2$. The expected value of the loss function

$$R = \mathrm{E}[L(\underline{c}, \underline{c}_n)]$$

is called a *mean risk*.

**Estimation problem:** For the given function *h* describing the influence of the noise on the result of the measurement $w_n$, the given densities $f_c(c)$, $f_z(z)$ and the loss function *L*, one should determine the estimation algorithm $G_n(\overline{w}_n)$ minimizing the mean risk *R*.

According to the definition

$$R = \int_C \int_{\overline{W}_n} L[c, G_n(\overline{w}_n)] f(c, \overline{w}_n) dc d\overline{w}_n \tag{7.29}$$

where $C$ is a space of vectors $c$, $\overline{W}_n$ is a set of all sequences $\overline{w}_n$, $f(c, \overline{w}_n)$ is a joint probability density of the variables $\underline{c}$ and $\underline{\overline{w}}_n = (\underline{w}_1, \underline{w}_2, ..., \underline{w}_n)$; $\underline{w}_i = h(c, \underline{z}_i)$. The formula (7.29) defines a functional which to the functions $G_n$ assigns the numbers $R$. The determination of the optimal algorithm $G_n$ is then a problem of a functional minimization. It may be shown that the optimal algorithm $G_n$ can be obtained as a result of minimization with respect to $c_n$ of so called *conditional risk*

$$r(c_n, \overline{w}_n) = \int_C L(c, c_n) f_c(c \mid \overline{w}_n) dc \,, \tag{7.30}$$

i.e. minimization with respect to $c_n$ of the conditional expected value for the given $\overline{w}_n$. In the relationship (7.30), $f_c(c| \overline{w}_n)$ denotes the conditional probability density. According to *Bayesian rule*

$$f_c(c \mid \overline{w}_n) = \frac{f_c(c) f_{wn}(\overline{w}_n \mid c)}{f_n(\overline{w}_n)} \tag{7.31}$$

where $f_{wn}$ denotes the conditional density of $\underline{\overline{w}}_n$ for the given $c$, and $f_n$ denotes the marginal density of $\underline{\overline{w}}_n$. Since $f_n(\overline{w}_n)$ does not depend on $c$ and $c_n$, it is sufficient to minimize the function

$$r(c_n, \overline{w}_n) f_n(\overline{w}_n) = \int_C L(c, c_n) f_c(c) f_{wn}(\overline{w}_n \mid c) dc \triangleq \bar{r}(c_n, \overline{w}_n) \tag{7.32}$$

with respect to $c_n$. Since $\underline{z}_n$ are stochastically independent for different $n$, the same may be said for $\underline{w}_n$. Then

$$f_{wn}(\overline{w}_n \mid c) = \prod_{i=1}^{n} f_w(w_i \mid c) \tag{7.33}$$

where $f_w(w_i|c)$ (i.e. the conditional density of the individual variable $\underline{w}_i$) may be determined for the given function (7.27) and the known $f_c(c)$. The procedure of the determining of the estimation algorithm is the following:

1. For the given $h$ and $f_c$ we find $f_{wn}$ according to (7.33).
2. We determine $\bar{r}(c_n, \overline{w}_n)$ according to (7.32).

3. We minimize $\bar{r}(c_n, \overline{w}_n)$ with respect to $c_n$ and obtain $c_n = G_n(w_n)$.

The computational problems may be similar to those connected with the determination and minimization of the integral (7.3). In simple cases it is possible to obtain the result in an analytical form. For $L(c, c_n) = -\delta(c - c_n)$ (Dirac delta), the risk (7.30) is reduced to

$$r(c_n, \overline{w}_n) = -f_c(c_n \mid \overline{w}_n) .$$

The optimal estimate $c_n$ is then the value $c$ maximizing the density (7.31), i.e.

$$c_n = \arg\max_c f_c(c) f_{wn}(\overline{w}_n \mid c) \triangleq G_n(\overline{w}_n) . \tag{7.34}$$

In the above considerations we use *a priori* distribution $f_c(c)$ presenting the information on $c$ before the observations have been started, and *a posteriori* distribution $f_c(c \mid \overline{w}_n)$ presenting the information on $c$ after $n$ measurements. The minimum risk method in the case (7.34) may be shortly called a *maximum probability method.* The name is fully justified when $c$ is a discrete random variable. In the case of a continuous random variable under consideration, it is a method using maximum *a posteriori* probability density and consisting in the determination of the estimate $c_n$ maximizing this density. It is interesting and useful to compare this method with a maximum likelihood method which is used when $f_c(c)$ is unknown or when there are no reasons to assume that the value $c$ has been taken randomly from any set, i.e. when the probability distribution does not exist. The maximum likelihood method consists in finding the estimate

$$c_n = \arg\max_c f_{wn}(\overline{w}_n; c) , \tag{7.35}$$

i.e. the estimate maximizing so called likelihood function $f_{wn}(\overline{w}_n; c) \triangleq L(c; \overline{w}_n)$. The form of the function $f_{wn}(\overline{w}_n; c)$ is the same as $f_{wn}(\overline{w}_n \mid c)$, it is then the density of $\overline{w}_n$ for the fixed $c$, which cannot be called a conditional density if $c$ is not a value of a random variable. The comparison of the estimates (7.34) and (7.35) shows the difference between the most probable and the most likely estimation.

**Example 7.2.** Let us determine the control algorithm $\Psi_n(\overline{w}_n)$ for the one-dimensional plant $y_n = cu_n$ where $c$ is measured with an additive noise, i.e.

$w_n = c + z_n$. Assume that $\underline{c}$ has Gaussian distribution with the expected value $\tilde{c}$ and the variance $\sigma_c^2$, i.e.

$$f_c(c) = \frac{1}{\sigma_c\sqrt{2\pi}} \exp[-\frac{(c-\tilde{c})^2}{2\sigma_c^2}] \triangleq N(\tilde{c}, \sigma_c), \tag{7.36}$$

and $z_n$ has Gaussian distribution with the expected value $\tilde{z} = 0$ and the variance $\sigma_z^2$, i.e.

$$f_z(z) = \frac{1}{\sigma_z\sqrt{2\pi}} \exp(-\frac{z^2}{2\sigma_z^2}).$$

Assume the performance index

$$\varphi(y, y^*) = (y - y^*)^2$$

which is reduced to the requirement $y = y^*$. Then

$$u = H(c) = \frac{y^*}{c}.$$

Let us find the most probable estimate $c_n$. For this purpose one should determine $f_{wn}(\overline{w}_n \mid c)$. According to (7.33) for $w_i = c + z_i$ we have

$$f_{wn}(\overline{w}_n \mid c) = \prod_{i=1}^{n} f_z(w_i - c) = \frac{1}{(\sigma_z\sqrt{2\pi})^n} \exp[-\frac{1}{2\sigma_z^2}\sum_{i=1}^{n}(w_i - c)^2]. \tag{7.37}$$

After substituting (7.36) and (7.37) into (7.34) and omitting the coefficients independent on $c$ we obtain

$$c_n = \arg\min_c [\frac{(c-\tilde{c})^2}{2\sigma_c^2} + \frac{1}{2\sigma_z^2}\sum_{i=1}^{n}(w_i - c)^2]. \tag{7.38}$$

Differentiating the function in the bracket with respect to $c$ and equating the derivative to zero we obtain the following estimation algorithm:

$$c_n = \frac{\tilde{c} + (\frac{\sigma_c}{\sigma_z})^2 \sum_{i=1}^{n} w_i}{1 + (\frac{\sigma_c}{\sigma_z})^2 n}. \tag{7.39}$$

If the ratio of the variance of $c$ to the variance of the noise is small and the number of measurements $n$ is small then $c_n \approx \tilde{c}$, i.e. we accept that the unknown parameter is approximately equal to the expected value $\tilde{c}$, and the influence of the measurement data is small. If this ratio is great then for large $n$

$$c_n \approx \frac{1}{n}\sum_{i=1}^{n} w_i ,$$

i.e. we accept that the approximate value of the unknown parameter is equal to the arithmetic mean of the results of the measurement, and the knowledge of $\tilde{c}$, $\sigma_c$ and $\sigma_z$ does not take a role. The formula (7.39) shows how to use both the *a priori information* ($\tilde{c}$, $\sigma_c$ and $\sigma_z$) and the *current information* obtained as a result of the measurements ($w_1$, $w_2$, ..., $w_n$) to determine the estimate minimizing the mean risk. □

The result (7.39) can be extended to an important case of a linear multi-input plant

$$y_n = c^{\mathrm{T}} u_n = c_1 u_n^{(1)} + c_2 u_n^{(2)} + \ldots + c_p u_n^{(p)}$$

with the vector of the unknown parameters $c^{\mathrm{T}} = [c_1\ c_2\ \ldots c_p]$. The distribution of the additive noise $\underline{z}_n$ is Gaussian as in the previous case and the vector $c$ has the multi-dimensional Gaussian distribution

$$f_c(c) = \frac{1}{2\pi^{\frac{p}{2}} (\det M)^{\frac{1}{2}}} \exp[-\frac{1}{2}(c-\tilde{c})^{\mathrm{T}} M^{-1}(c-\tilde{c})]$$

where $\tilde{c}$ is a vector of the expected values $\mathrm{E}(\underline{c}^{(i)})$ and

$$M = [\mathrm{E}\{(\underline{c}^{(i)} - \tilde{c}^{(i)})(\underline{c}^{(j)} - \tilde{c}^{(j)})\}]_{\substack{i=1,\ldots,p \\ j=1,\ldots,p}}$$

is a covariance matrix (see e.g. [14]). Let us note that for the single-input plant considered above, the maximum likelihood method is reduced to a *least square method*, i.e. to the determination of

$$c_n = \arg\min_c \sum_{i=1}^{n} (w_i - c)^2 .$$

It follows from the fact that the first component in the formula (7.38) is

equal to zero. Consequently, in this case the most likely estimate is equal to the arithmetic mean of the measurement results.

### 7.3.2 Direct Approach

Now we shall not decompose our problem and find the solution via the estimation of the unknown parameter but we shall determine directly the control algorithm $\Psi_n$ in the system presented in Fig. 7.3, minimizing the value of the performance index $\varphi(y,y^*)$.

**Decision making (control) problem:** For the given $\Phi, h, f_c, f_z$ and $\varphi$ one should determine the algorithm $u_n = \Psi_n(\overline{w}_n)$ minimizing for every $n$ the probabilistic performance index

$$Q = \underset{\underline{y}}{\mathrm{E}}[(\varphi(\underline{y}_n, y^*)].$$

According to the definition

$$Q = \int_C \int_{\overline{W}_n} \varphi\{\Phi[\Psi_n(\overline{w}_n), c], y^*\} f(c, \overline{w}_n) dc d\overline{w}_n . \tag{7.40}$$

The minimization of the functional (7.40) with respect to the function $\Psi_n$ may be replaced by the minimization of the conditional expected value

$$q(u_n, \overline{w}_n) = \mathrm{E}[\varphi(\underline{y}_n, y^*) \mid \overline{w}_n] = \int_C \varphi[\Phi(u_n, c), y^*] f_c(c \mid \overline{w}_n) dc$$

with respect to $u_n$. According to Bayesian rule and after omitting the denominator in (7.31), we minimize the function

$$\overline{q}(u_n, \overline{w}_n) \triangleq q(u_n, \overline{w}_n) f_n(\overline{w}_n) = \int_C \varphi[\Phi(u_n, c), y^*] f_c(c) f_{wn}(\overline{w}_n \mid c) dc$$

with respect to $u_n$ and we obtain $u_n = \Psi_n(w_n)$.

In general, the result obtained via the indirect approach differs from the result obtained by applying the direct method. It is then the result worse in general, i.e. giving the value of the probabilistic performance index $Q$ defined by the formula (7.40) greater than the minimum value obtained by applying the algorithm obtained with the help of the direct method. However, the direct method is usually much more complicated from the computational point of view. In the linear-quadratic problems with

putational point of view. In the linear-quadratic problems with Gaussian distributions the both approaches give the same result.

## 7.4 Application of Games Theory

Let us come back to the plant

$$y_n = \Phi(u_n, z_n) \tag{7.41}$$

in which the disturbance $z_n$ is not measured. This case has been considered in Sect.7.1 under the assumption that the distribution $f_z(z)$ is given. Then the decision problem has consisted in the determination of the constant decision $u^*$ minimizing the expected value of the performance index (7.8). Let us assume now that the probability density $f_z(z)$ is not known and consequently, the determination of the decision $u^*$ minimizing the index (7.8) is not possible. Then we can apply so called game approach or, more precisely speaking, a *two-person zero-sum game theory*. One player is the controlling system generating the decision $u_n$ (Player A) while the environment generating the disturbance $z_n$ is treated as the other player (Player B). The source of the disturbances is then "personificated" (considered as a person) and treated as a partner playing with a decision maker. Let us assume that sets of possible decisions for the both partners are finite. These are the sets

$$\overline{U} = \{\overline{u}^{(1)}, \overline{u}^{(2)}, ..., \overline{u}^{(M)}\}$$

and

$$\overline{Z} = \{\overline{z}^{(1)}, \overline{z}^{(2)}, ..., \overline{z}^{(N)}\}$$

for the partner A and B, respectively, which means that $u_n \in \overline{U}$ and $z_n \in \overline{Z}$ for every $n$. If the elements in the sets are determined and the elements are marked by their indexes, it is sufficient to use the indexes $i = 1, 2, ..., M$ and $j = 1, 2, ..., N$ where $i$ denotes the $i$-th decision and $j$ denotes the $j$-th disturbance. Choosing of the values $(u_n, z_n)$ or the indexes $(i_n, j_n)$ in a successive stage is called a move. An effect of the move is the respective value of the performance index

$$\varphi(y_n, y^*) = \varphi[\Phi(u_n, z_n), y^*] \triangleq G(u_n, z_n) = v_n.$$

We assume that the loss $v_n$ of Player A is equal to the profit of Player B. For all possible moves, i.e. all pairs $(u_n, z_n)$, the function $G$ determines the following table of the respective values $v$:

| $u$ \ $z$ | 1 | 2 | ... | $j$ | ... | $N$ |
|---|---|---|---|---|---|---|
| 1 | $v_{11}$ | $v_{12}$ | ... | $v_{1j}$ | ... | $v_{1N}$ |
| 2 | $v_{21}$ | $v_{22}$ | ... | $v_{2j}$ | ... | $v_{2N}$ |
| ⋮ | ⋮ | ⋮ | ⋮ | ⋮ | ⋮ | ⋮ |
| $i$ | $v_{i1}$ | $v_{i2}$ | ... | $v_{ij}$ | ... | $v_{iN}$ |
| ⋮ | ⋮ | ⋮ | ⋮ | ⋮ | ⋮ | ⋮ |
| $M$ | $v_{M1}$ | $v_{M2}$ | ... | $v_{Mj}$ | ... | $v_{MN}$ |

In the table $v_{ij} = G(\bar{u}^{(i)}, \bar{z}^{(j)})$. In a concrete game this table may be given directly and not by the function $G$ as in the decision problem under consideration. The value $v_n$ may denote an amount of money which Player A pays to Player B if A has chosen the index $i$ and B has chosen the index $j$. Of course, the players do not know each other choices before making their own choice. The above table, i.e.

$$M = [v_{ij}]_{\substack{i=1,2,...,M \\ j=1,2,...,N}}$$

is called a *payoff matrix*. The game denotes a sequence of successive moves. For the large number of moves, Player B would like the sum of payments (his winnings) to be as great as possible, while Player B would like this sum (his loss) to be as small as possible. Of course, Player A would take part in the game with such rules only if the chances were not evident, i.e. if some numbers in the payoff matrix were negative which would mean that in fact in such a case it is Player B who pays. A strategy of Player A is presented by a sequence of probabilities

$$(p_1, p_2, ..., p_M) \stackrel{\Delta}{=} p$$

according to which he will make the choices in the successive stages. Let us denote by

$$(q_1, q_2, ..., q_N) \stackrel{\Delta}{=} q$$

the respective strategy of Player B. A main idea of the game approach

from Player A point of view consists in the assumption that Player B chooses his strategy $q$ in such a way as to maximize the expected winning. Then Player A determines his strategy $p$ in such a way as to minimize the expected winning of Player B (the expected payment), i.e. to minimize the function

$$\max_{q_1,\ldots,q_N} (\sum_{i=1}^{M} p_i \sum_{j=1}^{N} v_{ij} q_j) = \max_{q_1,\ldots,q_N} \sum_{j=1}^{N} (\sum_{i=1}^{M} v_{ij} p_i) q_j \tag{7.42}$$

with respect to $p_1, p_2, \ldots, p_M$, subject to constraints

$$\bigwedge_i (p_i \geq 0), \qquad \sum_{i=1}^{M} p_i = 1. \tag{7.43}$$

The identical constraints concerning $q$ should be taken into account for the maximization with respect to $q$. As it is seen from (7.42), Player A, not knowing the strategy of Player B, chooses the strategy $p$ in such a way as to minimize the worst situation.

Using the game approach we can formulate the following decision problem for the plant (7.41).

**Decision making (control) problem:** For the given function $\Phi$, the set $\overline{U}$ and the set $\overline{Z}$ one should determine the strategy $p_1, p_2, \ldots, p_M$ minimizing (7.42) and satisfying the constraints (7.43) where

$$v_{ij} = \varphi [\Phi(\bar{u}^{(i)}, \bar{z}^{(j)}), y^*]. \tag{7.44}$$

Let us note that with the constraints concerning $q$, the result of the maximization with respect to $q$ in (7.42) is such that $q_j = 1$ if the coefficient at $q_j$ is the greatest, and $q_j = 0$ at the other coefficients. Hence, the following function:

$$\max_j (\sum_{i=1}^{M} v_{ij} p_i) \triangleq V \tag{7.45}$$

should be minimized with respect to $p_1, p_2, \ldots, p_M$. The minimization of $V$ with the constraints (7.43) is similar to a linear programming problem. To its solution existing iterative algorithms and respective computer programs may be used. For the determined strategy $p$ one may calculate the value $V$ by substituting into (7.45). It will be so called guaranteed expected performance index, what means that the average value $v$ (for many repetitions of the decision making) cannot be greater than $V$. A simplified scheme of

the algorithm determining the strategy $p$ and the value $V$ is illustrated in Fig. 7.4.

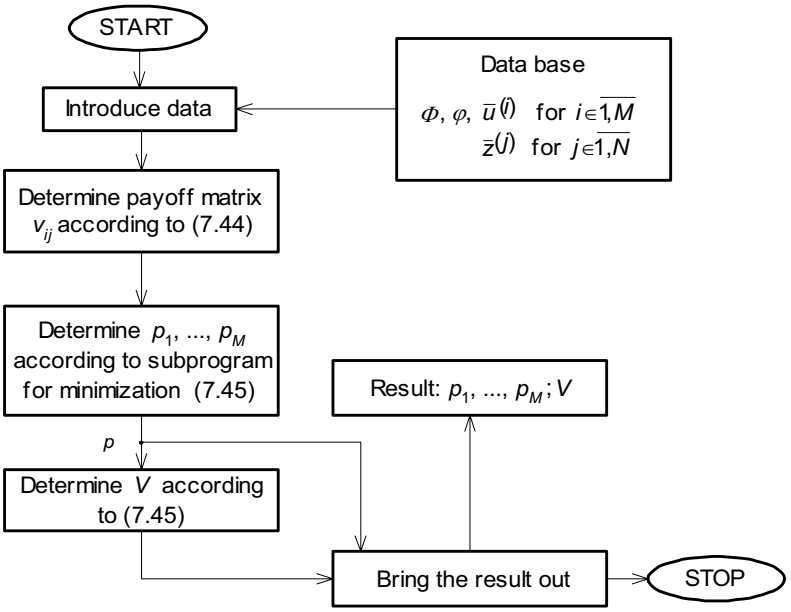


**Fig. 7.4.** Simplified scheme of algorithm determining control strategy

The execution of the determined strategy requires two generators of random numbers for the random choosing of the decision $u_n$, and a discretization of the disturbance for the determination of the set $\overline{Z}$. If $z_n$ is one-dimensional disturbance varying in the interval $[\alpha, \beta]$ then this interval may be divided into $(m-1)$ equal parts with a length $\bar{z}$ and accept

$$\bar{z}^{(j)} = \alpha + (j-1)\bar{z}, \quad j = 1, 2, ..., m.$$

Consequently, the computer program will contain three blocks (Fig. 7.5): the discretization block, the subprogram determining the strategy and the generator of random numbers needed for the execution of the random strategy. It is a subprogram which generates the decisions $u_n \in \overline{U}$ according to the given probability distribution $p_1, p_2, ..., p_M$. In the figure, the transferring of the information on the range of variation $[\alpha, \beta]$ is marked.

The determination of the control strategy is simplified for $M = 2$, i.e. for $i = 1, 2$. Then $p_2 = 1 - p_1$,

$$p_1 = \arg \min_p \max \{v_{11}p + v_{21}(1-p),\ v_{12}p + v_{22}(1-p),$$
$$..., v_{1N}p + v_{2N}(1-p)\} \tag{7.46}$$

where max denotes the greatest number in the brackets. Finally, let us note that the dependence of the performance index $v = \bar{y}$ (see Sect. 4.1) upon $u$ can be formulated directly, without using the required value $y^*$. It means the direct formulation of the model $v_n = \Phi(u_n, z_n)$ for an extremal control (static optimization) plant with an input $u_n$ and one-dimensional output to be minimized. An example of an application of the game approach to the determination of decisions in a production management may be found in [20].

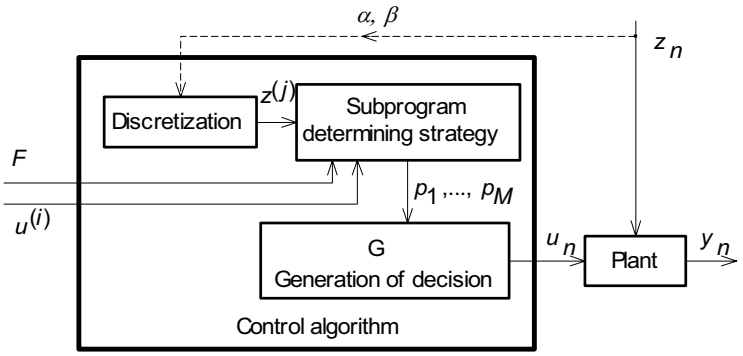


**Fig. 7.5.** System determining and executing control strategy

**Example 7.3.** A relationship between a profit $v$ and a production size $u$ in a certain production process may have the form

$$v = cu - F(u)$$

where $c$ is a unit price of the product, $cu$ is an income from the sale and $F$ is so called *cost function*. Let us assume that the production is repeated every day and at the beginning of the day one should plan the daily production size $\bar{u}^{(1)} = 500$ or $\bar{u}^{(2)} = 600$ units. Every day one of two sorts of a raw material is supplied but the quality (the sort) may be estimated after receiving the product only. The price $c$ and a parameter of the cost function depend on the sort of the raw material. The result of the daily profit calculation (in determined money units) is as follows:

| | Sort I | Sort II |
|---|---|---|
| $u = 500$ | 20 | 70 |
| $u = 600$ | 40 | 15 |

Let us denote by $p$ the probability of the choice $u = 500$. If the sort I was supplied every day then the expected profit would be

$$\mathrm{E}(\underline{v}) = 20p + 40(1 - p) = -20p + 40.$$

If however the sort II was supplied then

$$\mathrm{E}(\underline{v}) = 70p + 15(1 - p) = 55p + 15.$$

According to (7.46), the optimal value $p$ can be obtained by solving the equation

$$-20p + 40 = 55p + 15$$

and as a result $p = \frac{1}{3}$. In many days of the random daily choosing of the production sizes according to the probabilities $\frac{1}{3}$ and $\frac{2}{3}$, the average daily profit will be not less than $\frac{55}{3} + 15 \approx 33.3$. □

## 7.5 Basic Problem for Dynamical Plant

One of the basic problems considered in Sect. 7.1 for the static plant $y_n = \Phi(u_n, z_n)$ consisted in the determination of the decision $u^*$ minimizing the expected value of the performance index (7.8) for the given probability density $f_z(z)$. Now we shall present an analogous problem for the discrete dynamic plant described by the equation.

$$x_{n+1} = f(x_n, u_n, z_n) \tag{7.47}$$

where $x_n \in X$ is the state vector, $u_n \in U$ is the control vector and $z$ is the vector of disturbances. We assume that $z_n$ is a value of a random variable $\underline{z}_n$, the variables $\underline{z}_n$ are stochastically independent for different $n$ and have the same probability density $f_z(z)$. Let us introduce the performance index

$$Q_N = \sum_{n=1}^{N} \varphi(x_n, u_{n-1}) .$$

**Problem of the control optimal in a probabilistic sense:** For the given $f$, $x_0$, $\varphi$ and $f_z$ one should determine the sequence of optimal control decisions $u_0^*$, $u_1^*$, ..., $u_{N-1}^*$ minimizing the expected value of the index $Q$, i.e.

$$(u_0^*, u_1^*, ..., u_{N-1}^*) = \arg \min_{u_0, ..., u_{N-1}} \mathrm{E}_{\underline{z}_0, ..., \underline{z}_{N-1}} [\sum_{n=0}^{N-1} g(\underline{x}_n, u_n, \underline{z}_n)] \qquad (7.48)$$

where

$$g(x_n, u_n, z_n) = \varphi [f(x_n, u_n, z_n), u_n].$$

This is a probabilistic version of the optimal control problem described in Sect. 4.2 for the deterministic discrete plant. In the formula (7.48) it has been taken into account that $\underline{x}_n$ is a random variable what follows from the assumption that $\underline{z}_n$ is a random variable.

The determination of the sequence of the optimal decisions directly from the definition (7.48) is, in general, very difficult. One may make the solution easier by applying dynamic programming procedure described in Sect. 4.3 and consisting in a decomposition of the problem into separate stages considered from the end, i.e. from $n = N$. As a result one obtains the relationships $u_n = \Psi_n(x_n)$ or the control algorithm in a closed-loop system. Let us introduce the notation

$$V_{N-n}(x_n) = \min_{u_n, ..., u_{N-1}} \{ \mathrm{E}_{\underline{z}_n} [g(\underline{x}_n, u_n, \underline{z}_n)] + \mathrm{E}_{\underline{z}_{n+1}, ..., \underline{z}_{N-1}} [\sum_{i=n+1}^{N-1} g(\underline{x}_i, u_i, \underline{z}_i) \mid x_n] \}.$$

For $n = N - 1$

$$V_1(x_{N-1}) = \min_{u_{N-1}} \int_Z g(x_{N-1}, u_{N-1}, z) f_z(z)\, dz.$$

As a result we obtain a relationship between the minimizing value $u_{N-1}^*$ and the state $x_{N-1}$, which we denote by $\Psi_{N-1}$, i.e.

$$u_{N-1}^* = \Psi_{N-1}(x_{N-1}).$$

For two stages from the end

$$V_2(x_{N-2}) = \min_{u_{N-2}} \int_Z \{ g(x_{N-2}, u_{N-2}, z) + V_1[f(x_{N-2}, u_{N-2}, z)] \} f_z(z)\, dz.$$

As a result we obtain

$$u^*_{N-2} = \Psi_{N-2}(x_{N-2}).$$

Hence, the algorithm of the determination of the optimal control may be presented in the form of the following recursive procedure:

$$V_{N-n}(x_n) = \min_{u_n} \int_Z \{ g(x_n, u_n, z_n) + V_{N-n-1}[f(x_n, u_n, z)] f_z(z)\, dz \}, \qquad (7.49)$$

$$n = N-1, N-2, \ldots, 0, \qquad V_0 = 0.$$

As a result we obtain the relationships $u^*_n = \Psi_n(x_n)$, that is the control algorithm in a closed-loop system for the measurable plant. Applying the decomposition described above we have used the property that the conditional probability distribution of the random variable $\underline{x}_{n+1}$ for the given $u_n$ depends on $x_n$ only, but does not depend on the former states, i.e.

$$f_x(x_{n+1} \mid u_n, x_n, \ldots, x_0) = f_x(x_{n+1} \mid u_n, x_n).$$

This property follows from the fact that the stochastic process described by the equation

$$\underline{x}_{n+1} = f(\underline{x}_n, u_n, \underline{z}_n)$$

is a *discrete Markov process*.

The procedure (7.49) determining the multistage decision process optimal in a probabilistic sense may be called a *probabilistic version of dynamic programming*. In a similar way as in deterministic situations, the algorithm may be obtained analytically in simple cases only. Most often it is necessary to apply numerical successive approximation procedures. The determination of the multistage decision process with the probabilistic description of uncertainty has numerous practical applications not only to the control of technological processes but also in a management, in particular to the determination of a business plan, to the planning of investments processes etc. [20].

In a way analogous to that in Sect. 4.3, one may present the determina-

tion of the multistage decision process for the terminal control, i.e. the determination of the decision sequence minimizing the expected value of the performance index

$$Q_N = \varphi(x_N) = g(x_{N-1}, u_{N-1}, z_{N-1}).$$

Let us introduce the notation

$$V_{N-n}(x_n) = \min_{u_n,\ldots,u_{N-1}} \mathop{\mathrm{E}}_{\underline{z}_n,\ldots,\underline{z}_{N-1}} [g(\underline{x}_{N-1}, u_{N-1}, \underline{z}_{N-1}) \mid x_n].$$

For $n = N-1$

$$V_1(x_{N-1}) = \min_{u_{N-1}} \int_Z g(x_{N-1}, u_{N-1}, z) f_z(z) dz .$$

As a result we obtain the relationship

$$u^*_{N-1} = \Psi_{N-1}(x_{N-1}) .$$

For two stages from the end

$$V_2(x_{N-2}) = \min_{u_{N-2}} \int_Z V_1[f(x_{N-2}, u_{N-2}, z) f_z(z) dz ,$$

$$u^*_{N-2} = \Psi_{N-2}(x_{N-2}) .$$

Consequently, the algorithm of the determination of the optimal control may be presented in the form of the following recursive procedure:

$$V_{N-n}(x_n) = \min_{u_n} \int_Z V_{N-n-1}[f(x_n, u_n, z)] f_z(z) dz ,$$

$$n = N-1, N-2, \ldots, 0 ; \qquad V_0 = g(x_{N-1}, u_{N-1}, z) .$$

As a result we obtain the relationships $u^*_n = \Psi_n(x_n),$ that is the control algorithm in a closed-loop system.

It is worth noting that in the cases considered in this section it is not possible to obtain concrete values of the control decisions $u_0, u_1, \ldots, u_{N-1}$ by applying the second series of calculations *from the beginning to the end* as it was described in Sect. 4.3. It is caused by the presence of the disturbance $z_n$ in the plant equation, which makes it impossible to determine $x_{n+1}$ for the given $x_n$ and $u_n$.

## 7.6 Stationary Stochastic Process

For dynamical plants with a random parameter $c$ one may apply the parametric optimization in a way similar to that presented in Chap. 5. Now the performance index $Q = \Phi(c, a)$ is a function of the unknown plant parameter $c$ and the parameter $a$ in the control algorithm, which is to be determined. The problem is then reduced to a static probabilistic optimization and consists in the determination of the value $a$ minimizing the expected value $\mathrm{E}[\Phi(\underline{c}, a)]$. It is possible to apply another approach consisting in the determination of the control algorithm for the fixed $c$ treated as a known parameter. Consequently, $c$ will appear as a parameter in the control algorithm. Such an algorithm with the random parameter $c$ may be called a *random control algorithm* in the case under considerations, or shortly – a *random controller* in an open-loop or closed-loop system. The deterministic algorithm may be obtained as a result of the determinization consisting in the determination of the decision equal to the expected value of the controller output. We shall return to this concept in Chap. 8.

The problem is much more complicated if the incomplete knowledge of the plant concerns unknown time-varying disturbances which are not measured and are assumed to be random disturbances. In the next sections we shall consider a special probabilistic optimization problem for dynamical plants, namely an optimization problem for linear closed-loop control system with constant parameters and stationary random disturbances. For this purpose the basic information concerning a stationary stochastic process will be shortly presented. Let us consider a variable $x(t) \in X$ (in general, a vector variable) varying randomly. In every moment $t$ the value $x(t)$ is assumed to be a value of a random variable $\underline{x}(t)$ characterized by a probability distribution, in general depending on $t$. It means that for different $t$ the corresponding value $x(t)$ is chosen randomly from different sets characterized by different probability distributions. In the further considerations we shall assume that there exists a probability density, as for the static plant in the previous sections. As it was already mentioned, a function assigning to the variable $t$ a corresponding random variable $\underline{x}(t)$ is called a *stochastic process*. That is why a concrete function $x(t)$ is called a *realization* or an observation of the stochastic process. A more general description of $\underline{x}(t)$ is given by a joint probability density of the random variables $\underline{x}(t_1) = \underline{x}_1$, $\underline{x}(t_2) = \underline{x}_2$, ..., $\underline{x}(t_n) = \underline{x}_n$ in selected $n$ moments, which we denote by $f_n(x_1, x_2, ..., x_n; t_1, t_2, ..., t_n)$. A stochastic process $\underline{x}(t)$ is called *stationary* if this probability density depends only on the distances be-

tween the points $x_1, ..., x_n$ and does not depend on their location on the axis of the time, i.e. for any $\tau$

$$f_n(x_1, x_2, ..., x_n; t_1+\tau, t_2+\tau, ..., t_n+\tau) = f_n(x_1, x_2, ..., x_n; t_1, t_2, ..., t_n).$$

In particular, $f_1(x) = f(x)$ does not depend on $t$. We may say that the statistical properties of the stationary process are constant in time. For example, if in the case of one-dimensional process, using many observed realizations we shall determine the arithmetic mean of the value $x$ observed in the same moment $t$, or having one long realization we shall determine the arithmetic mean of the values $x(t)$ in many moments $t_1, t_2, ..., t_n$, then these mean values will be approximately equal. In general, the *mean value with respect to a set*, i.e.

$$\tilde{x} = \mathrm{E}(\underline{x}) = \int_X x f(x) dx$$

is equal with probability 1 to the *mean value with respect to time*

$$\bar{x} = \lim_{T\to\infty} \frac{1}{2T} \int_{-T}^{T} x(t) dt$$

(except in very special cases rather not occurring in practice). This is so called *ergodic property* which is also satisfied for functions of the variables $x_1, x_2, ..., x_n$. If the stationary stochastic process is an object of linear transformations (in other words, the stationary stochastic signal is put at an input of a linear dynamical system), its description in the form of so called correlation functions and spectral densities are more convenient than the description in the form of the probability densities. Let us define these descriptions for a one-dimensional process $\underline{x}(t)$. An *autocorrelation function* of the process $\underline{x}(t)$ is defined as follows:

$$R_{xx}(\tau) = \mathrm{E}[\underline{x}_1 \cdot \underline{x}_2] = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} x_1 x_2 f_2(x_1, x_2; \tau) dx_1 dx_2 \tag{7.50}$$

where $x_1 = x(t)$, $x_2 = x(t + \tau)$. According to the ergodic property

$$R_{xx}(\tau) = \lim_{T\to\infty} \frac{1}{2T} \int_{-T}^{T} x(t) x(t+\tau) dt. \tag{7.51}$$

In the further considerations we shall assume $\mathrm{E}(\underline{x}) = 0$. Then, according to

(7.50), the value of the autocorrelation function for the fixed $\tau$ is a covariance (a correlation moment) of the random variables $\underline{x}_1$ and $\underline{x}_2$, i.e. is a measure of a correlation between two values $x(t)$ and $x(t+\tau)$. Then

$$\lim_{\tau\to\infty} R_{xx}(\tau) = 0 .$$

According to the definition, $R_{xx}(\tau) = R_{xx}(-\tau)$ and

$$R_{xx}(0) = \lim_{T\to\infty} \frac{1}{2T} \int_{-T}^{T} x^2(t)dt \stackrel{\Delta}{=} \bar{x}^2 .$$

The number $\bar{x}^2$ is called a *mean-square value* of the signal $x(t)$. Using a power interpretation we may say that this is a value proportional to the power of the signal. Frequently used simple examples of the autocorrelation functions are as follows (Fig. 7.6):

$$R_{xx}(\tau) = ce^{-\alpha\tau^2}, \qquad R_{xx}(\tau) = ce^{-\alpha|\tau|} .$$

A *spectral density* $S_{xx}(\omega)$ of the process $\underline{x}(t)$ may be defined as Fourier transformation of the autocorrelation function

$$S_{xx}(\omega) = \int_{-\infty}^{\infty} R_{xx}(\tau)e^{-j\omega\tau}d\tau . \tag{7.52}$$

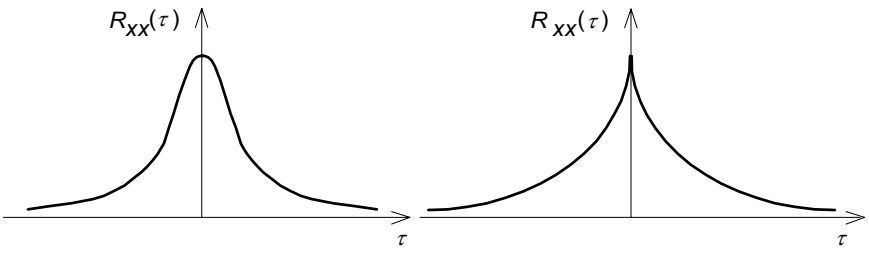


**Fig. 7.6.** Examples of autocorrelation functions

The descriptions $R_{xx}(\tau)$ and $S_{xx}(\omega)$ are equivalent and

$$R_{xx}(\tau) = \frac{1}{2\pi} \int_{-\infty}^{\infty} S_{xx}(\omega)e^{j\omega\tau}d\omega . \tag{7.53}$$

Using the definition (7.52) and the property $R_{xx}(\tau) = R_{xx}(-\tau)$, it is easy to

note that $S_{xx}(\omega) = S_{xx}(-\omega)$ and that $S_{xx}(\omega)$ has real values. According to (7.51) and (7.53)

$$\bar{x}^2 = \frac{1}{2\pi}\int_{-\infty}^{\infty} S_{xx}(\omega)d\omega = \frac{1}{\pi}\int_{0}^{\infty} S_{xx}(\omega)d\omega\,. \tag{7.54}$$

The formula (7.54) shows how to determine the mean-square value using the spectral density. Applying the power interpretation one may say that the integral of the spectral density (7.54) is proportional to the power of the signal or that $S_{xx}(\omega)$ presents a power distribution in a frequency domain. The signal with the autocorrelation function $R_{xx}(\tau) = c\delta(\tau)$ (Dirac delta) is called a *white noise*. It is a *fully random* signal with no correlation between the neighbour values $x$ for an arbitrarily small $\tau \neq 0$. According to (7.52), for the white noise $S_{xx} = \text{const} = c$. It is then a signal with an infinitely great power, which may be approximated by so called *practical white noise* for which $R_{xx}(\tau)$ is a very short and very high impulse in the neighbourhood of $\tau = 0$ and $S_{xx}(\omega)$ is constant in the large interval of $\omega$ starting from $\omega = 0$.

For two stationary stochastic processes $\underline{x}(t)$ and $\underline{y}(t)$ let us introduce a cross-correlation function $R_{xy}(\tau)$ and a corresponding spectral density:

$$R_{xy}(\tau) = \mathrm{E}[\underline{x}(t)\underline{y}(t+\tau)] = \lim_{T\to\infty}\frac{1}{2T}\int_{-T}^{T} x(t)y(t+\tau)d\tau,$$

$$S_{xy}(\mathrm{j}\omega) = \int_{-\infty}^{\infty} R_{xy}(\tau)e^{-\mathrm{j}\omega\tau}d\tau. \tag{7.55}$$

It is easy to note that $R_{yx}(\tau) = R_{xy}(-\tau)$ and the values of $S_{xy}(\mathrm{j}\omega)$ do not have to be real. The presented descriptions may be generalized for the vector signals $x(t)$ and $y(t)$. In this case

$$R_{xx}(\tau) = \mathrm{E}[\underline{x}(t)\underline{x}^{\mathrm{T}}(t+\tau)] = \lim_{T\to\infty}\frac{1}{2T}\int_{-T}^{T} x(t)x^{\mathrm{T}}(t+\tau)d\tau\,,$$

$$R_{xy}(\tau) = \mathrm{E}[\underline{x}(t)\underline{y}^{\mathrm{T}}(t+\tau)] = \lim_{T\to\infty}\frac{1}{2T}\int_{-T}^{T} x(t)y^{\mathrm{T}}(t+\tau)d\tau.$$

Now the autocorrelation function $R_{xx}(\tau)$ is a matrix. In its principal diagonal there are the autocorrelation functions of the components of the vector $x$ and outside the diagonal – the cross-correlation functions of the different components, i.e.

$$R_{xx}(\tau) = [R_{x^{(i)}x^{(j)}}(\tau)]_{\substack{i=1,2,\ldots,k \\ j=1,2,\ldots,k}}$$

where $x^{(i)}$ is the $i$-th component of the vector $x$ with $k$ components. The matrix $R_{xy}(\tau)$ has a similar form. If the pairs of the different components of the vector $x(t)$ are uncorrelated then $R_{xx}(\tau)$ is a diagonal matrix. In the multi-dimensional case under consideration, the spectral densities $S_{xx}(\mathrm{j}\omega)$ and $S_{xy}(\mathrm{j}\omega)$ are matrices defined by the formulas (7.52) and (7.55), respectively, where multiplying by $e^{-\mathrm{j}\omega\tau}$ and integrating refer to the particular entries of the matrix $R_{xx}(\tau)$ and $R_{xy}(\tau)$. Considerations concerning one-dimensional and multi-dimensional correlation functions and their application in an identification problem for control plants may be found e.g. in [14, 100].

## 7.7 Analysis and Parametric Optimization of Linear Closed-loop Control System with Stationary Stochastic Disturbances

Let us consider a continuous, one-dimensional linear stationary system with two inputs $x(t)$ and $z(t)$, and the output $y(t)$, described by the transmittances

$$Y(s) = K_1(s)X(s) + K_2(s)Z(s)\,. \tag{7.56}$$

Assume that $x(t)$ and $z(t)$ are realizations of stationary stochastic processes. Consequently, after passing a transit process caused by putting $x(t)$ and $z(t)$ at the input, the response $y(t)$ is also a realization of a stationary stochastic process. For such a system, the analysis problem cannot consist in the determination of the response $y(t)$ for the given functions $x(t)$ and $z(t)$, but it can consist in the determination of the mean-square value $\overline{y}^2$ for the given correlation functions of the inputs and the given transmittances of the system.

**Analysis problem for the dynamical system:** For the given $K_1(s)$, $K_2(s)$,

$R_{xx}(\tau)$, $R_{zz}(\tau)$, $R_{xz}(\tau)$ one should find $\bar{y}^2$.

Instead of the correlation functions one may use the equivalent descriptions in the form of the respective spectral densities. It may be shown that the relationship between the spectral densities of the inputs and the output is the following:

$$S_{yy}(\omega) = |K_1(\mathrm{j}\omega)|^2 S_{xx}(\omega) + |K_2(\mathrm{j}\omega)|^2 S_{zz}(\omega) + K_1(-\mathrm{j}\omega)K_2(\mathrm{j}\omega)S_{xz}(\mathrm{j}\omega) + K_1(\mathrm{j}\omega)K_2(-\mathrm{j}\omega)S_{zx}(\mathrm{j}\omega). \quad (7.57)$$

If the processes $x(t)$ and $z(t)$ are uncorrelated then only the first and the second components will occur in the formula (7.57). According to the formula (7.54) we find

$$\bar{y}^2 = \frac{1}{\pi}\int_0^\infty S_{yy}(\omega)d\omega. \quad (7.58)$$

In typical cases the spectral density is a rational function of $\omega^2$ and may be presented in the form

$$S_{yy}(\omega) = \frac{G_p(\omega)}{H_p(\omega)H_p(-\omega)} \quad (7.59)$$

where

$$G_p(\omega) = b_0\omega^{2p-2} + b_1\omega^{2p-4} + ... + b_{p-1},$$

$$H_p(\omega) = a_0\omega^p + a_1\omega^{p-1} + ... + a_p,$$

and all zeros of the polynomial $H_p(\omega)$ lie in the upper half-plane. The integrals (7.58) for the function (7.59) have been found for small degrees $p$. The results for $p = 1, 2, 3$ are the following:

$$I_1 = -\mathrm{j}\frac{b_0}{2a_0a_1}, \qquad I_2 = \mathrm{j}\frac{-b_0 + \frac{a_0 b_1}{a_2}}{2a_0a_1}, \quad (7.60)$$

$$I_3 = -\mathrm{j}\frac{-a_2b_0 + a_0b_1 - \frac{a_0a_1b_2}{a_3}}{2a_0(a_0a_3 - a_1a_2)} \quad (7.61)$$

where $I_p$ denotes the integral (7.58) for the spectral density (7.59).

The method of the analysis for the dynamical system presented above

may be applied to the analysis and the parametric optimization of the linear closed-loop control system (Fig.7.7) with the stationary stochastic disturbances $y^*(t)$ and $z(t)$, in which the plant and the controller are described by the transmittances $K_{\mathrm{O}}(s)$ and $K_{\mathrm{R}}(s)$, respectively, i.e.

$$Y(s) = K_{\mathrm{O}}(s)U(s) + K_{\mathrm{Z}}(s)Z(s),$$

$$U(s) = K_{\mathrm{R}}(s)E(s)$$

where

$$E(s) = Y^*(s) - Y(s).$$

Hence,

$$E(s) = K_1(s)Y^*(s) + K_2(s)Z(s)$$

where

$$K_1(s) = \frac{1}{1 + K_{\mathrm{O}}(s)K_{\mathrm{R}}(s)},$$

$$K_2(s) = \frac{-K_{\mathrm{Z}}(s)}{1 + K_{\mathrm{O}}(s)K_{\mathrm{R}}(s)}.$$

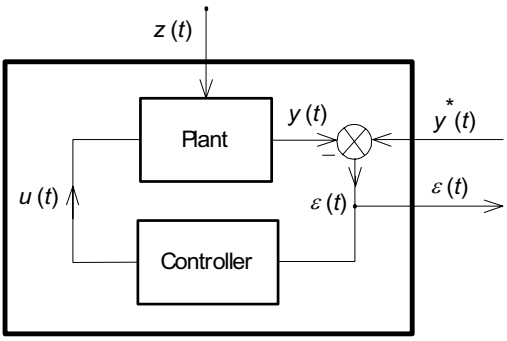


**Fig. 7.7.** Scheme of control system under consideration

**Analysis problem for the closed-loop control system:** For the given $K_{\mathrm{O}}(s)$, $K_{\mathrm{Z}}(s)$, $K_{\mathrm{R}}(s)$, $R_{y^*y^*}(\tau)$, $R_{zz}(\tau)$, $R_{y^*z}(\tau)$ one should find the mean-square control error $\overline{\varepsilon}^2$.

The closed-loop control system may be treated as a dynamical system with the inputs $y^*(t)$ and $z(t)$, and the output $\varepsilon(t)$. Then, according to (7.58)

$$\bar{\varepsilon}^2 = \frac{1}{\pi}\int_0^\infty S_{\varepsilon\varepsilon}(\omega)d\omega$$

where $S_{\varepsilon\varepsilon}(\omega)$ should be determined according to the formula (7.57) with $y^*$ in the place of $x$ and with $\varepsilon$ in the place of $y$.

In a similar way as in deterministic cases considered in Chap. 5, the analysis may be used as the first stage of the parametric optimization of the control system. If we assume a determined form of the controller transmittance $K_R(s; a)$ where $a$ is a vector of parameters, then as a result of the analysis we have the dependence of the performance index $\bar{\varepsilon}^2 \triangleq Q$ upon $a$. Then the design of the controller consists in the determination of the value $a^*$ minimizing the function $Q(a)$. If the function $Q(a)$ is differentiable with respect to $a$ then $a^*$ may be determined from the equation

$$\operatorname*{grad}_a Q(a) = \bar{0} \tag{7.62}$$

under the assumption that $a^*$ is a unique solution of the equation (7.62) and that in this point the function $Q(a)$ takes its local minimum.

The analysis problem for the dynamical system with two inputs $x$, $z$ and one output $y$ may be generalized for a multi-dimensional system with the input vector $x$ and the output vector $y$, described by the relationship

$$Y(s) = K(s)X(s)$$

where $K(s)$ denotes a matrix transmittance. Then

$$S_{yy}(\mathrm{j}\omega) = K(\mathrm{j}\omega)S_{xx}(\mathrm{j}\omega)K^{\mathrm{T}}(-\mathrm{j}\omega)$$

where $S_{xx}(\mathrm{j}\omega)$ and $S_{yy}(\mathrm{j}\omega)$ denote matrix spectral densities. Now the mean-square value of the output

$$\bar{y}^2 = \overline{y^{\mathrm{T}}y} = \lim_{T\to\infty}\frac{1}{2T}\int_{-T}^{T} y^{\mathrm{T}}(t)y(t)dt$$

and according to (7.54)

$$\bar{y}^2 = \frac{1}{\pi}\int_0^\infty \bar{S}_{yy}(\omega)d\omega$$

where $\overline{S}_{yy}(\omega)$ denotes the sum of entries in the principal diagonal of the matrix $S_{yy}(\omega)$. The generalization for a multi-dimensional dynamical system presented above may be used to the analysis and the parametric optimization of multi-dimensional closed-loop control system considered as a specific dynamical system, in a way similar to that used for the one-dimensional system illustrated in Fig. 7.7.

**Example 7.4.** Let us determine $\overline{\varepsilon}^2$ in the closed-loop control system (Fig. 7.7) where

$$K_{\mathrm{O}}(s) = \frac{k_{\mathrm{O}}}{sT+1}, \qquad K_{\mathrm{R}}(s) = k_{\mathrm{R}},$$

$$S_{y^* y^*}(\omega) = \frac{2\alpha}{(\omega^2+\alpha^2)\omega^2}.$$

It is easy to prove that $y^*(t)$ is a result of integrating a signal with autocorrelation function

$$R(\tau) = e^{-\alpha|\tau|}.$$

Assume that the disturbance $z(t)$ is a white noise, i.e. $S_{zz}(\omega) = c$, $R_{y^* z}(\tau) = 0$. Assume that $z(t)$ is an additive noise added to $y^*(t)$. Then

$$K_{\mathrm{Z}}(s) = K_{\mathrm{O}}(s)K_{\mathrm{R}}(s).$$

Applying the formula (7.57) for $y^*$ and $\varepsilon$ we obtain

$$S_{\varepsilon\varepsilon}(\omega) = \frac{1}{\left|1+K_{\mathrm{O}}(\mathrm{j}\omega)K_{\mathrm{R}}(\mathrm{j}\omega)\right|^2} \cdot \frac{2\alpha}{(\omega^2+\alpha^2)\omega^2} + \frac{\left|K_{\mathrm{O}}(\mathrm{j}\omega)K_{\mathrm{R}}(\mathrm{j}\omega)\right|^2}{\left|1+K_{\mathrm{O}}(\mathrm{j}\omega)K_{\mathrm{R}}(\mathrm{j}\omega)\right|^2} c.$$

The first term $S^{\mathrm{I}}_{\varepsilon\varepsilon}(\omega)$ after substituting $K_{\mathrm{O}}$ and $K_{\mathrm{R}}$ may be presented in the form

$$S^{\mathrm{I}}_{\varepsilon\varepsilon}(\omega) = \frac{2\alpha(\omega^2T^2+1)}{H_3(\omega)H_3(-\omega)}$$

where

$$H_3(\omega) = \mathrm{j}T\omega^3 - \alpha T\omega^2 - \mathrm{j}(\alpha + k)\omega + \alpha k, \qquad k = k_O k_R.$$

In a similar way, the second term $S_{\varepsilon\varepsilon}^{\mathrm{II}}(\omega)$ may be reduced to the form

$$S_{\varepsilon\varepsilon}^{\mathrm{II}}(\omega) = \frac{c}{H_2(\omega)H_2(-\omega)}$$

where

$$H_2(\omega) = -T\omega^2 + \mathrm{j}\omega + k.$$

Using the formula (7.58) for $\varepsilon$ and the formulas (7.60), (7.61) in order to determine the integrals of the both terms, after some transformations we obtain

$$\bar{\varepsilon}^2 = \frac{Tk+1}{2\alpha^2 k} + \frac{k}{2}. \qquad \square$$

## 7.8 Non-parametric Optimization of Linear Closed-loop Control System with Stationary Stochastic Disturbances

For the linear stationary system (Fig. 7.7) with stationary stochastic disturbances, the non-parametric optimization is sometimes called a *synthesis* of an optimal controller and consists in the determination of a linear controller minimizing a mean-square control error. As in the case of analysis, we shall start with a more general problem concerning the optimization of a linear dynamical system. Let us assume that a signal $x(t)$ is a realization of a stationary stochastic process and we want to obtain a signal which will be a result of a linear transformation of the signal $x(t)$, determined by a transmittance $H(s)$, i.e. we want to obtain a signal $v(t)$ such that $V(s) = H(s)X(s)$. For example, $H(s) = s$ means that we want to differentiate the signal $x(t)$. If the transmittance $H(s)$ is physically realizable (a degree of the numerator is not greater than a degree of the denominator) and the signal $x(t)$ is available without a noise then it is sufficient to put $x(t)$ at the input of a system described by $H(s)$ and to obtain $v(t)$ at the output. If $H(s)$ is not realizable and (or) only the signal with a stationary stochastic noise $x(t) + z(t)$ is available then we can try to determine such a transmittance $K(s)$ that the signal $x(t) + z(t)$ put at the input of a system described by $K(s)$, gives at the output the signal $w(t)$ which is the best approximation of

$v(t)$ , minimizing the mean-square error (Fig. 7.8). It is so called *Wiener's problem.* The typical and the most frequently considered cases of this problem are the following:

**1. Filtration**

$$H(s) = 1, \quad \text{i.e.} \quad v(t) = x(t).$$

**2. Differentiation**

$$H(s) = s, \quad \text{i.e.} \quad v(t) = \dot{x}(t).$$

**3. Prediction**

$$H(s) = e^{sT}, \quad \text{i.e.} \quad v(t) = x(t + T)$$

where $T > 0$.

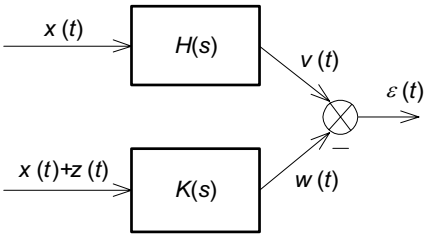


**Fig. 7.8.** Illustration of the approximation problem under consideration

**Optimization problem for the dynamical system:** For the given $H(s)$, $R_{xx}(\tau)$, $R_{zz}(\tau)$, $R_{xz}(\tau)$ one should determine the transmittance $K(s)$ minimizing the mean-square approximation error $\bar{\varepsilon}^2$ where $\varepsilon(t) = v(t) - w(t)$.

The given functions determine a functional which assigns the numbers $\bar{\varepsilon}^2$ to the functions $K(s)$. By applying a variatonal calculus to the minimization of this functional one may show that the result should satisfy the following integral equation:

$$R_{vm}(\tau) = \int_0^\infty R_{mm}(\tau - \lambda) k_i(\lambda) d\lambda, \qquad \tau \geq 0 \tag{7.63}$$

where $m(t) = x(t) + z(t)$, and $k_i(t)$ is the impulse response of the system to be determined, i.e. $K(s)$ is the Laplace transform of the function $k_i(t)$. The integral equation (7.63) with the unknown function $k_i(\cdot)$ is called a *Wie-*

*ner-Hopf equation*. It may be shown that its solution in the form of the frequency transmittance $K(\mathrm{j}\omega)$ satisfying the realizability condition is as follows:

$$K(\mathrm{j}\omega) = \frac{1}{2\pi \Psi_1(\mathrm{j}\omega)} \int_0^\infty e^{-\mathrm{j}\omega t} dt \int_{-\infty}^\infty \frac{S_{vm}(\mathrm{j}\omega)}{\Psi_2(\mathrm{j}\omega)} e^{\mathrm{j}\omega t} d\omega \tag{7.64}$$

where

$$\Psi_1(\mathrm{j}\omega)\,\Psi_2(\mathrm{j}\omega) = S_{mm}(\omega), \tag{7.65}$$

$\Psi_2(\mathrm{j}\omega) = \Psi_1(-\mathrm{j}\omega)$, all poles and zeros of the function $\Psi_1(\mathrm{j}\omega)$ lie in the upper half-plane. The presentation of $S_{mm}$ in the form (7.65) is always possible because $S_{mm}(\omega)$ is a real rational function and $S_{mm}(\omega) = S_{mm}(-\omega)$. Consequently, for each pole (zero) of this function there exist three other poles (zeros) lying symmetrically with respect to the both coordinate axes.

In the filtration problem $S_{vm} = S_{xx} + S_{xz}$, $S_{mm} = S_{xx} + S_{zz} + S_{xz} + S_{zx}$. Let us note that

$$\frac{1}{2\pi} \int_{-\infty}^\infty \frac{S_{vm}(\omega)}{\Psi_2(\mathrm{j}\omega)} e^{\mathrm{j}\omega t} d\omega \triangleq \beta(t)$$

is an inverse Fourier transform of the function $\dfrac{S_{vm}(\omega)}{\Psi_2(\mathrm{j}\omega)}$, and

$$\int_0^\infty \beta(t) e^{-\mathrm{j}\omega t} dt \triangleq B(\mathrm{j}\omega)$$

is the Fourier transform of the function $\beta(t)$ for $t > 0$. Consequently, in order to determine $B(\mathrm{j}\omega)$ and then to determine

$$K(\mathrm{j}\omega) = \frac{B(\mathrm{j}\omega)}{\Psi_1(\mathrm{j}\omega)}$$

according to (7.64), one should present $\dfrac{S_{vm}(\omega)}{\Psi_2(\mathrm{j}\omega)}$ as a sum of partial fractions and take into account only the fractions corresponding to the poles in the upper half-plane.

**Optimization problem for the closed-loop control system:** For the given

$K_O(s)$, $K_Z(s)$, $R_{y^*y^*}(\tau)$, $R_{zz}(\tau)$, $R_{y^*z}(\tau)$ in the system presented in Fig. 7.7, one should determine the transmittance $K_R(s)$ minimizing the mean-square control error $\bar{\varepsilon}^2$.

Without loss of generality one may assume that $z(t)$ is an additive noise added to $y^*(t)$, i.e. $K_Z = -K_O K_R$. Then it is not the control error $\varepsilon$ but the signal $e = y^* + z - y$ that is put at the input of the controller. Denote by $\bar{z}(t)$ a disturbance acting on the plant according to the transmittance $K_Z$ (as in the analysis problem in Sect. 7.7). The disturbance $\bar{z}(t)$ may be replaced by the disturbance $z(t)$ added to $y^*$, and

$$Z(s) = \frac{K_Z(s)}{K_O(s)K_R(s)}\bar{Z}(s).$$

The closed-loop control system with the input $y^* + z$ and the output $y$ may be treated as a filter. Consequently, we can determine the transmittance of the closed-loop system as a whole

$$\frac{K_O(s)K_R(s)}{1+K_O(s)K_R(s)} \triangleq \bar{K}(s) \tag{7.66}$$

in the same way as the transmittance of an optimal filter in the problem considered in the first part of this section. Then, from the equation (7.66) for the given transmittances $\bar{K}(s)$ and $K_O(s)$ one should determine $K_R(s)$, i.e. such transmittance of the controller that the control system as a whole acts as an optimal filter. It is worth noting that in some cases the transmittance $K_R(s)$ determined in this way may be unrealizable. The considerations for multi-dimensional systems with matrix correlation functions and matrix spectral densities are similar but much more complicated. Analogous problems and methods may be formulated and applied for discrete systems in which *discrete correlation functions*

$$R_{xx}(m) = \lim_{N\to\infty} \frac{1}{2N+1} \sum_{n=-N}^{N} x_n x_{n+m}$$

and *discrete spectral densities*

$$S_{xx}(e^{j\omega}) = \sum_{m=-\infty}^{\infty} R_{xx}(m)e^{-j\omega m}$$

occur. Wide considerations concerning the analysis and optimization problems for linear systems with stationary stochastic disturbances may be found e.g. in [100] and for discrete systems also in [102].

**Example 7.5.** Let us consider the control system with the plant $K_O(s) = k_O$ and the stationary stochastic input $y^*(t) + z(t)$. Assume that

$$R_{y^*y^*}(\tau) = \frac{1}{2}e^{-|\tau|},$$

$z(t)$ is a white noise, i.e. $S_{zz}(\tau) = c$, the signals $y^*$ and $z$ are uncorrelated. One should find the optimal controller $K_R(s)$ minimizing the mean-square control error.

Using Fourier transformation of the function $R_{y^*y^*}$ yields

$$S_{y^*y^*} = \frac{1}{\omega^2 + 1}.$$

In the case under consideration $m(t) = y^*(t) + z(t)$, $v(t) = y^*(t)$ (the filtration problem) and

$$S_{vm} = S_{y^*y^*} = \frac{1}{\omega^2 + 1},$$

$$S_{mm} = S_{y^*y^*} + S_{zz} = \frac{1}{\omega^2 + 1} + c = \frac{(\sqrt{1+c^2} + \mathrm{j}\omega c)(\sqrt{1+c^2} - \mathrm{j}\omega c)}{(1+\mathrm{j}\omega)(1-\mathrm{j}\omega)}.$$

Then

$$\Psi_1(\mathrm{j}\omega) = \frac{\sqrt{1+c^2} + \mathrm{j}\omega c}{1+\mathrm{j}\omega}.$$

Consequently

$$\frac{S_{vm}(\omega)}{\Psi_1(-\mathrm{j}\omega)} = \frac{1}{(1+\mathrm{j}\omega)\sqrt{1+c^2} - \mathrm{j}\omega c} = \frac{1}{c+\sqrt{1+c^2}}\left(\frac{1}{1+\mathrm{j}\omega} + \frac{c}{\sqrt{1+c^2} - \mathrm{j}\omega c}\right).$$

In the above expression only the first fraction corresponds to the pole in the upper half-plane ($\omega = \mathrm{j}$), i.e.

$$B(\mathrm{j}\omega) = \frac{1}{(c + \sqrt{1+c^2})(1 + \mathrm{j}\omega)}$$

and the optimal transmittance of the closed-loop control system as a whole is the following:

$$\overline{K}(s) = \frac{B(s)}{\Psi_1(s)} = \frac{1}{(c + \sqrt{1+c^2})(\sqrt{1+c^2} + sc)}.$$

Solving the equation (7.66) with respect to $K_\mathrm{R}(s)$ for the given $K_\mathrm{O}(s)$, we obtain

$$K_\mathrm{R}(s) = \frac{k_\mathrm{R}}{sT + 1}$$

where

$$k_\mathrm{R} = \frac{1}{k_\mathrm{O}[(c + \sqrt{1+c^2})\sqrt{1+c^2} - 1]},$$

$$T = \frac{c(c + \sqrt{1+c^2})}{(c + \sqrt{1+c^2})\sqrt{1+c^2} - 1}.$$

In this way it is possible to determine the transmittance of the optimal controller if

$$(c + \sqrt{1+c^2})\sqrt{1+c^2} > 1$$

i.e. if $c$ is sufficiently large. □

## 7.9 Relational Plant with Random Parameter

Let us turn back now to the static plant with the input $u \in U$ and the output $y \in Y$, and let us consider a relational plant described by the relation

$$R(u, y; c) \subset U \times Y$$

where $c$ is an unknown parameter. Assume that $c$ is a value of a random variable $\underline{c}$ described by the probability density $f_c(c)$. This is a case of so called *second order uncertainty* or two-level uncertainty. The first (lower)

level denotes the uncertainty concerning the plant and described by the relation $R$ which is not a function. For example, for one-dimensional case the description

$$cu \le y \le 2cu \tag{7.67}$$

means that the plant is non-deterministic and for the same value $u$, different values $y$ satisfying the inequality (7.67) may occur at the output. In our consideration $c$ is an unknown parameter which means an expert's uncertainty (see remarks in Sect. 6.1). This is the second uncertainty level described here by the probability distribution $f_c(c)$. Hence, the relational plant with a random parameter is a plant with the second order uncertainty, where a plant uncertainty is described in a relational way and an expert uncertainty is characterized in a probabilistic way. The second description means that the parameter $c$ in the plant has been randomly chosen from the set of values $C$ described by the probability density $f_c(c)$ or that the plant has been randomly chosen from the set of plants with different values $c$ in the relation $R$. If the relation $R(u, y; c)$ is reduced to the function $y = \Phi(u, c)$ then for the fixed $u$ the output $y$ is a value of the random variable $\underline{y} = \Phi(u, \underline{c})$ and the analysis problem may consist in the determination of the probability density $f_y(y; u)$ for the fixed $u$. If $R$ is not a function, such a formulation of the analysis problem is not possible. Now, for the fixed $u$ the set of possible outputs

$$D_y(u; c) = \{y \in Y: (u, y) \in R(u, y; c)\}$$

may be found (see (6.6)). Consequently, $u$ does not determine a random variable $\underline{y}$ but a random set $D_y(u; \underline{c})$. Then the analysis problem for the given $y$ may consist in the determination of the probability that this value **may occur** at the output of the plant. The problem may be generalized for a set of output values $\Delta_y \subset Y$ given by a user.

**Analysis problem:** For the given $R, f_c(c)$, $u$ and $\Delta_y$ one should determine

$$P[\Delta_y \subseteq D_y(u; \underline{c})] \stackrel{\Delta}{=} p(\Delta_y, u), \tag{7.68}$$

i.e. the probability that every value $y$ belonging to the set $\Delta_y$ given by a user may appear at the output of the plant.

Let us note that

$$P[\Delta_y \subseteq D_y(u; \underline{c})] = P[\underline{c} \in D_c(\Delta_y, u)] \tag{7.69}$$

where

$$D_c(\Delta_y, u) = \{c \in C: \Delta_y \subseteq D_y(u; c)\}.$$

Consequently,

$$p(\Delta_y, u) = \int_{D_c(\Delta_y, u)} f_c(c)dc\,. \tag{7.70}$$

In particular for $\Delta_y = \{y\}$ (a singleton), the probability that the given value $y$ may appear at the output of the plant is the following:

$$p(y,u) = \int_{D_c(y,u)} f_c(c)dc$$

where

$$D_c(y, u) = \{c \in C: y \in D_y(u; c)\}.$$

**Decision making (control problem):** For the given $R$, $f_c(c)$ and $D_y$ formulated by a user one should determine the decision $u^*$ maximizing the probability

$$P[D_y(u;\ \underline{c}\,) \subseteq D_y] \triangleq p(u). \tag{7.71}$$

This is one of possible formulations of the decision problem, consisting in the determination of the decision $u^*$ maximizing the probability that the set of possible outputs belongs to the given set $D_y$, i.e. that $y$ not belonging to $D_y$ will not appear at the output. Another version of the decision problem is presented in [52]. Since

$$P[D_y(u; \underline{c}\,) \subseteq D_y] = P[\,\underline{c} \in D_c(D_y, u)] \tag{7.72}$$

where

$$D_c(D_y, u) = \{c \in C : D_y(u; c) \subseteq D_y\}$$

then

$$u^* = \arg \max_u \int_{D_c(D_y, u)} f_c(c)dc\,. \tag{7.73}$$

The above considerations can be extended for the plant described by the relation $R(u, y, z; c)$ where $z$ is the disturbance which is measured. Then, for the given $R$, $f_c(c)$, $z$ and $D_y$, the decision making problem consists in

finding the decision $u$ maximizing the probability

$$P\,[D_y(u, z\,;\, \underline{c}\,) \subseteq D_y] \stackrel{\Delta}{=} p(u, z)$$

where

$$D_y(u, z; c) = \{y \in Y : (u, y, z) \in R(u, y, z; c)\}.$$

In the way similar to that for the plant without the disturbances, we shall determine $u = \Psi(z)$, i.e. the control algorithm in an open-loop system. It will be a control algorithm determined directly by using the knowledge of the plant KP = $< R, f_c >$, i.e. a control algorithm *based on the knowledge of the plant*. For the fixed $c$ and $z$, in the same way as in (6.19) one can determine the largest set $D_u(z; c)$ for which the implication

$$u \in D_u(z; c) \rightarrow y \in D_y$$

is satisfied. Since $u \in D_u(z; c)$ and $D_y(u, z; c) \subseteq D_y$ are equivalent properties, the relationship $u = \Psi(z)$ may be also obtained by maximization of the probability

$$P[u \in D_u(z; \underline{c}\,)] = \int_{D_c} f_c(c)dc$$

where

$$D_c = D_c(D_y, u, z) = \{c \in C : u \in D_u(z, c)\}.$$

Such a way of obtaining the decision $u = \Psi(z)$ or the decision $u^*$ in the case without disturbances, makes it possible to present a more understandable practical interpretation of the result: This is a decision which with the greatest probability belongs to the set of decisions $D_u$ for which the requirement $y \in D_y$ is satisfied. It is worth noting that it is not a probability that $y \in D_y$ because the properties $u \in D_u$ and $y \in D_y$ are not equivalent. Then the implication inverse to $u \in D_u \rightarrow y \in D_y$ may not be satisfied and consequently, for $y$ the probability distribution $f_y(y)$ does not exist, i.e. $y$ is not a value of a random variable under the assumption that $R$ is not a function.

**Example 7.6.** Let us determine the optimal decision $u^*$ for one-dimensional plant and the following data:

$$cu \le y \le 2cu, \qquad u \ge 0, \qquad D_y = [y_1, y_2], \qquad y_1 \le \frac{y_2}{2},$$

$$f_c(c) = \begin{cases} \lambda e^{-\lambda c} & \text{for} \quad c \ge 0 \\ \\ 0 & \text{for} \quad c \le 0. \end{cases}$$

In this case the set $D_c(D_y, u)$ is determined by the inequality

$$\frac{y_1}{u} \le c \le \frac{y_2}{2u}.$$

Then

$$p(u) = \int_{\frac{y_1}{u}}^{\frac{y_2}{u}} f_c(c)dc = \exp(-\lambda \frac{y_1}{u}) - \exp(-\lambda \frac{y_2}{2u}).$$

From the equation

$$\frac{dp(u)}{du} = 0$$

after some transformations we shall obtain

$$u^* = \arg\max_u p(u) = \frac{\lambda(\frac{y_2}{2} - y_1)}{\ln y_2 - \ln 2y_1}. \qquad \square$$

# 8 Uncertain Variables and Their Applications

In Chap. 7 we assumed that values of unknown quantities (parameters or signals) were values of random variables, i.e. that they had been chosen randomly from determined sets and the descriptions of the uncertainty had a form of probability distributions. Now we shall present the applications of two non-probabilistic descriptions of uncertainty, given by an expert and characterizing his (or her) subjective opinions on values of the unknown quantities. These will be descriptions using so called uncertain variables and fuzzy variables. We shall assume in the first case that values of the unknown quantities are values of uncertain variables, and in the second case that they are values of fuzzy variables. Consequently, we shall speak about uncertain plants and uncertain control algorithms in the first case, and about fuzzy plants and fuzzy control algorithms in the second case. In the wide sense of the word an uncertain system is understood in the book as a system containing any kind and any form of uncertainty in its description (see remarks on the uncertainty in Chaps. 6 and 7). In a narrow sense, an uncertain system is understood as a system with the description based on uncertain variables. In this sense, such names as "random, uncertain and fuzzy knowledge" or "random, uncertain and fuzzy controllers" will be used. Additional remarks will be introduced, if necessary, to avoid misunderstandings.

This chapter concerns the first part of non-probabilistic descriptions of the uncertainty and is devoted to the applications of uncertain variables to analysis and decision making in uncertain control systems. The applications of fuzzy variables will be presented in Chap. 9. Foundations of the uncertain variables theory and their applications to analysis and decision making in uncertain systems may be found in two books [43, 52] and in a lot of papers [26, 32–37, 41, 44, 47, 51, 55, 56].

## 8.1 Uncertain Variables

Let $\omega \in \Omega$ denote an element of a certain set $\Omega$ for which the function

$\bar{x} = g(\omega) \triangleq \bar{x}(\omega)$ determines the value of a certain numerical feature assigned to the element $\omega$. For example, $\Omega$ is a determined set of persons and $\bar{x}(\omega)$ denotes the age of the person $\omega$, or $\Omega$ is a set of resistors and $\bar{x}(\omega)$ denotes the resistance of the resistor $\omega$. Let us assume that the expert does not know the exact value of $\bar{x}$ for the fixed given $\omega$, but using some information concerning $\bar{x}$ and his experience, he gives different approximate values $x_1, x_2, ..., x_m$ of $\bar{x}$ and for each of them presents a degree of certainty $v(x_i)$ that $x_i$ is an approximate value of the feature $\bar{x}$. The estimation $v(x_i)$ will be called a *certainty index* that $\bar{x}$ is approximately equal to $x_i$.

For example, the expert looking at the person $\omega$ characterizes the age of $\omega$ as follows: $v(46) = 0.3$, $v(47) = 0.6$, $v(48) = 0.7$, $v(49) = 0.7$, $v(50) = 1.0$, $v(51) = 0.9$, $v(52) = 0.8$, $v(53) = 0.5$; which means: "$\omega$ is approximately 46 years old" with the certainty index 0.3, "$\omega$ is approximately 47 years old" with the certainty index 0.6 etc.

Let us note that the sentence "$\bar{x}$ is approximately equal to $x$" for the fixed $x$ is not a proposition in two-valued logic, i.e. it is not possible to say whether it is true (its logic value is equal to 1) or false (its logic value is equal to 0). Two-valued propositional logic deals with propositions ($\alpha_1$, $\alpha_2$, ...) whose logic values $w(\alpha) \in \{0,1\}$, and the logic values of negation $\neg\alpha$, disjunction $\alpha_1 \vee \alpha_2$ and conjunction $\alpha_1 \wedge \alpha_2$ are defined by using $w(\alpha)$, $w(\alpha_1)$ and $w(\alpha_2)$. The set $\{0,1\}$ with the definitions of the operations mentioned above is called a *two-valued logic algebra*.

In multi-valued logic we consider the propositions for which the logic value $w(\alpha) \in [0,1]$, i.e. may be any number from the set [0,1]. The operations in the set of logic values may be defined as follows:

$$\left.\begin{aligned} w(\neg\alpha) &= 1 - w(\alpha), \\ w(\alpha_1 \vee \alpha_2) &= \max\{w(\alpha_1), w(\alpha_2)\}, \\ w(\alpha_1 \wedge \alpha_2) &= \min\{w(\alpha_1), w(\alpha_2)\} \end{aligned}\right\} \tag{8.1}$$

where max (min) denotes the greater (the less) from the values in the brackets. It is easy to note that the definitions (8.1) are the same as the known definitions of the operations $\neg$, $\vee$, $\wedge$ in two-valued logic algebra. So the algebra [0,1] with the definitions (8.1) is an extension of two-valued logic algebra to the set [0,1]. In Sect. 6.1 a predicate in two-valued logic has been defined. In multi-valued logic a *predicate* is such a property $\varphi(x)$

concerning the variable $x \in X$, which for a fixed value $x$ is a proposition in multi-valued logic, i.e. $w[\varphi(x)] \in [0,1]$ for every $x$. If $w[\varphi(x)] \in \{0,1\}$ for every $x$ then $\varphi(x)$ will be called a *crisp property*. Otherwise, $\varphi(x)$ will be called a *soft property*. There exist different interpretations of the logic value in multi-valued logic. In our considerations $w[\varphi(x)]$ denotes a degree of the expert's certainty that for the fixed $x$ the property $\varphi(x)$ is satisfied. It will be denoted by $v[\varphi(x)]$ and called a *certainty index* of the property $\varphi(x)$.

Now we shall present formal definitions of two versions of an uncertain variable (in general – a vector uncertain variable). Let $X \subseteq R^k$ denote real number vector space and $g$: $\Omega \to \overline{X}$ denote a function $\overline{x} = g(\omega) \stackrel{\Delta}{=} \overline{x}(\omega)$ where $\overline{X} \subseteq R^k$ and $\overline{X} \supseteq X$.

Let us introduce two soft properties:

1. The property "$\overline{x} \cong x$" which means: "$\overline{x}$ is approximately equal to $x$". The equivalent formulations are the following: "$x$ is an approximate value of $\overline{x}$" or "$x$ belongs to a small neighbourhood of $\overline{x}$". This is a soft property in $X$. Denote by $h(x)$ the logic value of this property

$$w(\overline{x} \cong x) = v(\overline{x} \cong x) \stackrel{\Delta}{=} h(x)$$

and assume that

$$\max_{x \in X} h(x) = 1.$$

2. The property "$\overline{x} \,\widetilde{\in}\, D_x$" where $D_x \subseteq X$, which means: "an approximate value of $\overline{x}$ belongs to $D_x$" or "$\overline{x}$ approximately belongs to $D_x$". This is a soft property of a family of sets $D_x$, generated by "$\overline{x} \cong x$" and the crisp property "$x \in D_x$". The variable $\overline{x}$ will be called an *uncertain variable*. the complete definition contains $h(x)$ and the definitions of the certainty indexes $v(\overline{x} \,\widetilde{\in}\, D_x)$, $v(\overline{x} \,\widetilde{\notin}\, D_x)$, $v(\overline{x} \,\widetilde{\in}\, D_1 \vee \overline{x} \,\widetilde{\in}\, D_2)$, $v(\overline{x} \,\widetilde{\in}\, D_1 \wedge \overline{x} \,\widetilde{\in}\, D_2)$ where $D_1, D_2 \subseteq X$ [36, 37, 52].

**Definition 8.1** (*uncertain variable*). An uncertain variable $\overline{x}$ is defined by the set of values $X$, the function $h(x) = v(\overline{x} \cong x)$ (i.e. the certainty index that $\overline{x} \cong x$, given by an expert) and the following definitions:

$$v(\overline{x} \,\widetilde{\in}\, D_x) = \begin{cases} \max\limits_{x \in D_x} h(x) & \text{for } D_x \neq \varnothing \\ 0 & \text{for } D_x = \varnothing \end{cases}, \tag{8.2}$$

$$v(\bar{x} \,\tilde{\notin}\, D_x) = 1 - v(\bar{x} \,\tilde{\in}\, D_x)\,, \tag{8.3}$$

$$v(\bar{x} \,\tilde{\in}\, D_1 \vee \bar{x} \,\tilde{\in}\, D_2) = \max\{v(\bar{x} \,\tilde{\in}\, D_1), v(\bar{x} \,\tilde{\in}\, D_2)\}\,, \tag{8.4}$$

$$v(\bar{x} \,\tilde{\in}\, D_1 \wedge \bar{x} \,\tilde{\in}\, D_2) = \begin{cases} \min\{v(\bar{x} \,\tilde{\in}\, D_1), v(\bar{x} \,\tilde{\in}\, D_2)\} & \text{for } D_1 \cap D_2 \neq \varnothing \\ 0 & \text{for } D_1 \cap D_2 = \varnothing\,. \end{cases} \tag{8.5}$$

The function $h(x)$ will be called a *certainty distribution*. □

In particular, two cases can occur: the discrete case when $X = \{x_1, x_2, ..., x_m\}$ and the continuous case when $h(x)$ is a continuous function.

In the definition of the uncertain variable not only the formal description but also its interpretation (semantics) is important. The semantics are provided in the following: for the given $\omega$ it is not possible to state whether the crisp property "$x \in D_x$" is true or false because the function $g(\omega)$ and consequently the value of $\bar{x}$ corresponding to $\omega$ are unknown. The exact information, i.e. the knowledge of the function $g$ is replaced by the certainty distribution $h(x)$, which for the given $\omega$ characterizes the different possible approximate values of $\bar{x}(\omega)$. The expert giving the function $h(x)$ in this way determines for the different values $x$ his degree of certainty that $\bar{x}$ is approximately equal to $x$. The certainty index may be given directly by an expert or may be determined when $\bar{x}$ is a known function of an uncertain variable $\bar{y}$ described by a certainty distribution $h_y(y)$ given by an expert.

Using the definitions (8.2)–(8.5) one may prove the following theorem concerning the property $v(\bar{x} \,\tilde{\in}\, D_x)$ [52].

**Theorem 8.1.**

$$v(\bar{x} \,\tilde{\in}\, D_1 \cup D_2) = \max\{v(\bar{x} \,\tilde{\in}\, D_1), v(\bar{x} \,\tilde{\in}\, D_2)\}, \tag{8.6}$$

$$v(\bar{x} \,\tilde{\in}\, D_1 \cap D_2) \leq \min\{v(\bar{x} \,\tilde{\in}\, D_1), v(\bar{x} \,\tilde{\in}\, D_2)\}, \tag{8.7}$$

$$v(\bar{x} \,\tilde{\in}\, \overline{D}_x) \geq v(\bar{x} \,\tilde{\notin}\, D_x) = 1 - v(\bar{x} \,\tilde{\in}\, D_x) \tag{8.8}$$

where $\overline{D}_x$ is a complement of $D_x$, i.e. $\overline{D}_x = X - D_x$. □

It is worth noting that the certainty index $v$ of the property "$\bar{x}$ approximately belongs to the complement of $D_x$" may be greater than the certainty index $v$ of the property "$\bar{x}$ does not belong approximately to $D_x$"

or "an approximate value of $\bar{x}$ does not belong to $D_x$".

We shall define now another version of the uncertain variable called a *C*-uncertain variable. In this version the logic value $w(\bar{x} \,\tilde{\in}\, D_x)$ denoted by $v_c$ is defined in another way.

**Definition 8.2** (*C-uncertain variable*). A *C*-uncertain variable $\bar{x}$ is defined by the set of values *X*, the function $h(x) = v(\bar{x} \cong x)$ given by an expert and the following definitions:

$$v_c(\bar{x} \,\tilde{\in}\, D_x) = \frac{1}{2}[\max_{x \in D_x} h(x) + 1 - \max_{x \in \overline{D}_x} h(x)], \tag{8.9}$$

$$v_c(\bar{x} \,\tilde{\notin}\, D_x) = v_c(\bar{x} \,\tilde{\in}\, \overline{D}_x), \tag{8.10}$$

$$v_c(\bar{x} \,\tilde{\in}\, D_1 \vee \bar{x} \,\tilde{\in}\, D_2) = v_c(\bar{x} \,\tilde{\in}\, D_1 \cup D_2) \ , \tag{8.11}$$

$$v_c(\bar{x} \,\tilde{\in}\, D_1 \wedge \bar{x} \,\tilde{\in}\, D_2) = v_c(\bar{x} \,\tilde{\in}\, D_1 \cap D_2) \ . \tag{8.12}$$

□

According to (8.2) and (8.3)

$$\begin{aligned} v_c(\bar{x} \,\tilde{\in}\, D_x) &= \frac{1}{2}[v(\bar{x} \,\tilde{\in}\, D_x) + v(\bar{x} \,\tilde{\notin}\, \overline{D}_x)] \\ &= \frac{1}{2}[v(\bar{x} \,\tilde{\in}\, D_x) + 1 - v(\bar{x} \,\tilde{\in}\, \overline{D}_x)]. \end{aligned} \tag{8.13}$$

In the formulation of $v(\bar{x} \,\tilde{\in}\, D_x)$, the certainty index of the property $\bar{x} \,\tilde{\in}\, D_x$ is defined "in a positive way", and in the formulation of $v(\bar{x} \,\tilde{\notin}\, \overline{D}_x) \triangleq v_n(\bar{x} \,\tilde{\in}\, D_x)$ – is defined "in a negative way" as a certainty index that $\bar{x}$ does not belong approximately to the complement of $D_x$. The certainty index $v_c$ is defined "in a complex way" taking into account the both properties "$\bar{x} \,\tilde{\in}\, D_x$" and "$\bar{x} \,\tilde{\notin}\, \overline{D}_x$" which are equivalent for *C*-uncertain variable (see (8.10)).

For example, if $\max\,[h(x)\colon\ x \in D_x] = 0.8$ and $\max\,[h(x)\colon\ x \in \overline{D}_x] = 1$ (Fig. 8.1) then $v(\bar{x} \,\tilde{\in}\, D_x) = 0.8$, $v_n(\bar{x} \,\tilde{\in}\, D_x) = 1 - v(\bar{x} \,\tilde{\in}\, \overline{D}_x) = 0$, $v_c(\bar{x} \,\tilde{\in}\, D_x) = 0.4$.

Thus, in the definition of $v_c(\bar{x} \,\tilde{\in}\, D_x)$ the values of $h(x)$ in the set $\overline{D}_x$ are also taken into account. It is also worth noting that in the case of *C*-

uncertain variable the logic operations (negation, disjunction and conjunction) correspond to the operations in the family of sets $D_x$ (complement, union and intersection). On the other hand, one should note that the certainty indexes $v_c$ for disjunction and conjunction are not determined by $v_c(\bar{x} \,\tilde{\in}\, D_1)$ and $v_c(\bar{x} \,\tilde{\in}\, D_2)$, i.e. the determination of these indexes cannot be reduced to the operations in sets of the indexes $v_c(\bar{x} \,\tilde{\in}\, D_x)$.

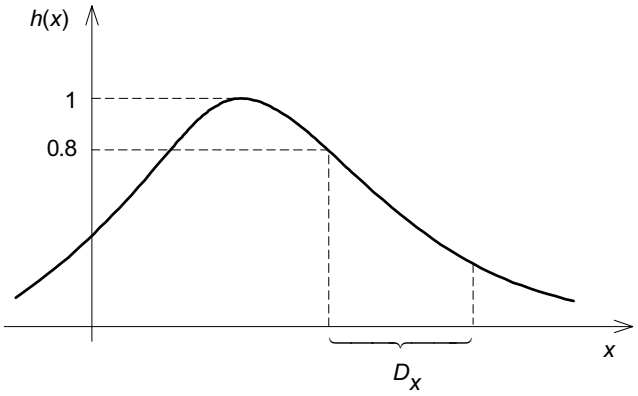


**Fig. 8.1.** Example of certainty distribution

The theorem presenting relationships between the operations for $v_c$, analogous to Theorem 8.1, is as follows:

**Theorem 8.2.**

$$v_c(\bar{x} \,\tilde{\in}\, D_1 \cup D_2) \geq \max\,\{v_c(\bar{x} \,\tilde{\in}\, D_1),\, v_c(\bar{x} \,\tilde{\in}\, D_2)\},$$

$$v_c(\bar{x} \,\tilde{\in}\, D_1 \cap D_2) \leq \min\,\{v_c(\bar{x} \,\tilde{\in}\, D_1),\, v_c(\bar{x} \,\tilde{\in}\, D_2)\},$$

$$v_c(\bar{x} \,\tilde{\notin}\, D_x) = 1 - v_c(\bar{x} \,\tilde{\in}\, D_x). \qquad \square$$

The formula (8.9) or (8.13) can be presented in the form

$$v_c(\bar{x} \,\tilde{\in}\, D_x)$$

$$= \begin{cases} \dfrac{1}{2}\max\limits_{x\in D_x} h(x) = \dfrac{1}{2} v(\bar{x} \,\tilde{\in}\, D_x) & \text{if } \max\limits_{x\in \overline{D}_x} h(x) = 1 \\ 1 - \dfrac{1}{2}\max\limits_{x\in \overline{D}_x} h(x) = v(\bar{x} \,\tilde{\in}\, D_x) - \dfrac{1}{2} v(\bar{x} \,\tilde{\in}\, \overline{D}_x) & \text{otherwise}. \end{cases} \tag{8.14}$$

In particular, for $D_x = \{x\}$ (a singleton), the function $v_c(\bar{x} \cong x) \triangleq h_c(x)$ may be called a *C*-certainty distribution. It is easy to note that in a continuous case $h_c(x) = \frac{1}{2}h(x)$, and in a discrete case

$$h_c(x_i) = \begin{cases} \frac{1}{2}h(x_i) & \text{if } \max_{x \neq x_i} h(x) = 1 \\ 1 - \frac{1}{2}\max_{x \neq x_i} h(x) & \text{otherwise.} \end{cases} \tag{8.15}$$

The *C*-certainty distribution $h_c(x)$ does not determine the certainty index $v_c(\bar{x} \cong D_x)$. In order to determine $v_c$ one should use $h(x)$ given by an expert and apply the formula (8.9) or (8.13). For the uncertain variable one can define a *mean value* $\mathrm{M}(\bar{x})$ in a similar way as an expected value for a random variable. In the discrete case

$$\mathrm{M}(\bar{x}) = \sum_{i=1}^{m} x_i \bar{h}(x_i)$$

where

$$\bar{h}(x_i) = \frac{h(x_i)}{\sum_{j=1}^{m} h(x_j)}.$$

In the continuous case

$$\mathrm{M}(\bar{x}) = \int_X x\bar{h}(x)dx,$$

$$\bar{h}(x) = \frac{h(x)}{\int_X h(x)dx}$$

under the assumption that the respective integrals exist. For *C*-uncertain variable the definition of $\mathrm{M}_c(\bar{x})$ is identical, with $h_c(x)$ instead of $h(x)$. It is easy to note that in the continuous case $\mathrm{M}_c = \mathrm{M}$.

Let us now consider a pair of uncertain variables $(\bar{x}, \bar{y}) = < X \times Y, h(x,y) >$ where $h(x,y) = v[(\bar{x}, \bar{y}) \cong (x,y)]$ is given by an expert and is called a *joint certainty distribution*. Then, using (8.1) for the disjunction in multi-valued logic, we have the following *marginal certainty distributions*:

$$h_x(x) = v(\bar{x} \cong x) = \max_{y \in Y} h(x, y), \tag{8.16}$$

$$h_y(y) = v(\bar{y} \cong y) = \max_{x \in X} h(x, y). \tag{8.17}$$

If the certainty index $v[\bar{x}(\omega) \cong x]$ given by an expert depends on the value of $y$ for the same $\omega$ (i.e. if the expert changes the value $h_x(x)$ when he obtains the value $y$ for the element $\omega$ "under observation") then $h_x(x \mid y)$ may be called a *conditional certainty distribution*. The variables $\bar{x}, \bar{y}$ are called independent when

$$h_x(x \mid y) = h_x(x), \qquad h_y(y \mid x) = h_y(y).$$

Using (8.1) for the conjunction in multi-valued logic we obtain

$$\begin{aligned} h(x, y) = v(\bar{x} \cong x \wedge \bar{y} \cong y) &= \min\{h_x(x), h_y(y \mid x)\} \\ &= \min\{h_y(y), h_x(x \mid y)\}. \end{aligned} \tag{8.18}$$

**Remark 8.1.** The definitions of two versions of the uncertain variables are based on the definitions of so called *uncertain logics* described in [43, 52]. □

**Example 8.1.** Let $X = \{1, 2, 3, 4, 5, 6, 7\}$ and the respective values of $h(x)$ be (0.5, 0.8, 1, 0.6, 0.5, 0.4, 0.2), i.e. $h(1) = 0.5$, $h(2) = 0.8$ etc. Using (8.15) we obtain $h_c(1) = 0.25$, $h_c(2) = 0.4$, $h_c(3) = 1 - \frac{0.8}{2} = 0.6$, $h_c(4) = 0.3$, $h_c(5) = 0.25$, $h_c(6) = 0.2$, $h_c(7) = 0.1$.

Let

$$D_1 = \{1, 2, 4, 5, 6\}, \qquad D_2 = \{3, 4, 5\}.$$

Then $D_1 \cup D_2 = \{1, 2, 3, 4, 5, 6\}$, $D_1 \cap D_2 = \{4, 5\}$, $v(\bar{x} \,\tilde{\in}\, D_1) = \max\{0.5, 0.8, 0.6, 0.5, 0.4\} = 0.8$, $v(\bar{x} \,\tilde{\in}\, D_2) = 1$, $v(\bar{x} \,\tilde{\in}\, D_1 \cup D_2) = \max\{0.5, 0.8, 1, 0.6, 0.5, 0.4\} = 1$, $v(\bar{x} \,\tilde{\in}\, D_1 \vee \bar{x} \,\tilde{\in}\, D_2) = \max\{0.8, 1\} = 1$, $v(\bar{x} \,\tilde{\in}\, D_1 \cap D_2) = \max\{0.6, 0.5\} = 0.6$, $v(\bar{x} \,\tilde{\in}\, D_1 \wedge \bar{x} \,\tilde{\in}\, D_2) = \min\{0.8, 1\} = 0.8$.

Using (8.14) we have

$$v_c(\bar{x} \,\tilde{\in}\, D_1) = \frac{1}{2} v(\bar{x} \,\tilde{\in}\, D_1) = 0.4,$$

$$v_c(\bar{x} \,\tilde{\in}\, D_2) = 1 - \frac{1}{2} v(\bar{x} \,\tilde{\in}\, D_2) = 1 - \frac{0.8}{2} = 0.6,$$

$$v_c(\bar{x} \,\tilde{\in} D_1 \vee \bar{x} \,\tilde{\in} D_2) = v_c(\bar{x} \,\tilde{\in} D_1 \cup D_2) = 1 - \frac{0.2}{2} = 0.9,$$

$$v_c(\bar{x} \,\tilde{\in} D_1 \wedge \bar{x} \,\tilde{\in} D_2) = v_c(\bar{x} \,\tilde{\in} D_1 \cap D_2) = \frac{0.6}{2} = 0.3.$$

In the example considered, for $D_1$ as well as for $D_2$ we have $v_c(\bar{x} \,\tilde{\in} D) = \max h_c(x)$ for $x \in D$. Let $D = \{2, 3, 4\}$. Now

$$v_c(\bar{x} \,\tilde{\in} D) = 1 - \frac{0.5}{2} = 0.75$$

and

$$\max h_c(x) = \max \{0.4, 0.6, 0.3\} = 0.6 < v_c. \qquad \square$$

## 8.2 Application of Uncertain Variables to Analysis and Decision Making (Control) for Static Plant

Now let us consider shortly decision making problems using uncertain variables, analogous to the problems considered in Sects. 7.1, 7.2 and 7.9 for the static plant with a probabilistic description of uncertainty. The decision problems will be preceded by a short presentation of the analysis problems.

### 8.2.1 Parametric Uncertainty

Let us consider a plant described by a function

$$y = \Phi(u, z, c)$$

where $z \in Z$ is a vector of the disturbances which can be measured (see (7.9)) and $c \in C$ is an unknown vector parameter which is assumed to be a value of an uncertain variable with the certainty distribution $h_c(c)$ given by an expert. Consequently, $y$ is a value of an uncertain variable $\bar{y} = \Phi(u, z, \bar{c})$.

**Analysis problem:** For the given $\Phi$, $h_c(c)$, $u$ and $z$ find the certainty distribution $h_y(y)$.

According to (8.2)

$$h_y(y;u,z) = v(\bar{y} \,\tilde{=}\, y) = v[\bar{c} \,\tilde{\in} D_c(y;u,z)] = \max_{c \in D_c(y;u,z)} h_c(c) \qquad (8.19)$$

where

$$D_c(y;u,z) = \{c \in C : \Phi(u,z,c) = y\}.$$

Having $h_y(y;u,z)$ one can determine the mean value

$$\mathrm{M}_y(\bar{y};u,z) = \int_Y y h_y(y;u,z)dy \cdot [\int_Y h_y(y;u,z)dy]^{-1} \stackrel{\Delta}{=} \Phi_b(u,z) \qquad (8.20)$$

(for the continuous case) and

$$\hat{y} = \arg\max_{y \in Y} h_y(y;u,z),$$

i.e. such a value $\hat{y}$ that $h_y(\hat{y};u,z) = 1$. If $\Phi$ as a function of $c$ is one-to-one mapping and $c = \Phi^{-1}(u,z,y)$ then

$$h_y(y;u,z) = h_c[\Phi^{-1}(u,z,y)] \qquad (8.21)$$

and $\hat{y} = \Phi(u,z,\hat{c})$ where $\hat{c}$ satisfies the equation $h_c(c) = 1$. It is easy to note that $\hat{y} = \hat{y}_c$ where

$$\hat{y}_c = \arg\max_{y \in Y} h_{cy}(y;u,z)$$

and $h_{cy}$ is a certainty distribution for the *C*-uncertain variable.

**Decision problem:** For the given $\Phi$, $h_c(c)$, $z$ and $y^*$

I. One should find $u \stackrel{\Delta}{=} u_a$ maximizing $v(\bar{y} \cong y^*)$.

II. One should find $u \stackrel{\Delta}{=} u_b$ such that $\mathrm{M}_y(\bar{y}) = y^*$.

In version I

$$u_a = \arg\max_{u \in U} \Phi_a(u,z) \stackrel{\Delta}{=} \Psi_a(z) \qquad (8.22)$$

where $\Phi_a(u,z) = h_y(y^*;u,z)$ and $h_y$ is determined according to (8.19). The result $u_a$ is a function of $z$ if $u_a$ is a unique value maximizing $\Phi_a$ for the given $z$.

In version II one should solve the equation

$$\Phi_b(u,z) = y^* \qquad (8.23)$$

where the function $\Phi_b$ is determined by (8.20). If equation (8.23) has a unique solution with respect to $u$ for the given $z$ then as a result one obtains $u_b = \Psi_b(z)$. The functions $\Psi_a$ and $\Psi_b$ are two versions of the deci-

sion algorithm $u = \Psi(z)$ in an open-loop decision system. It is worth noting that $u_a$ is a decision for which $v(\overline{y} \cong y^*) = 1$.

The functions $\Phi_a$, $\Phi_b$ are the results of two different ways of *determinization* of the uncertain plant, and the functions $\Psi_a$, $\Psi_b$ are the respective decision algorithms based on the *knowledge of the plant* (KP):

$$\text{KP} = < \Phi, h_c > . \tag{8.24}$$

Assume that the equation

$$\Phi(u,z,c) = y^*$$

has a unique solution with respect to $u$:

$$u \stackrel{\Delta}{=} \Phi_d(z,c) . \tag{8.25}$$

The relationship (8.25) together with the certainty distribution $h_c(c)$ may be considered as a *knowledge of the decision making* (KD):

$$\text{KD} = < \Phi_d, h_c > , \tag{8.26}$$

obtained by using KP and $y^*$. Equation (8.25) together with $h_c$ may also be called an *uncertain decision algorithm* in the open-loop decision system. The determinization of this algorithm leads to two versions of the deterministic decision algorithm $\Psi_d$, corresponding to versions I and II of the decision problem:

*Version I.*

$$u_{ad} = \arg\max_{u \in U} h_u(u;z) \stackrel{\Delta}{=} \Psi_{ad}(z) \tag{8.27}$$

where

$$h_u(u;z) = \max_{c \in D_c(u;z)} h_c(c) \tag{8.28}$$

and

$$D_c(u;z) = \{c \in C : u = \Phi_d(z,c)\} .$$

*Version II.*

$$u_{bd} = \text{M}_u(\overline{u};z) \stackrel{\Delta}{=} \Psi_{bd}(z) . \tag{8.29}$$

The decision algorithms $\Psi_{ad}$ and $\Psi_{bd}$ are based directly on the knowledge of the decision making. Two concepts of the determination of deterministic decision algorithms are illustrated in Figs. 8.2 and 8.3. In the first case (Fig. 8.2) the decision algorithms $\Psi_a(z)$ and $\Psi_b(z)$ are obtained via the determinization of the knowledge of the plant KP. In the

second case (Fig. 8.3) the decision algorithms $\Psi_{ad}(z)$ and $\Psi_{bd}(z)$ are based on the determinization of the knowledge of the decision making KD obtained from KP for the given $y^*$. The results of these two approaches may be different.

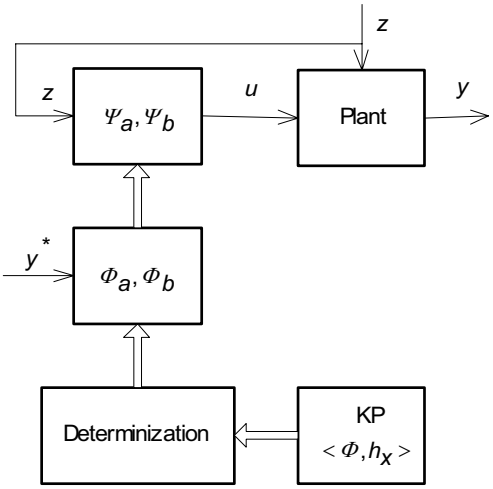


**Fig. 8.2.** Decision system with determinization – the first case

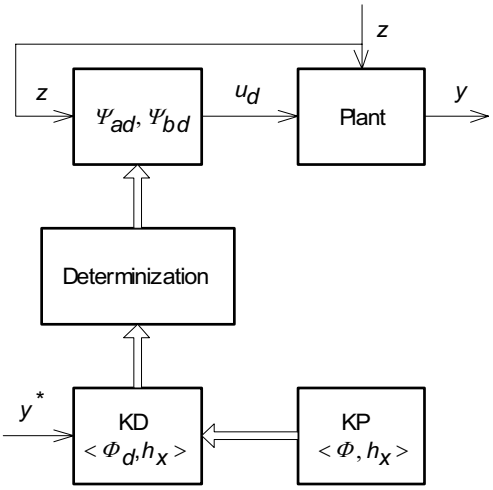


**Fig. 8.3.** Decision system with determinization – the second case

**Theorem 8.3.** For the plant described by KP in the form (8.24) and for KD in the form (8.26), if there exists an inverse function $c = \Phi^{-1}(u,z,y)$ then

$$\Psi_a(z) = \Psi_{ad}(z).$$

*Proof*: According to (8.21) and (8.27)

$$h_y(y^*;u,z) = h_c[\Phi^{-1}(u,z,y^*)],$$

$$h_u(u;z) = h_c[\Phi^{-1}(u,z,y^*)].$$

Then, by using (8.22) and (8.27) we obtain $\Psi_a(z) = \Psi_{ad}(z)$. □

**Example 8.2.** Let $u, y, c, z \in R^1$ and

$$y = cu + z.$$

Then

$$\mathrm{M}_y(\bar{y}) = u\,\mathrm{M}_c(\bar{c}) + z$$

and from the equation $\mathrm{M}_y(\bar{y}) = y^*$ we obtain

$$u_b = \Psi_b(z) = \frac{y^* - z}{\mathrm{M}_c(\bar{c})}.$$

The uncertain decision algorithm is

$$u = \Phi_d(z,c) = \frac{y^* - z}{c}$$

and after the determinization

$$u_{bd} = \Psi_{bd}(z) = (y^* - z)\mathrm{M}_c(\bar{c}^{-1}) \neq \Psi_b(z).$$ □

This very simple example shows that the deterministic decision algorithm $\Psi_b(z)$ obtained via the determinization of the uncertain plant may differ from the deterministic decision algorithm $\Psi_{bd}(z)$ obtained as a result of the determinization of the uncertain decision algorithm.

### 8.2.2 Non-parametric Uncertainty

Now we shall present a non-parametric decision problem analogous to that described in Sect. 7.2 for the probabilistic description of uncertainty. Consider a static plant with input vector $u \in U$, output vector $y \in Y$ and a vector of external disturbances $z \in Z$, and assume that $(u,y,z)$ are values

of the uncertain variables $(\overline{u}, \overline{y}, \overline{z})$. The plant is described by

$$\mathrm{KP} = < h_y(y|u,z) >$$

where $h_y(y|u,z)$ is a conditional certainty distribution given by an expert. The decision problem consists in finding an uncertain decision (control) algorithm in the form of $h_u(u|z)$ for the required $h_y(y)$ given by a user.

**Decision (control) problem:** For the given $\mathrm{KP} = < h_y(y|u,z) >$ and $h_y(y)$ required by a user one should determine $h_u(u|z)$.

The determination of $h_u(u|z)$ may be decomposed into two steps. In the first step, one should find the function $h_{uz}(u,z)$ satisfying the equation

$$h_y(y) = \max_{u \in U,\, z \in Z} \min\{h_{uz}(u,z), h_y(y|u,z)\} \tag{8.30}$$

and the conditions for a certainty distribution:

$$\bigwedge_{u \in U} \bigwedge_{z \in Z} h_{uz}(u,z) \geq 0, \quad \max_{u \in U, z \in Z} h_{uz}(u,z) = 1 .$$

In the second step, one should determine the function $h_u(u|z)$ satisfying the equation

$$h_{uz}(u,z) = \min\{h_z(z), h_u(u|z)\} \tag{8.31}$$

where

$$h_z(z) = \max_{u \in U} h_{uz}(u,z), \tag{8.32}$$

and the conditions for a certainty distribution:

$$\bigwedge_{u \in U} \bigwedge_{z \in Z} h_u(u|z) \geq 0, \qquad \bigwedge_{z \in Z} \max_{u \in U} h_u(u|z) = 1 .$$

The solution may be not unique. The function $h_u(u|z)$ may be considered as a knowledge of the decision making $\mathrm{KD} = < h_u(u|z) >$ or an *uncertain decision algorithm* (the description of an *uncertain controller* in the open-loop control system). Having $h_u(u|z)$, one can obtain the deterministic decision algorithm $\Psi(z)$ as a result of the determinization of the uncertain decision algorithm $h_u(u|z)$. Two versions corresponding to the versions presented in Sect. 7.2 are the following:

*Version I.*

$$u_a = \arg\max_{u \in U} h_u(u|z) \triangleq \Psi_a(z). \tag{8.33}$$

*Version II.*

$$u_b = \mathrm{M}_u(\overline{u}|z) = \int_U u h_u(u|z)du \cdot [\int_U h_u(u|z)du]^{-1} \triangleq \Psi_b(z). \tag{8.34}$$

The deterministic decision algorithms $\Psi_a(z)$ or $\Psi_b(z)$ are based on the knowledge of the decision making $\mathrm{KD} = < h_u(u|z) >$, which is determined from the knowledge of the plant KP for the given $h_y(y)$.

**Theorem 8.4.** The set of functions $h_u(u|z)$ satisfying equation (8.31) is determined as follows:

$$h_u(u|z) = \begin{cases} = h_{uz}(u,z) & \text{for} \quad (u,z) \notin D(u,z) \quad (8.35) \\ \geq h_{uz}(u,z) & \text{for} \quad (u,z) \in D(u,z) \quad (8.36) \end{cases}$$

where

$$D(u,z) = \{(u,z) \in U \times Z: \quad h_z(z) = h_{uz}(u,z)\}.$$

*Proof*: From (8.31) it follows that

$$\bigwedge_{u \in U} \bigwedge_{z \in Z} [h_z(z) \geq h_{uz}(u,z)].$$

If $h_z(z) > h_{uz}(u,z)$ then, according to (8.31), $h_{uz}(u,z) = h_u(u|z)$. If $h_z(z) = h_{uz}(u,z)$, i.e. $(u,z) \in D(u,z)$ then $h_u(u|z) \geq h_{uz}(u,z)$. □

In general, the solution of the problem in the second step is not unique, i.e. we can choose any function $h_u(u|z)$ satisfying the condition (8.36) for $(u,z) \in D(u,z)$, such that

$$\bigwedge_{z \in Z} \max_{u \in U} h_u(u|z) = 1.$$

For the fixed $z$, the set

$$D_u(z) = \{u \in U: (u,z) \in D(u,z)\}$$

is a set of values $u$ maximizing $h_{uz}(u,z)$. If $D_u(z) = \{\hat{u}(z)\}$ (a singleton), then

$$\hat{u}(z) = \arg\max_{u \in U} h_{uz}(u,z)$$

and $h_u(\hat{u}|z)=1$, i.e.

$$h_u(u|z)=\begin{cases} h_{uz}(u,z) & \text{for} \quad u\neq\hat{u}(z) \\ 1 & \text{for} \quad u=\hat{u}(z). \end{cases} \tag{8.37}$$

It is easy to note that $h_u(u|z)$ determined by (8.37) is a continuous function for every $z\in Z$ if and only if

$$\bigwedge_{z\in D_z}[h_z(z)=1],$$

i.e.

$$\bigwedge_{z\in D_z}[\max_{u\in U} h_{uz}(u,z)=1] \tag{8.38}$$

where

$$D_z=\{z\in Z:\bigvee_{u\in U} h_{uz}(u,z)\neq 0\}.$$

If the condition (8.38) is satisfied then $h(u|z)=h_{uz}(u,z)$. In this case , according to (8.33) the decision $u_a$ does not depend on $z$ and the decision $u_b$ (8.34) does not depend on $z$ if $u_b=u_a$. It is worth noting that if $D_u(z)$ is a continuous domain, we may obtain a continuous function $h_u(u|z)$ and the decisions $u_a$, $u_b$ depending on $z$.

**Remark 8.2.** The distribution $h_y(y\,|\,u,z)$ given by an expert and/or the result $h_u(u\,|\,z)$ may not satisfy the condition $\max h=1$ (see Example 8.3). The normalization in the form

$$\bar{h}_u(u\,|\,z)=\frac{h_u(u\,|\,z)}{\max\limits_{u\in U} h_u(u\,|\,z)} \tag{8.39}$$

is not necessary if we are interested in the deterministic decisions $u_a$ and $u_b$, which are the same for $h_u$ and $\bar{h}_u$. □

In a way analogous to that for the probabilistic description (Sect. 7.2), we may formulate two versions of the non-parametric decision problem for the deterministic requirement $y=y^*$:

*Version I.*

$$u_a = \max_{u \in U} h_y(y^* \mid u,z) \stackrel{\Delta}{=} \overline{\Psi}_a(z).$$

*Version II.* $u_b \stackrel{\Delta}{=} \overline{\Psi}_b(z)$ is a solution of the equation

$$\int_Y y\, h_y(y \mid u,z)dy \cdot [\int_Y h_y(y \mid u,z)dy]^{-1} = y^*.$$

The deterministic algorithms $\Psi_a(z)$ and $\Psi_b(z)$ are based on the determinization of the plant and, in general, differ from the algorithms $\Psi_a(z)$ and $\Psi_b(z)$ in (8.33) and (8.34), obtained via the determinization of the uncertain decision algorithm $h_u(u \mid z)$.

**Example 8.3.** Consider a plant with $u, y, z \in R^1$, described by the conditional certainty distribution given by an expert:

$$h_y(y \mid u,z) = -(y-d)^2 + 1 - u - (b-z) \tag{8.40}$$

for

$$0 \le u \le \frac{1}{2}, \qquad b - \frac{1}{2} \le z \le b,$$

$$-\sqrt{1-u-(b-z)} + d \le y \le \sqrt{1-u-(b-z)} + d,$$

and $h_y(y \mid u,z) = 0$ otherwise.

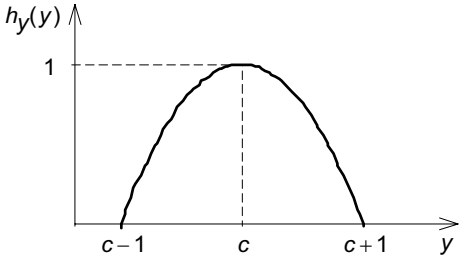


**Fig. 8.4.** Parabolic certainty distribution

For the certainty distribution required by a user (Fig. 8.4):

$$h_y(y) = \begin{cases} -(y-c)^2 + 1 & \text{for } c-1 \le y \le c+1 \\ 0 & \text{otherwise}, \end{cases}$$

one should determine the uncertain decision algorithm in the form

$$h_u(u \mid z) = h_{uz}(u,z)\,.$$

Let us assume that $b > 0,\ c > 1$ and

$$c + 1 \le d \le c + 2\,. \tag{8.41}$$

Then the equation

$$h_y(y) = h_y(y \mid u,z)$$

has a unique solution $y(u,z)$, which is reduced to the solution of the equation

$$-(y-c)^2 + 1 = -(y-d)^2 + 1 - u - (b-z)$$

and

$$y(u,z) = \frac{d^2 - c^2 + u + b - z}{2(d-c)} = \frac{1}{2}(d + c + \frac{u+b-z}{d-c})\,. \tag{8.42}$$

Using (8.42) and (8.41) we obtain

$$h_{uz}(u \mid z) = h_{uz}(u,z) = h_y[y(u,z)]$$

$$= \begin{cases} -[\dfrac{(d-c)^2 + u + b - z}{2(d-c)}]^2 + 1 & \text{for} \quad u \le 1 - [(d-c)-1]^2 - (b-z), \\ & 0 \le u \le \dfrac{1}{2},\ b - \dfrac{1}{2} \le z \le b \\ 0 & \text{otherwise}\ . \end{cases}$$

The values of $h_{uz}(u,z)$ may be greater than zero (i.e. the solution of our decision problem exists) if for every $z$

$$1 - [(d-c)-1]^2 - (b-z) > 0\,. \tag{8.43}$$

Taking into account the inequality

$$0 \le b - z \le \frac{1}{2},$$

and (8.41), we obtain from (8.43) the following condition:

$$d - c < 1 + \frac{1}{\sqrt{2}}\,.$$

Note that the description (8.40) given by an expert and the solution $h_u(u \mid z) = h_{uz}(u,z)$ do not satisfy the condition $\max h = 1$ (see Remark 8.2). □

## 8.3 Relational Plant with Uncertain Parameter

Let us consider the plant such as in Sect. 7.9, i.e. described by the relation

$$R(u, y; c) \subset U \times Y$$

with an unknown parameter $c \in C$ which is assumed to be a value of an uncertain variable $\overline{c}$ with the certainty distribution $h_c(c)$ given by an expert. As in Sect. 7.9 let us introduce the set of possible outputs for the given $u$

$$D_y(u; c) = \{y \in Y: (u, y) \in R(u, y; c)\}. \tag{8.44}$$

**Analysis problem:** For the given $R$, $h_c(c)$, $u$ and $\Delta_y \subset Y$ one should determine

$$v[\Delta_y \mathrel{\tilde{\subseteq}} D_y(u; \overline{c})] \stackrel{\Delta}{=} g(\Delta_y, u), \tag{8.45}$$

i.e. the certainty index that every value $y$ belonging to the set $\Delta_y$ given by a user may appear at the output of the plant. In other words, this is the certainty index that for the given $u$ the approximate set of possible outputs contains all values from the set $\Delta_y$, or that the set of possible outputs approximately contains the set $\Delta_y$.

Let us note that

$$v[\Delta_y \mathrel{\tilde{\subseteq}} D_y(u; \overline{c})] = v[\overline{c} \mathrel{\tilde{\in}} D_c(\Delta_y, u)] \tag{8.46}$$

where

$$D_c(\Delta_y, u) = \{c \in C: \ \Delta_y \subseteq D_y(u; c)\}.$$

Then

$$g(\Delta_y, u) = \max_{c \in D_c(\Delta_y, u)} h_c(c). \tag{8.47}$$

In particular, for $\Delta_y = \{y\}$ the certainty index that the given value may appear at the output of the plant is the following:

$$g(y,u) = \max_{c \in D_c(y,u)} h_c(c)$$

where

$$D_c(y, u) = \{c \in C: \ y \in D_y(u; c)\}.$$

The formulas (8.45), (8.46) and (8.47) are analogous to the formulas (7.68), (7.69) and (7.70) for the probabilistic description.

**Decision making problem:** For the given $R$, $h_c(c)$ and $D_y \subset Y$ formulated by a user one should determine the decision $u^*$ maximizing

$$v[D_y(u; \bar{c}\,) \,\tilde{\subseteq}\, D_y] \triangleq v(u) \tag{8.48}$$

where $D_y(u; c)$ is defined by the formula (8.44).

It is one of possible formulations of a decision problem, consisting in the determination of the decision $u^*$ giving the greatest certainty index that the approximate set of possible outputs (i.e. the set of possible outputs for an approximate value of $\bar{c}$ ) belongs to the given set $D_y$, or that the set of possible outputs approximately belongs to $D_y$. Since

$$v[D_y(u; \bar{c}\,) \,\tilde{\subseteq}\, D_y] = v[\bar{c} \,\tilde{\in} D_c(D_y, u)] = \max_{c \in D_c(D_y, u)} h_c(c) \tag{8.49}$$

where

$$D_c(D_y, u) = \{c \in C:\ D_y(u; c) \subseteq D_y\}$$

then

$$u^* = \arg \max_u v(u) = \arg \max_u \max_{c \in D_c(D_y, u)} h_c(c). \tag{8.50}$$

The formulas (8.48), (8.49) and (8.50) are analogous to the formulas (7.71), (7.72) and (7.73) for the probabilistic description. It is worth noting that the solution may be not unique, i.e. we may obtain the set of decisions $D_u$ (8.50). Denote by $\tilde{c}$ the value maximizing $h_c$, i.e. $h_c(\tilde{c}\,) = 1$. Then

$$D_u = \{u \in U:\ \tilde{c} \in D_c(D_y, u)\} \tag{8.51}$$

and for every $u \in D_u$ the maximum value of the certainty index $v(u) = 1$. Hence, to determine $D_u$ it is not necessary to know the form of $h_c(c)$ but it is sufficient to know $\tilde{c}$ only. If $\bar{c}$ is considered as $C$-uncertain variable then one should determine $v$ according to the formula (8.49) and

$$\max_{c \in \overline{D}_c(D_y, u)} h_c(c) \triangleq \bar{v} \tag{8.52}$$

where $\overline{D}_c(D_y, u)$ is a complement of the set $D_c(D_y, u)$. Then, according to (8.13)

$$v_c[D_y(u, \bar{c}\,) \,\tilde{\subseteq}\, D_y] = v_c[\bar{c} \,\tilde{\in} D_c(D_y, u)] = \frac{1}{2}(v + 1 - \bar{v}) \triangleq v_c(u) \tag{8.53}$$

and

$$u_c^* = \arg \max_u \; v_c[D_y(u, \bar{c}) \mathrel{\tilde{\subseteq}} D_y].$$

Now the knowledge of $h_c(c)$ is necessary to determine the optimal decision.

The above considerations can be extended to the plant described by the relation $R(u, y, z; c)$ where the disturbance $z$ can be measured.

**Decision making problem:** For the given $R$, $h_c(c)$, $z$ and $D_y$ one should determine $u$ maximizing

$$v[D_y(u, z; \bar{c}) \mathrel{\tilde{\subseteq}} D_y] \triangleq \Phi(u, z)$$

where $D_y(u, z; c) = \{y \in Y: (u, y, z) \in R(u, y, z; c)\}$. In the same way as for the plant without the disturbance we determine the control algorithm in an open-loop system in the form of the set of optimal decisions $D_u(z)$ dependent on $z$. For every $u \in D_u(z)$ the certainty index $\Phi(u, z) = 1$. In the case of a unique solution, the control algorithm is reduced to the function $u = \Psi(z)$. This is a control algorithm based on the knowledge of the plant $\mathrm{KP} = < R, h_c >$.

For the fixed $c$ and $z$ one can solve the decision problem such as in Sect. 6.3, i.e. determine the largest set $D_u(z; c)$ such that the implication $u \in D_u(z; c) \rightarrow y \in D_y$ is satisfied. According to (6.19)

$$D_u(z; c) = \{u \in U: \; D_y(u, z; c) \subseteq D_y\} \triangleq \bar{R}(z, u; c).$$

Then we can find the decision

$$u_d = \arg \max_u v[u \mathrel{\tilde{\in}} D_u(z; \bar{c})] \triangleq \Psi_d(z) \tag{8.54}$$

where

$$v[u \mathrel{\tilde{\in}} D_u(z; \bar{c})] = v[\bar{c} \mathrel{\tilde{\in}} D_{cd}(D_y, u, z)] = \max_{c \in D_{cd}} h_c(c),$$

$$D_{cd}(D_y, u, z) = \{c \in C: \; u \in D_u(z; c)\}.$$

In a similar way as in the former case, we can obtain not one decision $u_d = \Psi_d(z)$ but the set of decisions, maximizing the certainty index in (8.54). Let us note that $\Psi_d(z)$ is a decision algorithm (a control algorithm in an open-loop system) based on the knowledge of the decision making $\mathrm{KD} = < \bar{R}, h_c >$. The relation $\bar{R}$ or the set $D_u(z; c)$ is an uncertain control algorithm in our case. For a concrete measured $z$, it is the set of all possi-

ble control decisions or the set of all $u$ for which the requirement is satisfied. It is easy to note that now $u_d = u$ for every $z$, i.e. $\Psi_d(z) = \Psi(z)$. This equality follows from the fact that the properties $u \in D_u(z;\ c)$ and $D_y(u, z;\ c) \subseteq D_y$ are equivalent. Consequently, the certainty indexes that these properties are approximately satisfied are identical. This remark provides a clearer interpretation of the decision $u = u_d$ or the decision $u^*$ (8.50) in the case without disturbances: This is a decision which with the greatest certainty index belongs approximately to the decision set $D_u$ for which the requirement $y \in D_y$ is satisfied. When the standard version of an uncertain variable (i.e. not $C$-uncertain variable) is applied, this greatest certainty index is equal to 1. It is worth noting that we cannot determine and maximize directly a certainty index that $y$ approximately belongs $D_y$ because the properties $u \in D_u$ and $y \in D_y$ are not equivalent, i.e. the implication inverse to $u \in D_u \to y \in D_y$ may not be satisfied. In other words, for $y$ the distribution $h_y(y)$ does not exist, i.e. $y$ is not a value of an uncertain variable.

More details on uncertain variables and their applications to analysis and decision making in uncertain systems may be found in [36, 37, 43, 52].

**Example 8.4.** Consider the plant such as in Example 7.6 in Sect. 7.9 and assume that the value of an unknown parameter $c$ is a value of an uncertain variable with the triangular certainty distribution presented in Fig. 8.5. One should determine the decision $u^*$.

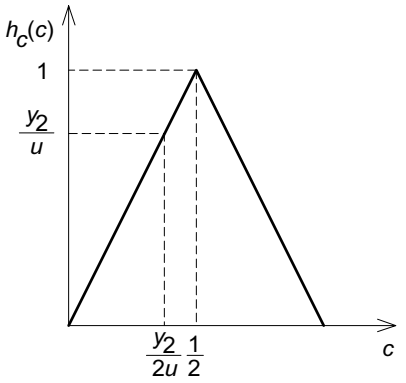


**Fig. 8.5.** Example of certainty distribution

In this case

$$D_c(D_y, u) = [\frac{y_1}{u}, \frac{y_2}{2u}], \qquad \widetilde{c} = \frac{1}{2}.$$

By applying the index $v$ (a standard version) we obtain the following set $D_u$ of the decisions $u^*$ (8.53):

$$D_u = \{u \in U: \ \widetilde{c} \in [\frac{y_1}{u}, \frac{y_2}{2u}]\} = [\frac{y_1}{\widetilde{c}}, \frac{y_2}{2\widetilde{c}}] = [2y_1, y_2].$$

For every decision from this set, the certainty index that this decision belongs to the set of decisions for which the requirement $y \in [y_1, y_2]$ is satisfied – is equal to 1. Let us assume now that $\overline{c}$ is $C$-uncertain variable and determine $v_c(u)$. In order to find it we determine $v(u)$ according to the formula (8.49):

$$v(u) = \begin{cases} \frac{y_2}{u} & \text{for} \quad u \geq y_2 \\ 1 & \text{for} \quad 2y_1 \leq u \leq y_2 \\ 2(1 - \frac{y_1}{u}) & \text{for} \quad y_1 \leq u \leq 2y_1 \\ 0 & \text{for} \quad u \leq y_1 \end{cases}.$$

The case $u \geq y_2$, i.e. $\frac{y_2}{2u} \leq \frac{1}{2}$ has been illustrated in Fig. 8.5. In a similar way we determine $\overline{v}(u)$ according to the formula (8.52):

$$\overline{v}(u) = \begin{cases} 1 & \text{for} \quad u \geq y_2 \\ 2 - \frac{y_2}{u} & \text{for} \quad y_1 + \frac{y_2}{2} \leq u \leq y_2 \\ \frac{2y_1}{u} & \text{for} \quad 2y_1 \leq u \leq y_1 + \frac{y_2}{2} \\ 1 & \text{for} \quad u \leq 2y_1 \end{cases}.$$

According to (8.53), after some transformations we obtain

$$v_c(u) = \begin{cases} \frac{y_2}{2u} & \text{for} \quad u \geq y_1 + \frac{y_2}{2} \\ 1 - \frac{y_1}{u} & \text{for} \quad y_1 \leq u \leq y_1 + \frac{y_2}{2} \\ 0 & \text{for} \quad u \leq y_1 \end{cases}$$

and

$$u_c^* = \arg\max\ v_c(u).$$

For example, for $y_1 = 2$ and $y_2 = 12$ we obtain $u^* \in [4, 12]$, $v = 1$ in the first case and $u_c^* = 8$, $v_c = 0.75$ in the second case. The function $v_c(u)$ is illustrated in Fig. 8.6. □

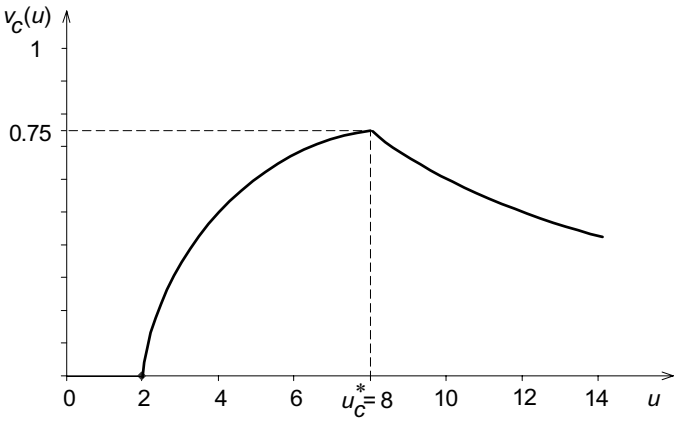


**Fig. 8.6.** Illustration of relationship $v_c(u)$

## 8.4 Control for Dynamical Plants. Uncertain Controller

The description based on uncertain variables can be used to control problems for a dynamical plant in a way analogous to that for the dynamical plant with a probabilistic description. In particular, one can consider a multistage decision problem (a control of a discrete dynamical plant) analogous to that described in Sect. 7.5, in which the certainty distribution $h_z(z)$ will occur in the place of $f_z(z)$ and an expected value $\mathrm{E}(\underline{Q})$ will be replaced by a mean value $\mathrm{M}(\overline{Q})$.

In this section the considerations will be limited to two basic problems for a dynamical plant with a parametric uncertainty, analogous to the problems mentioned at the beginning of Sect. 7.6 for the probabilistic case. For the plant with an uncertain parameter $c$ one may apply the parametric optimization in a way similar to that presented in Chap. 5. Now, the performance index $Q = \Phi(c, a)$ is a function of the unknown parameter $c$ and the

parameter $a$ in the control algorithm, which is to be determined. The closed-loop control system is then considered as a static plant with the input $a$, the output $Q$ and the unknown parameter $c$, for which we can formulate and solve the decision problems described in Sect. 8.2.1. The control problem consisting in the determination of $a$ (in general, a vector parameter) may be formulated as follows.

**Control problem:** For the given models of the plant and the controller find the value $a^*$ minimizing $\mathrm{M}(\overline{Q})$, i.e. the mean value of the performance index.

The procedure for solving the problem is then the following:

1. To determine the function $Q = \Phi(a,c)$.
2. To determine the certainty distribution $h_q(q;a)$ for $\overline{Q}$ using the function $\Phi$ and the distribution $h_c(c)$ in the same way as in the formula (8.19) for $\overline{y}$.
3. To determine the mean value $\mathrm{M}(\overline{Q};a)$.
4. To find $a^*$ minimizing $\mathrm{M}(\overline{Q};a)$.

In order to apply the second case of the determinization, corresponding to the determination of $\Psi_d$ for the static plant (see Sect. 8.2.1), it is necessary to find the value $a(c)$ minimizing $Q = \Phi(a,c)$ for the fixed $c$. The control algorithm with the uncertain parameter $a(c)$ may be considered as a knowledge of the control in our case, and the controller with this parameter is an *uncertain controller* in the closed-loop system. To obtain the deterministic control algorithm, one should substitute $\mathrm{M}(\overline{a})$ in place of $a(c)$ in the uncertain control algorithm, where the mean value $\mathrm{M}(\overline{a})$ should be determined by using the function $a(c)$ and the certainty distribution $h_c(c)$.

Assume that the state of the plant $x(t)$ is put at the input of the controller. Then the uncertain controller has a form

$$u = \Psi(x,c)$$

which may be obtained as a result of non-parametric optimization, i.e. $\Psi$ is the optimal control algorithm for the given model of the plant with the fixed $c$ and for the given form of a performance index. Then

$$u_d = \mathrm{M}(\overline{u};x) \triangleq \Psi_d(x)$$

where $\mathrm{M}(\overline{u};x)$ is determined by using the distribution

$$h_u(u;x) = v[\bar{c} \,\tilde{\in}\, D_c(u;x)] = \max_{c \in D_c(u;x)} h_c(c)$$

and

$$D_c(u;x) = \{c \in C : u = \Psi(x,c)\} .$$

**Example 8.5.** The data for the linear control system under consideration (Fig. 8.7) are the following:

$$K_{\mathrm{O}}(s;c) = \frac{c}{(sT_1+1)(sT_2+1)}, \qquad K_{\mathrm{R}}(s;a) = \frac{a}{s},$$

$z(t) = 0$ for $t < 0$, $z(t) = 1$ for $t \geq 0$, $h_c(c)$ has a triangular form presented in Fig. 8.8.

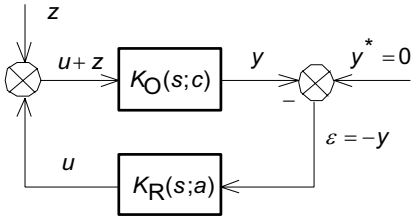


**Fig. 8.7.** Closed-loop control system

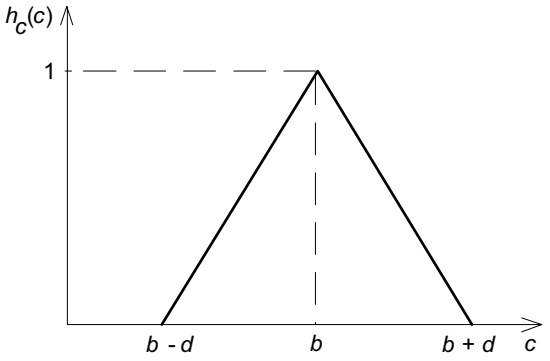


**Fig. 8.8.** Example of certainty distribution

In Example 5.1 we determined the function $Q = \Phi(c,a)$ for the optimization problem considered and we found the optimal parameter of the controller

$$a = \frac{\alpha}{c}, \qquad \alpha = \frac{T_1 + T_2}{T_1 T_2}.$$

The uncertain controller is then described by

$$K_{\mathrm{R}}(s) = \frac{a(c)}{s} = \frac{\alpha}{cs}. \tag{8.55}$$

The certainty distribution $h_a(a)$ is as follows:

$$h_a(a) = \begin{cases} 0 & \text{for} \quad 0 < a \le \dfrac{\alpha}{b+d} \\ \dfrac{ba-\alpha}{da}+1 & \text{for} \quad \dfrac{\alpha}{b+d} \le a \le \dfrac{\alpha}{b} \\ \dfrac{-ba+\alpha}{da}+1 & \text{for} \quad \dfrac{\alpha}{b} \le a \le \dfrac{\alpha}{b-d} \\ 0 & \text{for} \quad \dfrac{\alpha}{b-d} \le a < \infty. \end{cases}$$

From the definition of a mean value we obtain

$$\mathrm{M}(\overline{a}) = \frac{\alpha d(d^2 + 2b^2)}{2b^2 \ln \dfrac{b^2}{b^2 - d^2}}. \tag{8.56}$$

Finally, the deterministic controller is described by

$$K_{\mathrm{R},d}(s) = \frac{\mathrm{M}(\overline{a})}{s}. \qquad \square$$

To apply the first approach described in the previous section, it is necessary to find the certainty distribution for $Q$ using the function $Q = \Phi(c,a)$ determined in Example 5.1, and the distribution $h_c(c)$, then to determine $\mathrm{M}(\overline{Q};a)$ and to find the value $a^*$ minimizing $\mathrm{M}(\overline{Q};a)$. It may be shown that $a^* \neq \mathrm{M}(\overline{a})$ given by the formula (8.56).

**Example 8.6.** Let us consider the time-optimal control of the plant with $K_{\mathrm{O}}(s;c) = cs^{-2}$ (Fig. 8.9), subject to constraint $|u(t)| \le M$. It is well known that the optimal control algorithm $u = \Psi(x,c)$ is the following:

$$u(t) = M \operatorname{sgn}(\varepsilon + \frac{|\dot{\varepsilon}|\dot{\varepsilon}}{2cM})$$

where $x=[\varepsilon,\dot{\varepsilon}]$. For the given $h_c(c)$ we can determine $h_u(u;\varepsilon,\dot{\varepsilon})$, which is reduced to three values $v_1=v(\overline{u}\cong M)$, $v_2=v(\overline{u}\cong -M)$, $v_3=v(\overline{u}\cong 0)$. Then

$$u_d(t)=\mathrm{M}(\overline{u})=M(v_1-v_2)(v_1+v_2+v_3)^{-1}.$$

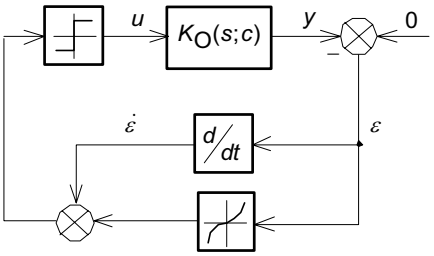


**Fig. 8.9.** Example of control system

It is easy to see that

$$v_1=\max_{c\in D_{c1}} h_c(c), \qquad v_2=\max_{c\in D_{c2}} h_c(c)$$

where

$$D_{c1}=\{c: c\,\mathrm{sgn}\,\varepsilon > -|\dot{\varepsilon}|\dot{\varepsilon}(2M|\varepsilon|)^{-1}\},$$

$$D_{c2}=\{c: c\,\mathrm{sgn}\,\varepsilon < -|\dot{\varepsilon}|\dot{\varepsilon}(2M|\varepsilon|)^{-1}\}$$

and

$$v_3=h_c\left(\frac{-|\dot{\varepsilon}|\dot{\varepsilon}}{2M\varepsilon}\right).$$

Assume that the certainty distribution of $\overline{c}$ is the same as in Example 8.5. For $\varepsilon>0$, $\dot{\varepsilon}<0$ and $c_g<b$ it is easy to obtain the following control algorithm

$$u_d=\mathrm{M}(\overline{u})=\begin{cases} M & \text{for} \quad d\le b-c_g \\ M\dfrac{b-c_g}{3d-2(b-c_g)} & \text{for} \quad d\ge b-c_g \end{cases}$$

where $c_g=(\dot{\varepsilon})^2(2M\varepsilon)^{-1}$. For example, for $M=0.5$, $\dot{\varepsilon}=-3$, $\varepsilon=1$, $b=16$ and $d=10$ we obtain $u_d=0.2$. □

# 9 Fuzzy Variables, Analogies and Soft Variables

This chapter concerns the second part of non-probabilistic descriptions of the uncertainty. The first part of the chapter presents the application of fuzzy variables to non-parametric problems for a static plant, analogous to those described for random and uncertain variables. In Sect. 9.1, a very short description of fuzzy variables (see e.g. [52, 75, 81, 82, 95]) is given in the form needed to formulate our problems and to indicate analogies for non-parametric problems based on random, uncertain and fuzzy variables. These analogies lead to a generalization in the form of soft variables and their applications to non-parametric decision problems. The considerations are completed with a presentation of a fuzzy controller in a closed-loop control system and with some remarks concerning so called descriptive and prescriptive approaches.

## 9.1 Fuzzy Sets and Fuzzy Numbers

Let us consider a universal set $X$ and a property (a predicate) $\varphi(x)$ defined on a set $X$, i.e. a property concerning the variable $x \in X$. If $\varphi(x)$ is a crisp property, then for the fixed value $x$ the logic value $w[\varphi(x)] \in \{0, 1\}$ and the property $\varphi(x)$ defines a set

$$D_x = \{x \in X : w[\varphi(x)] = 1\} \stackrel{\Delta}{=} \{x \in X : \varphi(x)\}$$

(see Sect. 8.1). If $\varphi(x)$ is a soft property then, for the fixed $x$, $\varphi(x)$ forms a proposition in multi-valued logic and $w[\varphi(x)] \in [0, 1]$. The logic value $w[\varphi(x)]$ denotes the *degree of truth*, i.e. for the fixed $x$ the value $w[\varphi(x)]$ shows to what degree the property $\varphi(x)$ is satisfied. The determination of the value $w[\varphi(x)]$ for every $x \in X$ leads to the determination of a function

$$\mu : X \to [0,1], \quad \text{i.e. } w[\varphi(x)] \stackrel{\Delta}{=} \mu(x).$$

In two-valued logic

$$\mu(x) \stackrel{\Delta}{=} I(x) \in \{0,1\}$$

and the set $D_x$ is defined by the pair $X$, $I(x)$:

$$D_x = < X, I(x) > = \{x \in X : I(x) = 1\}. \tag{9.1}$$

The function $\mu(x)$ is called a *membership function* and the pair $< X, \mu(x) >$ is called a *fuzzy set*. This is a generalization of the function $I(x)$ and the set (9.1), respectively. To every element, the membership function assigns the value $\mu(x)$ from the set $[0, 1]$. In practical interpretations it is necessary to determine the property $\varphi(x)$ for which the membership function is given. We assume that the membership function is given by an expert and describes his/her subjective opinions concerning the degree of truth (degree of satisfaction) of the property $\varphi(x)$ for the different elements $x \in X$. For example, let $X$ denote a set of women living in some region. Consider two properties (predicates):

1. $\varphi(x)$ = "the age of $x$ is less than 30 years".
2. $\varphi(x)$ = "$x$ is beautiful".

The first predicate is a crisp property because for the fixed woman $x$ the sentence $\varphi(x)$ is true or false, i.e. $w[\varphi(x)] \in \{0, 1\}$. The property $\varphi(x)$ determines the set of women (the subset $D_x \subset X$) who are less than 30 years old. The second predicate is a soft property and $w[\varphi(x)] = \mu(x) \in [0, 1]$ may denote a degree of beauty assigned to a woman $x$ by an expert. The property $\varphi(x)$ together with the function $\mu(x)$ determines the set of beautiful women. This is a fuzzy set, and for every $x$ the function $\mu(x)$ determines a degree of membership to this set. In the first case (for the crisp property $\varphi(x)$) the expert, not knowing the age of the woman $x$, may give his/her estimate $\overline{\mu}(x) \in [0,1]$ of the property $\varphi(x)$. Such an estimate is not a membership function of the property $\varphi(x)$ but a value of a certainty index characterizing the expert's uncertainty. Such a difference is important for the proper understanding of fuzzy numbers and their comparison with uncertain variables, presented in Sect. 9.3. We may say that the estimate $\overline{\mu}(x)$ is a membership function of the property "it seems to me that $x$ is less than 30 years old", formulated by the expert.

Let us consider another example: the points $x$ on a plane are red to different degrees: from definitely red via different degrees of pink to definitely white. The value $\mu(x)$ assigned to the point $x$ denotes the degree of

red colour of this point. If the definitely red points are concentrated in some domain and the further from this domain they are less red (more white), then the function $\mu(x)$ (the surface upon the plane) reaches its maximum value equal to 1 in this domain and decreases to 0 for the points far from this domain.

According to (8.1), for the determined $X$ and any two functions $\mu_1(x), \mu_2(x)$ (i.e. any two fuzzy sets)

$$\mu_1(x) \vee \mu_2(x) = \max\{\mu_1(x), \mu_2(x)\}, \tag{9.2}$$

$$\mu_1(x) \wedge \mu_2(x) = \min\{\mu_1(x), \mu_2(x)\}, \tag{9.3}$$

$$\neg\mu_1(x) = 1 - \mu_1(x). \tag{9.4}$$

These are definitions of the basic operations in the algebra of fuzzy sets $< X, \mu(x) >$. The relation

$$\mu_1(x) \le \mu_2(x)$$

denotes the inclusion for fuzzy sets, which is a generalization of the inclusion $I_1(x) \le I_2(x)$, i.e. $D_{x1} \subseteq D_{x2}$. It is worth noting that except (8.1) one considers other definitions of the operations $\vee$ and $\wedge$ in the set [0, 1], and consequently – other definitions of the operations (9.2) and (9.3).

If $X$ is a subset of $R^1$ (the set of real numbers) then the pair $< X, \mu(x) > \stackrel{\Delta}{=} \hat{x}$ is called a *fuzzy number*. In further considerations $\hat{x}$ will be called a *fuzzy variable* to indicate the analogy with random and uncertain variables, and the equation $\hat{x} = x$ will denote that the variable $\hat{x}$ takes a value $x$. The function $\mu(x)$ is now the membership function of a soft property $\varphi(x)$ concerning a number. The possibilities of the formulation of such properties are rather limited. They may be the formulations concerning the size of the number, e.g. for positive numbers, "$x$ is small", "$x$ is very large" etc. and for real numbers, "$x$ is small positive", "$x$ is large negative" etc. Generally, for the property "$\hat{x}$ is $d$", the value $\mu(x)$ denotes to what degree this property is satisfied for the value $\hat{x} = x$. For the interpretation of the fuzzy number described by $\mu(x)$ it is necessary to determine the property $\varphi(x)$ for which $\mu(x)$ is given. One assumes that

$$\max_{x \in X} \mu(x) = 1.$$

Usually one considers two cases: the discrete case with

$X = \{x_1, x_2, \ldots, x_m\}$ and the continuous case in which $\mu(x)$ is a continuous function. In the case of fuzzy variables the determinization is called a *defuzzification*. In a way similar to that for random and uncertain numbers, it may consist in replacing the uncertain variable $\hat{x}$ by its deterministic representation

$$x^* = \arg\max_{x \in X} \mu(x)$$

on the assumption that $x^*$ is a unique point such that $\mu(x^*) = 1$, or by the mean value $\mathrm{M}(\hat{x})$. In the discrete case

$$\mathrm{M}(\hat{x}) = \frac{\sum_{i=1}^{m} x_i \mu(x_i)}{\sum_{i=1}^{m} \mu(x_i)} \tag{9.5}$$

and in the continuous case

$$\mathrm{M}(\hat{x}) = \frac{\int_{-\infty}^{\infty} x\mu(x)dx}{\int_{-\infty}^{\infty} \mu(x)dx} \tag{9.6}$$

on the assumption that the respective integrals exist.

Let us consider two fuzzy numbers defined by sets of values $X \subseteq R^1$, $Y \subseteq R^1$ and membership functions $\mu_x(x)$, $\mu_y(y)$, respectively. The membership function $\mu_x(x)$ is the logic value of the soft property $\varphi_x(x) =$ "if $\hat{x} = x$ then $\hat{x}$ is $d_1$" or shortly "$\hat{x}$ is $d_1$", and $\mu_y(y)$ is the logic value of the soft property $\varphi_y(y) =$ "$\hat{y}$ is $d_2$", i.e.

$$w[\varphi_x(x)] = \mu_x(x), \qquad w[\varphi_y(y)] = \mu_y(y)$$

where $d_1$ and $d_2$ denote the size of the number, e.g. $\varphi_x(x) =$ "$\hat{x}$ is small", $\varphi_y(y) =$ "$\hat{y}$ is large". Using the properties $\varphi_x$ and $\varphi_y$ we can introduce the property $\varphi_x \to \varphi_y$ (e.g. "if $\hat{x}$ is small, then $\hat{y}$ is large") with the respective membership function

$$w[\varphi_x \to \varphi_y] \stackrel{\Delta}{=} \mu_y(y \mid x),$$

and the properties

$$\varphi_x \vee \varphi_y \qquad \text{and} \qquad \varphi_x \wedge \varphi_y = \varphi_x \wedge [\varphi_x \to \varphi_y]$$

for which the membership functions are defined as follows:

$$w[\varphi_x \vee \varphi_y] = \max\{\mu_x(x), \mu_y(y)\},$$

$$w[\varphi_x \wedge \varphi_y] = \min\{\mu_x(x), \mu_y(y \mid x)\} \stackrel{\Delta}{=} \mu_{xy}(x,y). \tag{9.7}$$

If we assume that

$$\varphi_x \wedge [\varphi_x \to \varphi_y] = \varphi_y \wedge [\varphi_y \to \varphi_x]$$

then

$$\mu_{xy}(x,y) = \min\{\mu_x(x), \mu_y(y \mid x)\} = \min\{\mu_y(y), \mu_x(x \mid y)\}. \tag{9.8}$$

The properties $\varphi_x$, $\varphi_y$ and the corresponding fuzzy numbers $\hat{x}$, $\hat{y}$ are called *independent* if

$$w[\varphi_x \wedge \varphi_y] = \mu_{xy}(x,y) = \min\{\mu_x(x), \mu_y(y)\}.$$

Using (9.8) it is easy to show that

$$\mu_x(x) = \max_{y \in Y} \mu_{xy}(x,y), \tag{9.9}$$

$$\mu_y(y) = \max_{x \in X} \mu_{xy}(x,y). \tag{9.10}$$

The equations (9.8), (9.9) and (9.10) describe the relationships between $\mu_x$, $\mu_y$, $\mu_{xy}$, $\mu_x(x \mid y)$ as being analogous to the relationships (8.18), (8.16), (8.17) for uncertain variables, in general defined in the multi-dimensional sets $X$ and $Y$. For the given $\mu_{xy}(x,y)$, the set of functions $\mu_y(y \mid x)$ is determined by equation (9.7) in which

$$\mu_x(x) = \max_{y \in Y} \mu_{xy}(x,y).$$

**Theorem 9.1.** The set of functions $\mu_y(y \mid x)$ satisfying equation (9.7) is determined as follows:

$$\mu_y(y \mid x) \begin{cases} = \mu_{xy}(x,y) & \text{for} \quad (x,y) \notin D(x,y) \\ \geq \mu_{xy}(x,y) & \text{for} \quad (x,y) \in D(x,y) \end{cases} \tag{9.11}$$

where

$$D(x,y) = \{(x,y) \in X \times Y : \ \mu_x(x) = \mu_{xy}(x,y)\}.$$

*Proof:* From (9.7) it follows that

$$\bigwedge_{x\in X}\bigwedge_{y\in Y}[\mu_x(x)\geq\mu_{xy}(x,y)].$$

If $\mu_x(x)>\mu_{xy}(x,y)$ then, according to (9.7), $\mu_{xy}(x,y)=\mu_y(y\,|\,x)$. If $\mu_x(x)=\mu_{xy}(x,y)$, i.e. $(x,y)\in D(x,y)$ then $\mu_y(y\,|\,x)\geq\mu_{xy}(x,y)$. □

In particular, as one of the solutions of equation (9.7), i.e. one of the possible definitions of the membership function for an implication we may accept

$$\mu_y(y\,|\,x)=\mu_{xy}(x,y). \tag{9.12}$$

If $\mu_{xy}(x,y)=\min\{\mu_x(x),\mu_y(y)\}$ then according to (9.12)

$$\mu_y(y\,|\,x)=\min\{\mu_x(x),\mu_y(y)\}$$

and according to (9.7)

$$\mu_y(y\,|\,x)=\mu_y(y).$$

Except $\varphi_x(x)\to\varphi_y(y)$ (i.e. the property $\varphi_y(y)$ under the condition $\varphi_x$), we can consider the property $\varphi_y(y)$ for the given value $\hat{x}=x$ (i.e. the property $\varphi_y(y)$ under the condition $\hat{x}=x$):

$$\text{“}\hat{x}=x\to\varphi_y(y)\text{”}\stackrel{\Delta}{=}\text{“}\varphi_y(y)\,|\,x\text{”},$$

and the membership function of this property

$$\begin{aligned}w[\varphi_y(y)\,|\,x]&=w[\hat{x}=x\to\varphi_y(y)]=w\{[\hat{x}=x\wedge\varphi_x(x)]\wedge[\varphi_x(x)\to\varphi_y(y)]\}\\&=\min\{\mu_x(x)\wedge\mu_y(y\,|\,x)\}=\mu_{xy}(x,y).\end{aligned}$$

Then $\mu_{xy}(x,y)$ may be interpreted as a conditional membership function of the property $\varphi_y(y)$ for the given $x$, determined with the help of the property $\varphi_x(x)$. Such an interpretation is widely used in the description of fuzzy controllers in closed-loop systems.

It is worth noting that, according to (9.11), we may use the different functions $\mu_y(y\,|\,x)$ for the given $\mu_{xy}(x,y)$ and, consequently, for the fixed

$$\mu_x(x)=\max_{y\in Y}\mu_{xy}(x,y)$$

and

$$\mu_y(y) = \max_{x \in X} \mu_{xy}(x, y) .$$

In other words, the membership function of the implication

$$w[\varphi_x(x) \to \varphi_y(y)] = \mu_y(y \mid x)$$

may be defined in different ways.

For the fixed $x$, the set

$$D_y(x) = \{y \in Y : (x, y) \in D(x, y)\}$$

is a set of values $x$ maximizing $\mu_{xy}(x, y)$. If $D_y(x) = \{y^*(x)\}$ (a singleton), then

$$y^*(x) = \arg\max_{y \in Y} \mu_{xy}(x, y)$$

and $\mu_y(y^* \mid x) = 1$, i.e.

$$\mu_y(y \mid x) = \begin{cases} \mu_{xy}(x, y) & \text{for} \quad y \neq y^*(x) \\ 1 & \text{for} \quad y = y^*(x). \end{cases} \tag{9.13}$$

It is easy to note that $\mu_y(y \mid x)$ determined by (9.13) is a continuous function for every $x \in X$ if and only if

$$\bigwedge_{x \in D_x} [\mu_x(x) = 1] ,$$

i.e.

$$\bigwedge_{x \in D_x} [\max_{y \in Y} \mu_{xy}(x, y) = 1] \tag{9.14}$$

where

$$D_x = \{x \in X : \bigvee_{y \in Y} \mu_{xy}(x, y) \neq 0\} .$$

If the condition (9.14) is satisfied then $\mu_y(y \mid x) = \mu_{xy}(x, y)$.

## 9.2 Application of Fuzzy Description to Decision Making (Control) for Static Plant

### 9.2.1 Plant without Disturbances

The description concerning the pair of fuzzy variables may be directly applied to a one-dimensional static plant with single input $u \in U$ and single output $y \in Y$ $(U, Y \subseteq R^1)$. The non-parametric description of uncertainty using fuzzy variables may be formulated by introducing two soft properties $\varphi_u(u)$ and $\varphi_y(y)$. This description (the knowledge of the plant KP) is given by an expert in the form of the membership function

$$w[\varphi_u \to \varphi_y] = \mu_y(y \mid u).$$

For example, the expert says that "if $\hat{u}$ is large then $\hat{y}$ is small" and gives the membership function $\mu_y(y \mid u)$ for this property. In this case the analysis problem may consist in the determination of the membership function $\mu_y(y)$ characterizing the *output property* $\varphi_y$ for the given membership function $\mu_u(u)$ characterizing the *input property*. The decision problem may be stated as an inverse problem, consisting in finding $\varphi_u(u)$ for a desirable membership function $\mu_y(y)$ given by a user. From a formal point of view, the formulations of these problems and the respective formulas are similar to those for random variables (see Sect. 7.2) and for uncertain variables (see Sect. 8.2.2).

The essential difference is the following:
The descriptions in the form of $f_u(u)$ or $h_u(u)$, and in the form of $f_y(y)$ or $h_y(y)$, are concerned directly with *values of the input and output*, respectively, and the descriptions in the form of $\mu_u(u)$ and $\mu_y(y)$ are concerned with determined *input and output properties*, respectively. In particular, in the decision problem the functions $f_y(y)$ or $h_y(y)$ describe the user's requirement characterizing directly the *value of the output*, and the function $\mu_y(y)$ required by the user characterizes the determined *output property* $\varphi_y(y)$. Consequently, the solution $\mu_u(u)$ concerns the determined *input property* $\varphi_u(u)$, and not directly the input value $u$ as in the case of $f_u(u)$ or $h_u(u)$.

**Analysis problem:** For the determined properties $\varphi_u(u)$, $\varphi_y(y)$, the given $\mathrm{KP} = <\mu_y(y \mid u)>$ and $\mu_u(u)$ find the membership function $\mu_y(y)$.

According to (9.10) and (9.7) with $u$ in place of $x$

$$\mu_y(y) = \max_{u \in U} \min\{\mu_u(u), \mu_y(y \mid u)\}. \tag{9.15}$$

We can also formulate the **analysis problem for the given input**: Find

$$\mu_{uy}(u, y) = w[\varphi_y(y) \mid u] = \min\{\mu_u(u), \mu_y(y \mid u)\}.$$

Having $\mu_{uy}(u, y)$, one can determine the value of $y$ maximizing $\mu_{uy}(u, y)$ or the conditional mean value for the given $u$:

$$\mathrm{M}(\hat{y} \mid u) = \frac{\int_{-\infty}^{+\infty} y \mu_{uy}(u, y) dy}{\int_{-\infty}^{+\infty} \mu_{uy}(u, y) dy}.$$

**Decision problem:** For the determined properties $\varphi_u(u)$, $\varphi_y(y)$, the given $\mathrm{KP} = <\mu_y(y \mid u)>$ and $\mu_y(y)$ find the membership function $\varphi_u(u)$.

To find the solution one should solve equation (9.15) with respect to the function $\mu_u(u)$ satisfying the conditions for a membership function:

$$\bigwedge_{u \in U} \mu_u(u) \geq 0, \qquad \max_{u \in U} \mu_u(u) = 1.$$

The membership function $\mu_u(u)$ may be called a *fuzzy decision*. The deterministic decision may be obtained via a determinization which consists in finding the value $u_a$ maximizing the membership function $\mu_u(u)$ or the mean value $u_b = \mathrm{M}(\hat{u})$.

Assume that the function

$$\mu_{uy}(u, y) = \min\{\mu_u(u), \mu_y(y \mid u)\}$$

for the given $y$ takes its maximum value at one point

$$\hat{u}(y) = \arg\max_{u \in U} \min\{\mu_u(u), \mu_y(y \mid u)\}.$$

**Theorem 9.2.** For the continuous case (i.e. continuous membership functions), assume that:
1. The function $\mu_u(u)$ has one local maximum for

$$u^* = \arg\max_{u \in U} \mu_u(u)$$

and it is a unique point such that $\mu_u(u^*) = 1$.
2. For every $y \in Y$ the membership function $\mu_y(y \mid u)$ as a function of $u$ has at most one local maximum equal to 1, i.e. the equation

$$\mu_y(y \mid u) = 1$$

has at most one solution

$$\tilde{u}(y) = \arg\max_{u \in U} \mu_y(y \mid u).$$

Then

$$\hat{u}(y) = \arg \max_{u \in D_u(y)} \mu_y(y \mid u)$$

where $D_u(y)$ is a set of values $u$ satisfying the equation

$$\mu_u(u) = \mu_y(y \mid u). \qquad \square$$

The proof of Theorem 9.2 may be found in [52]. The procedure for the determination of $\mu_u(u)$ for the fixed $u$ is the following:
1. To solve the equation

$$\mu_u(u) = \mu_y(y \mid u)$$

with respect to $y$ and to obtain a solution $y(u)$ (in general, a set of solutions $D_y(u)$).
2. To determine

$$\overline{\mu}_u(u) = \mu_y[y(u)] = \mu_y[y(u) \mid u]. \tag{9.16}$$

3. To prove whether

$$\mu_y(y) = \max_{u \in \tilde{D}_u(y)} \mu_y(y \mid u) \tag{9.17}$$

where $\tilde{D}_u(y)$ is a set of values $u$ satisfying the equation

$$\overline{\mu}_u(u) = \mu_y(y \mid u).$$

4. To accept the solution $\overline{\mu}_u(u) = \mu_u(u)$ for which (9.17) is satisfied.

**Remark 9.1.** The same considerations concerning non-parametric analysis

and decision problems may be presented for the description based on uncertain variables with the distributions $h$ instead of the membership functions $\mu$. The same remark concerns the plant with the disturbances, described in the next section. □

**Example 9.1.** Consider a plant with $u, y \in R^1$, described by the membership function

$$\mu_y(y \mid u) = \begin{cases} -4(y-u)^2 + 1 & \text{for} \quad u - \frac{1}{2} \le y \le u + \frac{1}{2} \\ 0 & \text{otherwise}. \end{cases}$$

For the membership function required by a user (Fig. 9.1)

$$\mu_y(y) = \begin{cases} -(y-c)^2 + 2 & \text{for} \quad c - \sqrt{2} \le y \le c - 1 \\ & \text{or} \quad c + 1 \le y \le c + \sqrt{2} \\ 1 & \text{for} \quad c - 1 \le y \le c + 1 \\ 0 & \text{otherwise}, \end{cases} \tag{9.18}$$

one should determine the fuzzy decision in the form of the membership function $\mu_u(u)$.

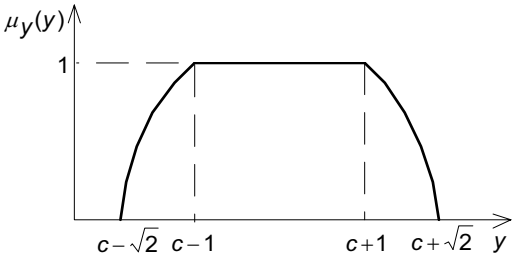


**Fig. 9.1.** Example of the membership function

The solution of the equation

$$\mu_y(y) = \mu_y(y \mid u) \tag{9.19}$$

has the following form:
1. For

$$c - 1 \le u \le c + 1$$

equation (9.19) has one solution:

$$y(u) = u\,.$$

2. For

$$c - \sqrt{2} - \frac{1}{2} < u < c - 1 \qquad \text{or} \qquad c + 1 < u < c + \sqrt{2} + \frac{1}{2}$$

equation (9.19) is reduced to the equation

$$4(y-u)^2 - (y-c)^2 + 1 = 0$$

which has one solution such that $h_y(y) > 0$:

$$y(u) = \begin{cases} \dfrac{4u - c + \sqrt{\Delta}}{3} & \text{for} \quad c - \sqrt{2} - \dfrac{1}{2} < u < c - 1 \\ \dfrac{4u - c - \sqrt{\Delta}}{3} & \text{for} \quad c + 1 < u < c + \sqrt{2} + \dfrac{1}{2} \end{cases}$$

where

$$\Delta = 4(u-c)^2 + 3\,.$$

3. Otherwise, equation (9.19) has no solution such that $h_y(y) > 0$.

Then, according to (9.16) and (9.18)

$$\overline{\mu}_u(u) = \mu_y[y(u)] = \begin{cases} (\dfrac{4u - 4c + \sqrt{\Delta}}{3})^2 + 2 & \text{for} \quad c - \sqrt{2} - \dfrac{1}{2} \le u \le c - 1 \\ 1 & \text{for} \quad c - 1 \le u \le c + 1 \\ (\dfrac{4u - 4c - \sqrt{\Delta}}{3})^2 + 2 & \text{for} \quad c + 1 \le u \le c + \sqrt{2} + \dfrac{1}{2} \\ 0 & \text{otherwise.} \end{cases}$$

□

**Remark 9.2.** The properties $\varphi_u(u)$ and $\varphi_y(y)$ considered in the example may be introduced by using additional descriptions. For example, if

$$\mu_y(y \mid u) = \begin{cases} -4(y-u)^2 + 1 & \text{for} \quad u - \dfrac{1}{2} \le y \le u + \dfrac{1}{2} \\ & \text{and} \quad u > \dfrac{1}{2} \\ 0 & \text{otherwise} \end{cases}$$

and $c > \sqrt{2} + \frac{1}{2}$, then we can say that

$$\varphi_u(u) = \text{“}u \text{ is medium positive”},$$

$$\varphi_y(y) = \text{“}y \text{ is medium positive”}$$

and $\mu_y(y \mid u)$ is a membership function of the property:

$\varphi_u(u) \to \varphi_y(y)$ = “If $u$ is medium positive then $y$ is medium positive”.

If we introduce a new variable $\overline{u} = u - c$ with the respective constraint then

$\varphi_u(u) \to \varphi_y(y)$ = “if $|\overline{u}|$ is small then $y$ is medium positive”. □

### 9.2.2 Plant with External Disturbances

Consider a static plant with single input $u \in U$, single output $y \in Y$ and single disturbance $z \in Z$ $(U, Y, Z \subseteq R^1)$. Now the non-parametric description of uncertainty using fuzzy variables may be formulated by introducing three soft properties: $\varphi_u(u), \varphi_z(z)$ and $\varphi_y(y)$. This description is given by an expert in the form of the membership function

$$w[\varphi_u \wedge \varphi_z \to \varphi_y] = \mu_y(y \mid u, z),$$

i.e. the knowledge of the plant

$$\text{KP} = < \mu_y(y \mid u, z) > .$$

For example, the expert says that “if $\hat{u}$ is large and $\hat{z}$ is medium then $\hat{y}$ is small” and gives the membership function $\mu_y(y \mid u, z)$ for this property. For such a plant the analysis and decision problems may be formulated as extensions of the problems described in the previous section.

**Analysis problem:** For the given $\text{KP} = < \mu_y(y \mid u, z) >$, $\mu_u(u \mid z)$ and $\mu_z(z)$ find the membership function $\mu_y(y)$.

According to (9.10) and (9.7)

$$u_y(y) = \max_{u \in U, z \in Z} \mu_y(y, u, z) \tag{9.20}$$

where

$$\mu_y(y,u,z) = w[\varphi_u \wedge \varphi_z \wedge \varphi_y]$$

i.e.

$$\mu_y(y,u,z) = \min\{\mu_{uz}(u,z), \mu_y(y\,|\,u,z)\}\,. \tag{9.21}$$

Putting

$$\mu_{uz}(u,z) = \min\{\mu_z(z), \mu_u(u\,|\,z)\} \tag{9.22}$$

and (9.21) into (9.20) yields

$$\mu_y(y) = \arg \max_{u\in U, z\in Z} \min\{\mu_z(z), \mu_u(u\,|\,z), \mu_y(y\,|\,u,z)\}\,. \tag{9.23}$$

**Decision problem:** For the given $\mathrm{KP} = <\mu_y(y\,|\,u,z)>$ and $\mu_y(y)$ required by a user one should determine $\mu_u(u\,|\,z)$.

The determination of $\mu_u(u\,|\,z)$ may be decomposed into two steps. In the first step, one should find the function $\mu_{uz}(u,z)$ satisfying the equation

$$\mu_y(y) = \max_{u\,\in\,U, z\,\in\,Z} \min\{\mu_{uz}(u,z), \mu_y(y\,|\,u,z)\} \tag{9.24}$$

and the conditions for a membership function:

$$\bigwedge_{u\in U}\ \bigwedge_{z\in Z} \mu_{uz}(u,z) \ge 0\,, \qquad \max_{u\,\in\,U, z\,\in\,Z} \mu_{uz}(u,z) = 1\,.$$

In the second step, one should determine the function $\mu_u(u\,|\,z)$ satisfying the equation

$$\mu_{uz}(u,z) = \min\{\mu_z(z), \mu_u(u\,|\,z)\} \tag{9.25}$$

where

$$\mu_z(z) = \max_{u\,\in\,U} \mu_{uz}(u,z)\,,$$

and the conditions for a membership function:

$$\bigwedge_{u\in U}\ \bigwedge_{z\in Z} \mu_u(u\,|\,z) \ge 0\,, \qquad \bigwedge_{z\in Z} \max_{u\in U} \mu_u(u\,|\,z) = 1\,.$$

The solution may not be unique. The function $\mu_u(u\,|\,z)$ may be considered as a knowledge of the decision making $\mathrm{KD} = <\mu_u(u\,|\,z)>$ or a *fuzzy decision algorithm* (the description of a *fuzzy controller* in the open-loop control system). It is important to remember that the description of the fuzzy

controller is concerned with the determined *input and output properties,* i.e.

$$\mu_u(u|z) = w\,[\varphi_z(z) \to \varphi_u(u)]$$

where the properties $\varphi_z(z)$ and $\varphi_u(u)$ have been used in the description of the plants. Having $\mu_u(u|z)$, one can obtain the deterministic decision algorithm $\Psi(z)$ as a result of the determinization (defuzzification) of the fuzzy decision algorithm $\mu_u(u|z)$. Two versions corresponding to versions I and II in Sect. 8.2.2 are the following:

*Version I.*

$$u_a = \arg\max_{u \in U} \mu_u(u|z) \stackrel{\Delta}{=} \Psi_a(z).$$

*Version II.*

$$u_b = \mathrm{M}_u(\hat{u}|z) = \int_U u\mu_u(u|z)du \;[\int_U \mu_u(u|z)du\;]^{-1} \stackrel{\Delta}{=} \Psi_b(z). \qquad (9.26)$$

Using $\mu_u(u|z)$ or $\mu_{uz}(u,z)$ with the fixed $z$ in the determination of $u_a$ or $u_b$, one obtains two versions of $\Psi(z)$. In the second version the fuzzy controller has the form $\mathrm{KD} = < w[\varphi_u(u)|z] > = < \mu_{uz}(u,z) >$ and the second step with equation (9.25) is not necessary. Both versions are the same if we assume that $\mu_u(u|z) = \mu_{uz}(u,z)$. Let us note that in the analogous problems for random variables (Sect. 7.2) and for uncertain variables (Sect. 8.2.2) it is not possible to introduce two versions of KD considered here for fuzzy numbers. It is caused by the fact that $\mu_{uz}(u,z)$ and $\mu_u(u|z)$ do not concern directly the *values* of the variables (as probability distributions or certainty distributions) but are concerned with the *properties* $\varphi_u$, $\varphi_z$ and

$$\mu_{uz}(u,z) = w[\varphi_u \wedge \varphi_z], \quad \mu_u(u|z) = w[\varphi_z \to \varphi_u] = w[\varphi_u \,|\, \varphi_z].$$

The deterministic decision algorithms $\Psi_a(z)$ or $\Psi_b(z)$ are based on the knowledge of the decision making $\mathrm{KD} = < \mu_u(u|z) >$, which is determined from the knowledge of the plant KP for the given $\mu_y(y)$. It is worth noting that the deterministic decision algorithms $\Psi_a(z)$ or $\Psi_b(z)$ have no clear practical interpretation.

From a formal point of view the considerations in this section are the same as in Sect. 8.2.2 for uncertain variables. Then we can repeat here Theorem 8.4 and the next considerations including Remark 8.2 with

$\mu_u(u)$, $\mu_y(y)$, $\mu_z(z)$, $\mu_{uz}(u,z)$, $\mu_u(u|z)$, $\mu_y(y|u,z)$ in place of $h_u(u)$, $h_y(y)$, $h_z(z)$, $h_{uz}(u,z)$, $h_u(u|z)$, $h_y(y|u,z)$, respectively.

Let us note that the condition

$$\bigwedge_{z \in D_z} [\mu_z(z) = 1]$$

corresponding to the condition (8.38) means that $\varphi_z(z)$ is reduced to a crisp property "$z \in D_z$".

The considerations may be extended to the multi-dimensional case with vectors $u, y, z$. To formulate the knowledge of the plant one introduces soft properties of the following form: $\varphi_{ui}(j)$ = "$u^{(i)}$ is $d_j$", $\varphi_{zi}(j)$ = "$z^{(i)}$ is $d_j$", $\varphi_{yi}(j)$ = "$y^{(i)}$ is $d_j$" where $u^{(i)}, z^{(i)}, y^{(i)}$ denote the $i$-th components of $u, z, y$, respectively. The determinization of the fuzzy algorithm may be made according to versions I and II presented for the one-dimensional case. In particular, version II consists in the determination of $\mathrm{M}(\hat{u}^{(i)})$ for the fixed $z$ and each component of the vector $u$, using the membership functions $\mu_{ui}(u^{(i)}, z)$ or $\mu_{ui}(u^{(i)}|z)$ where

$$\mu_{ui}(u^{(i)}, z) = \max\{\mu_{ui}(1,z), \mu_{ui}(2,z), ..., \mu_{ui}(m,z)\}$$

and $\mu_{ui}(j,z)$ corresponds to $\varphi_{ui}(j)$ = "$u^{(i)}$ is $d_j$".

**Example 9.2.** Consider a plant with $u, y, z \in R^1$ described by the following KP:
"If $u$ is small non-negative and $z$ is large but not greater than $b$ (i.e. $b - z$ is small non-negative) then $y$ is medium". Then
$\varphi_u(u)$ = "$u$ is small non-negative",
$\varphi_z(z)$ = "$z$ is large, not greater than $b$",
$\varphi_y(y)$ = "$y$ is medium".

The membership function $w[\varphi_u \wedge \varphi_z \to \varphi_y]$ is as follows:

$$\mu_y(y|u,z) = -(y-d)^2 + 1 - u - (b - z)$$

for

$$0 \le u \le \frac{1}{2}, \qquad b - \frac{1}{2} \le z \le b,$$

$$-\sqrt{1-u-(b-z)}+d \le y \le \sqrt{1-u-(b-z)}+d$$

and $\mu_y(y \mid u,z)=0$ otherwise.

For the membership function required by a user

$$\mu_y(y)=\begin{cases}-(y-c)^2+1 & \text{for } c-1 \le y \le c+1\\ 0 & \text{otherwise,}\end{cases}$$

one should determine the fuzzy decision algorithm in the form $\mu_u(u \mid z)=\mu_{uz}(u,z)$.

Let us assume that $b>0, c>1$ and

$$c+1 \le d \le c+2 .$$

Then the equation $\mu_y(y)=\mu_y(y \mid u,z)$ has a unique solution which is reduced to the solution of the equation

$$-(y-c)^2+1=-(y-d)^2+1-u-(b-z) .$$

Further considerations are the same as in Example 8.3, which is identical from the formal point of view. Consequently, we obtain the following result:

$$\mu_{uz}(u,z)=\mu_u(u \mid z)$$

$$=\begin{cases}-\left[\dfrac{(d-c)^2+u+b-z}{2(d-c)}\right]^2+1 & \text{for } u \le 1-[(d-c)-1]^2-(b-z)\\ & 0 \le u \le \dfrac{1}{2},\ b-\dfrac{1}{2} \le z \le b\\ 0 & \text{otherwise .}\end{cases}$$

By applying the determinization (defuzzification) we can determine the deterministic decision algorithm in an open-loop decision system:

$$u_a=\arg\max_{u \in U}\mu_u(u \mid z) \stackrel{\Delta}{=} \Psi_a(z)$$

or

$$u_b=\mathrm{M}_u(\hat{u} \mid z) \stackrel{\Delta}{=} \Psi_b(z) . \qquad \square$$

**Remark 9.3.** The description $\mu_y(y \mid u,z)$ given by an expert and the solution $\mu_u(u \mid z)=\mu_{uz}(u,z)$ do not satisfy the condition $\max\mu=1$. The normalization in the form analogous to (8.39) is not necessary if we are interested in the deterministic decisions $u_a$ or $u_b$, which are the same for $\mu_u(u \mid z)$ and the normalized form $\overline{\mu}_u(u \mid z)$. $\square$

## 9.3 Comparison of Uncertain Variables with Random and Fuzzy Variables

The formal part of the definitions of a random variable, a fuzzy number and an uncertain variable is the same: $< X, \mu(x) >$, that is a set $X$ and a function $\mu : X \to R^1$ where $0 \le \mu(x)$ for every $x \in X$. For the fuzzy number, the uncertain variable and for the random variable in the discrete case, $\mu(x) \le 1$. For the random variable the property of additivity is required, which in the discrete case $X = \{x_1, x_2, ..., x_m\}$ is reduced to the equality $\mu(x_1) + \mu(x_2) + ... + \mu(x_m) = 1$. Without any additional description, one can say that each variable is defined by a fuzzy set $< X, \mu(x) >$. In fact, each definition contains an additional description of semantics which discriminates the respective variables. To compare the uncertain variables with probabilistic and fuzzy approaches, take into account the definitions for $X \subseteq R^1$, using $\Omega$, $\omega$ and $g(\omega) = \bar{x}(\omega)$ introduced in Sect. 8.1. *The random variable* $\tilde{x}$ is defined by $X$ and probability distribution $\mu(x) = F(x)$ (or probability density $f(x) = F'(x)$ if this exists) where $F(x)$ is the probability that $\tilde{x} \le x$. In the discrete case $\mu(x_i) = p(x_i) = P(\tilde{x} = x_i)$ (probability that $\tilde{x} = x_i$). For example, if $\Omega$ is a set of $100$ persons and $20$ of them have the age $\bar{x}(\omega) = 30$, then the probability that a person chosen randomly from $\Omega$ has $\bar{x} = 30$ is equal to $0.2$. In general, the function $p(x)$ (or $f(x)$ in a continuous case) is an *objective* characteristic of $\Omega$ as a whole and $h_\omega(x)$ is a subjective characteristic given by an expert and describes his or her individual opinion of the fixed particular $\omega$.

To compare uncertain variables with fuzzy numbers, let us recall three basic definitions of the fuzzy number in a wide sense of the word, that is the definitions of the fuzzy set based on the number set $X = R^1$.

1. The fuzzy number $\hat{x}(d)$ for the given fixed value $d \in X$ is defined by $X$ and the membership function $\mu(x, d)$, which may be considered as a logic value (*degree of truth*) of the soft property "if $\hat{x} = x$ then $\hat{x} \cong d$".
2. The linguistic fuzzy variable $\hat{x}$ is defined by $X$ and a set of membership functions $\mu_i(x)$ corresponding to different descriptions of the size of $\hat{x}$ (small, medium, large, etc.). For example, $\mu_1(x)$ is a logic value of the soft property "if $\hat{x} = x$ then $\hat{x}$ is small".

3. The fuzzy number $\hat{x}(\omega)$ (where $\omega \in \Omega$ was introduced at the beginning of Sect. 8.1) is defined by $X$ and the membership function $\mu_\omega(x)$, which is a logic value (*degree of possibility*) of the soft property "it is possible that the value $x$ is assigned to $\omega$".

In the first two definitions the membership function does not depend on $\omega$; in the third case there is a family of membership functions (a family of fuzzy sets) for $\omega \in \Omega$. The difference between $\hat{x}(d)$ or the linguistic fuzzy variable $\hat{x}$ and the uncertain variable $\bar{x}(\omega)$ is quite evident. The variables $\hat{x}(\omega)$ and $\bar{x}(\omega)$ are formally defined in the same way by the fuzzy sets $< X, \mu_\omega(x) >$ and $< X, h_\omega(x) >$, respectively, but the interpretations of $\mu_\omega(x)$ and $h_\omega(x)$ are different. In the case of the uncertain variable there exists a function $\bar{x} = g(\omega)$, the value $\bar{x}$ is determined for the fixed $\omega$ but is unknown to an expert who formulates the degree of certainty that $\bar{x}(\omega) \cong x$ for the different values $x \in X$. In the case of $\hat{x}(\omega)$ the function $g$ may not exist. Instead we have a property of the type "it is possible that $P(\omega, x)$" (or, briefly, "it is possible that the value $x$ is assigned to $\omega$") where $P(\omega, x)$ is such a property concerning $\omega$ and $x$ for which it makes sense to use the words "it is possible". Then $\mu_\omega(x)$ for fixed $\omega$ means the degree of possibility for the different values $x \in X$ given by an expert. The example with persons and age is not adequate for this interpretation. In the popular example of the possibilistic approach $P(\omega, x) =$ "John $(\omega)$ ate $x$ eggs at his breakfast".

From the point of view presented above, $\bar{x}(\omega)$ may be considered as a special case of $\hat{x}(\omega)$ (when the relation $P(\omega, x)$ is reduced to the function $g$), with a specific interpretation of $\mu_\omega(x) = h_\omega(x)$. A further difference is connected with the definitions of $w(\bar{x} \,\tilde{\in}\, D_x)$, $w(\bar{x} \,\tilde{\notin}\, D_x)$, $w(\bar{x} \,\tilde{\in}\, D_1 \vee \bar{x} \,\tilde{\in}\, D_2)$ and $w(\bar{x} \,\tilde{\in}\, D_1 \wedge \bar{x} \,\tilde{\in}\, D_2)$. The function $w(\bar{x} \,\tilde{\in}\, D_x) \triangleq m(D_x)$ may be considered as a *measure* defined for the family of sets $D_x \subseteq X$. Two measures have been defined in the definitions of the uncertain variables: $v(\bar{x} \,\tilde{\in}\, D_x) \triangleq \overline{m}(D_x)$ and $v_c(\bar{x} \,\tilde{\in}\, D_x) \triangleq m_c(D_x)$. Let us recall the following special cases of monotonic non-additive measures (see for example [81]) and their properties for every $D_1$, $D_2$.

1. If $m(D_x)$ is a *belief measure*, then

$$m(D_1 \cup D_2) \geq m(D_1) + m(D_2) - m(D_1 \cap D_2).$$

2. If $m(D_x)$ is a *plausibility measure*, then

$$m(D_1 \cap D_2) \leq m(D_1) + m(D_2) - m(D_1 \cup D_2).$$

3. A *necessity measure* is a belief measure for which

$$m(D_1 \cap D_2) = \min\{m(D_1), m(D_2)\}.$$

4. A *possibility measure* is a plausibility measure for which

$$m(D_1 \cup D_2) = \max\{m(D_1), m(D_2)\}.$$

Taking into account the properties of $\overline{m}$ and $m_c$ presented in Sect. 8.1, it is easy to see that $\overline{m}$ is a possibility measure, that $m_n \stackrel{\Delta}{=} 1 - v(\overline{x} \cong \overline{D}_x)$ is a necessity measure and that $m_c$ is neither a belief nor a plausibility measure.

The interpretation of the membership function $\mu(x)$ as a logic value $w$ of a given soft property $P(x)$, that is $\mu(x) = w[P(x)]$, is especially important and necessary if we consider two fuzzy numbers $(x, y)$ and a relation $R(x, y)$ or a function $y = f(x)$. Consequently, it is necessary if we formulate analysis and decision problems. The formal relationships (see for example [95])

$$\mu_y(y) = \max_x [\mu_x(x) : f(x) = y]$$

for the function and

$$\mu_y(y) = \max_x [\mu_x(x) : (x, y) \in R]$$

for the relation do not determine evidently $P_y(y)$ for the given $P_x(x)$. If $\mu_x(x) = w[P_x(x)]$ where $P_x(x) =$ "if $\hat{x} = x$ then $\hat{x} \cong d$", then we can accept that $\mu_y(y) = w[P_y(y)]$ where $P_y(y) =$ "if $\hat{y} = y$ then $\hat{y} \cong f(\hat{x})$" in the case of the function, but in the case of the relation $P_y(y)$ is not determined. If $P_x(x) =$ "if $\hat{x} = x$ then $\hat{x}$ is small", then $P_y(y)$ may not be evident even in the case of the function, for example $y = \sin x$. For the uncertain variable $\mu_x(x) = h_x(x) = v(\overline{x} \cong x)$ with the definitions (8.2)–(8.5), the property $P_y(y)$ such that $\mu_y(y) = v[P_y(y)]$ is deter-

mined precisely: in the case of the function, $\mu_y(y) = h_y(y) = v(\bar{y} \cong y)$ and, in the case of the relation, $\mu_y(y)$ is the certainty index of the property $P_y(y) =$ "there exist $\bar{x}$ such that $(\bar{x}, \bar{y}) \cong R(x, y)$ ".

Consequently, using uncertain variables it is possible not only to formulate the analysis and decision problems in the form considered in Chap. 8, but also to define precisely the meaning of these formulations and solutions. This corresponds to the two parts of the definition of the uncertain variable mentioned in Sect. 8.1 after the Definition 8.1: a formal description and its interpretation. The remark concerning $\omega$ in this definition is also very important because it makes it possible to interpret precisely the source of the information about the unknown parameter $\bar{x}$ and the term "certainty index".

In the theory of fuzzy sets and systems there exist other formulations of analysis and decision problems (see for example [75]), different from those presented in this chapter. The decision problem with a fuzzy goal is usually based on the given $\mu_y(y)$ as the logic value of the property "$\hat{y}$ is satisfactory" or related properties.

The statements of analysis and decision problems in Chap. 8 for the system with the *known relation* $R$ *and unknown parameter* $c$ considered as an uncertain variable are similar to analogous approaches for the probabilistic model and together with the deterministic case form a unified set of problems. For $y = \Phi(u, c)$ and given $y$ the decision problem is as follows:

1. If $c$ is known (the deterministic case), find $u$ such that $\Phi(u, c) = y$.
2. If $c$ is a value of random variable $\tilde{c}$ with given certainty distribution, find $u$, maximizing the probability that $\tilde{y} = y$ (for the discrete variable), or find $u$ such that $\mathrm{E}(\tilde{y}, u) = y$ where E denotes the expected value of $\tilde{y}$.
3. If $c$ is a value of uncertain variable $\bar{c}$ with given certainty distribution, find $u$, maximizing the certainty index of the property $\bar{y} \cong y$, or find $u$ such that $\mathrm{M}_y(u) = y$ where M denotes the mean value of $\bar{y}$.

The definition of the uncertain variable has been used to introduce a *C*-uncertain variable, especially recommended for analysis and decision problems with unknown parameters because of its advantages mentioned in Sect. 8.1. Not only the interpretation but also a formal description of the *C*-uncertain variable differ in an obvious way from the known definitions of fuzzy numbers (see Definition 8.2 and the remark concerning the meas-

ure $m_c$ in this section).

## 9.4 Comparisons and Analogies for Non-parametric Problems

To indicate analogies and differences between the descriptions based on random, uncertain and fuzzy variables let us present together basic non-parametric problems (i.e. the problems based on the non-parametric descriptions), discussed in Sects. 6.3, 7.2, 8.2.2 and 9.2.2. The general approach to the decision problem is illustrated in Fig. 9.2, for a static plant with input vector $u \in U$, output vector $y \in Y$ and vector of external disturbances $z \in Z$. The knowledge of the decision making KD is determined from the knowledge of the plant KP and the requirement concerning $y$, given by a user. The deterministic decision algorithm $u_d = \Psi(z)$ is obtained as a result of the determinization of KD. For simplicity, we shall recall only the mean value as a result of the determinization.

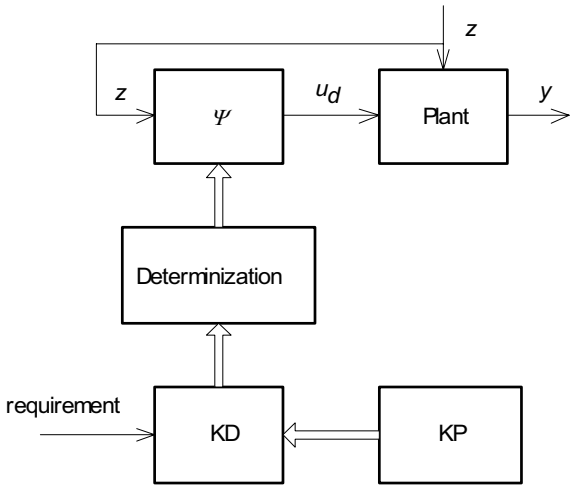


**Fig. 9.2.** General idea of the decision system under consideration

**A. Relational system**

The knowledge of the plant KP has the form of a relation

$$R(u, y, z) \subset U \times Y \times Z ,$$

which determines the set of possible outputs for the given $u$ and $z$:

$$D_y(u,z) = \{y \in Y : (u,y,z) \in R\}. \tag{9.27}$$

**Analysis problem:** For the given $D_y(u,z)$, $D_u \subset U$ and $D_z \subset Z$ one should determine the smallest set $D_y \subset Y$ for which the implication

$$(u \in D_u) \wedge (z \in D_z) \rightarrow y \in D_y$$

is satisfied.

According to (6.5) and (9.27)

$$D_y = \bigcup_{u \in D_u} \bigcup_{z \in D_z} D_y(u,z). \tag{9.28}$$

**Decision problem:** For the given $D_y(u,z)$ and $D_y$ required by a user one should determine the largest set $D_u(z)$ such that for the given $z$ the implication

$$u \in D_u(z) \rightarrow y \in D_y$$

is satisfied.

According to (6.19)

$$D_u(z) = \{u \in U : D_y(u,z) \subseteq D_y\} \stackrel{\Delta}{=} \overline{R}(z,u). \tag{9.29}$$

The knowledge of the decision making $\mathrm{KD} = < \overline{R}(z,u) >$ has been called a *relational decision algorithm* (the description of a *relational controller* in the open-loop control system). The determinization in the form of a mean value gives the deterministic decision algorithm

$$u_d = \int_{D_u(z)} u du \cdot [\int_{D_u(z)} du\ ]^{-1} \stackrel{\Delta}{=} \Psi_d(z).$$

The deterministic decision algorithm $\Psi_d(z)$ is based on the knowledge of the decision making KD, which is determined from the knowledge of the plant KP (reduced to $D_y(u,z)$), for the given $D_y$.

**B. Description based on random variables**

The knowledge of the plant has the form of a conditional probability density

$$\mathrm{KP} = < f_y(y \mid u,z) >. \tag{9.30}$$

**Analysis problem:** For the given $\mathrm{KP} = < f_y(y \mid u,z) >$, $f_u(u \mid z)$ and $f_z(z)$ find the probability density $f_y(y)$:

$$f_y(y) = \int_U \int_Z f_z(z) f_u(u|z) f_y(y|u,z)\, du\, dz\,. \tag{9.31}$$

**Decision problem:** For the given $\mathrm{KP} = < f_y(y|u,z) >$ and $f_y(y)$ required by a user one should determine $f_u(u|z)$.

The determination of $f_u(u|z)$ may be decomposed into two steps. In the first step one, should find the function $f_{uz}(u,z)$ satisfying the equation

$$f_y(y) = \int_U \int_Z f_{uz}(u,z) f_y(y|u,z)\, du\, dz \tag{9.32}$$

and the conditions for a probability density:

$$\bigwedge_{u\in U} \bigwedge_{z\in Z} f_{uz}(u,z) \geq 0\,, \qquad \int_U \int_Z f_{uz}(u,z)\, du\, dz = 1\,.$$

In the second step, one should determine the function $f_u(u|z)$:

$$f_u(u|z) = \frac{f_{uz}(u,z)}{\int_U f_{uz}(u,z)\, du}\,. \tag{9.33}$$

The knowledge of the decision making $\mathrm{KD} = < f_u(u|z) >$ has been called a *random decision algorithm* (the description of a *random controller* in the open-loop control system). The deterministic decision algorithm

$$u_d = \int_U u\, f_u(u|z)\, du \triangleq \Psi_d(z)$$

is based on KD determined from KP, for the given $f_y(y)$.

**C. Description based on uncertain variables**

The knowledge of the plant has the form of a conditional certainty distribution given by an expert:

$$\mathrm{KP} = < h_y(y|u,z) >\,. \tag{9.34}$$

**Analysis problem:** For the given $\mathrm{KP} = < h_y(y|u,z) >$, $h_u(u|z)$ and $h_z(z)$ find the certainty distribution $h_y(y)$.

According to (8.30)

$$h_y(y) = \max_{u \in U,\, z \in Z} \min\{h_z(z), h_u(u \mid z), h_y(y \mid u, z)\}. \tag{9.35}$$

**Decision problem:** For the given $\mathrm{KP} = < h_y(y \mid u, z) >$ and $h_y(y)$ required by a user one should determine $h_u(u \mid z)$.

According to (8.30) and (8.31), the determination of $h_u(u \mid z)$ may be decomposed into two steps. First, one should find the function $h_{uz}(u, z)$ satisfying the equation

$$h_y(y) = \max_{u \in U,\, z \in Z} \min\{h_{uz}(u, z),\, h_y(y \mid u, z)\} \tag{9.36}$$

and the conditions for a certainty distribution

$$\bigwedge_{u \in U} \bigwedge_{z \in Z} h_{uz}(u, z) \geq 0, \qquad \max_{u \in U,\, z \in Z} h_{uz}(u, z) = 1.$$

Then, one should determine the function $h_u(u \mid z)$ satisfying the equation

$$h_{uz}(u, z) = \min\{\max_{u \in U} h_{uz}(u, z), h_u(u \mid z)\} \tag{9.37}$$

and the conditions for a certainty distribution. The knowledge of the decision making $\mathrm{KD} = < h_u(u \mid z) >$ has been called an *uncertain decision algorithm* (the description of an *uncertain controller* in the open-loop control system). The deterministic decision algorithm

$$u_d = \int_U u\, h_u(u \mid z)\, du \cdot [\int_U h_u(u \mid z)\, du\,]^{-1} \stackrel{\Delta}{=} \Psi_d(z)$$

is based on KD determined from KP, for the given $h_y(y)$.

**D. Description based on fuzzy variables**

For the determined soft properties $\varphi_u(u)$, $\varphi_z(z)$ and $\varphi_y(y)$, the knowledge of the plant has the form of a membership function

$$\mathrm{KP} = < \mu_y(y \mid u, z) >. \tag{9.38}$$

**Analysis problem:** For the given $\mathrm{KP} = < \mu_y(y \mid u, z) >$, $\mu_u(u \mid z)$ and $\mu_z(z)$ find the membership function $\mu_y(y)$.

The solution is given by the formula (9.23).

**Decision problem:** For the given $\mathrm{KP} = < \mu_y(y \mid u, z) >$ and $\mu_y(y)$ required by a user one should determine $\mu_u(u \mid z)$.

Two steps of the solution are described by the formulas (9.24) and (9.25). The deterministic decision algorithm

$$u_d = \int_U u\,\mu_u(u\,|\,z)\,du \cdot [\int_U \mu_u(u\,|\,z)\,du\,]^{-1} \stackrel{\Delta}{=} \Psi_d(z)$$

is based on the *fuzzy decision algorithm* (the description of a *fuzzy controller* in the open-loop control system) $\mathrm{KD} = <\mu_u(u\,|\,z)>$, and is determined from KP for the given $\mu_y(y)$ with $\Psi_d$ in place of $\Psi_a$, $\Psi_b$.

**Remark 9.4.** In special cases of the decision problem considered in Sects. 8.2.2 and 9.2.2, when the solution in the first step in the form of $h_{uz}(u,z)$ or $\mu_{uz}(u,z)$ is not unique, the distribution $h_z(z)$ or $\mu_z(z)$ may be given a priori. □

The different cases of KP are described by (9.27), (9.30), (9.34), (9.38) and the respective results of the analysis problem are given by (9.28), (9.31), (9.35), (9.23). The solution of the decision problem (9.29) corresponds to the solution in two steps described by (9.32) and (9.33) for the random variables, by (9.36) and (9.37) for the uncertain variables, and by (9.24) and (9.25) for the fuzzy variables. The essential differences are the following:

1. Cases A, B are based on the objective descriptions of KP, and cases C, D are based on the subjective descriptions given by an expert.
2. The descriptions in cases B, C are concerned directly with values of $(u,y,z)$, and the description in case D is concerned with determined properties of $(u,y,z)$.

## 9.5 Introduction to Soft Variables

The uncertain, random and fuzzy variables may be considered as special cases of a more general description of the uncertainty in the form of *soft variables* and *evaluating functions* [50, 52], which may be introduced as a tool for a unification and generalization of non-parametric analysis and decision problems based on the uncertain knowledge representation. The definition of a soft variable should be completed with the determination of relationships for the pair of soft variables.

**Definition 9.1** (*soft variable and the pair of soft variables*). A soft variable $\overset{\vee}{x} = <X, g(x)>$ is defined by the set of values $X$ (a real number vector

space) and a bounded evaluating function $g: X \to R^+$, satisfying the following condition:

$$\int_X xg(x) < \infty$$

for the continuous case and

$$\sum_{i=1}^{\infty} x_i g(x_i) < \infty$$

for the discrete case.

Let us consider two soft variables $\overset{\vee}{x} =< X, g_x(x) >$, $\overset{\vee}{y} =< Y, g_y(y) >$ and the variable $(\overset{\vee}{x}, \overset{\vee}{y})$ described by $g_{xy}(x,y): X \times Y \to R^+$. Denote by $g_y(y \mid x)$ the evaluating function of $\overset{\vee}{y}$ for the given value $x$ (the conditional evaluating function). The pair $(\overset{\vee}{x}, \overset{\vee}{y})$ is defined by $g_{xy}(x,y)$ and two operations:

$$g_{xy}(x,y) = O_1[g_x(x), g_y(y \mid x)], \tag{9.39}$$

$$g_x(x) = O_2[g_{xy}(x,y)], \tag{9.40}$$

i.e.

$$O_1 : D_{gx} \times D_{gy} \to D_{g,xy}, \qquad O_2 : D_{g,xy} \to D_{g,x}$$

where $D_{gx}$, $D_{gy}(x)$ and $D_{g,xy}$ are sets of the functions $g_x(x)$, $g_y(y \mid x)$ and $g_{xy}(x,y)$, respectively. The mean value $\mathrm{M}(\overset{\vee}{x})$ is defined in the same way as for an uncertain variable (see Sect.8.1), with $g_x(x)$ in place of $h_x(x)$. □

The evaluating function may have different practical interpretations. In the random case, a soft variable is a random variable described by the probability density $g(x) = f(x)$ or by probabilities $g(x_i) = P(\tilde{x} = x_i)$. In the case of an uncertain variable, $g(x) = h(x)$ is the certainty distribution. In the case of the fuzzy description, a soft variable is a fuzzy variable described by the membership function $g(x) = \mu(x) = w[\varphi(x)]$ where $w$ denotes a logic value of a given soft property $\varphi(x)$. In general, we can say

that $g(x)$ describes an evaluation of the set of possible values $X$, characterizing for every value $x$ its significance (importance or weight). This description presents a knowledge concerning the variable $\overset{\vee}{x}$, which may be given by an expert describing his / her subjective opinion, or may have an objective character such as in the case of a random variable.

The non-parametric decision (control) problems considered for random, uncertain and fuzzy variables may be written together and generalized by using soft variables. For the plant with $u \in U$, $y \in Y$ and $z \in Z$ we assume that $(u, y, z)$ are values of soft variables $(\overset{\vee}{u}, \overset{\vee}{y}, \overset{\vee}{z})$ and the knowledge of the plant has the form of a conditional evaluating function

$$\mathrm{KP} = < g_y(y \,|\, u, z) > .$$

**Decision problem:** For the given $\mathrm{KP} = < g_y(y \,|\, u, z) >$ and $g_y(y)$ required by a user one should determine $g_u(u \,|\, z)$.

The determination of $g_u(u \,|\, z)$ may be decomposed into two steps. In the first step, one should find the evaluating function $g_{uz}(u, z)$ satisfying the equation

$$g_y(y) = O_2\{O_1[\, g_{uz}(u, z), g_y(y \,|\, u, z)]\}.$$

In the second step, one should determine the function $g_u(u \,|\, z)$ satisfying the equation

$$g_{uz}(u, z) = O_1[\, g_z(z), g_u(u \,|\, z)]$$

where

$$g_z(z) = O_2[\, g_{uz}(u, z)\,] .$$

The function $g_u(u \,|\, z)$ may be called a knowledge of the decision making $\mathrm{KD} = < g_u(u \,|\, z) >$ or a *soft decision algorithm* (the description of a *soft controller* in the open-loop control system). Having $g_u(u \,|\, z)$ one can obtain the deterministic decision algorithm as a result of the determinization of the soft decision algorithm. Two versions of the determinization are the following:

*Version I.*

$$u_a = \arg \max_{u \,\in\, U} \; g_u(u \,|\, z) \overset{\Delta}{=} \Psi_a(z) .$$

*Version II.*

$$u_b = \mathrm{M}(\overset{\vee}{u}\,|\,z) = \int_U u\, g_u(u\,|\,z)\,du \cdot [\int_U g_u(u\,|\,z)\,du\,]^{-1} \overset{\Delta}{=} \Psi_b(z)\,.$$

The deterministic decision algorithms $\Psi_a(z)$ or $\Psi_b(z)$ are based on the knowledge of the decision making $\mathrm{KD} = < g_u(u\,|\,z) >$ determined from the knowledge of the plant KP for the given $g_y(y)$.

## 9.6 Descriptive and Prescriptive Approaches. Quality of Decisions

In the analysis and design of knowledge-based uncertain systems it may be important to investigate a relation between two concepts concerning two different subjects of the knowledge given by an expert, which have been mentioned in Sect. 6.5 and Sect. 7.2 [39, 52]. In the *descriptive approach* an expert gives the knowledge of the plant KP, and the knowledge of the decision making KD is obtained from KP for the given requirement. This approach is widely used in the traditional decision and control theory. The deterministic decision algorithm may be obtained via the determinization of KP or the determinization of KD based on KP. Such a situation is illustrated in Figs. 6.8 and 6.9 for the relational description and in Figs. 8.2 and 8.3 for the formulation based on uncertain variables. In the *prescriptive approach* the knowledge of the decision making $\overline{\mathrm{KD}}$ is given directly by an expert. This approach is used in the design of fuzzy controllers where the deterministic control algorithm is obtained via the defuzzification of the knowledge of the control given by an expert. The descriptive approach to the decision making based on the fuzzy description may be found in [75].

Generally speaking, the descriptive and prescriptive approaches may be called *equivalent* if the deterministic decision algorithms based on KP and $\overline{\mathrm{KD}}$ are the same. Different particular cases considered in the previous chapters may be illustrated in Figs. 9.3 and 9.4 for two different concepts of the determinization. Fig. 9.5 illustrates the prescriptive approach. In the first version (Fig. 9.3) the approaches are equivalent if $\Psi(z) = \overline{\Psi}_d(z)$ for every $z$. In the second version (Fig. 9.4) the approaches are equivalent if $\mathrm{KD} = \overline{\mathrm{KD}}$. Then $\Psi_d(z) = \overline{\Psi}_d(z)$ for every $z$.

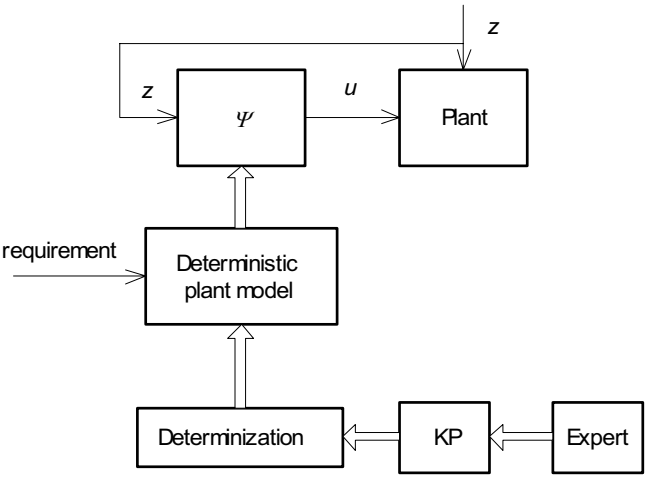


**Fig. 9.3.** Illustration of descriptive approach – the first version

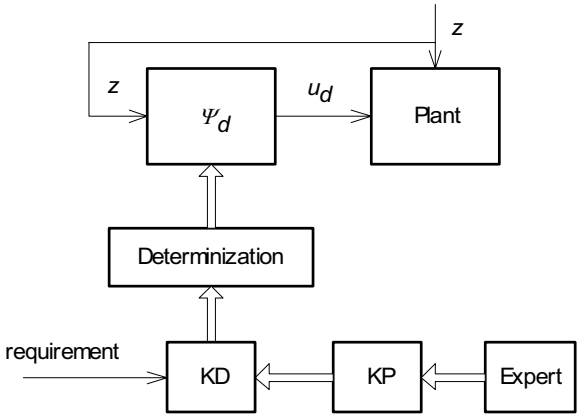


**Fig. 9.4.** Illustration of descriptive approach – the second version

Let us consider more precisely version I of the decision problem described in Sect. 8.2.1. An expert formulates $\mathrm{KP} = < \Phi, h_c >$ (the descriptive approach) or $\overline{\mathrm{KD}} = < \overline{\Phi}_d, h_c >$ (the prescriptive approach). In the first version of the determinization illustrated in Fig. 8.2, the approaches are equivalent if $\Psi_a(z) = \overline{\Psi}_{ad}(z)$ for every $z$, where $\Psi_a(z)$ is determined by

(8.22) and $\overline{\Psi}_{ad}(z)$ is determined by (8.27) with $\overline{\Phi}_d(z,c)$ instead of $\Phi_d(z,c)$ obtained as a solution of the equation

$$\Phi(u,z,c) = y^* . \tag{9.41}$$

In the second version of the determinization illustrated in Fig. 8.3, the approaches are equivalent if the solution of equation (9.41) with respect to $u$ has the form $\overline{\Phi}_d(z,c)$, i.e.

$$\Phi[\overline{\Phi}_d(z,c),z,c] = y^* .$$

For the non-parametric problem described in Sect. 8.2.2 only the second version of the determinization may be applied.

The similar formulation of the equivalency may be given for the random and fuzzy descriptions presented in Chap. 7 and in this chapter, respectively. The generalization for the soft variables and evaluating functions described in Sect. 9.5 may be formulated as a principle of equivalency.

**Principle of equivalency:** If the knowledge of the decision making $\overline{\text{KD}}$ given by an expert has the form of an evaluating function $\hat{g}_u(u \mid z)$ and $\hat{g}_u(u \mid z) \in D_{gu}(z)$ where $D_{gu}(z)$ is the set of all solutions of the decision problem presented in Sect. 9.5, then the decision algorithm based on the knowledge of the decision making given by an expert is equivalent to one of the decision algorithms based on the knowledge of the plant. □

For the non-parametric cases described together in Sect. 9.4, descriptive and prescriptive approaches are equivalent if:

1. $\overline{f}_u(u \mid z)$ given by an expert satisfies equation (9.31).
2. $\overline{h}_u(u \mid z)$ given by an expert satisfies equation (9.35).
3. $\overline{\mu}_u(u \mid z)$ given by an expert satisfies equation (9.23).
4. $\overline{g}_u(u \mid z)$ given by an expert satisfies equation

$$g_y(y) = O_2\{O_1[\, O_1(g_z(z), g_u(u \mid z)), g_y(y \mid u,z)]\} .$$

It is worth noting that the determination of the decision algorithm $\overline{\Psi}$ based on $\overline{\text{KD}}$ means the solution of the analysis problem for the unit (the plant) described by $\overline{\text{KD}}$ and for the given input of this unit: $z$ in the open-loop system and $x$ in the closed-loop system. It may be useful to present together the determinizations of $\overline{\text{KD}}$ in non-parametric cases for static plants in an open-loop system (see Fig. 9.5).

**A. Description based on random variables**

For the given $\bar{f}_u(u \mid z)$ one should determine

$$\bar{u}_d = \mathrm{E}(\tilde{u} \mid z) = \int_U u\bar{f}_u(u \mid z)du \stackrel{\Delta}{=} \bar{\Psi}_d(z) .$$

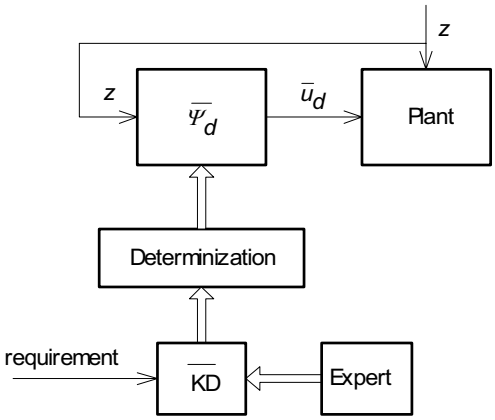


**Fig. 9.5.** Illustration of prescriptive approach

**B. Description based on uncertain variables**

For the given $\bar{h}_u(u \mid z)$ one should determine

$$\bar{u}_d = \mathrm{M}(\bar{u} \mid z) = \int_U u\bar{h}_u(u \mid z)du \cdot [\int_U \bar{h}_u(u,z)du]^{-1} \stackrel{\Delta}{=} \bar{\Psi}_d(z) .$$

**C. Description based on fuzzy variables ($U = R^1$)**

In this case we can consider two versions (see Sects. 9.1 and 9.2):

1. For the given $\bar{\mu}_u(u \mid z) = w[\varphi_z(z) \to \varphi_u(u)]$ one should determine

$$\bar{u}_d = \mathrm{M}(\hat{u} \mid z) = \int_{-\infty}^{\infty} u\bar{\mu}_u(u \mid z)du \cdot [\int_{-\infty}^{\infty} \bar{\mu}_u(u,z)du]^{-1} \stackrel{\Delta}{=} \bar{\Psi}_d(z) . \qquad (9.42)$$

2. For the given

$$\bar{\mu}_{uz}(u,z) = w[\hat{z} = z \to \varphi_u(u)] = \min\{\bar{\mu}_z(z), \bar{\mu}_u(u \mid z)\}$$

one should determine

$$\bar{u}_d = \mathrm{M}(\hat{u} \mid z) = \int_{-\infty}^{\infty} u\bar{\mu}_{uz}(u \mid z)du \cdot [\int_{-\infty}^{\infty} \bar{\mu}_{uz}(u \mid z)du]^{-1} \stackrel{\Delta}{=} \bar{\Psi}_d(z) . \qquad (9.43)$$

Cases 1 and 2 are equivalent if $\overline{\mu}_u(u \mid z) = \overline{\mu}_{uz}(u \mid z)$.

Instead of the mean value E or M we can use the value of $u$ maximizing the distribution, e.g. in case A

$$\overline{u}_d = \arg\max_{u \in U} \bar{f}_u(u \mid z).$$

Denote by $a$ the vector of parameters in the description given by an expert. They may be parameters in $\bar{f}_u(u \mid z)$, $\overline{h}_u(u \mid z)$ or $\overline{\mu}_u(u \mid z)$. Consequently, the deterministic decision algorithm $\overline{u}_d = \overline{\Psi}_d(z,a)$ depends on $a$. Then the problem of a *parametric optimization* consisting in choosing $a^*$ minimizing the performance index $Q$, and the problem of *adaptation* consisting in adjusting $a$ to $a^*$, may be considered (see Sect. 11.4).

The cases corresponding to A, B, C may be listed for the dynamical plant with $x$ instead of $z$. Note that the description for the fuzzy variables is concerned with a simple one-dimensional case. In the next section we shall present it for a multi-dimensional case in the closed-loop system.

Different approaches to the determination of the deterministic decision (control) algorithm, based on different formal descriptions of the uncertainty (and including descriptive and prescriptive approaches) may be verified and compared by evaluating the quality of decisions based on an uncertain description and applied to a concrete deterministic plant with a known description. Consider a plant described by a function $y = \Phi(u,z)$ and introduce the performance index evaluating the quality of the decision $u$ for the given $z$

$$Q(u,z) = (y - y^*)^{\mathrm{T}}(y - y^*) = [\Phi(u,z) - y^*]^{\mathrm{T}}[\Phi(u,z) - y^*]$$

where $y^*$ denotes a desirable value of the output. Assume that the function $\Phi$ (i.e. the exact deterministic description of the plant) is unknown, $(u,y,z)$ are values of uncertain variables $(\overline{u},\overline{y},\overline{z})$ and a user presents the requirement in the form of a certainty distribution $h_y(y)$ in which

$$\arg\max_y h_y(y) = y^*.$$

If $u_d = \Psi(z,a)$ is the deterministic decision algorithm obtained as a result of a determinization of the uncertain decision algorithm $h_u(u \mid z)$ obtained from KP or given directly by a user, then

$$Q(z) = [\Phi(\Psi(z,a),z) - y^*]^{\mathrm{T}}[\Phi(\Psi(z,a),z) - y^*] \triangleq \overline{\Phi}(z) \qquad (9.44)$$

where $a$ is the vector of parameters in the certainty distribution $h_y(y \mid z)$ (in the descriptive approach) or directly in the certainty distribution $h_u(u \mid z)$ (in the prescriptive approach). For the given $z$, the performance index (9.44) evaluates the quality of the decision $u_d$ based on the uncertain knowledge and applied to the real plant described by $\Phi$. To evaluate the quality of the algorithm $\Psi_d$ for different possible values of $z$, one can use the mean value

$$\mathrm{M}(\overline{Q}) = \int_0^{+\infty} q\, h_q(q)dq \cdot [\int_0^{+\infty} h_q(q)dq]^{-1} \tag{9.45}$$

where $h_q(q)$ is the certainty distribution of $\overline{Q} = \overline{\Phi}(\bar{z})$ which should be determined for the given function $\overline{\Phi}$ and the certainty distribution

$$h_z(z) = \max_{u \in U} h_{uz}(u,z),$$

and $h_{uz}(u,z)$ is the distribution obtained in the first step of the decision problem solution (see (8.30) in Sect. 8.2.2). In the prescriptive approach $h_z(z)$ should be given by an expert. The performance index (9.44) or (9.45) may be used in:

1. Investigation of the influence of the parameter $a$ in the description of the uncertain knowledge on the quality of the decisions based on this knowledge.
2. Comparison of the descriptive and prescriptive approaches in the case when $h_u(u \mid z)$ and $\overline{h}_u(u \mid z)$ have the same form with different values of the parameter $a$.
3. Parametric optimization and adaptation, when

$$a^* = \arg\min_a Q(a)$$

is obtained by the adaptation process consisting in *step by step* changing of the parameters of the controller in an open-loop decision system with a simulator of the plant described by the model $\Phi$.

The considerations for fuzzy controllers are analogous, with $\mu_y(y \mid u,z)$, $\mu_z(z)$ and $\mu_u(u \mid z)$ in place of $h_y(y \mid u,z)$, $h_z(z)$ and $h_u(u \mid z)$.

## 9.7 Control for Dynamical Plants. Fuzzy Controller

Let us consider the closed-loop control system with a dynamical plant (continuous- or discrete-time) in which the state $x$ is put at the input of the controller. In the simple one-dimensional case the knowledge $\overline{\text{KD}}$ (or the fuzzy controller) given by an expert consists of two parts:
1. The *rule*

$$\varphi_x(x) \to \varphi_u(u) \tag{9.46}$$

with the determined properties $\varphi_x(x)$ and $\varphi_u(u)$: "$x$ is $d_x$" and "$u$ is $d_u$", i.e. "if $\hat{x} = x$ then $x$ is $d_x$" and "if $\hat{u} = u$ then $u$ is $d_u$" (see Sect. 9.1).
2. In the first version corresponding to (9.42) for the open-loop system – the membership function $\mu_u(u \mid x)$ of the property (9.46). In the second version corresponding to (9.43) – the membership function $\mu_u(u \mid x)$ and the membership function $\mu_x(x)$ of the property $\varphi_x(x)$, or directly the membership function

$$\mu_{ux}(u,x) = w[\hat{x} = x \to \varphi_u(u)] = \min\{\mu_x(x), \mu_u(u \mid x)\}. \tag{9.47}$$

Then the deterministic control algorithm (or deterministic controller) is described by the following procedure:
1. Put $x$ at the input of the controller.
2. In the first version, determine the decision

$$u_d = \int_{-\infty}^{\infty} u\mu_u(u \mid x)du \cdot [\int_{-\infty}^{\infty} \mu_u(u \mid x)du]^{-1}.$$

In the second version, for the given $\mu_u(u \mid x)$ and $\mu_x(x)$ determine $\mu_{ux}(u,x)$ according to (9.47) and find the decision

$$u_d = \int_{-\infty}^{\infty} u\mu_{ux}(u,x)du \cdot [\int_{-\infty}^{\infty} \mu_{ux}(u,x)du]^{-1}.$$

Instead of the mean value we can determine and use the decision $u_d$ maximizing the membership function $\mu_u(u \mid x)$ or $\mu_{ux}(u,x)$.

Let us present the extension of the second version to the controller with $k$ inputs $x^{(1)}, x^{(2)}, ..., x^{(k)}$ (the components of the state vector $x$) and one output $u$. The description of the *fuzzy controller* ($\overline{\text{KD}}$) given by an expert has a form analogous to that for the fuzzy description of a multi-dimensional static plant (see Sect. 9.2.2) and contains two parts:

1. The set of *rules*

$$\varphi_{j1}(x^{(1)}) \wedge \varphi_{j2}(x^{(2)}) \wedge ... \wedge \varphi_{jk}(x^{(k)}) \to \varphi_{ju}(x^{(u)}), \quad j = 1,2,...,N \tag{9.48}$$

where $N$ is a number of rules, $\varphi_{ji}(x^{(i)}) =$ "$x^{(i)}$ is $d_{ji}$" and $\varphi_{ju}(u) =$ "$u$ is $d_j$", $d_{ji}$ and $d_j$ denote the size of the numbers as in Sects. 9.1 and 9.2.2. The meaning of the rules (9.48) is then as follows:

IF ($x^{(1)}$ is $d_{j1}$) AND ($x^{(2)}$ is $d_{j2}$) AND ... AND ($x^{(k)}$ is $d_{jk}$)

THEN $u$ is $d_j$.

For example ($k = 3$)

($x^{(1)}$ is small positive) $\wedge$ ($x^{(2)}$ is large negative) $\wedge$ ($x^{(3)}$ is small negative) $\to$ $u$ is medium positive.

2. The matrix of the membership functions

$$\mu_{xji}(x^{(i)}) = w[\varphi_{ji}(x^{(i)})], \quad \begin{matrix} i = 1,2,...,k, \\ j = 1,2,...,N \end{matrix}$$

and the sequence of the membership functions

$$\mu_{uj}(u \mid x^{(1)}, x^{(2)}, ..., x^{(k)}), \qquad j = 1,2,...,N$$

for the properties (9.48).

The *deterministic controller* (i.e. the deterministic control algorithm obtained as a result of the determinization of $\overline{\text{KD}}$) is described by the following procedure:

1. Put $x$ at the input of the controller.
2. Find the sequence of values

$$\mu_{xj}(x) = \min\{\mu_{xj1}(x^{(1)}), \mu_{xj2}(x^{(2)}), ..., \mu_{xjk}(x^{(k)})\}, \quad j = 1,2,...,N.$$

3. From each rule determine the membership function

$$\mu_{ux,j}(u,x) = w[\hat{x} = x \to u \text{ is } d_j] = \min\{\mu_{xj}(x), \mu_{uj}(u \mid x)\},$$
$$j = 1,2,..., N.$$

4. Determine the membership function $\mu_u(x)$ of the property

$$(\hat{x} = x) \to (u \text{ is } d_1) \vee (u \text{ is } d_2) \vee ... \vee (u \text{ is } d_N).$$

Then

$$\mu_u(x) = \max\{\mu_{ux,1}(u,x), \mu_{ux,2}(u,x), ..., \mu_{ux,N}(u,x)\} .$$

5. Determine the decision $u_d$ as a result of the determinization (defuzzification) of $\mu_u(x)$:

$$u_{ad} = \arg\max_u \mu_u(x) \stackrel{\Delta}{=} \overline{\Psi}_{ad}(x)$$

or

$$u_{bd} = \int_{-\infty}^{+\infty} u\mu_u(x)du \,[\int_{-\infty}^{+\infty} \mu_u(x)du]^{-1} \stackrel{\Delta}{=} \overline{\Psi}_{bd}(x) .$$

In a discrete case the integrals are replaced by the sums (see (9.5)). For simplicity, it may be assumed that the membership function of the implication (9.48) does not depend on $x$. Then

$$\mu_{uj}(u|x) \stackrel{\Delta}{=} \overline{\mu}_{uj}(u)$$

and

$$\mu_{ux,j}(u,x) = w[\hat{x} = x \to u \text{ is } d_j] = \min\{\mu_{xj}(x), \overline{\mu}_{uj}(u)\} .$$

The relations between $\overline{\mu}_{uj}$, $\mu_{xj}$ and $\mu_{uxj}$ are illustrated in Fig. 9.6.

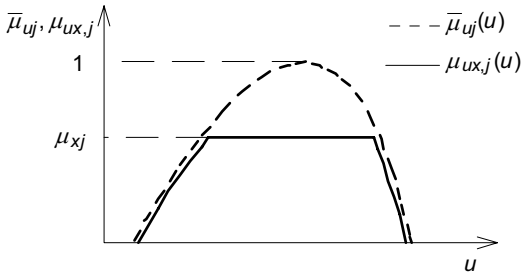


**Fig. 9.6.** Example of $\overline{\mu}_{uj}$ and $\mu_{ux,j}$

If for a single-output continuous dynamical plant $x^{\mathrm{T}} = [\varepsilon(t), \dot{\varepsilon}(t), ..., \varepsilon^{(k-1)}(t)]$ (where $\varepsilon(t)$ is the control error put at the input of the controller), then the properties in the rules and the corresponding membership functions concern the control error and its derivatives. If $u$ is a vector (in the case of a multi-input plant) then the knowledge given by an expert and the procedure of finding the decision are for each component of $u$ the same as for one-dimensional $u$ considered above. There exist different versions and modifications of fuzzy controllers described in the literature (e.g. [62]). To characterize the fuzzy controllers based on the knowledge

of the control given by an expert, the following remarks should be taken into account:

1. In fact, the control decisions $u_d$ are determined by the *deterministic controller* $\Psi_d$ in the closed-loop control system.
2. The deterministic control algorithm $\Psi_d$ has a form of the *procedure* presented in this section, based on the description of the *fuzzy controller* (i.e. the knowledge of the control $\overline{\text{KD}}$) given by an expert (Fig. 9.7).
3. The deterministic control algorithm $u_d = \Psi_d(x,a)$ where $a$ is the vector of parameters of the membership functions in $\overline{\text{KD}}$ – may be considered as a parametric form of a deterministic controller. This form is determined by the forms of rules and membership functions in $\overline{\text{KD}}$, i.e. is proposed indirectly by an expert.
4. The parametric form $u_d = \Psi_d(x,a)$ is proposed in a rather arbitrary way, not reasoned by the description of the plant. Besides, it is a rather complicated form (in comparison with traditional and given directly parametric forms of a deterministic controller) and the decisions $u_d$ may be very sensitive to changes of forms and parameters of the membership functions in $\overline{\text{KD}}$.
5. It is reasonable and recommended to apply the *parametric optimization* described in Chap. 5 and *adaptation* presented in Sect. 11.4, to achieve the value $a^*$ optimal for the accepted form $\Psi_d$, i.e. for the forms of rules and membership functions in $\overline{\text{KD}}$ given by an expert.

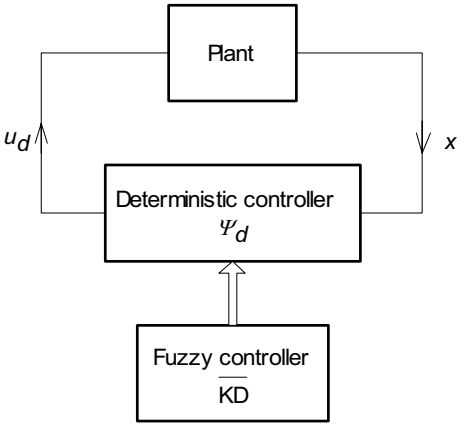


**Fig. 9.7.** Control system based on fuzzy controller

# 10 Control in Closed-loop System. Stability

Chapters 10 and 11 form the **fourth part** of the book, which is devoted to control under uncertainties, as the former part. Unlike the **third part** containing Chaps. 6, 7, 8 and 9, now we shall consider two concepts of using information obtained during the control process in a closed-loop system: to the direct determination of control decision (Chap. 10) and to *step by step* improving of a basic decision algorithm in an adaptation and learning process (Chap. 11).

## 10.1 General Problem Description

In Chaps. 7 and 8 we considered control plants with unknown parameters with the description of the uncertainty in the form of probability distributions or certainty distributions. These have been the descriptions of *a priori* information on the unknown parameters, i.e. the information known at the stage of a design, before starting the control process. Only in Sect. 7.3 we considered a case when the information on the unknown parameter was obtained during the control process and was used to current modifications of control decisions. Obtaining the information had there a direct character and consisted in a direct observation of the unknown parameter $c$, more precisely – in the measurement of this parameter with the presence of random noises. As a result, the information on the parameter $c$ could be formulated in an *explicit* form (directly and precisely), i.e. in the form of *a priori* probability density $f_c(c)$ and *a posteriori* probability density $f_c(c \mid \overline{w}_n)$. Now we shall consider a concept consisting in obtaining the information on the plant during the control process in an indirect way, via observations of control results in a closed-loop control system. In such a case, it is important to use effects of the earlier control decisions for the determination of the proper next decisions and to design the closed-loop system in such a way as to assure the convergence of the control process to the values required. This is the main idea of the design and the performance of a feed-back system. Let us note that obtaining the information as a result of the direct observation of the unknown parameters does not require

a simultaneous control, i.e. has a *passive* character, while obtaining the information by the observations of control results requires a variation of the plant input, i.e. has an *active* character.

Let us present more precisely the above concept for a static plant

$$y = \Phi(u), \qquad u \in U, \quad y \in Y,$$

described in Sect. 3.1. Finding the solution $u^*$ of the equation $y^* = \Phi(u)$ for the determination of the decision $u = u^*$ satisfying the requirement $y = y^*$ may be obtained by using the successive approximation method, according to the algorithm

$$u_{n+1} = u_n + K\,[\,y^* - \Phi(u_n)] \tag{10.1}$$

where $u_n$ is the $n$-th approximation of the solution, $K$ is the matrix of coefficients whose values should be chosen in such a way that $u_n \to u^*$ for $n \to \infty$. The algorithm (10.1) may be executed in the closed-loop control system (Fig. 10.1). It means that the substituting of the approximation $u_n$ into the formula $\Phi$ and calculating the value $\Phi(u_n)$ is replaced by putting the value $u_n$ at the input of the plant and measuring the output $y_n$. Then, $u_n$ is now the control decision in the $n$-th period of the control and, according to (10.1), the control algorithm in the closed-loop system is a follows:

$$u_{n+1} = u_n + K\varepsilon_n \tag{10.2}$$

where $\varepsilon_n = y^* - y_n$ denotes the control error. Consequently, the control system as a whole is a discrete dynamical system described by the equation $u_{n+1} = F(u_n)$ where

$$F(u_n) = u_n + K[y^* - \Phi(u_n)].$$

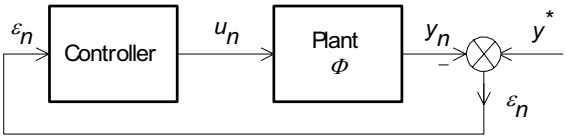


**Fig. 10.1.** Closed-loop control system with static plant

The value $u^*$ satisfying the equation $u = F(u)$ may be called an *equilibrium state* of the system and the property $u_n \to u^*$ for $n \to \infty$ may be called a

*stability of the equilibrium state* or shortly – a *stability of the system*. So, instead of speaking about the convergence of the approximation process in an *approximation system* in which the successive approximations $u_n$ are executed starting with the initial value $u_0$, one can speak about the stability of the control system, meaning the convergence of $u_n$ to the equilibrium state $u^*$, i.e. the returning to the equilibrium state. The initial state $u_0 \neq u^*$ is an effect of a disturbance which acted before the moment $n$ and removed the system from the equilibrium state.

In the further considerations we shall use the term *stability*, remembering that the stability conditions under consideration have a wider meaning and may be used as convergence conditions in an approximation system containing a plant of approximation and an approximation algorithm. This is a uniform approach to convergence problems in different systems realizing recursive approximation processes, such as a computational system determining successive approximations of a solution, a system of identification, control, recognition, self-optimization seeking an extremum, adaptation etc. [8].

Let us assume now that an unknown parameter $c$ occurs in the function $\Phi$, i.e. $y = \Phi(u, c)$ where in general $c$ is a vector ($c \in C$), and note that the exact knowledge of the value $c$ is not necessary for the determination of the solution of the equation $y^* = \Phi(u)$ by the successive approximation procedure (10.1) with any initial value $u_0$, or for the satisfaction of the stability condition in the respective control system with any initial state (see the remark at the end of Sect. 3.1). In other words, by the proper choosing of $K$, the property of the convergence (stability) can be satisfied for a set of different values $c$. Then, for the proper choosing of the matrix $K$, the exact knowledge of $c$ is not required; it is sufficient to know the set of all possible values $c$. So, the matrix $K$ assuring the convergence may be determined for an uncertain plant with the description of the uncertainty in the form of a set of all possible values $c$.

For example, let in the one-dimensional case $y = cu$. Then according to (10.1)

$$\bar{u}_{n+1} = \bar{u}_n - kc\bar{u}_n = (1 - kc)\bar{u}_n$$

where $\bar{u}_n = u_n - u^*$. Hence, the inequality $|1 - kc| \leq 1$, or $0 < kc < 2$ is the necessary and sufficient condition of the convergence of $\bar{u}_n$ to 0 for any $u_0$. If it is known that the unknown parameter $c \in (0, \bar{c}\,]$ then this condition will be fulfilled for every $k \in (0, \frac{2}{\bar{c}})$, i.e. every $k$ satisfying the inequality

$0 < k < \frac{2}{\bar{c}}$.

The above considerations may be generalized for the dynamical plant in the closed-loop control system

$$\left.\begin{aligned} x_{\mathrm{O},n+1} &= f_{\mathrm{O}}(x_{\mathrm{O}n}, u_n), \qquad & y_n &= \eta_{\mathrm{O}}(x_{\mathrm{O}n}), \\ x_{\mathrm{R},n+1} &= f_{\mathrm{R}}(x_{\mathrm{R}n}, y_n), \qquad & u_n &= \eta_{\mathrm{R}}(x_{\mathrm{R}n}) \end{aligned}\right\} \tag{10.3}$$

where $x_{\mathrm{O}n}$ is the state vector of the plant, $x_{\mathrm{R}n}$ is the state of the controller, $u_n$ is the input vector of the plant and $y_n$ is the output vector of the plant.

By substituting $u_n = \eta_{\mathrm{R}}(x_{\mathrm{R}n})$ into $f_{\mathrm{O}}$ and $y_n = \eta_{\mathrm{O}}(x_{\mathrm{O}n})$ into $f_{\mathrm{R}}$, the set of equations (10.3) may be reduced to one equation

$$x_{n+1} = F(x_n), \qquad x_n \in X = R^k \tag{10.4}$$

where

$$x_n = \begin{bmatrix} x_{\mathrm{O}n} \\ x_{\mathrm{R}n} \end{bmatrix}$$

is the state vector of the control system. The solution $x \stackrel{\Delta}{=} x_e$ of the equation $x = F(x)$ is called an *equilibrium state*. Let us assume that the system described by (10.4) has one and only one equilibrium state.

**Definition 10.1** (*stability*). The system (10.4) (or its equilibrium state $x_e$) is called *globally asymptotically stable* in the domain $D_x \subseteq X$ if and only if $\lim_{n\to\infty} x_n = x_e$ for every $x_0 \in D_x$. □

In the further considerations we shall speak only about asymptotic stability. Sometimes, the system globally stable for $D_x = X$ is called totally stable. For practical reasons, for the system (10.4) with the fixed input $\bar{u}_n$ and output $\bar{y}_n$, i.e. described by the equation

$$x_{n+1} = F(x_n, \bar{u}_n), \qquad \bar{y}_n = \eta(x_n)$$

we introduce a property called *input-output stability*. This property means that if $\bar{u}_n \to \bar{u}$ for $n \to \infty$ then $\bar{y}_n \to \eta(x_e) \stackrel{\Delta}{=} \bar{y}$ where $x_e$ is the equilibrium state for $u_n = \text{const.} = \bar{u}$, i.e. is a solution of the equation $x = F(x, \bar{u})$. In other words, if a disturbance $\bar{u}_n - \bar{u}$ acting at the input is converging to 0 then the response at the output is converging to the value $\bar{y}$ correspond-

ing to the equilibrium state. In the next considerations we shall assume that $x_e = \overline{0}$, $\overline{u} = \overline{0}$ and $\overline{y} = \overline{0}$. The input-output stability depends on the equations describing the system and on the choice of the pair $(\overline{u}_n, \overline{y}_n)$. If the system is stable in the sense determined in Definition 10.1 for $D_x = X$ then the property of the input-output stability is satisfied for any disturbance $\overline{u}_n$ converging to $\overline{0}$. The inverse theorem is true if the system is fully controllable and observable, i.e. the disturbance $\overline{u}_n$ can remove the system from the equilibrium state and it can be observed by measuring $\overline{y}_n$.

In the next text, speaking about *input-output stability* we shall use the term *stability* only, which means that we shall speak only about the stability of the controllable and observable part of the system, or that we shall assume full controllability and observability of the system as a whole. Let us assume that until the moment $n = 0$ the control system was in the equilibrium state (the control error $\varepsilon_n = \overline{0}$ for $n < 0$) and in the moment $n = 0$ a step disturbance $z_n = \overline{z} \cdot \mathbf{1}(n)$ was put at the plant (which means that $z_n = \text{const} = \overline{z}$ for $n \geq 0$) and/or the required value of the plant output changed: $y_n^* = \overline{y}^* \cdot \mathbf{1}(n)$ ($y_n^* = \overline{0}$ for $n < 0$). The stability of the control system (exactly speaking, input-output stability) means that as a result of the action of the controller, the control error $\varepsilon_n$ converges to a constant value $\varepsilon_\infty$, in particular to $\overline{0}$.

For the static plant described by $y = \Phi(u)$ it is not possible to satisfy the requirement $y = y^*$ by putting directly the proper decision $u^*$ at the input as it was presented in the case of the full information on the plant in Sect. 3.1, but it is possible to achieve the value $u^*$ as a result of *step by step* approximation process in a stable closed-loop control system with the control algorithm (10.2). Similarly, for the system (10.3) it is not possible to achieve a required state of the plant in a finite time as it was described in Chap. 3 in the case of the full information on the plant, but it is possible to achieve it for $n \to \infty$, and practically after a sufficiently long time, such that after this time the distance $\|x_n - x^*\|$ is already less than the given small number.

Analogous terms and definitions are applied to a continuous dynamical closed-loop control system described by the equations

$$\left.\begin{aligned} \dot{x}_{\mathrm{O}}(t) &= f_{\mathrm{O}}[x_{\mathrm{O}}(t), u(t)], \qquad & y(t) &= \eta_{\mathrm{O}}[x_{\mathrm{O}}(t)], \\ \dot{x}_{\mathrm{R}}(t) &= f_{\mathrm{R}}[x_{\mathrm{R}}(t), y(t)], \qquad & u(t) &= \eta_{\mathrm{R}}[x_{\mathrm{R}}(t)], \end{aligned}\right\} \tag{10.5}$$

i.e.

$$\dot{x}(t) = F[x(t)] \tag{10.6}$$

where

$$x(t) = \begin{bmatrix} x_{\mathrm{O}}(t) \\ x_{\mathrm{R}}(t) \end{bmatrix}$$

is the state of the control system and the equilibrium state $x = x_e$ is a solution of the equation $F(x) = \bar{0}$. In the stable control system, the control error $\varepsilon(t)$ caused by a step disturbance converges to a constant value $\varepsilon_\infty$, in particular to $\bar{0}$. The designing of the closed-loop control system with the plant containing an unknown parameter $c$ and with an assumed form of the controller containing a parameter $a$ consists in the choice of the value $a$ such that for every value $c$ from the set of all possible values, the system is stable. The requirement of the system stability replaces now the requirement of the minimization of the performance index considered in Chap. 5 which is possible to satisfy with the full information on the plant, i.e. with the knowledge of $c$. It is a *parametric* design problem as well as an optimization problem considered in Chap. 5 but with a weaker requirement caused by the incomplete information on the plant. As a result, usually a solution of the problem is not unique, i.e. one obtains a set of values $a$ which together with the value $c$ satisfy the stability condition for the given forms of the plant model and the control algorithm. In order to determine this set, it is necessary to determine the stability condition concerning $(c, a)$.

## 10.2 Stability Conditions for Linear Stationary System

### 10.2.1 Continuous System

Let the system (10.6) be the linear and stationary system (i.e. with constant parameters)

$$\dot{x}(t) = Ax(t), \qquad x \in X = R^k. \tag{10.7}$$

For the given initial condition $x(0) = x_0$ one can find the concrete solution of the equation (10.7)

$$x(t) = e^{At}x_0. \tag{10.8}$$

By using (10.8), the following theorem may be proved:

**Theorem 10.1.** The system (10.7) is stable if and only if

$$\bigwedge_{i\in\overline{1,k}} \operatorname{Re} s_i < 0 \tag{10.9}$$

where $s_i$ are eigenvalues of the matrix $A$, i.e. the roots of the characteristic equation

$$\det(A - sI) = 0. \tag{10.10}$$

□

The property $x(t) \to \overline{0}$ for (10.8) does not depend on $x_0$, then (10.9) is the condition of the global stability for $D_x = X$. The equation (10.10) may be presented in the form

$$s^k + a_{k-1}s^{k-1} + ... + a_1 s + a_0 = 0. \tag{10.11}$$

Consequently, the system is stable if and only if the roots of the linear algebraic equation (10.11) are all located in the left half-plane of the $s$-plane. The condition may be proved without solving the equation (10.11), by applying so called Hurwitz criterion. For this purpose we consider the following $k$-th degree determinant:

$$\Delta_k = \begin{vmatrix} a_{k-1} & 1 & 0 & 0 & 0 & 0 & 0 & ... & 0 & 0 \\ a_{k-3} & a_{k-2} & a_{k-1} & a_k & 0 & 0 & 0 & ... & 0 & 0 \\ a_{k-5} & a_{k-4} & a_{k-3} & a_{k-2} & a_{k-1} & a_k & 0 & ... & 0 & 0 \\ ..... & & & & & & & & & \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & ... & 0 & a_0 \end{vmatrix}.$$

All the roots of the equation (10.11) lie in the left half-plane if and only if all principal subdeterminants (minors) $\Delta_1, \Delta_2, ..., \Delta_k$ with the form

$$\Delta_1 = a_{k-1}, \quad \Delta_2 = \begin{vmatrix} a_{k-1} & 1 \\ a_{k-3} & a_{k-2} \end{vmatrix}, \quad \Delta_3 = \begin{vmatrix} a_{k-1} & 1 & 0 \\ a_{k-3} & a_{k-2} & a_{k-1} \\ a_{k-5} & a_{k-4} & a_{k-3} \end{vmatrix},$$

etc., are positive. The stability analysis may be based on transmittances. For one-dimensional closed-loop control system (see (2.24))

$$E(s) = \frac{Y^*(s)}{1 + K(s)}$$

where the function $K(s) = K_R(s)K_O(s)$ is a transmittance of an open-loop control system. For $y^*(t) = y^* \cdot \mathbf{1}(t)$, the form of $\varepsilon(t)$ is determined by the characteristic roots of the closed-loop system, i.e. the roots of the characteristic equation

$$L(s) + M(s) = 0$$

where $L(s)$ and $M(s)$ denote the polynomials in the numerator and the denominator of the transmittance $K(s)$, respectively. The location of all characteristic roots of the closed-loop system in the left half-plane is a sufficient and necessary stability condition. It follows directly from the forms of components (addends) of the function $\varepsilon(t)$, corresponding to the real roots or to the pair of the complex roots (conjugate to each other) of the equation $L(s) + M(s) = 0$.

It is worth noting that the function $K(s)$ may be considered as a mapping transferring all roots of the equation $K(s) + 1 = 0$ from $s$-plane into one point (–1, j0) in $K(s)$-plane, and transferring the half-axis $j\omega$ for $0 < \omega < \infty$ into the graph of the frequency transmittance $K(j\omega)$. Then, instead to investigate the location of the roots of equation $K(s) + 1 = 0$ with respect to axis $j\omega$ in $s$-plane, one can investigate the location of the point (–1, j0) with respect to the graph of $K(j\omega)$ in $K(s)$-plane. Since $K(j\omega)$ is symmetric with respect to real number axis, i.e. $K(-j\omega)$ and $K(j\omega)$ are conjugate to each other, it is sufficient to investigate the location of the point (–1, j0) with respect to $K(j\omega)$ for $0 < \omega < \infty$. From these considerations so called *frequency stability criterion* or Nyquist criterion follows. In the simple case, under the assumption that all poles of $K(s)$ (i.e. roots of the equation $M(s) = 0$) lie in the left half-plane of $s$-plane, the control system is stable if and only if the graph of $K(j\omega)$ for $0 < \omega < \infty$ does not encircle the point (–1, j0).

### 10.2.2 Discrete System

Stability conditions for the discrete linear system are analogous to those for the continuous system. The solution of the equation

$$x_{n+1} = Ax_n \tag{10.12}$$

has the following form:

$$x_n = A^n x_0.$$

By using this solution, the following theorem may be proved:

**Theorem 10.2.** The system (10.12) is stable if and only if

$$\bigwedge_{i\in\overline{1,k}} |z_i| < 1 \tag{10.13}$$

where $z_i$ are the roots of the equation

$$\det(A - zI) = 0. \tag{10.14}$$

□

This is a global stability condition for $D_x = X$. After transformations the equation (10.14) takes the form (10.11) with the unknown $z$ instead of $s$. In order to determine the condition that all the roots of this equation satisfy the property (10.13), i.e. lie inside the circle with radius 1 in $z$-plane – one may apply the transformation

$$z = \frac{w+1}{w-1}$$

which transfers the left half-plane in $w$-plane into the circle with the radius 1 in $z$-plane. Substituting this expression in the place of $z$, after some transformations we obtain the $k$-th degree linear algebraic equation with the unknown $w$, for which we may apply Hurtwitz criterion.

For one-dimensional closed-loop system with the transmittance of the open-loop system $K(z) = K_R(z)K_O(z)$, the condition (10.13) concerns the roots of the equation

$$L(z) + M(z) = 0$$

where $L(z)$ and $M(z)$ denote the polynomials in the numerator and the denominator of $K(z)$, respectively. The function $K(z)$ transfers all roots of the equation $K(z) + 1 = 0$ from $z$-plane into one point $(-1, j0)$ in $K(z)$-plane, and the circle $e^{j\omega}$ for $-\pi < \omega < \pi$ into the graph of the discrete frequency transmittance $K(e^{j\omega})$. Thus, the discrete form of Nyquist criterion may be applied. In the simplest case, under the assumption that all roots of the equation $M(z) = 0$ lie inside the circle with radius 1, the control system is stable if and only if the graph of $K(e^{j\omega})$ does not encircle the point $(-1, j0)$.

**Example 10.1.** Let us determine the stability condition for the closed-loop control system with the following transmittances of the plant and of the controller

$$K_{\mathrm{O}}(s) = \frac{k_{\mathrm{O}}}{(sT_1 + 1)}, \qquad K_{\mathrm{R}}(s) = \frac{k_{\mathrm{R}}}{s(sT_2 + 1)}.$$

The characteristic equation of the closed-loop system is as follows:

$$k + s(sT_1 + 1)\,(sT_2 + 1) = T_1T_2s^3 + (T_1 + T_2)\mathrm{s}^2 + s + k = 0$$

where $k = k_{\mathrm{O}}k_{\mathrm{R}}$ is an amplification factor of the open-loop system. Then we have

$$a_2 = \frac{T_1 + T_2}{T_1T_2}, \quad a_1 = \frac{1}{T_1T_2}, \quad a_0 = \frac{k}{T_1T_2}.$$

Applying Hurwitz criterion, i.e. the inequalities $\Delta_1 > 0$, $\Delta_2 > 0$, $\Delta_3 > 0$, we obtain $a_2 > 0$, $a_1a_2 - a_0 > 0$, $a_0\Delta_2 > 0$ or $a_0 > 0$. Since $T_1, T_2 > 0$, the stability condition is the following:

$$0 < k < \frac{1}{T_1} + \frac{1}{T_2}. \tag{10.15}$$

The condition $k > 0$ is evident because it means that the feed-back must be negative. The right-hand side of (10.15) means that the amplification factor should be sufficiently small and that for too great values of $T_1$ and (or) $T_2$ the stability limit $k = \frac{1}{T_1} + \frac{1}{T_2}$ may be exceeded. Having the condition (10.15), for the given numerical data $k$, $T_1$, $T_2$, we can prove whether the system is stable.

The application of this condition in the designing of the controller consists in the proper choice of the values $k_{\mathrm{R}}$ and $T_2$ by a designer (see remarks in Sect. 10.1). If the exact values of the plant parameters are unknown but it is known that $k_{\mathrm{O}} \le k_{\mathrm{O,max}}$ and $T_1 \le T_{1\max}$, then one should choose such values $k_{\mathrm{R}}$ and $T_2$ that the condition

$$k_{\mathrm{R}} < \frac{1}{k_{\mathrm{O,max}}}\left(\frac{1}{T_{1\max}} + \frac{1}{T_2}\right)$$

is satisfied. The condition (10.15) can be also obtained by applying Nyquist criterion. For this purpose one should find the value $\overline{\omega}$ from the equation $\mathrm{Im}K(\mathrm{j}\omega) = 0$ and use the condition $\mathrm{Re}K(\mathrm{j}\,\overline{\omega}) > -1$, what is illustrated in Fig. 10.2. After some transformation we obtain

$$\mathrm{Re}K(\mathrm{j}\,\overline{\omega}) = \frac{-kT_1T_2}{T_1+T_2}$$

and consequently the condition (10.15). □

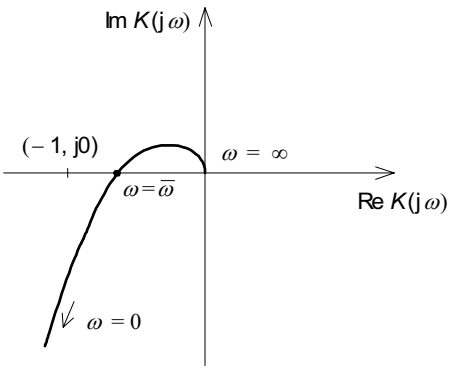


**Fig. 10.2.** Example of frequency transmittance

**Example 10.2.** Let us determine the stability condition for the discrete closed-loop control system in which the plant and the controller are described by the equations

$$y_{n+1} - \alpha y_n = k_{\mathrm{O}} u_{n+1}, \quad \varepsilon_{n+1} = \frac{1}{k_{\mathrm{R}}}(u_{n+1} - u_n),$$

respectively. The discrete transmittances are then as follows:

$$K_{\mathrm{O}}(z) = \frac{k_{\mathrm{O}} z}{z-\alpha}, \qquad K_{\mathrm{R}}(z) = \frac{k_{\mathrm{R}} z}{z-1}.$$

The equation $L(z) + M(z) = 0$ is here the following:

$$z^2 + a_1 z + a_0 = 0 \tag{10.16}$$

where

$$a_1 = -\frac{\alpha+1}{k+1}, \quad a_0 = \frac{\alpha}{k+1}.$$

Substituting $z = \dfrac{w+1}{w-1}$ we obtain

$$(1 + a_1 + a_0)w^2 + 2(1 - a_0)w + 1 - a_1 + a_0 = 0.$$

For the 2-nd degree equation, the conditions in Hurwitz criterion are satisfied if the coefficients in this equation are positive. Hence,

$$a_0 < 1, \quad a_0 > a_1 - 1, \quad a_0 > -a_1 - 1 \tag{10.17}$$

and after some transformations it gives the following stability condition:

$$k > \max\{0, \alpha - 1, -2(\alpha + 1)\}.$$

Consequently, $k > 0$ if $|\alpha| < 1$ (the plant is stable), $k > -2(\alpha + 1)$ if $\alpha \le -1$, and $k > \alpha - 1$ if $\alpha \ge 1$. For example, if it is known that $1 \le \alpha \le 7$ and $k_O \ge 3$ then the designer should choose $k_R > 2$. □

## 10.3 Stability of Non-linear and Non-stationary Discrete Systems

A general method of the determination of sufficient stability conditions for non-linear and non-stationary (time-varying) discrete systems is based on so called *principle of contraction mapping* [7, 10, 12]. The function $F$ in the formula (10.4) is called a *contraction* mapping if for any two vectors $x$, $\bar{x} \in X$

$$\| F(\bar{x}) - F(x) \| < \| \bar{x} - x \|$$

and $\| \cdot \|$ denotes a norm of the vector. If $F(x)$ is a contraction mapping then the equation $x = F(x)$ has one and only one solution (so called fixed point of the mapping $F$) equal to the limit of the recursive sequence (10.4). Let us consider the non-linear and non-stationary system

$$x_{n+1} = F(c_n, x_n) \tag{10.18}$$

where $c_n \in C$ is a vector of time-varying parameters. If the system is stationary then the equation (10.18) is reduced to (10.4). Let us present (10.18) in the form

$$x_{n+1}^{(i)} = F_i(c_n, x_n), \quad i = 1, 2, ..., k$$

and assume that the functions $F_i$ have the following form:

$$F_i(c_n, x_n) = \sum_{j=1}^{k} a_{ij}(c_n, x_n)\, x_n^{(j)},$$

i.e.

$$x_{n+1} = A(c_n , x_n )\, x_n \tag{10.19}$$

where the matrix $A(c_n, x_n) = [a_{ij}(c_n, x_n)] \in R^{k\times k}$. According to the earlier assumption about the solution of the equation $x = F(x)$, for every $c \in C$ the equation $x = A(c, x)x$ has the unique solution $x_e = \overline{0}$ (the equilibrium point). For the linear system

$$x_{n+1} = A(c_n)x_n$$

and for the stationary system

$$x_{n+1} = A(x_n)x_n.$$

It is convenient to formulate the principle of contraction mapping for $F(x) = Ax$ by using a norm of the matrix. A norm of the matrix $\| A \|$ for the determined norm of the vector $\| x \|$ is defined as follows:

$$\| A \| = \max_{x\in\Delta_x} \frac{\| Ax \|}{d}, \quad \Delta_x = \{x \in X : \| x \| = d\}. \tag{10.20}$$

Hence, it is the maximum ratio of the length of the vector $Ax$ to the length of the vector $x$, for different vectors $x$ with the same length. The following norms are most frequently used:

1. $\| x \| = \sqrt{x^{\mathrm{T}} x}$ (*Euclidean norm*)

$$\| A \| \stackrel{\Delta}{=} \| A \|_2 = \sqrt{\lambda_{\max}(A^{\mathrm{T}} A)} \tag{10.21}$$

where $\lambda_{\max}$ is the maximum eigenvalue of the matrix $A^{\mathrm{T}}A$.

2. If

$$\| x \| = \max_{1\le i\le k} | x^{(i)} | \tag{10.22}$$

or

$$\| x \| = \sum_{i=1}^{k} | x^{(i)} | \tag{10.23}$$

then

$$\| A \| \stackrel{\Delta}{=} \| A \|_1 = \max_{1\le i\le k} \sum_{j=1}^{k} | a_{ij} |, \tag{10.24}$$

$$\| A \| \stackrel{\Delta}{=} \| A \|_\infty = \max_{1\le j\le k} \sum_{i=1}^{k} | a_{ij} |, \tag{10.25}$$

respectively.

The following theorems are based on [7, 12].

**Theorem 10.3.** If there exists a norm $\|\cdot\|$ such that

$$\bigwedge_{n\geq 0}\bigwedge_{x\in X} \| A(c_n, x)\| < 1 \tag{10.26}$$

then the system (10.19) is globally stable for $D_x = X$, i.e. is totally stable. □

**Theorem 10.4.** If there exists a norm $\|\cdot\|$ and a non-singular matrix $P \in R^{k\times k}$ such that

$$\bigwedge_{n\geq 0}\bigwedge_{x\in X} \| P^{-1} A(c_n, x) P\| < 1 \tag{10.27}$$

then the system (10.19) is totally stable. □

**Theorem 10.5.** Denote by $\lambda_i(A) = \lambda_i(c_n, x)$ the eigenvalues of the matrix $A$ ($i$ = 1, 2, ..., $k$). If $A(c_n, x)$ is a symmetric matrix and

$$\bigwedge_{n\geq 0}\bigwedge_{x\in X} \max_i \left|\lambda_i(c_n, x)\right| < 1 \tag{10.28}$$

then the system (10.19) is totally stable. □

Theorem 10.3 follows from the fact that under the assumption (10.26) $A(c_n, x)x$ is a contraction mapping in $X$. The condition (10.27) is obtained by introducing the new state vector $v_n = P^{-1}x_n$ and using Theorem 10.3 for the equation

$$v_{n+1} = P^{-1} A(c_n, x_n) P v_n .$$

Theorem 10.5 may be easily proved (see [12]) by using the norm (10.21). If $A$ is a symmetric matrix then $\lambda_{\max}(A^{\mathrm{T}} A) = \max\limits_i |\lambda_i(A)|^2$.

**Remark 10.1.** For the norm $\|\cdot\|_2$ and $(P^{-1})^{\mathrm{T}} P^{-1} \triangleq Q$, the condition (10.27) is reduced to the following statement. If there exists a positive definite matrix $Q$ such that

$$\bigwedge_{n\geq 0}\bigwedge_{x\in X} A^{\mathrm{T}}(c_n,x)QA(c_n,x)-Q<0$$

then the system (10.19) is totally stable. □

Condition (10.26) may be presented in the form

$$\bigwedge_{n\geq 0}\bigwedge_{x\in X} A(c_n,x)\in \mathcal{A} \tag{10.29}$$

where $\mathcal{A}$ is a set of $k\times k$ matrices $\hat{A}$ defined as

$$\mathcal{A}=\{\hat{A}:\|\hat{A}\|<1\}\,. \tag{10.30}$$

We shall also use another form of the stability condition (10.26)

$$\bigwedge_{n\geq 0} c_n\in \overline{D}_c \tag{10.31}$$

where

$$\overline{D}_c=\{c\in C:\ \bigwedge_{x\in X}\ \|A(c,x)\|<1\}\,. \tag{10.32}$$

Conditions (10.27) and (10.28) may be presented in an analogous form with $\|P^{-1}\hat{A}P\|$ or $\max_i|\lambda_i(\hat{A})|$ instead of $\|\hat{A}\|$ in (10.30) and with $\|P^{-1}A(c,x)P\|$ or $\max_i|\lambda_i(c,x)|$ instead of $\|A(c,x)\|$ in (10.32).

Theorem 10.5 shows that if $A(c_n,x)$ is a symmetric matrix then for the non-linear and non-stationary system one may apply the condition such as for a linear and stationary system.

Let us note that Theorems 10.3, 10.4, 10.5 formulate sufficient stability conditions only. The satisfaction of these conditions assures a monotonic convergence of $\|x_n\|$ to 0, which is not necessary for the stability. When the condition (10.26) will not be satisfied, we do not know whether the system is stable. For the different norms, different particular sufficient conditions (10.26) may be obtained, and by a proper choice of the matrix $P$ one can try to obtain a weaker condition (10.26). We can use the basic condition (10.26) in two ways:

1. We try to determine the total stability condition for the parameters of the system, and consequently – for the control parameters $a$. If it is not possible to choose the value $a$ as to satisfy the condition (10.26), we try to determine the domain of global stability $D_x$ for the fixed $a$.

2. We determine the global stability domain $D_x$, i.e. such a set $D_x$ containing the equilibrium state $\overline{0}$ that if $x_0 \in D_x$ then $x_n \to \overline{0}$. Let us note that if $x_N \in \overline{D}_x$ where

$$\overline{D}_x = \{x \in X: \bigwedge_{n \geq N} \| A(c_n, x) \| < 1 \}$$

then $\| x_{N+1} \| < \| x_N \|$. On the other hand, if $x_N \in D_x$ then $x_n$ converges to $\overline{0}$ for $n > N$. The set $D_x$ is then the maximum domain determined by the inequality $\| x \| \leq d$ and contained in the domain $\overline{D}_x$, i.e.

$$D_x = \{x \in X: \ \| x \| \leq d\}$$

for the maximum $d$ such that

$$\bigwedge_{\| x \| \leq d} (x \in \overline{D}_x).$$

The convergence problem is more complicated when the output of the plant in a closed-loop system is measured with a random noise $z_n$. In such a case one can apply so called *stochastic approximation* algorithm which for the static plant $y = \Phi(u)$ considered in Sect. 10.1 takes the form

$$u_{n+1} = u_n + \gamma_n(y^* - \overline{y}_n)$$

where $\overline{y}_n = y_n + z_n$ is the result of the output measurement. Under some very general assumptions concerning the function $\Phi$ and the noise $z_n$, usually satisfied in practice – it can be proved that such a process in a probabilistic sense (see Sec. 7.1) converges to $u^*$, i.e. to the solution of the equation $\Phi(u) = y^*$, if $\gamma_n > 0$ for every $n$, the sequences $\gamma_n$ converges to 0 and satisfies the conditions

$$\sum_{n=0}^{\infty} \gamma_n = \infty, \qquad \sum_{n=0}^{\infty} \gamma_n^2 < \infty.$$

In order to assure the convergence of the approximation process so called *degressive feed-back* [9, 11] should be applied, i.e. a feed-back acting weaker and weaker (with less and less $\gamma_n$) for increasing $n$. The conditions presented above are satisfied by the sequence $\gamma_n = \frac{\gamma}{n}$. The stochastic approximation is widely applied in approximation processes for control and

identification as well as adaptation and learning which will be described in Chap. 11. More precise information on the stochastic approximation and its applications may be found in [14, 103].

**Example 10.3.** Let us consider a one-dimensional feed-back control system with a continuous plant consisting of a non-linear static part described by the function $w = \Phi(u)$ and the linear dynamical part described by the transmittance

$$K_{\mathrm{O}}(s) = \frac{k_{\mathrm{O}}}{s(s+1)}.$$

The plant is controlled in a discrete way via zero-order hold (Fig. 10.3), $u(t) = k_{\mathrm{R}}\varepsilon(t)$, $u_n = u(nT)$ where $T$ is a sampling period (see the description of a continuous plant controlled in a discrete way, presented in Chap. 2). It is easy to show that, choosing the state variables $x_n^{(1)} = y(nT) = -\varepsilon(nT)$, $x_n^{(2)} = \dot{y}(nT)$, one obtains the following equation:

$$\begin{bmatrix} x_{n+1}^{(1)} \\ x_{n+1}^{(2)} \end{bmatrix} = \begin{bmatrix} 1 - g(-x_n^{(1)})(T - 1 + e^{-T}) & (1 - e^{-T}) \\ -g(-x_n^{(1)})(1 - e^{-T}) & e^{-T} \end{bmatrix} \begin{bmatrix} x_n^{(1)} \\ x_n^{(2)} \end{bmatrix}$$

where

$$\frac{1}{k_{\mathrm{O}}k_{\mathrm{R}}}\, g(-x^{(1)}) = g(u) = \begin{cases} \dfrac{\Phi(u)}{u} & \text{for} \quad u \neq 0 \\ \lim\limits_{u \to 0} \dfrac{\Phi(u)}{u} & \text{for} \quad u = 0. \end{cases}$$

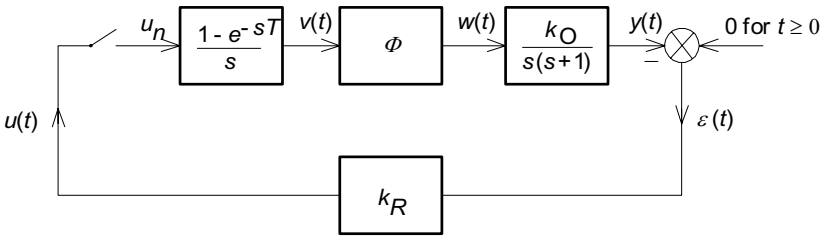


**Fig. 10.3.** Block scheme of the control system under consideration

Applying the condition (10.26) with the norm (10.24) yields

$$|\, 1 - k_{\mathrm{O}}k_{\mathrm{R}}\, g(u)(T - 1 + e^{-T})| + (1 - \mathrm{e}^{-T}) < 1,$$

$$| k_{\mathrm{O}} k_{\mathrm{R}}\, g(u)(1 - e^{-T}) | + \mathrm{e}^{-T} < 1$$

and finally

$$\left.\begin{array}{c} \dfrac{1-e^{-T}}{k_{\mathrm{O}} k_{\mathrm{R}}\,(T-1+e^{-T})} < g(u) < \dfrac{1+e^{-T}}{k_{\mathrm{O}} k_{\mathrm{R}}\,(T-1+e^{-T})}, \\ -1 < g(u) < 1, \end{array}\right\} \qquad (10.33)$$

under the assumption that $T - 1 + e^{-T} > 0$. The inequalities (10.33) determine the bounds $g_1$ and $g_2$ for $g(u)$. If $g_1 \le g(u) \le g_2$ for every $u$, i.e. the characteristic $w = \Phi(u)$ lies between the lines $w = g_1 u$ and $w = g_2 u$ (Fig. 10.4) then the system is totally stable. Sometimes in this case we use the term *absolute stability condition*, i.e. the condition concerning the whole set of non-linear characteristics. If the given characteristic $w = \Phi(u)$ is located between the lines mentioned and it is known that $k_{\mathrm{O,min}} \le k_{\mathrm{O}} \le k_{\mathrm{O,max}}$ then the choice of $k_{\mathrm{R}}$ satisfying the condition

$$\frac{1+e^{-T}}{g_1 k_{\mathrm{O,min}}\,(T-1+e^{-T})} < k_{\mathrm{R}} < \frac{1+e^{-T}}{g_2 k_{\mathrm{O,max}}\,(T-1+e^{-T})}$$

assures the stability. It is also a condition for a non-stationary system under the assumption that for every $n \ge 0$

$$k_{\mathrm{O,min}} \le k_{\mathrm{O},n} \le k_{\mathrm{O,max}}. \qquad \square$$

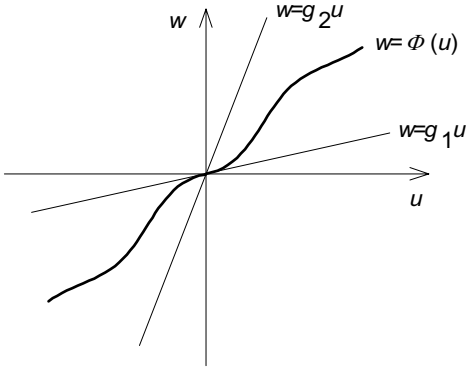


**Fig. 10.4.** Characteristic of static non-linear element

## 10.4 Stability of Non-linear and Non-stationary Continuous Systems

Let us assume that the continuous system

$$\dot{x} = F[c(t), x(t)] = A[c(t), x(t)]x(t) \tag{10.34}$$

has one equilibrium state $x_e = \overline{0}$, i.e. for every $c \in C$ the equation $A(c, x)x = \overline{0}$ has the unique solution $x_e = \overline{0}$. The considerations are now analogous to those for a discrete case. The requirement $\| x_{n+1} \| < \| x_n \|$ may be replaced by the requirement for $\| x(t) \|$ to be a decreasing function of $t$. If it is a differentiable function then we may determine the condition assuring for every $t > 0$ the inequality

$$\frac{d \| x(t) \|}{dt} < 0 , \tag{10.35}$$

i.e.

$$[\operatorname{grad}_x \| x \| |_{x=x(t)}]^{\mathrm{T}} \cdot \dot{x}(t) = [\operatorname{grad}_x \| x \| |_{x=x(t)}]^{\mathrm{T}} A[c(t), x(t)]x(t) < 0 . \tag{10.36}$$

The requirement that $\| x(t) \|$ is a differentiable function restricts the choice of the norm $\| x \|$. It is however worth noting that instead of a norm one may use any function $V(x)$ assigning non-negative real number to a vector $x$ and satisfying the following properties:

1. $V(x) = 0 \Leftrightarrow x = \overline{0}$.

2. For every sequence $x_n$ such that $\| x_n \| \to \infty$, the sequence $V(x_n) \to \infty$.

Of course, $\| x \|$ is a function $V(x)$ for any norm. Now the inequalities (10.35) and (10.36) take the forms

$$\frac{dV[x(t)]}{dt} < 0 , \tag{10.37}$$

$$[\operatorname{grad}_x V(x)]^{\mathrm{T}} A[c(t), x]x < 0 \tag{10.38}$$

under the assumption that the function $V(x)$ is differentiable with respect to $x$. If for every $t$ the condition (10.37) is satisfied then for $t \to \infty$ the value $V$ converges to 0, and consequently $x(t) \to \overline{0}$. Then the following theorem, analogous to Theorem 10.3, is true.

**Theorem 10.6.** If there exists a function $V(x)$ such that the inequality

(10.38) is satisfied for every $t > 0$ and every $x \in X$ then the system (10.34) is globally stable for $D_x = X$. □

Consequently, theorems analogous to Theorems 10.4 and 10.5 are also true. In the first case, in the place of $A$ in (10.38) $P^{-1}AP$ occurs, and in the second case the property $|\lambda_i(c_n, x)| < 1$ is replaced by the inequality $\mathrm{Re}\lambda_i[c(t), x] < 0$, i.e. the condition is the same as for a linear stationary system. The function $V(x)$ satisfying for every $t$ the condition (10.37) is called a *Lyapunov function*, and the respective method of the stability analysis is called the *second Lyapunov method* (the first Lyapunov method concerns the investigation of a local stability, based on a linear approximation). The approach described in Sect. 10.3 may be called the second *Lyapunov method in a discrete form*, with the function $V(x)$ equal to $\| x \|$ or $(\| x \|)^2$.

If $V(x) = x^{\mathrm{T}}Qx$ where $Q$ is a positive definite matrix then

$$[\operatorname{grad}_x V(x)]^{\mathrm{T}} Ax = 2x^{\mathrm{T}}QAx = x^{\mathrm{T}}(QA + A^{\mathrm{T}}Q)x .$$

Thus, if there exists a positive definite matrix $Q$ such that for every $t > 0$ and every $x \in X$ the matrix

$$QA[c(t), x] + A^{\mathrm{T}}[c(t), x]Q \tag{10.39}$$

is negative definite then, according to Theorem 10.6 the system is globally stable for $D_x = X$. This condition is analogous to the condition in Remark 10.1 for the discrete case.

## 10.5 Special Case. Describing Function Method

Let us consider a special case of a closed-loop control system, namely one-dimensional system containing two parts: a non-linear static part with the characteristic $v = \Phi(\varepsilon)$ and a linear dynamical part described by the transmittance $K(s)$ (Fig. 10.5).

In order to apply the second Lyapunov method one can introduce the state vector in the form $x^{\mathrm{T}} = [\varepsilon, \dot{\varepsilon}, ..., \varepsilon^{(k-1)}]$ where $\varepsilon^{(k-1)}$ denotes the $(k-1)$-th derivative of $\varepsilon(t)$ and $k$ is the order of the plant. Frequently, as a Lyapunov function in this case one chooses

$$V(x) = x^{\mathrm{T}}Qx + \int_0^{x^{(1)}} \Phi(\varepsilon)d\varepsilon .$$

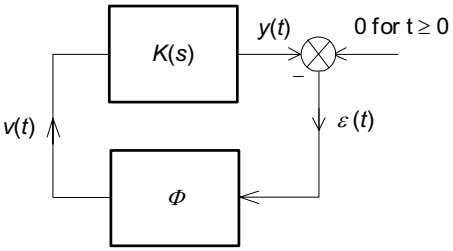


**Fig. 10.5.** Control system with static non-linear part

Then

$$\operatorname*{grad}_{x} V(x) = 2Qx + \alpha(x)$$

where $\alpha(x)$ is a zero vector except the first component equal to $\Phi(x^{(1)})$. The condition (10.38) takes now the form

$$x^{\mathrm{T}}[QA(c, x) + A^{\mathrm{T}}(c, x)Q]x + \Phi(x^{(1)})w_1(c, x)x < 0 \tag{10.40}$$

where $c$ denotes a parameter of the transmittance $K(s)$ and $w_1(x, c)$ denotes the first row of the matrix $A(c, x)$ which should be determined by transferring the initial description of the system into the description using the state vector. As a result we may obtain a condition concerning $\Phi(x^{(1)})$, i.e. $\Phi(\varepsilon)$.

The using of the condition (10.40) may be difficult in more complicated cases and may not give an effective result, i.e. the total stability condition. That is why in this case one often applies an approximate method consisting in a *harmonic linearization* and called a describing function method. Let us assume that the system under consideration is not stable and there occur oscillations in the system, i.e. $\varepsilon(t)$, $v(t)$, $y(t)$ are periodic functions. Assume that $\varepsilon(t)$ is approximately equal to $A\sin\omega t$, expand the function $v(t) = \Phi(A\sin\omega t)$ in Fourier series and take into account the first term (a fundamental harmonic) only

$$v(t) \approx v_1(t) = B\sin(\omega t + \varphi).$$

The approximation is acceptable if the linear dynamical part is a low pass filter (what usually occurs in practice) and the higher harmonics of $v(t)$ are much smaller than the fundamental one. If the signal $v_1(t)$ at the input of the part $K(s)$ is sinusoidal then the signal at the output $y(t) = -\varepsilon(t)$ is also sinusoidal with the amplitude $B|K(\mathrm{j}\omega)|$ and the phase $\varphi + \arg K(\mathrm{j}\omega)$. Then

$$B|K(\mathrm{j}\omega)|\sin[\omega t + \varphi + \arg K(\mathrm{j}\omega)] = -A\sin\omega t = A\sin(\omega t + \pi).$$

Consequently,

$$B|K(\mathrm{j}\omega)| = A, \quad \varphi + \arg K(\mathrm{j}\omega) = \pi,$$

what may be written in the form

$$K(\mathrm{j}\omega)J(A) = -1 \tag{10.41}$$

where

$$J(A) = \frac{B}{A} e^{\mathrm{j}\varphi} \tag{10.42}$$

is called a *describing function* of the element $\Phi$; $|J(A)|$ is a ratio of the amplitude of the first harmonic of the output signal to the amplitude of the sinusoidal input signal, and $\arg J(A) = \varphi$ is the phase of the output with respect to the input. If the relationship between $\varepsilon$ and $v$ is a function $v = \Phi(\varepsilon)$ (i.e. $\varepsilon$ uniquely defines $v$) then $\varphi = 0$. The phase $\varphi \neq 0$ if so called *hisieresis* occurs in the non-linear element, which means that the value $v$ depends not only on $\varepsilon$ but also on whether the fixed value $\varepsilon$ has been achieved by increasing or decreasing of $\varepsilon(t)$. The equality (10.41) defines the condition for the existence of oscillations in our system. In fact (10.41) contains two equations: for $|J(A)|$ and $\arg J(A)$ or for real and imaginary parts. From these equations one can find the approximate values of $\omega$ and $A$ for the oscillations.

For the linear static element $v = \Phi(\varepsilon) = k\varepsilon$ the describing function $J(A) = k$. Then the condition for the oscillations (10.41) takes the form $\overline{K}(\mathrm{j}\omega) + 1 = 0$ where $\overline{K}(\mathrm{j}\omega) = kK(\mathrm{j}\omega)$ is the frequency transmittance of the open-loop control system. This form corresponds to the stability limit (see Nyquist criterion in Sect. 10.2), i.e. the system is stable if $K(\mathrm{j}\omega)$ does not encircle the point $(-\frac{1}{k}, \mathrm{j}0)$. In the non-linear system, the point $-\frac{1}{k}$ is "expanded" to the curve $-\frac{1}{J(A)}$. In the linear system the amplitude of possible oscillations is not determined by a description and parameters of the system but depends on initial conditions. In the non-linear system the amplitude of the oscillations called a *limit cycle* depends on the description and parameters of the system. Substitution of $\Phi(\varepsilon)$ by $J(A)$ may be treated as a kind of a linearization of the non-linear element for the fixed $A$, consisting in omitting the higher harmonics.

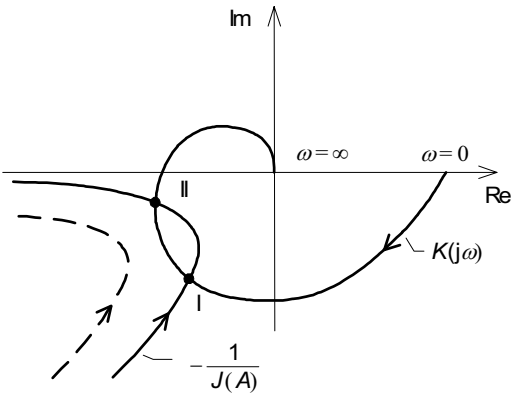


**Fig. 10.6.** Illustration of describing function method

Let us consider the graphs of $K(\mathrm{j}\omega)$ and $-\dfrac{1}{J(A)}$ presented in Fig. 10.6 where the arrow in the second curve indicates increasing of $A$, i.e. if $A$ is increasing then the respective point is moving in the direction indicated by the arrow. As it is illustrated in the figure, the condition (10.41) may be satisfied for two pairs ($A$, $\omega$), i.e. two limit cycles corresponding to the intersection points I and II. It can be proved that only one of these cycles is stable, i.e. may exist after disappearing of a disturbance removing the system from this cycle. In an approximate way it may be explained as follows: If in the state (oscillation regime) II a transit disappearing disturbance causes a small increase of the amplitude, the point of the curve $-\dfrac{1}{J(A)}$ will not be encircled by the graph of $K(\mathrm{j}\omega)$, then the system will be stable, the amplitude of the oscillations will decrease and the oscillations will return to the limit cycle II. The similar return will occur in the case of a transit decrease of the amplitude because the point of the curve $-\dfrac{1}{J(A)}$ will be encircled by the graph of $K(\mathrm{j}\omega)$ and the amplitude will increase after disappearing of a disturbance. If in the state (oscillation regime) I a transit disappearing disturbance causes a small increase of the amplitude, the point of the curve $-\dfrac{1}{J(A)}$ will be encircled by the graph of $K(\mathrm{j}\omega)$, then the system will be unstable, the amplitude of the oscillations will increase and the system will remove to the state II. If the disturbance causes a small decrease of the amplitude, the amplitude will continue to decrease and the oscillations will disappear.

The situation may be summarized as follows:
1. The system is stable for small disturbances (such that their effect, i.e. the initial state for the further process, is sufficiently near to the equilibrium state), and is unstable for the greater disturbances. Then the limit oscillations corresponding to the point II will occur in the system.
2. The system is totally stable if the graph of $K(\mathrm{j}\omega)$ does not encircle the curve $-\dfrac{1}{J(A)}$ (the case indicated by the broken line in Fig. 10.6).

As an illustration of the describing function method let us analyze the stability of *three-position control system* in which the signal $v$ may take three values only: $v = D$ for $\varepsilon > M$, $v = -D$ for $\varepsilon < -M$, $v = 0$ for $|\varepsilon| \le M$. The values of $J(A)$ are now real positive, the points of the curve $-\dfrac{1}{J(A)}$ lie in the negative real half-axis, the function $J(A) = 0$ for $A \le M$, then it increases and after taking the maximum converges to 0 for $A \to \infty$. To determine the stability condition it is sufficient to find the maximum of this function. After finding the amplitude of the first harmonic one can determine

$$\max_A J(A) = \frac{2D}{\pi M}.$$

Thus, the system is totally stable if the point

$$-\frac{1}{\max\limits_A J(A)}$$

lies on the left from the point in which the graph $K(\mathrm{j}\omega)$ intersects the imaginary axis, i.e.

$$\frac{\pi M}{2D} > -\mathrm{Re}\,K(\mathrm{j}\overline{\omega})$$

and $\overline{\omega}$ can be find by solving the equation $\mathrm{Im}K(\mathrm{j}\omega) = 0$.

## 10.6 Stability of Uncertain Systems. Robustness

Let us recall that according to the concept described in Sect.10.1, stability conditions are defined in order to choose parameters of the control algorithm (the controller) assuring stability of the system for every plant from a determined set of plants including the plant considered. In this case, the determination of the set of possible plants means the description of an un-

certainty. The choice of a suitable control algorithm assuring the stability for every plant from the determined set means the designing of a stable control system for an uncertain plant or a *stabilization* of the system. Usually the considered design task is a parametric problem, i.e. one should determine suitable parameters $a$ in a given form of the control algorithm. As a rule, one does not obtain a unique value $a$ but a set of possible values such that for every $a$ from this set and every plant from the set of possible plants the system is stable. In general, the description of the uncertainty may have a non-parametric form of the set of possible plants, or a parametric form of the set of possible values of a plant parameter $c$ in a given form of the plant model.

Usually, the feature consisting in satisfying by a system a certain property for a fixed set of its elements (and in the parametric formulation – for a fixed set of values of parameters) is called a *robustness*, and the system is called *robust*. The main idea of a robust system design is as follows: A designer wants to design the system satisfying a determined property $W$ (e.g. stability, controllability, observability). The satisfaction of this property depends on an existing system parameter $c \in C$ and on a parameter $a \in A$ which is to be chosen by the designer. The sufficient condition of this property is formulated in the form of a set $D_{a,c} \subset A \times C$:

$$(a, c) \in D_{a,c} \rightarrow W.$$

The designer knows the set $D_c$ of all possible values of $c$ (i.e. the description of the uncertainty in this case). Then the designer should determine the largest set $D_a \subset A$ such that for every $a \in D_a$ and every $c \in D_c$ the sufficient condition $(a, c) \in D_{a,c}$ is satisfied. Hence,

$$D_a = \{a \in A: \bigwedge_{c \in D_c} [(a, c) \in D_{a,c}]\}.$$

Such a procedure can be applied in the task of designing the stable system with the uncertain plant, considered in this section. The uncertainty may concern the function $A(c,x)$ and the sequence $c_n$. In general, it may be formulated as

$$\bigwedge_{n \geq 0} \bigwedge_{x \in X} A(c_n, x) \in \mathcal{A}_u \tag{10.43}$$

where $\mathcal{A}_u$ is a given set of the matrices $\hat{A} \in R^{k \times k}$. Then the general condition of the total stability for the uncertain system corresponding to

(10.26) is $\mathcal{A}_u \subseteq \mathcal{A}$ and may be expressed in the following way.

**Theorem 10.7.** If $\| A \| < 1$ for every $A \in \mathcal{A}_u$ then the system is globally stable for $D_x = X$. □

When the function $A(c,x)$ is known and the uncertainty concerns only the sequence $c_n$, it may be formulated as

$$\bigwedge_{n \geq 0} c_n \in D_c \tag{10.44}$$

where $D_c \subset C$ is a given subset of $C$.

**Theorem 10.8.** If $D_c \subseteq \overline{D}_c$ where $\overline{D}_c$ is determined by (10.32), i.e. if

$$\bigwedge_{c \in D_c} \bigwedge_{x \in X} \| A(c,x) \| < 1 \tag{10.45}$$

then the system is globally stable for $D_x = X$. □

The theorem follows immediately from (10.31), (10.32) and (10.44). The conditions corresponding to (10.27) and (10.28) have the analogous form.

For simplicity let us denote $A(c_n, x_n)$ by $A_n$. In the case of an *additive uncertainty* $A_n = A + \overline{A}_n$, i.e.

$$x_{n+1} = (A + \overline{A}_n) x_n . \tag{10.46}$$

The uncertainty concerns the matrix $\overline{A}_n$ and is formulated by one of the three forms denoted by (10.47), (10.48) and (10.49):

$$\bigwedge_{n \geq 0} \bigwedge_{x \in X} \overline{A}_n^+ \leq A_M \tag{10.47}$$

where $A_M$ is a given non-negative matrix (i.e. all entries of $A_M$ are non-negative) and $\overline{A}_n^+$ is the matrix obtained by replacing the entries of $\overline{A}_n$ by their absolute values,

$$\bigwedge_{n \geq 0} \bigwedge_{x \in X} \| \overline{A}_n \| \leq \beta \ , \tag{10.48}$$

$$\bigwedge_{n\geq 0}\bigwedge_{x\in X}\max_{i}\ |\lambda_i(\overline{A}_n)|\leq\overline{\beta} \tag{10.49}$$

where $\beta$ and $\overline{\beta}$ are given positive numbers. The inequalities (10.47), (10.48) and (10.49) define the set $\mathcal{A}_u$ in (10.43) for the cases under consideration.

**Lemma 1.** If $A$ and $B$ are quadratic matrices with non-negative entries (some of the entries are positive) and $A\geq B$ (i.e. $a_{ij}\geq b_{ij}$ for each $i$ and $j$) then $\|A\|\geq\|B\|$ for the norm (10.21), (10.24) and (10.25).

*Proof:* For (10.24) and (10.25) the lemma follows immediately from the definition of the norm. Denote

$$\overline{x}\stackrel{\Delta}{=}\arg\max_{x\in X}\ (\|Ax\|_2:\|x\|_2=1)$$

i.e.,

$$\|A\|_2=\|A\overline{x}\|_2 \tag{10.50}$$

where

$$\|Ax\|_2^2=\sum_{i=1}^{k}(a_{i1}x^{(1)}+a_{i2}x^{(2)}+\ldots+a_{ik}x^{(k)})^2=\sum_{i=1}^{k}\sum_{l=1}^{k}\sum_{j=1}^{k}a_{il}a_{ij}x^{(l)}x^{(j)}\ . \tag{10.51}$$

Suppose that there exist $l$ and $j$ such that $\overline{x}^{(l)}>0$ and $\overline{x}^{(j)}<0$. Then from (10.51) under the assumption about the entries of $A$

$$\|A\hat{x}\|_2>\|A\overline{x}\|_2 \tag{10.52}$$

where $\hat{x}^{(i)}=\overline{x}^{(i)}$ for $i\neq j$ and $\hat{x}^{(i)}=-\overline{x}^{(i)}$ for $i=j$. From (10.52) we see that $\overline{x}$ does not maximize $\|Ax\|_2$. Hence

$$\bigwedge_{l,j}\ \overline{x}^{(l)}\ \overline{x}^{(j)}\geq 0 \tag{10.53}$$

and from (10.51) it follows that the norm (10.50) is an increasing function of its entries, which proves the lemma. □

**Theorem 10.9.** Assume that $A$ has distinct eigenvalues. Then the system (10.46) with the uncertainty (10.47) is globally stable for $D_x=X$ if

$$\alpha\ +\ \|(M^{-1})^{+}A_M M^{+}\|<1 \tag{10.54}$$

where $\|\cdot\|$ is one of the norms (10.21), (10.24), (10.25), $M$ is the modal matrix of $A$ (i.e. the columns of $M$ are the eigenvectors of $A$) and

$$\alpha = \max_i |\lambda_i(A)|.$$

*Proof:* Let us use Theorem 10.4 with $P = M$ and the equality $M^{-1}AM = \operatorname{diag}[\lambda_1(A), \lambda_2(A), ..., \lambda_k(A)]$. Then

$$\| M^{-1}(A + \overline{A}_n)M \| = \| M^{-1}AM + M^{-1}\overline{A}_n M \| \le \alpha + \| M^{-1}\overline{A}_n M \|. \quad (10.55)$$

It is easy to see that for any matrices $A$ and $B$

$$(AB)^+ \le A^+ B^+. \quad (10.56)$$

It is known that for any matrix $A$

$$\| A \|_2 \le \| A^+ \|_2. \quad (10.57)$$

For the norms (10.24) and (10.25) the equality $\| A \| = \| A^+ \|$ follows directly from the definitions of the norms. Then, using (10.55), (10.56), (10.57) (or the equality for the norms $\|\cdot\|_1$, $\|\cdot\|_\infty$), Lemma 1 and (10.47), we obtain

$$\| M^{-1}(A + \overline{A}_n)M \| \le \alpha + \| (M^{-1}\overline{A}_n M)^+ \| \le \alpha + \| (M^{-1})^+ A_M M^+ \|.$$

Finally, using Theorem 10.4 yields the desired result. □

The result (10.54) and the other conditions described in this section have been presented in [27, 38]. It has been shown that by using Theorem 10.4 based on the general principle of the contraction mapping it is possible to obtain a more general result than conditions presented earlier in the literature, for special cases of non-linear and time-varying systems.

**Corollary 1.** Assume that $A$ has distinct eigenvalues and all eigenvalues of $(M^{-1})^+ A_M M^+$ are real. Then the system (10.46) with the uncertainty (10.47) is globally stable for $D_x = X$ if

$$\alpha + \lambda_{\max}[(M^{-1})^+ A_M M^+] < 1 \quad (10.58)$$

where $\lambda_{\max}$ is the maximum eigenvalue of $(M^{-1})^+ A_M M^+$.

*Proof:* Let $N$ be a diagonal matrix with real positive entries. Then $MN$ is also the modal matrix of $A$. It is known that

$$\inf_{N} \| N^{-1}(M^{-1})^{+} A_M M^{+} N \|_2 = \lambda_{\max}[(M^{-1})^{+} A_M M^{+}]. \quad (10.59)$$

Condition (10.58) follows from (10.54) and (10.59). □

**Theorem 10.10.** If $A$ is a symmetric matrix and

$$\alpha + \| A_M \|_2 < 1 \quad (10.60)$$

then the system (10.46) with the uncertainty (10.47) is globally stable for $D_x = X$.

*Proof:* If $A$ is a symmetric matrix then

$$\| A \|_2 = \max_i \; |\lambda_i(A)| = \alpha. \quad (10.61)$$

Using (10.61), (10.57) and Lemma 1 we obtain

$$\| A + \overline{A}_n \|_2 \le \| A \|_2 + \| \overline{A}_n \|_2 \le \alpha + \| \overline{A}_n^{+} \|_2 \le \alpha + \| A_M \|_2 .$$

Finally, using Theorem 10.3 yields the desired result. □

**Theorem 10.11.** If there exists a non-singular matrix $P \in R^{k \times k}$ such that

$$\| (P^{-1})^{+}(A^{+} + A_M)P^{+} \| < 1 \quad (10.62)$$

where $\| \cdot \|$ is one of the norms (10.21), (10.24) and (10.25), then the system (10.46) with the uncertainty (10.47) is globally stable for $D_x = X$.

*Proof:* Using (10.57) (or the equality for the norms $\| \cdot \|_1$, $\| \cdot \|_\infty$), (10.56) and Lemma 1, we obtain

$$\| P^{-1}(A + \overline{A}_n)P \| \le \| (P^{-1})^{+}(A^{+} + \overline{A}_n^{+})P^{+} \| \le \| (P^{-1})^{+}(A^{+} + A_M)P^{+} \| .$$

Consequently, (10.62) implies the inequality $\| P^{-1}(A + \overline{A}_n)P \| < 1$ and according to Theorem 10.4 the system is globally stable for $D_x = X$. □

In particular we can apply diagonal positive matrix $P$. If $P = I$ (identity matrix) then (10.62) becomes

$$\| A^{+} + A_M \| < 1 . \quad (10.63)$$

Other theorems and more details concerning the stability of uncertain systems may be found in [27, 38, 52].

**Example 10.4.** Let in (10.46) and (10.47) $k = 2$,

$$A = \begin{bmatrix} a_{11}+b & 0 \\ a_{21}+b & a_{22} \end{bmatrix}, \qquad A_M = \begin{bmatrix} a_{M11} & a_{M12} \\ a_{M21} & a_{M22} \end{bmatrix},$$

$a_{11}$, $a_{21}$, $a_{22}$, $b > 0$.

Applying the condition (10.63) with the norm $\|\cdot\|_1$ yields

$$a_{11} + b + a_{M11} + a_{M12} < 1,$$

$$a_{21} + b + a_{M21} + a_{22} + a_{M22} < 1$$

and finally

$$b < 1 - \max\{(a_{11} + a_{M11} + a_{M12}),\ (a_{21} + a_{22} + a_{M21} + a_{M22})\}. \quad (10.64)$$

Let us now apply the condition (10.54) with the norm $\|\cdot\|_1$. We have $\lambda_1(A) = a_{11} + b$, $\lambda_2(A) = a_{22}$, $\lambda_{\max}(A) = \max(a_{11} + b, a_{22})$. It is easy to show that

$$M = \begin{bmatrix} 1 & 0 \\ s & 1 \end{bmatrix}$$

with

$$s = \frac{a_{21} + b}{a_{11} + b - a_{22}}$$

is a modal matrix of $A$ and

$$(M^{-1})^{+} A_M M^{+}$$

$$= \begin{bmatrix} a_{M11} + a_{M12} & a_{M12} \\ a_{M11}|s| + a_{M12}|s|^2 + a_{M21} + a_{M22}|s| & a_{M12}|s| + a_{M22} \end{bmatrix}. \quad (10.65)$$

Suppose that $a_{11} \geq a_{22}$, i.e. $\alpha = a_{11} + b$. Applying (10.54) we obtain

$$b \leq 1 - a_{11} - \max\{[a_{M11} + 2a_{M12}], [a_{M11}|s| + a_{M12}(|s|^2 + |s|) + a_{M21} + a_{M22}(|s| + 1)]\}. \quad (10.66)$$

Since $s$ depends on $b$, the final condition for $b$ may be very complicated. To show that the condition (10.66) may be more conservative than

(10.64) assume that $a_{21} = a_{11} - a_{22}$, i.e. $s$ =1. Then (10.64) and (10.66) become

$$b < 1 - a_{11} - \max \{a_{M11} + a_{M12},\ a_{M21} + a_{M22}\} , \quad (10.67)$$

$$b < 1 - a_{11} - (a_{M11} + 2a_{M12} + a_{M21} + 2a_{M22}) . \quad (10.68)$$

Let us now use condition (10.58). The eigenvalues of the matrix (10.65) are

$$\lambda_{1,2} = \frac{a_{M11} + a_{M12}(1+|s|) + a_{M22}}{2} \pm \frac{\sqrt{[a_{M11} + a_{M12}(1+|s|) + a_{M22}]^2 + 4e}}{2} \quad (10.69)$$

where

$$e = a_{M12}^2(|s|^2 - |s|) + a_{M12}a_{M21} + a_{M12}a_{M22}(|s| - 1) - a_{M11}a_{M22}$$

and condition (10.58) becomes

$$b < 1 - a_{11} - \lambda_{\max} \quad (10.70)$$

where $\lambda_{\max}$ is obtained by putting + in the numerator of (10.69). To compare it with (10.67) and (10.68) let us put $s$ =1. Then

$$e = a_{M12}a_{M21} - a_{M11}a_{M22} .$$

If $e \geq 0$ and $a_{M11} + 2a_{M12} > a_{M21}$ then

$$\lambda_{\max} \geq a_{M11} + 2a_{M12} + a_{M22} > a_{M21} + a_{M22}$$

and the condition (10.68) is more conservative than (10.67). The condition (10.68) may be more conservative than (10.70) but is easier to obtain. When $a_{M21} = a_{M22} = 0$ (10.68) and (10.70) give the same result.

For numerical data $a_{11} = 0.4$, $a_{21} = 0.3$, $a_{22} = 0.1$, $a_{M11} = 0.1$, $a_{M12} = 0.1$, $a_{M21} = 0.1$ and $a_{M22} = 0.05$ we obtain

from condition (10.63), i.e. from (10.64): $b < 0.4$,

from condition (10.54), i.e. from (10.66): $b < 0.1$,

from condition (10.58), i.e. from (10.70): $b < 0.24$.

For $a_{11} = 0.65$ from (10.64) we obtain $b < 0.15$; positive $b$ satisfying

condition (10.54) or condition (10.58) does not exist.

The obtained conditions for $b$ may be applied to different forms of the matrix $A(c_n, x_n)$. Let us list the typical cases.

1. Linear time-varying system

$$x_{n+1} = \begin{bmatrix} a_{11} + b + c_n^{(1)} & c_n^{(2)} \\ a_{21} + b + c_n^{(3)} & a_{22} + c_n^{(4)} \end{bmatrix} x_n$$

with the uncertainties

$$\bigwedge_{n \geq 0} [(|\, c_n^{(1)} \,| \leq a_{M11}) \wedge (|\, c_n^{(2)} \,| \leq a_{M12}) \wedge (|\, c_n^{(3)} \,| \leq a_{M21}) \wedge (|\, c_n^{(4)} \,| \leq a_{M22})].$$

Now $c_n^{\mathrm{T}} = [c_n^{(1)} \;\; c_n^{(2)} \;\; c_n^{(3)} \;\; c_n^{(4)}]$.

2. Non-linear system

$$x_{n+1}^{(1)} = (a_{11} + b) x_n^{(1)} + F_1^{(1)}(x_n^{(1)}) + F_2^{(1)}(x_n^{(2)}),$$

$$x_{n+1}^{(2)} = (a_{21} + b) x_n^{(1)} + a_{22} x_n^{(2)} + F_1^{(2)}(x_n^{(1)}) + F_2^{(2)}(x_n^{(2)})$$

with the uncertainties

$$\bigwedge_{-\infty < x^{(1)} < \infty} \left( \left| \frac{F_1^{(1)}(x^{(1)})}{x^{(1)}} \right| \leq a_{M11} \right) \wedge \left( \left| \frac{F_1^{(2)}(x^{(1)})}{x^{(1)}} \right| \leq a_{M21} \right), \qquad (10.71)$$

$$\bigwedge_{-\infty < x^{(2)} < \infty} \left( \left| \frac{F_2^{(1)}(x^{(2)})}{x^{(2)}} \right| \leq a_{M12} \right) \wedge \left( \left| \frac{F_2^{(2)}(x^{(2)})}{x^{(2)}} \right| \leq a_{M22} \right). \qquad (10.72)$$

For $x = 0$ one should put

$$\lim_{x \to 0} \frac{F(x)}{x}$$

under the assumption that the limit exists.

3. Non-linear time-varying system

$$x_{n+1}^{(1)} = (a_{11} + b) x_n^{(1)} + F_1^{(1)}(c_n^{(1)}, x_n^{(1)}) + F_2^{(1)}(c_n^{(2)}, x_n^{(2)}),$$

$$x_{n+1}^{(2)} = (a_{21} + b)x_n^{(1)} + a_{22}x_n^{(2)} + F_1^{(2)}(c_n^{(3)}, x_n^{(1)}) + F_2^{(2)}(c_n^{(4)}, x_n^{(2)}),$$

with the uncertainties analogous to the statements (10.71) and (10.72) which should be satisfied for every $n \geq 0$. For example

$$\bigwedge_{n \geq 0} \bigwedge_{-\infty < x^{(1)} < \infty} \left( \left| \frac{F_1^{(1)}(c_n^{(1)}, x^{(1)})}{x^{(1)}} \right| \leq a_{M11} \right) \qquad (10.73)$$

means that the function $F_1^{(1)}(c^{(1)}, x^{(1)})$ and the sequence $c_n^{(1)}$ are such that (10.73) is satisfied, e.g. the function

$$F_1^{(1)}(c^{(1)}, x^{(1)}) = \begin{cases} c^{(1)}[1 - \exp(-2x^{(1)})] & \text{for } x^{(1)} \geq 0 \\ c^{(1)}[\exp(2x^{(1)}) - 1] & \text{for } x^{(1)} < 0 \end{cases}$$

and the sequence $c_n^{(1)}$ such that

$$\bigwedge_{n \geq 0} |c_n^{(1)}| \leq \frac{1}{2} a_{M11}$$

satisfy the condition (10.73). For the function $F_1^{(1)}(c^{(1)}, x^{(1)}) = c^{(1)} \overline{F}_1(x^{(1)})$, if

$$\bigwedge_{n \geq 0} |c_n^{(1)}| \leq \gamma, \quad \bigwedge_{-\infty < x^{(1)} < \infty} \left| \frac{\overline{F}_1(x^{(1)})}{x^{(1)}} \right| \leq \delta$$

and $\gamma \cdot \delta = a_{M11}$ then the condition (10.73) is satisfied. □

## 10.7 An Approach Based on Random and Uncertain Variables

Consider a non-linear time-varying system described by

$$x_{n+1} = A(c_n, b, x_n)x_n \qquad (10.74)$$

where $x_n \in X$ is the state vector, $c_n \in C$ is the vector of time-varying parameters, $b \in B$ is the vector of constant parameters; $X = R^k$, $C$ and $B$ are real number vector spaces. The matrix

$$A(c_n, b, x_n) = [a_{ij}(c_n, b, x_n)] \in R^{k \times k}.$$

Assume that for every $c \in C$ and $b \in B$ the equation $x = A(c, b, x)$ has a unique solution $x_e = \overline{0}$ (the vector with zero components). According to Definition 10.1, the system (10.74) (or the equilibrium state $x_e$) is globally asymptotically stable in $D_x \subset X$ iff $x_n$ converges to $\overline{0}$ for any $x_0 \in D_x$.

Assume now that the parameters $c_n$ and $b$ are unknown and the uncertainties concerning $c_n$ and $b$ are formulated as follows:

1. $$\bigwedge_{n \geq 0} (c_n \in D_c) \tag{10.75}$$

where $D_c$ is a given set in $C$.

2. $b$ is a value of random variable $\tilde{b}$ described by the probability density $f_b(b)$, and $f_b(b)$ is known.

Denote by $P_s$ the probability that the uncertain system (10.74), (10.75) is globally stable for $D_x = X$. The problem considered here consists in the determination of an estimation of $P_s$ [38, 51]. Let $W(b)$ and $V(b)$ denote properties concerning $b$ such that $W(b)$ is a sufficient condition and $V(b)$ is a necessary condition of the global asymptotic stability for the system (10.74), (10.75), i.e.

$W(b) \rightarrow$ the system (10.74), (10.75) is globally stable for $D_x = X$,

the system (10.74), (10.75) is globally stable for $D_x = X \rightarrow V(b)$.

Then

$$P_w \leq P_s \leq P_v \tag{10.76}$$

where

$$P_w = \int_{D_{bw}} f_b(b)db, \qquad P_v = \int_{D_{bv}} f_b(b)db, \tag{10.77}$$

$$D_{bw} = \{ b \in B: W(b)\}, \qquad D_{bv} = \{ b \in B: V(b)\},$$

$P_w$ is the probability that the sufficient condition is satisfied and $P_v$ is the probability that the necessary condition is satisfied. In general, $D_{bw} \subseteq D_{bv}$ and $D_{bv} - D_{bw}$ may be called a “grey zone”, which is a result of an additional uncertainty caused by the fact that $W(b) \neq V(b)$. The condition $V(b)$

may be determined as a negation of a sufficient condition that the system is not globally stable for $D_x = X$, i.e. such a property $V_{neg}(b)$ that

$$V_{neg}(b) \rightarrow \text{ there exists } c_n \text{ satisfying (10.75)}$$
$$\text{such that (10.74) is not globally stable for } D_x = X. \quad (10.78)$$

To estimate the probability $P_s$ according to (10.76), it is necessary to determine the conditions $W(b)$ and $V(b)$. The sufficient conditions for the uncertain system under consideration may have forms presented in the previous section, based on a general form (10.26). It is not possible to determine an analogous general necessary condition $V(b)$ or a sufficient condition of non-stability $V_{neg}(b)$. Particular forms of necessary conditions are presented in [52].

Let us consider one of the typical cases of uncertain systems (10.74), (10.75), when

$$D_c = \{c \in C : \bigwedge_{x \in X} [\underline{A}(b) \le A(c,b,x) \le \overline{A}(b)]\}, \quad (10.79)$$

$\underline{A}(b)$ and $\overline{A}(b)$ are given matrices and the inequality in (10.79) denotes the inequalities for the entries:

$$\underline{a}_{ij}(b) \le a_{ij}(c,b,x) \le \overline{a}_{ij}(b). \quad (10.80)$$

The definition (10.79) of the set $D_c$ means that if $c_n$ satisfies (10.75) then for every $n \ge 0$

$$\underline{A}(b) \le A(c_n,b,x_n) \le \overline{A}(b).$$

If we introduce the notation

$$A(b) = \frac{1}{2}[\underline{A}(b) + \overline{A}(b)], \qquad A(c,b,x) = A(b) + \overline{A}(c,b,x)$$

then the inequality in (10.79) may be replaced by

$$\overline{A}^{+}(c,b,x) \le A_M(b) \quad (10.81)$$

where $\overline{A}^{+}$ is the matrix obtained by replacing the entries of $\overline{A}$ by their absolute values and $A_M(b) = \overline{A}(b) - A(b)$. Then the inequality (10.81) corresponds to the form (10.47) with $A_M(b)$ in place of $A_M$. Consequently, we can use the sufficient conditions (10.54) and (10.63):

$$\alpha(b)+\| M^{-1}(b)]^{+} A_M(b)M^{+}(b) \| < 1$$

where $\alpha(b) = \max | \lambda_i[A(b)] |$, and

$$\| A^{+}(b) + A_M(b) \| < 1$$

which for $A(b) \geq 0$ (all entries of $A(b)$ are non-negative) is reduced to

$$\| \overline{A}(b) \| < 1 . \tag{10.82}$$

Under the assumption $\underline{A}(b) \geq 0$ it may be proved (see [52]) that if the system (10.74), (10.75) is globally stable for $D_x = X$ then

$$\bigvee_j [ \sum_{i=1}^{k} \underline{a}_{ij}(b) < 1] . \tag{10.83}$$

The considerations for the description based on uncertain variables are analogous to those presented for random variables. Assume that $b$ is a value of an uncertain variable $\overline{b}$ described by the certainty distribution $h_b(b)$ given by an expert. Denote by $v_s$ the certainty index that the uncertain system (10.74), (10.75) is globally stable for $D_x = X$. The problem considered here consists in the determination of an estimation of $v_s$. Using the sets $D_{bw}$ and $D_{bv}$ introduced above, one obtains

$$v_w \leq v_s \leq v_g$$

where

$$v_w = \max_{b \in D_{bw}} h_b(b), \quad v_g = \max_{b \in D_{bv}} h_b(b),$$

$v_w$ is the certainty index that the sufficient condition is satisfied and $v_g$ is the certainty index that the necessary condition is satisfied. Precisely speaking, they are the certainty indexes that the respective conditions are satisfied for approximate value of $b$, i.e., are "approximately satisfied". Choosing different sufficient and necessary conditions we may obtain different estimations of $v_s$. For example, if we choose the condition (10.82) with the norm $\| \cdot \|_\infty$ (see (10.25)) and the negation of (10.83), then

$$\max_{b \in D_{bw}} h_b(b) \leq v_s \leq \max_{b \in D_{bv}} h_b(b) ,$$

where

$$D_{bw} = \{b \in B : \bigwedge_{j} [\sum_{i=1}^{k} \overline{a}_{ij}(b) < 1]\},$$

$$D_{bv} = B - D_{b,neg}, \qquad D_{b,neg} = \{b \in B : \bigwedge_{j} [\sum_{i=1}^{k} \underline{a}_{ij}(b) \geq 1]\}.$$

More details on this subject are presented in [38, 45, 51, 52].

## 10.8 Convergence of Static Optimization Process

The convergence (stability) conditions presented in the previous sections may be also applied to the static optimization process for the plant $y = \Phi(u)$ with a single output, described in Sect. 4.1 and called an extremum searching process or extremal control (in Sect. 4.1 the notations $\bar{y}$ and $\overline{\Phi}$ have been applied to differ from the control plant with the required output). The convergence of the extremum searching process may be also called the stability of the closed-loop extremal control system. Assume that the function $y = \Phi(u)$ is differentiable with respect to $u$ and has one local minimum in the point $u^*$ (the considerations for the maximum are analogous), and that this is a unique point in which

$$\operatorname*{grad}_{u} \Phi(u) \stackrel{\Delta}{=} \widetilde{\Phi}(u) \stackrel{\Delta}{=} w = \overline{0}$$

(see 4.11). Then one may apply the optimization algorithm (4.13), i.e. the control algorithm in the closed-loop system for a substitutional plant $w = \widetilde{\Phi}(u)$ and the given output $w^* = \overline{0}$. If $u^* = \overline{0}$ and the description of the substitutional plant may be presented in the form $w = \overline{A}(u)\, u$ then, applying (4.13) we obtain the description of the closed-loop system

$$u_{n+1} = u_n - K\,\overline{A}\,(u_n) u_n = A(u_n) u_n \tag{10.84}$$

where $A(u_n) = I - K\,\overline{A}\,(u_n)$. If $y = u^{\mathrm{T}} P u$ where $P$ is a symmetric positive definite matrix then the substitutional plant is linear $w = 2Pu$ and for the linear stationary system (10.84) with the matrix $A = I - 2KP$ one can apply the stability condition: The extremum searching process (4.13) converges to $u^* = \overline{0}$ if and only if all eigenvalues of the matrix $I - 2KP$ lie inside the circle with radius 1. Using this condition and the information on an uncertain plant in the form of a set of possible matrices $P$, one can define a set

of the matrices $K$ such that for every $K$ belonging to this set the searching process is convergent. For the non-linear and (or) the non-stationary plant the condition (10.26) or related conditions presented in Sect. 10.6 may be applied.

In order to apply the algorithm (4.13) in the closed-loop control system it is necessary to obtain at the output of the substitutional plant the values $w_n$ in successive periods $n$. It is possible to obtain approximate values of the components of the vector $w_n$ replacing the components of the gradient by the ratio of increments obtained as results of *trial steps*. Then we use the algorithm (4.13) in which in the place of $w_n$ we put the approximate value of the gradient, with the following $i$-th component:

$$w_n^{(i)} \approx \frac{\Phi(u_n + \delta_i) - \Phi(u_n - \delta_i)}{2\sigma_i}, \qquad i = 1, 2, ..., p$$

where $p$ is a number of inputs, $\delta_i$ is a vector with zero components except the $i$-th component equal to $\sigma_i$ and $\sigma_i$ is a value of the trial step for the $i$-th input. Finally, the extremum searching algorithm for the current interval $n$ (and consequently, the program for a real-time controlling computer) is the following:

1. For the successive $i = 1, 2, ..., p$ put at the plant input (or execute) the decision $\bar{u}_n$ with the components

$$\bar{u}_n^{(j)} = \begin{cases} u_n^{(j)} & \text{for } j \neq i \\ u_n^{(j)} + \sigma_i & \text{for } j = i, \quad j = 1, 2, ..., p, \end{cases}$$

measure and put into memory the output value $\bar{y}_{n-1,\,i}$.

2. For the successive $i = 1, 2, ..., p$ put at the plant input the vector $\tilde{u}_n$ with the components

$$\tilde{u}_n^{(j)} = \begin{cases} u_n^{(j)} & \text{for } j \neq i \\ u_n^{(j)} - \sigma_i & \text{for } j = i, \quad j = 1,2,...,p, \end{cases}$$

measure and put into memory the output value $\tilde{y}_{n-1,\,i}$.

3. Find the next decision according to the formula (4.13) in which

$$w_n^{(i)} = \frac{\bar{y}_{n-1,\,i} - \tilde{y}_{n-1,\,i}}{2\sigma_i}.$$

As it can be seen, the determination of one decision $u_n$ requires $2p$ trial

steps, each of them consists in a trial changing of the $i$-th input ($i = 1, 2, ..., p$) and observing of the result in the form $\bar{y}_{n-1,i}$ or $\tilde{y}_{n-1,i}$. One should take into account that the extremum searching process may be long and the execution of the trial steps must be acceptable in practice. In order to determine the convergence condition, for the given function $\Phi$ one should find the function $w = \tilde{\Phi}(u)$ (in the same way as in the former considerations) and present it in the form $w = \bar{A}(u)\, u$.

If the plant output is measured with random noises, to the extremum searching process the stochastic approximation algorithm mentioned at the end of Sec. 10.3 can be applied. In the case of a gradient method this is the following algorithm:

$$u_{n+1} = u_n - \gamma_n w_n$$

where $w_n$ denotes the gradient of the function $\Phi$ with respect to $u$, and $\gamma_n$ is a sequence of coefficients presented in Sect. 10.3. For the approach with trial steps, the values of these steps should decrease in successive $n$ and converge to zero for $n \to \infty$ ( see [10, 103]).

# 11 Adaptive and Learning Control Systems

## 11.1 General Concepts of Adaptation

The present chapter as well as the previous one are devoted to problems connected with obtaining information on the plant (or decreasing an uncertainty concerning the plant) during the control process. Unlike the approach presented in Chap. 10, we shall now assume that there exists already a basic control algorithm and the additional information is used to gradual improving of this algorithm. As a rule, it is a parametric approach, i.e. improving consists in *step by step* changing of parameters $a$ in the basic algorithm. The improving is needed in order to adapt the basic control algorithm to the control plant. That is why such a control is called an *adaptive control* (more generally, an adaptive decision making). The adaptation is reasonable when, because of an uncertainty, a designer could not design the basic algorithm so as it would perform in an optimal way (more generally, determined requirements would be satisfied) for the concrete plant and disturbances, or when the plant is varying and data accepted by a designer in some time differ from the current data.

Consequently, the control algorithm in an adaptive system consists of two parts: the basic algorithm and the *algorithm of adaptation*, i.e. the procedure improving the basic algorithm. In other words, in the adaptive control system two levels may be distinguished (Fig. 11.1): the lower level with the basic controller (the executor of the control algorithm) directly receiving the data from the plant and determining the control decision $u$, and the upper level at which the *adaptator* (the executor of the adaptation algorithm) acts. The levels are called a *basic control level* and an *adaptation level*, respectively, and the two-level system in which the upper level improves the performance of the lower one – is sometimes called a *two-layer system*. Into the basic controller as well as into the adaptator, the results of the observation of the plant and (or) the environment in which the plant acts are introduced (in Fig. 11.1, for the basic control device they are the current values of $y$ and $z$). However, they are not obligatorily to be the

same data (as it is indicated in the figure). Besides, the data for the basic controller usually are introduced more frequently than the data introduced

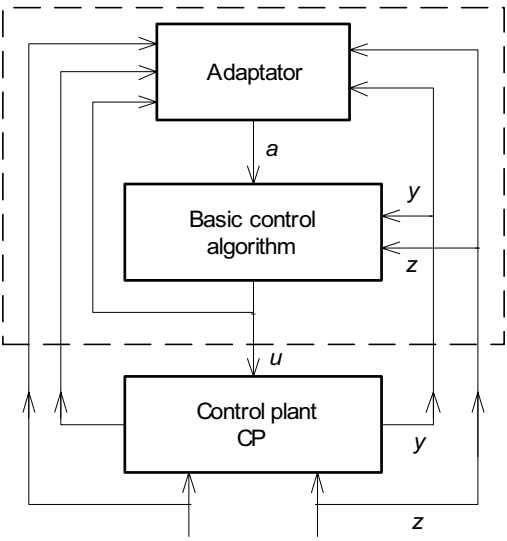


**Fig. 11.1.** Illustration of basic adaptation concept

into the adaptator because usually the adaptator acts much more slowly than the basic controller, i.e. the period of the basic control is short in comparison with the period of improving the basic control. The general idea of the adaptation presented here may be additionally characterized by the following remarks:

1. In the case of a parametric uncertainty concerning the plant parameter $c$, the idea of the adaptation is connected with the parametric optimization concept presented in Chap. 5. Now, it is not possible to find the optimum value of the parameter $a$ because the parameter $c$ is unknown. It is possible however to adapt this algorithm to the plant by changing the parameter $a$ currently during the control process.
2. From the adaptator level point of view, the adaptive system may be treated as a control system in which the basic control system is a plant with the input $a$, and the adaptator determining the decisions $a$ is a controller. The same adaptive system may also be treated as a control system with the basic control plant CP and a controller (a controlling system) consisting of two interconnected parts marked in Fig. 11.1: the basic controller and the adaptator. Thus, it is a control system with the plant CP and a complex control algorithm being a composition of two subalgorithms and using the information introduced to determine the decisions $u$.

3. One may try to determine directly the complex control algorithm. The concept of adaptation means a decomposition of this problem into two subproblems: the determination of the basic control algorithm (or only the determination of the values of parameters in the given form of this algorithm as it was discussed in Chap. 5) and the determination of the adaptation algorithm. Such a decomposition may be motivated by the fact that, as a rule, the subproblems are easier to solve and by other reasons which will be presented in Sect. 11.2.
4. Even if the complex algorithm is obtained via the decomposition, it can be executed in a coherent form as one computer control program. Sometimes, however, it may be reasonable to keep the separation of the subalgorithms (i.e. the basic control algorithm and the adaptation algorithm). They can be two cooperating subprograms implemented in one controlling computer or in two separate computers, or the program in computer adaptator improving the performance of a conventional controller at the lower level, as it occurs in practice, in the case of automatic control systems with plants being technical devices or industrial processes. A practical separation of the algorithms also occurs when the basic controller acted without improvements in the past and at some time an adaptator was inserted into the system. Sometimes it is worth keeping the separation because of reliability reasons: faults of the adaptator do not have to cause the break of the control, although the quality of the control may be decreasing at some time. Let us note that if the separation of the subalgorithms is liquidated or is not observable "from outside" then it will be possible to state that this is an adaptive system knowing how the system was composed (came into existence) but not by observing the final effect of this composition.
5. The basic problem of an adaptive control system design consists in the determination of the adaptation algorithm for the given existing basic control algorithm. In this chapter we shall present shortly the problem of the adaptation algorithm design for fixed forms of the basic algorithms, determined or described in the previous chapters.
6. All the remarks, concepts and algorithms of the adaptation concerning here the control, one may generally refer to decision making problems telling about methods and algorithms of the adaptive decision making and adaptive decision systems.

Various described and realized ideas of the adaptation most often can be reduced to one of two basic concepts:
a. Adaptation via identification.
b. Adaptation via adjustment of system parameters.
Let us explain them for a problem most frequently considered, in which the aim of changes of the parameter $a$ in the control algorithm is to achieve a minimum of the performance index $Q$, and these changes are needed be-

cause the plant parameter $c$ is changing. The first concept consists in a successive identification of the control plant (see [14]) and using the result in the form of the current value $c_m$ (as a rule, the approximate value of the real plant parameter $c$) to the determination of the value $a_m$. By $m$ the index of a successive period of adaptation has been denoted (usually it is a multiple of the basic control period denoted here by $n$), and $a_m$ denotes the value of $a$ minimizing the performance index $Q$ for $c = c_m$. For the basic control system treated as a control plant with the output $Q_m$, input $a_m$ and a disturbance $c_m$, this concept means a control in an open-loop system (Fig. 11.2). The second concept means an extremal control of the plant mentioned above in a closed-loop system (Fig. 11.3), i.e. changing $a_m$ as a result of observations of the former changes, i.e. the process of minimum $Q$ searching by a suitable change of $a$.

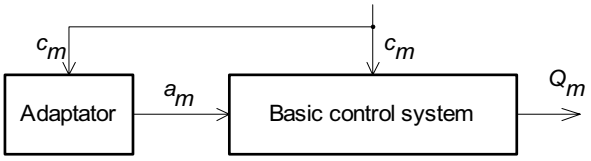


**Fig. 11.2.** Adaptation via identification in open-loop system

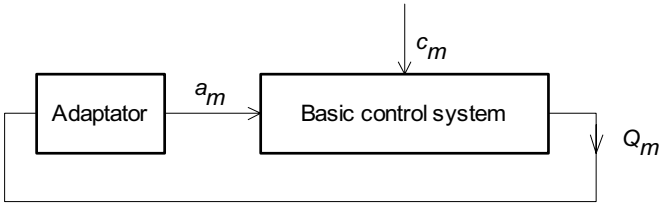


**Fig. 11.3.** Adaptation via adjustment in closed-loop system

The second part of the title of this chapter, i.e. a *learning* control system, is more difficult to define precisely and uniquely. This difficulty is caused by a great variety of different definitions, concepts and methods for which the term *learning* is used. Generally and roughly speaking, the learning process consists in a gradual improving (increasing) of an initial knowledge, based on additional information given by a trainer or obtained as a result of observations. For different methods of algorithmization and computerization of learning processes, developed in the first period mainly for needs of a classification and recognition – a common term *machine learning* has been used (see e.g. [89]). Generally and roughly speaking one can say that

adaptive control systems are systems in which a control learning process is performed. Thus, in wide sense of the word, every adaptive system is a learning system. In a more narrow sense, the term *learning control system* is understood in a different way. Usually, this term is used when in the control system at least one of the following features occurs:
1. The improvement of the control occurs as a result of the trainer's performance imitating.
2. In the adaptive system there is a third level (learning level) which improves the performance of the second level (adaptation level).
3. A knowledge representation of the plant (KP) or of the decision making or control (KD) differs from conventional mathematical models.
A short characteristic of learning control systems presented in Sects. 11.5 and 11.6 concerns the systems having the third feature mentioned above, i.e. systems with a knowledge representation, in which the learning process consists in *step by step* knowledge validation and updating and using the results of updating to the determination of current control decisions. They are then adaptive systems containing a knowledge representation considered as a generalization or a modification of traditional functional models. The first feature, i.e. learning with a trainer, may occur in such a system as well.

## 11.2 Adaptation via Identification for Static Plant

The value of the control algorithm parameter $a$ to be determined by a designer depends on the plant parameter $c$ (in general, $a$ and $c$ are vector parameters). Let us denote this relationship by $H$, i.e. $a = H(c)$. If the value $c$ is known then the respective value $a$ may be calculated by the designer. Otherwise, the designer can only give the relationship $H$ in the form of a formula or a computational procedure. Such a relationship has been considered for the static plant in Sect. 7.3 (see (7.28) and Fig. 7.3). In Chap. 5 we considered the determination of such relationship as a result of minimization of the performance index $Q$. In this case, the dependency of $Q$ upon $c$ and $u$, i.e.

$$Q = \Phi(c, a)$$

is a description of the adaptation plant, or the plant of the extremal control mentioned in the previous section. The relationship $a = H(c)$ is obtained as a result of minimization of the function $\Phi$ with respect to $a$, with the fixed $c$. In Sect. 7.3 it has been assumed that $c$ is directly measured with noises and the estimation of $c$ is put into the formula $a = H(c)$ in the place

of $c$. Now we consider a case when the estimation of $c$ is determined as a result of the plant identification, i.e. using a sequence of successive measurements of the plant input and output. What is more, it does not have to be an estimate of the unknown parameter in the known plant description, based on input and output measurements with noises – but the best parameter in a model accepted as an approximate description of the plant. Then the identification consists in the determination of the optimal model parameter ([17]). The value $c_m$ in a certain moment $m$ is determined by using a sequence of measurement results, according to an identification algorithm $G$, properly designed. By substituting the current value $c_m$ into the relationship $H$ we obtain the value $a_m = H(c_m)$ in the successive $m$-th adaptation period.

Figure 11.4 illustrates the concept of the adaptation via identification more precisely than Fig. 11.2. The adaptator consists of two parts: the identifier executing an identification algorithm and the part determining the current value $a_m = H(c_m)$ which is introduced into the basic control algorithm, i.e. is set in the basic controller.

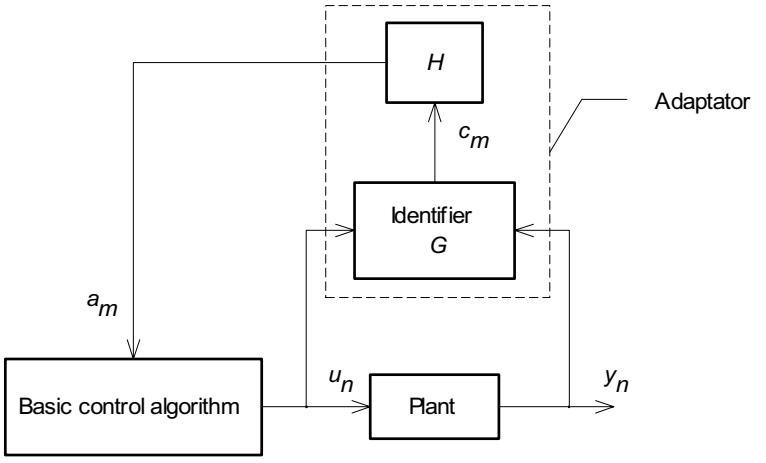


**Fig. 11.4.** Block scheme of system with adaptation via identification

Figure 11.4 presents the adaptation for a closed-loop system, but the concept described is general and may be applied to basic systems with different structures. Its essence consists in the fact that the difficult problem of determining directly the parameter $a_m$ based on observations of previous plant inputs and outputs is decomposed into two simpler problems: the determination of the identification algorithm $G$ and the designing algorithm

*H*. Thus, the identification problem is separated here in a similar way as the estimation problem of the parameter *c* based on its direct observation was separated in Sect. 7.3.

Nevertheless, designing the adaptation system with the decomposition mentioned above may be connected with some difficulties concerning the determination of the length of the adaptation period (how frequently one should determine new values of *c* and improve *a*), and concerning the convergence of $c_m$ to the unknown value *c* under the assumption that *c* is constant during the identification process, and consequently the convergence of $a_m$ to $a = H(c)$. The main reason of these difficulties is the fact that the identification is performed in a control system, in a specific situation when the plant inputs may not be changed in an arbitrary way but are determined by the basic controller updated as a result of the previous identification. Usually the formalisms describing the control process in the adaptive system are so complicated that an analytical investigation of the adaptive process convergence and the adaptive control quality is not possible and computer simulations should be applied.

The problems described above are relatively simple for static plants in which making the decision *u* considered in Sect. 3.1 is evaluated in one period. Hence, the period of the evaluation and improving of the control (i.e. adaptation period) may be equal to the period of the basic control, and if the plant is stationary, all previous values *u* and *y* from the beginning of the control may be used for the identification, that is for the determination of $c_n$. In this case, it is convenient to present the identification algorithm as a recursive procedure finding $c_n$ on the basis of $c_{n-1}$ and the current results of the $(u, y)$ measurements.

Let us consider a one-dimensional plant for which a linear model $\bar{y} = cu$ is accepted. If one assumes an identification quality index in the form

$$Q_{\mathrm{I},\,n} = \sum_{i=1}^{n-1} (y_i - cu_i)^2$$

then the optimal value of $c_n$ minimizing $Q_{\mathrm{I},\,n}$ is

$$c_n = \frac{\sum_{i=1}^{n-1} u_i y_i}{\sum_{i=1}^{n-1} u_i^2}. \tag{11.1}$$

This is the identification algorithm showing how to find $c_n$ on the basis of

$(u_1, y_1)$, $(u_2, y_2)$, ..., $(u_{n-1}, y_{n-1})$. It can be presented in the recursive form

$$c_n = \frac{\sum_{i=1}^{n-2} u_i y_i + u_{n-1} y_{n-1}}{\sum_{i=1}^{n-2} u_i^2 + u_{n-1}^2} = \frac{c_{n-1} b_{n-1} + u_{n-1} y_{n-1}}{b_{n-1} + u_{n-1}^2}, \tag{11.2}$$

$$b_n = b_{n-1} + u_{n-1}^2, \qquad n = 2, 3, ..., \tag{11.3}$$

with the initial conditions $c_1 = b_1 = 0$. The relationship (11.2) can be also presented in another form:

$$c_n = c_{n-1}\left(1 - \frac{u_{n-1}^2}{b_{n-1} + u_{n-1}^2}\right) + \frac{u_{n-1} y_{n-1}}{b_{n-1} + u_{n-1}^2}.$$

Then

$$c_n = c_{n-1} + \gamma_{n-1}(y_{n-1} - c_{n-1} u_{n-1}) \tag{11.4}$$

where

$$\gamma_{n-1} = \frac{u_{n-1}}{b_{n-1} + u_{n-1}^2} \tag{11.5}$$

For the sequence $\gamma_n$, the recursive formula can be also given without using $b_n$. The form (11.4) is connected with the known concept of the identification with a reference model, where the correction of the model parameter $c$ depends on the difference between the plant output $y_{n-1}$ and the model output $\bar{y}_n = c_{n-1} u_{n-1}$. Let us note that the coefficient $\gamma_n$ converges to zero for $n \to \infty$. The correction is then performed in a system with *degressive feed-back* mentioned in Chap. 10.

For the requirement $y = y^* \neq 0$, the relationship $H$ is reduced to $u = \frac{y^*}{c}$, and the algorithm of the control with adaptation is the following:

$$u_n = \frac{y^*}{c_n} = \frac{y^* \sum_{i=1}^{n-1} u_i^2}{\sum_{i=1}^{n-1} u_i y_i}.$$

The recursive form of this algorithm is determined by the relationships $u_n = \frac{y^*}{c_n}$, (11.2) and (11.3), or (11.4), (11,5) and (11.3), which can be replaced by one relationship describing the dependency of $\gamma_n$ upon $\gamma_{n-1}$ and $u_{n-1}$. A question arises whether this control process is convergent and to what limit, if it is. For this purpose one should distinguish the case $y_n = c_n u_n + z_n$ (i.e. $c$ is the parameter to be estimated in the linear plant in which the output is measured with a noise) and the case when $c$ is the parameter in the model $\bar{y} = cu$, which has to approximate a real plant. In the first case $c_n$ may be convergent to $c$ and consequently, $y_n$ may be convergent to $y^*$. In the second case the limit of the sequence $c_n$ may not exist and a definition of a value of $c$ optimal for a real plant to which $c_n$ might converge is not unique and it is not always formulated.

Comparing the concept of a control with the adaptation described above with the concept of a control described at the beginning of Chap. 10, it is worth noting that we can consider two control processes for the same plant:

1. The control process based on the plant model, in which successive control decisions are determined on the basis of **comparison of the plant and the model outputs** in the previous period.
2. The control process in which successive control decisions are determined directly on the basis of **comparison of the plant output with a required value**.

It is not possible to answer uniquely the question which control process is better. The above difficulties and doubts arisen in a rather simple example considered here show that the application of the adaptation via identification without a deep formal analysis or simulations may give not precisely defined and sometimes poor effects of using rather complicated control algorithms.

The considerations concerning the adaptation via identification can be extended for the multi-input and single-output plant with the linear model

$$\bar{y} = c^{\mathrm{T}} u, \qquad u, c \in R^p.$$

The generalization of the identification algorithm (11.1) is the relationship

$$c_n = (U_{n-1} U_{n-1}^{\mathrm{T}})^{-1} U_{n-1} Y_{n-1}^{\mathrm{T}}$$

where

$$U_{n-1} = [u_1 \;\; u_2 \;\; \ldots \;\; u_{n-1}], \qquad Y_{n-1} = [y_1 \;\; y_2 \;\; \ldots \;\; y_{n-1}]$$

denote the matrices of the measurement results; $Y_{n-1}$ is a one-row matrix. The formulas and transformations analogous to (11.2), (11.3), (11.4) and (11.5) are now the following:

$$\begin{aligned} c_n &= [U_{n-2} U_{n-2}^{\mathrm{T}} + u_{n-1} u_{n-1}^{\mathrm{T}}]^{-1} (U_{n-2} Y_{n-2}^{\mathrm{T}} + u_{n-1} y_{n-1}) \\ &= (B_{n-1} + u_{n-1} u_{n-1}^{\mathrm{T}})^{-1} (B_{n-1} c_{n-1} + u_{n-1} y_{n-1}), \end{aligned} \tag{11.6}$$

$$B_n = B_{n-1} + u_{n-1} u_{n-1}^{\mathrm{T}}, \qquad n = p+1, p+2, \ldots, \tag{11.7}$$

$$c_p = \bar{0}, \quad B_p = U_{p-1} U_{p-1}^{\mathrm{T}},$$

$$c_n = [I - (B_{n-1} + u_{n-1} u_{n-1}^{\mathrm{T}})^{-1}] c_{n-1} + (B_{n-1} + u_{n-1} u_{n-1}^{\mathrm{T}})^{-1} u_{n-1} y_{n-1}.$$

Then

$$c_n = c_{n-1} + \gamma_{n-1} (y_{n-1} - c_{n-1}^{\mathrm{T}} u_{n-1}) \tag{11.8}$$

where

$$\gamma_{n-1} = (B_{n-1} + u_{n-1} u_{n-1}^{\mathrm{T}})^{-1} u_n. \tag{11.9}$$

The algorithm (11.8), frequently and in a rather complicated way described in the literature (e.g. [66]) has a substantial negative feature limiting its usefulness in the concept of adaptation presented here: The correction of $p$ parameters (components of the vector $c$) is based on one scalar value, that is on the difference $y_n - \bar{y}_n$. It is worth noting that in this case the basic control algorithm is not unique, i.e. there exist infinitely many solutions of the equation $a^{\mathrm{T}} u = y^*$. Returning to the identification one can say that it would be more reasonable to correct $c$ after a successive *sequence* of the input and output measurements, containing $p$ individual measurements (see [9, 11]).

Let $y_n = c^{\mathrm{T}} u + z_n$ where $z_n$ is a random noise. The identification problem is now reduced to the estimation problem, that is to the determination of an estimate $c_n$ of the unknown parameter $c$. Then finding $c_n$ means determining a successive approximation of the value

$$c^* = \arg \min_c \mathrm{E}[(y_n - c^{\mathrm{T}} u_n)^2],$$

that is a successive approximation of the value $c^*$ satisfying the equation

$$\mathrm{E}[\operatorname{grad}_c (y_n - c^{\mathrm{T}} u_n)] = 0.$$

The application of stochastic approximation method (see Sect. 10.7) yields the following recursive identification algorithm:

$$c_n = c_{n-1} + \bar{\gamma}_{n-1} u_n (y_{n-1} - c_{n-1}^{\mathrm{T}} u_{n-1}) \tag{11.10}$$

where the scalar coefficient $\bar{\gamma}_n$ converges to zero and satisfies other convergence conditions of the stochastic approximation process. The algorithm (11.10) has then the form (11.8) in which $\gamma_{n-1} = \bar{\gamma}_{n-1} u_n$. It means that in the case where $y_n = c^{\mathrm{T}} u + z_n$ and under the conditions of stochastic approximation convergence, in the algorithm (11.8) one may put the matrix $\bar{\gamma}_{n-1}^{-1} I$ in the place of $(B_{n-1} + u_{n-1} u_{n-1}^{\mathrm{T}})$. The algorithm is then much simpler and the convergence is assured but if the procedure (11.8) and (11.9) is convergent, it can give a better identification quality and consequently, a better adaptive control quality.

## 11.3 Adaptation via Identification for Dynamical Plant

Let us consider a one-dimensional linear stationary discrete plant described by the difference equation

$$y_{n+m} + a_{m-1} y_{n+m-1} + \ldots + a_1 y_{n+1} + a_0 y_n = b_{m-1} u_{n+m-1} + \ldots + b_1 u_{n+1} + b_0 u_n,$$

i.e.

$$y_n = b_{m-1} u_{n-1} + \ldots + b_1 u_{n-m+1} + b_0 u_{n-m} - a_{m-1} y_{n-1} - \ldots - a_1 y_{n-m+1} - a_0 y_{n-m}.$$

This equation may be written in the form

$$v_n = c^{\mathrm{T}} w_n \tag{11.11}$$

where

$$c^{\mathrm{T}} = [b_{m-1} \quad b_{m-2} \quad \ldots \quad b_1 \quad b_0 \quad a_{m-1} \quad a_{m-2} \quad \ldots \quad a_1 \quad a_0],$$

$$w_n^{\mathrm{T}} = [u_{n-1} \quad u_{n-2} \quad \ldots \quad u_{n-m} \quad -y_{n-1} \quad -y_{n-2} \quad \ldots \quad -y_{n-m}].$$

Consequently, in order to identify the parameters $c$, that is to estimate their values in the known plant description or to determine their values in the model approximating the plant – one may apply the same algorithm as for identification of a static plant with the input $w_n$ and the output $v_n$. Then, one may apply recursive algorithms (11.6), (11.8) or (11.10) in which one should put $w_n$ in the place of $u_n$ and $y_n$ in the place of $v_n$. If as a result of the parametric designing of the system with a fixed parameter $a$ one obtains the formula $a = H(c)$, then the application of an adaptation period equal to a basic control period leads to changing the parameters of a basic control algorithm according to the formula $a_n = H(c_n)$. When the parameter $c$ is constant during the identification and the control corrections, and the recursive sequence $c_n$ converges to $c$, then (under the assumption that $H$ is a continuous function, which has been assumed in previous considerations as well) $a_n$ converges to the value of $a$, for which the requirement assumed in a designing stage is fulfilled. In particular, in a parametric optimization problem considered in Chap. 5, $a_n$ converges to the value minimizing the performance index $Q = \Phi(c, a)$.

All the remarks and doubts concerning the convergence of an adaptation process and the quality of an adaptive control discussed in the previous section refer to the process mentioned above as well. For a dynamical plant, a concept of the choice of the adaptation interval as a sufficiently large multiple of the basic control interval is justified and frequently described. This means that the determination of a new parameter $a_m$ is not performed in every control period by using $y_n - \bar{y}_n$ but in a period sufficiently long to estimate a quality of the control with a constant parameter $a_{m-1}$, or, more precisely speaking, to determine the control performance index as a sum of local indexes for particular control periods within one adaptation period. In this case

$$Q_m = \sum_{n=(N-1)m+1}^{n=Nm} \varphi(y_n) \tag{11.12}$$

where using $N$ means that one adaptation period contains $N$ basic control periods, $c_m$ is a value of $c_n$ for $n = Nm$, i.e. the value determined at the

end of the ($m$−1)-th adaptation period, $a_m = H(c_m)$ and is constant during the $m$-th adaptation period. Now, the basic control process and the process of the improvement of the control are separated in time, in such a sense that the improvement is performed in every $N$ periods of the basic control. To design an adaptive control algorithm the knowledge of the recursive identification algorithm and the procedure $H$ (i.e. the procedure of a parametric design with a fixed parameter $c$) is needed. It is also necessary to investigate the convergence of an identification process in the control system under consideration, for which simulations may be useful.

More details on different cases of the adaptation via identification may be found in [66].

## 11.4 Adaptation via Adjustment of Controller Parameters

This concept consists in the application of a suitable extremum searching algorithm (extremal control) to the minimization of $Q = \Phi(c, a)$, that is to a basic control system considered as a static optimization plant with the input $a_m$ and output $Q_m$ (Fig. 11.3). Such an adaptive system is usually called a *self-adjusting* or self-tuning control system. The value $Q_m = \Phi(c, a_m)$ is defined by the formula (11.12) or by another formula corresponding to another definition of the performance index estimating the control during $N$ periods. We assume that the value of the parameter $c$ is constant during an adjustment process and if a convergence condition is satisfied, $a_m$ converges to the value $a = H(c)$ minimizing the performance index $Q = \Phi(c, a)$. To the determination of an adjustment algorithm, known algorithms of the extremal control in a closed-loop system can be used. Under some assumptions, one may apply the gradient algorithm

$$a_{m+1} = a_m - Kw_m \tag{11.13}$$

where

$$w_m = \operatorname*{grad}_a Q(c,a) \Big|_{a=a_m}$$

or the algorithm with trial steps (see Sect. 10.7), i.e. the algorithm (11.13) in which the $i$-th component of the vector $w_m$ is as follows

$$w_m^{(i)} = \frac{Q(c, a_m + \delta_i) - Q(c, a_m - \delta_i)}{2\sigma_i}$$

where $\delta_i$ is the vector with zero components except the $i$-th component equal to $\sigma_i$ (the value of trial step). The matrix $K$ denotes the matrix of coefficients in the adjustment algorithm which are to be chosen by a designer in such a way as to assure the convergence of the adjustment process. For this purpose, the convergence conditions described in Chap. 10 and the stochastic approximation algorithm in the case of the searching process with noises can be applied [10]. It should be noted that the function $Q = \Phi(c, a)$ may be very complicated and may not satisfy the assumptions necessary to apply the algorithms mentioned above, in particular – it may have many local extrema. Then, to determine $a$ *step by step*, one can apply so called *genetic algorithm*. The main idea of this approach is the following: at each step not a single "candidate" $a_m$ as a possible extremizing (minimizing) value but a set of such candidates is evaluated, and a way of decreasing this set in the successive steps is given.

To apply the gradient algorithm it is necessary to know the formula $Q = \Phi(c, a)$, i.e. the performance index should be presented in an analytical form as a function of $c$ and $a$. Such functions for linear systems and quadratic performance indexes have been considered in Chap. 5. An important advantage of the adjustment algorithm with trial steps is that the knowledge of the function $\Phi$ is not required and the adjustment process is performed during the control of a real plant or its simulator by the basic control algorithm being adjusted. The algorithm with trial steps has to be applied not only when the function $\Phi$ and consequently, the function $a_m = H(c_m)$ (i.e. the extremal control algorithm in an open-loop system) are difficult to determine, but also when this determination is not possible. Such a situation occurs when the form of the control algorithm with undefined parameters is less or more arbitrary given as a universal form which by adjusting may be adapted to concrete plants from a wide class, or as a form describing more or less reasoned expert's knowledge on the control. It concerns mainly a neuron-like controller mentioned in Chap. 5, which will be described more precisely in the next chapter, and a fuzzy controller in which parameters of membership functions are treated as the components of the vector $a$. An adaptive fuzzy controller (or an adaptive fuzzy control algorithm) denotes then a fuzzy controller with the parameters adjusted during the control process.

Finally, let us note that the adaptation process consisting in the adjustment with trial steps may be very long. To determine and perform one basic step it is necessary to perform $2p$ trial steps where $p$ is a number of parameters to be adjusted, which requires $2p$ adaptation periods, sufficiently long to estimate a quality control within one such period. That is why a

simulator of the plant should be used (if it is possible) to generate the relationship $a_m = H(c_m)$ in a form of a table containing a sequence of different values $c_m$ and a sequence of corresponding values $a_m$ obtained as a result of the adjustment using the simulator. This table forms a data base in a computer executing an adaptive control of a real plant. The *adaptive program* consists of the following parts:

1. Determination of $c_m$ according to a recursive identification algorithm.
2. Finding in the table mentioned above the value $\bar{c}_m$ for which $\| c_m - \bar{c}_m \|$ is the smallest for a given form of the norm.
3. Putting $a_m$ corresponding to $\bar{c}_m$ into a basic control algorithm.

Such an adaptation process is sometimes called an *adaptation with learning*. In this case obtaining the table by using the simulator may be considered as an effect of learning. In other words, this is a combination of a concept of adaptation via identification in an open-loop system with a concept of adjustment adapting the control algorithm to the plant simulator. This concept as well as other ideas concerning the adaptation we present in a rather descriptive informal way because, in principle, designing the control program consists here in designing of subprograms according to the algorithms described in a formal way in the previous chapters or in other books devoted to identification and static optimization.

## 11.5 Learning Control System Based on Knowledge of the Plant

According to the remark at the end of Sect. 11.1, this section concerns plants described by a knowledge representation in the form of relations with unknown parameters. The learning process consists here in *step by step* knowledge validation and updating [25, 26, 28–31, 42, 52, 53]. At each step one should prove if the current observation "belongs" to the knowledge representation determined before this step (*knowledge validation*) and if not – one should modify the current estimation of the parameters in the knowledge representation (*knowledge updating*). The results of the successive estimation of the unknown parameters are used in the current determination of the decisions in a learning decision making system. This approach may be considered as an extension of the known idea of adaptation via identification for the plants described by traditional mathematical models (see e.g. [14]). We shall consider two versions of learning systems. In the first version the knowledge validation and updating is concerned with the knowledge of the plant (i.e. the relation *R* describing the

plant), and in the second version – with the knowledge of the decision making (i.e. the set of decisions $D_u$). In both versions the learning algorithms based on the knowledge validation and updating will be presented.

Consider a static plant described by a relation

$$R(u, y; c) \subset U \times Y \tag{11.14}$$

where $c \in C$ is a vector parameter (a vector of parameters). As it was said in Chap. 6, the relation $R$ may by described by a set of equalities and/or inequalities concerning the components of $u$ and $y$, e.g.

$$u^{\mathrm{T}} u + y^2 \le c^2 \tag{11.15}$$

where $u$ is a column vector, $c$ and $y$ are one-dimensional variables, or

$$a^{\mathrm{T}} u \le y \le b^{\mathrm{T}} u$$

where $a$, $b$ are vectors of parameters, i.e. $c = (a, b)$. As a solution of the decision problem for the given $D_y \subset Y$ (see Sect. 6.3) we obtain a set of decisions

$$D_u(c) = \{u \in U : \ D_y(u; c) \subseteq D_y\} \tag{11.16}$$

where

$$D_y(u; c) = \{y \in Y : (u, y) \in R(u, y; c)\}$$

is a set of possible outputs for the given $u$, and $D_u(c)$ is the largest set such that the implication $u \in D_u \to y \in D_y$ is satisfied. For example, if in the case (11.15) we have the requirement $y^2 \le \alpha^2$ then the solution $D_u(c)$ is determined by the inequality

$$c^2 - \alpha^2 \le u^{\mathrm{T}} u \le c^2 .$$

For the further considerations we assume that $R(u, y; c)$ for every $c \in C$ is a continuous and closed domain in $U \times Y$. Assume now that the parameter $c$ in the relation $R$ has the value $c = \bar{c}$ and $\bar{c}$ is unknown.

### 11.5.1 Knowledge Validation and Updating

Let us assume that the sequence of observations

$$(u_1, y_1), (u_2, y_2), \ldots, (u_n, y_n), \quad \bigwedge_{i \in \overline{1,n}} [(u_i, y_i) \in R(u, y; \overline{c})]$$

is available and may be used for the estimation of $\overline{c}$. For the value $u_i$ at the input, the corresponding value $y_i$ is "generated" by the plant. The greater the variation of $(u_i, y_i)$ inside *R*, the better the estimation that may be obtained. Let us introduce the set

$$D_c(n) = \{c \in C : \bigwedge_{i} [(u_i, y_i) \in R(u, y; c)]\}. \tag{11.17}$$

It is easy to note that $D_c(n)$ is a closed set in *C*. The boundary $\Delta_c(n)$ of the set $D_c(n)$ may be proposed as the estimation of $\overline{c}$. In the example (11.15) the set $D_c(n) = [c_{\min,n}, \infty)$ and $\Delta_c(n) = \{c_{\min,n}\}$ (a singleton) where

$$c^2_{\min,n} = \max_i (u_i^{\mathrm{T}} u_i + y_i^2). \tag{11.18}$$

For one-dimensional *u,* the estimation of $\overline{c}$ is illustrated in Fig. 11.5.

Assume that the points $(u_i, y_i)$ occur randomly from $R(u, y; \overline{c})$ with probability density $f(u, y)$, i.e. that $(u_i, y_i)$ are the values of random variables $(\widetilde{u}, \widetilde{y})$ with probability density $f(u, y)$.

**Theorem 11.1.** If $f(u, y) > 0$ for every $(u, y) \in R(u, y; \overline{c})$ and for every $c \neq \overline{c}$ $R(u, y; c) \neq R(u, y; \overline{c})$ then $\Delta_c(n)$ converges to $\{\overline{c}\}$ with probability 1.

*Proof:* From (11.17)

$$D_c(n+1) = \{c \in C : \bigwedge_{i \in \overline{1,n}} [(u_i, y_i) \in R(u, y; c)] \wedge (u_{n+1}, y_{n+1}) \in R(u, y; c)\}.$$

Then $D_c(n+1) \subseteq D_c(n)$, which means that $D_c(n)$ is a convergent sequence of sets. We shall show that $D_c = \overline{D}_c$ with probability 1, where

$$D_c = \lim_{n \to \infty} D_c(n) = \{c \in C : \bigwedge_{i \in \overline{1,\infty}} [(u_i, y_i) \in R(u, y; c)]\}, \tag{11.19}$$

$$\overline{D}_c = \{c \in C : R(u, y; \overline{c}) \subseteq R(u, y; c)\}. \tag{11.20}$$

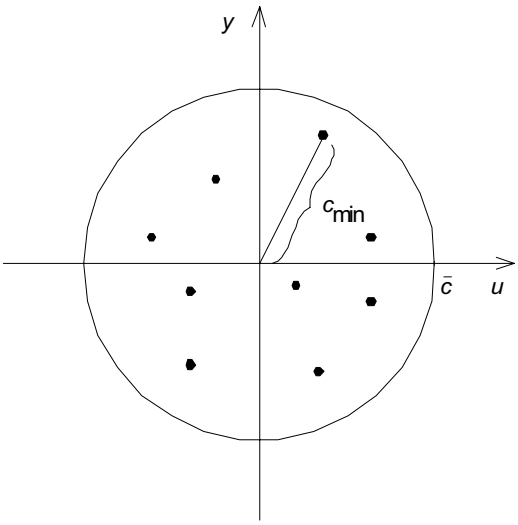


**Fig. 11.5.** Illustration of the estimation in example under consideration

Assume that $D_c \neq \overline{D}_c$, i.e. there exists $\hat{c} \in D_c$ such that $R(u, y; \overline{c}) \not\subset R(u, y; \hat{c})$. There exists then the subset of $R(u, y; \overline{c})$

$$R(u, y; \overline{c}) - R(u, y, \hat{c}) \stackrel{\Delta}{=} D_R \tag{11.21}$$

such that for every $i \in \overline{1,\infty}$ $(u_i, y_i) \notin D_R$. The probability of this property is the following:

$$\lim_{n \to \infty} p^n \stackrel{\Delta}{=} P_\infty$$

where

$$p = P[(\tilde{u}, \tilde{y}) \in U \times Y - D_R] = \int_{U \times Y - D_R} f(u, y) du dy .$$

From the assumption about $f(u, y)$ it follows that $p < 1$ and $P_\infty = 0$. Then $D_c = \overline{D}_c$ with probability 1. From (11.19)

$$\lim_{n\to\infty} \Delta_c(n) \stackrel{\Delta}{=} \Delta_c$$

where $\Delta_c$ is the boundary of $D_c$. Using the assumption about *R* it is easy to note from (11.20) that $\overline{\Delta}_c = \{\overline{c}\}$ where $\overline{\Delta}_c$ is the boundary of $\overline{D}_c$. Then with probability 1

$$\lim_{n\to\infty} \Delta_c(n) = \overline{\Delta}_c = \{\overline{c}\}. \qquad \square$$

The determination of $\Delta_c(n)$ may be presented in the form of the following recursive algorithm:

1. **Knowledge validation.** Prove if

$$\bigwedge_{c\in D_c(n-1)} [(u_n, y_n) \in R(u, y; c)]. \tag{11.22}$$

If yes then $D_c(n) = D_c(n-1)$ and $\Delta_c(n) = \Delta_c(n-1)$. If not then one should determine the new $D_c(n)$ and $\Delta_c(n)$, i.e. update the knowledge.

2. **Knowledge updating.**

$$D_c(n) = \{c \in D_c(n-1): \quad (u_n, y_n) \in R(u, y; c)\} \tag{11.23}$$

and $\Delta_c(n)$ is the boundary of $D_c(n)$. For $n = 1$

$$D_c(1) = \{c \in C: \quad (u_1, y_1) \in R(u, y; c)\}.$$

The successive estimations may be used in current updating of the solution of the decision problem in the open-loop learning system, in which the set $D_u(c_n)$ is determined by putting $c_n$ in (11.16), where $c_n$ is chosen randomly from $\Delta_c(n)$. For the random choice of $c_n$ a generator of random numbers is required.

### 11.5.2 Learning Algorithm for Decision Making in Closed-loop System

The successive estimations of $\overline{c}$ may be performed in a closed-loop learning system where $u_i$ is the sequence of the decisions. For the successive decision $u_n$ and its result $y_n$, knowledge validation and updating should be performed by using the algorithm presented in the first part of this sec-

tion. The next decision $u_{n+1}$ is based on the updated knowledge and is chosen randomly from $D_u(c_n)$. Finally, the decision making algorithm in the closed-loop learning system is the following:

1. Put $u_n$ at the input of the plant and measure $y_n$.
2. Prove the condition (11.22), determine $D_c(n)$ and $\Delta_c(n)$. If (11.22) is not satisfied, then knowledge updating according to (11.23) is necessary.
3. Choose randomly $c_n$ from $\Delta_c(n)$.
4. Determine $D_u(c_n)$ according to (11.16) with $c = c_n$.
5. Choose randomly $u_{n+1}$ from $D_u(c_n)$.

For $n = 1$ one should choose randomly $u_1$ from $U$ and determine $D_c(1)$. If for all $n < p$ the value $u_n$ is such that $y_n$ does not exist (i.e. $u_n$ does not belong to the projection of $R(u, y; \bar{c})$ on $U$), then the estimation starts from $n = p$. If $D_u(c_n)$ is an empty set (i.e. for $c = c_n$ the solution of the decision problem does not exist) then $u_{n+1}$ should be chosen randomly from $U$. The block scheme of the learning system is presented in Fig. 11.6. For the random choice of $c_n$ and $u_n$ the generators $G_1$ and $G_2$ are required. The probability distributions should be determined currently for $\Delta_c(n)$ and $D_u(c_n)$.

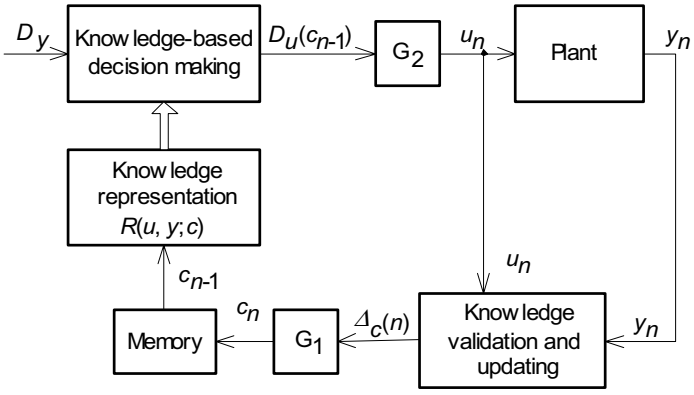


**Fig. 11.6.** Learning system based on the knowledge of the plant

Assume that the points $c_n$ are chosen randomly from $\Delta_c(n)$ with probability density $f_{cn}(c)$, the points $u_n$ are chosen randomly from $D_u(c_{n-1})$ with probability density $f_u(u \mid c_{n-1})$ and the points $y_n$ "are generated"

randomly by the plant with probability density $f_y(y \mid u_n; \bar{c})$ from the set $D_y(u;c) = \{y \in Y : (u, y) \in R(u, y; c)\}$ where $u = u_n$ and $c = \bar{c}$. It means that $(c_i, u_{i+1}, y_{i+1})$ are the values of random variables $(\tilde{c}_i, \tilde{u}_{i+1}, \tilde{y}_{i+1})$ with probability density $f_{ci}(c_i) \cdot f_u(u_{i+1} \mid c_i) \cdot f_y(y_{i+1} \mid u_{i+1}; \bar{c})$.

## 11.6 Learning Control System Based on Knowledge of Decisions

In this version the validation and updating directly concerns $D_u(c)$, i.e. the knowledge of the decision making. When the parameter $\bar{c}$ is unknown then for the fixed value $u$ it is not known if $u$ is a correct decision, i.e. if $u \in D_u(\bar{c})$ and consequently $y \in D_y$. Our problem may be considered as a classification problem with two classes. The point $u$ should be classified to class $j = 1$ if $u \in D_u(\bar{c})$ and to class $j = 2$ if $u \notin D_u(\bar{c})$. Assume that we can use the learning sequence

$$(u_1, j_1), (u_2, j_2), \ldots, (u_n, j_n) \stackrel{\Delta}{=} S_n$$

where $j_i \in \{1,2\}$ are the results of the correct classification given by an external trainer or obtained by testing the property $y_i \in D_y$ at the output of the plant. Let us assume for the further considerations that $D_u(c)$ is a continuous and closed domain in $U$, and consider the approaches analogous to those presented in the previous section.

### 11.6.1 Knowledge Validation and Updating

Let us denote by $\bar{u}_i$ the subsequence for which $j_i = 1$, i.e. $\bar{u}_i \in D_u(\bar{c})$ and by $\hat{u}_i$ the subsequence for which $j_i = 2$, and introduce the following sets in $C$:

$$\overline{D}_c(n) = \{c \in C : \bar{u}_i \in D_u(c) \text{ for every } \bar{u}_i \text{ in } S_n\}, \tag{11.24}$$

$$\hat{D}_c(n) = \{c \in C : \hat{u}_i \in U - D_u(c) \text{ for every } \hat{u}_i \text{ in } S_n\}. \tag{11.25}$$

It is easy to see that $\overline{D}_c$ and $\hat{D}_c$ are closed sets in $C$. The set

$$\overline{D}_c(n) \cap \hat{D}_c(n) \stackrel{\Delta}{=} \overline{\Delta}_c(n)$$

may be proposed as the estimation of $\bar{c}$. For example, if $D_u(\bar{c})$ is described by the inequality $u^{\mathrm{T}}u \le \bar{c}^2$ then

$$\overline{D}_c(n) = [c_{\min,n}, \infty), \quad \hat{D}_c(n) = [0, c_{\max,n}), \quad \overline{\Delta}_c(n) = [c_{\min,n}, c_{\max,n})$$

where

$$c^2_{\min,n} = \max_i \bar{u}_i^{\mathrm{T}} \bar{u}_i, \qquad c^2_{\max,n} = \min_i \hat{u}_i^{\mathrm{T}} \hat{u}_i.$$

Assume that the points $u_i$ are chosen randomly from $U$ with probability density $f(u)$.

**Theorem 11.2.** If $f(u) > 0$ for every $u \in U$ and $D_u(c) \ne D_u(\bar{c})$ for every $c \ne \bar{c}$ then $\overline{\Delta}_c(n)$ converges to $\{\bar{c}\}$ with probability 1 (w.p.1).

*Proof:* In the same way as for Theorem 11.1 we can prove that

$$\lim_{n\to\infty} \overline{D}_c(n) = \overline{D}_c, \qquad \lim_{n\to\infty} \hat{D}_c(n) = \hat{D}_c \tag{11.26}$$

w.p.1 where

$$\overline{D}_c = \{c \in C : D_u(\bar{c}) \subseteq D_u(c)\}, \quad \hat{D}_c = \{c \in C : D_u(c) \subseteq D_u(\bar{c})\}. \tag{11.27}$$

From (11.26) one can derive that $\overline{\Delta}_c(n)$ converges to $\overline{D}_c \cap \hat{D}_c \stackrel{\Delta}{=} \overline{\Delta}_c$ (the boundary of $\overline{D}_c$) w.p.1. Using the assumption about $D_u$ it is easy to note that $\overline{\Delta}_c = \{\bar{c}\}$. □

The determination of $\overline{\Delta}_c(n)$ may be presented in the form of the following recursive algorithm:

If $j_n = 1$ ($u_n = \bar{u}_n$).

1. **Knowledge validation** for $\bar{u}_n$. Prove if

$$\bigwedge_{c \in \overline{D}_c(n-1)} [u_n \in D_u(c)].$$

If yes then $\overline{D}_c(n) = \overline{D}_c(n-1)$. If not then one should determine the new

$\overline{D}_c(n)$, i.e. update the knowledge.

2. **Knowledge updating** for $\overline{u}_n$

$$\overline{D}_c(n) = \{c \in \overline{D}_c(n-1) : u_n \in D_u(c)\} .$$

Put $\hat{D}_c(n) = \hat{D}_c(n-1)$.

If $j_n = 2$ ($u_n = \hat{u}_n$).

3. **Knowledge validation** for $\hat{u}_n$. Prove if

$$\bigwedge_{c \in \hat{D}_c(n-1)} [u_n \in U - D_u(c)] .$$

If yes then $\hat{D}_c(n) = \hat{D}_c(n-1)$. If not then one should determine the new $\hat{D}_c(n)$, i.e. update the knowledge.

4. **Knowledge updating** for $\hat{u}_n$

$$\hat{D}_c(n) = \{c \in \hat{D}_c(n-1) : u_n \in U - D_u(c)\} .$$

Put $\overline{D}_c(n) = \overline{D}_c(n-1)$ and $\overline{\Delta}_c(n) = \overline{D}_c(n) \cap \hat{D}_c(n)$.

For $n = 1$, if $u_1 = \overline{u}_1$ determine

$$\overline{D}_c(1) = \{c \in C : u_1 \in D_u(c)\} ,$$

if $u_1 = \hat{u}_1$ determine

$$\hat{D}_c(1) = \{c \in C : u_1 \in U - D_u(c)\} .$$

If for all $i \le p$ $u_i = \overline{u}_i$ (or $u_i = \hat{u}_i$), put $\hat{D}_c(p) = C$ (or $\overline{D}_c(p) = C$).

### 11.6.2 Learning Algorithm for Control in Closed-loop System

The successive estimation of $\overline{c}$ may be performed in a closed-loop learning control system where $u_i$ is the sequence of the decisions. The control algorithm is as follows:

1. Put $u_n$ at the input of the plant and measure $y_n$.
2. Introduce $j_n$ given by a trainer.
3. Determine $\overline{\Delta}_c(n)$ using the estimation algorithm with knowledge vali-

dation and updating.

4. Choose randomly $c_n$ from $\overline{\Delta}_c(n)$, put $c_n$ into $R(u,y;c)$ and determine $D_u(c)$, or put $c_n$ directly into $D_u(c)$ if the set $D_u(c)$ may be determined from *R* in an analytical form.

5. Choose randomly $u_{n+1}$ from $D_u(c_n)$.

At the beginning of the learning process $u_i$ should be chosen randomly from *U*. The block scheme of the learning control system in the case when $c_n$ is put directly into $D_u(c)$ is presented in Fig. 11.7, and in the case when $D_u(c_n)$ is determined from $R(u,y;c_n)$ is presented in Fig. 11.8. The blocks $G_1$ and $G_2$ are the generators of random variables for the random choosing of $c_n$ from $\overline{\Delta}_c(n)$ and $u_{n+1}$ from $D_u(c_n)$, respectively.

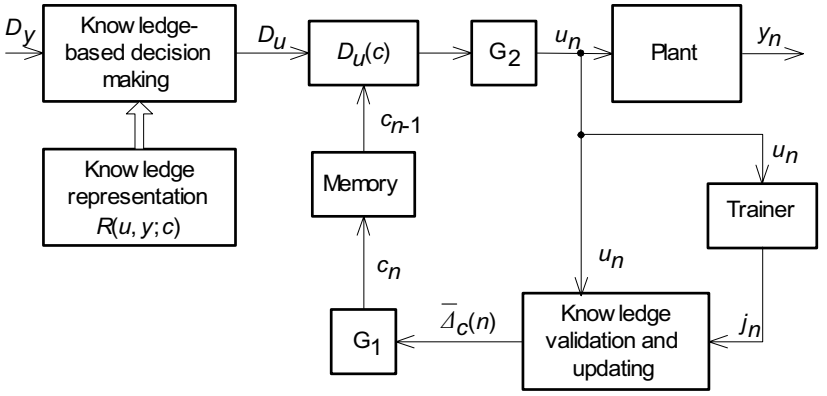


**Fig. 11.7.** Learning control system in the first version

Assume that the points $c_n$ are chosen randomly from $\overline{\Delta}_c(n)$ with probability density $f_{cn}(c)$ and the points $u_n$ are chosen randomly from $D_u(c_{n-1})$ with probability density $f_u(u \mid c_{n-1})$, i.e. $(c_i, u_{i+1})$ are the values of random variables $(\tilde{c}_i, \tilde{u}_{i+1})$.

**Theorem 11.3.** If

$$\bigwedge_i \bigwedge_{c \in \overline{\Delta}_c(i)} f_{ci}(c) > 0, \quad \bigwedge_{c \in C} \bigwedge_{u \in D_u(c)} f_u(u \mid c) > 0 \tag{11.28}$$

and for every $c \neq \overline{c}$

$$D_u(c) \neq D_u(\bar{c}) \tag{11.29}$$

then $\bar{\Delta}_c(n)$ converges to $\{\bar{c}\}$ w.p.1.

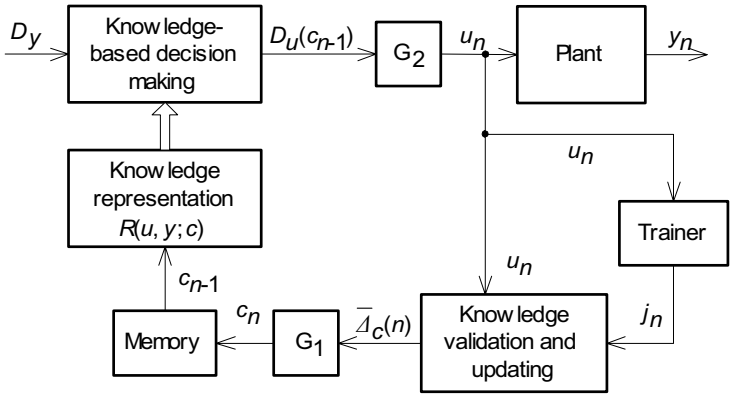


**Fig. 11.8.** Learning control system in the second version

*Proof*: From (11.24) it is easy to note that $\bar{D}_c(n+1) \subseteq \bar{D}_c(n)$, which means that $\bar{D}_c(n)$ is a convergent sequence of sets. We shall show that $D_c = \bar{D}_c$ w.p.1. where

$$D_c = \lim_{n\to\infty} \bar{D}_c(n)$$

and $\bar{D}_c$ is defined in (11.27). Assume that $D_c \neq \bar{D}_c$, i.e. there exists $\hat{c} \in D_c$ such that $D_u(\bar{c}) \not\subseteq D_u(\hat{c})$. There exists then the subset of $D_u(\bar{c})$

$$D_u(\bar{c}) - D_u(\hat{c}) \triangleq D_R$$

such that $\bar{u}_i \notin D_R$ for every $\bar{u}_i$ in $S_\infty$. The probability of this property is the following:

$$\lim_{n\to\infty} \prod_{i=1}^{n} p_i \triangleq P_\infty$$

where

$$p_i = P(\tilde{u}_i \in U - D_R) = \int_{U-D_R} f_{ui}(u)du\,,$$

$$f_{ui}(u) = \int_{\overline{\Delta}_c(i)} f_u(u \mid c) f_{ci}(c)\, dc \,. \tag{11.30}$$

Since $\overline{c} \in \overline{\Delta}_c(i)$ for every $i$ then from (11.28) and (11.30) it follows that $f_{ui}(u) > 0$ for every $u \in D_u(\overline{c})$ and consequently $f_{ui}(u) > 0$ for every $u \in D_R$. Thus, $p_i < 1$ for every $i$ and $P_\infty = 0$. Then $D_c = \overline{D}_c$ w.p.1. In the same way it may be proved that

$$\lim_{n\to\infty} \hat{D}_c(n) = \hat{D}_c \,, \text{ w.p.1}$$

where $\hat{D}_c$ is defined in (11.27). Consequently, $\overline{\Delta}_c(n)$ converges w.p.1 to $\overline{D}_c \cap \hat{D}_c = \overline{\Delta}_c$ (the boundary of $\overline{D}_c$). Using (11.29) it is easy to note that $\overline{\Delta}_c = \{\overline{c}\}$. □

**Remark 11.1.** Let us note that the decisions in a closed-loop learning system may be based on $j_n$ given by an external trainer, i.e. $j_n = 1$ if $u_n \in D_u(\overline{c})$ and $j_n = 2$ if $u_n \notin D_u(\overline{c})$, or may be obtained by testing the property $y_n \in D_y$. In this case,

$$\text{if } y_n \notin D_y \text{ then } j_n = 2 \text{ and } u_n \notin D_u(\overline{c}) \,,$$

$$\text{if } y_n \in D_y \text{ then } j_n = 1 \text{ and } u_n \in \widetilde{D}_u(\overline{c})$$

where

$$\widetilde{D}_u(\overline{c}) = \{u \in U : D_y(\overline{c}) \cap D_y \neq \varnothing\}$$

and

$$D_y(\overline{c}) = \{y \in Y : (u, y) \in R(u, y; \overline{c})\} \,.$$

Consequently, in (11.24) and in the first part of the recursive algorithm presented in Sect. 11.6.1 for $\overline{u}_n$, one should use $\widetilde{D}_u(c)$ instead of $D_u(c)$.

It is worth noting that Theorem 11.3 concerns the case with an external trainer. □

**Example 11.1.** Consider the single-output plant described by the inequality

$$0 \le y \le \frac{1}{\bar{c}} u^{\mathrm{T}} P u$$

where $P$ is a positive definite matrix. For the requirement $y \le \bar{y}$ we obtain

$$D_u(\bar{c}) = \{u \in U:\ u^{\mathrm{T}} P u \le \bar{c}\ \bar{y}\}\,. \tag{11.31}$$

According to (11.24) and (11.25)

$$\bar{D}_c(n) = [c_{\min,n}, \infty), \quad \hat{D}_c(n) = [0, c_{\max,n}), \quad \bar{\Delta}_c(n) = [c_{\min,n}, c_{\max,n})$$

where

$$c_{\min,n} = \bar{y}^{-1} \cdot \max_i \bar{u}_i^{\mathrm{T}} P \bar{u}_i\,, \qquad c_{\max,n} = \bar{y}^{-1} \cdot \min_i \hat{u}_i^{\mathrm{T}} P \hat{u}_i\,.$$

The decision making algorithm in the closed-loop learning system is the following:

1. Put $u_n$ at the input, measure $y_n$.
2. Introduce $j_n$ given by a trainer.
3. For $j_n = 1$ ($u_n = \bar{u}_n$), prove if

$$\bar{y}^{-1} u_n^{\mathrm{T}} P u_n \le c_{\min,n-1}\,.$$

If yes then $c_{\min,n} = c_{\min,n-1}$. If not, determine new $c_{\min,n}$

$$c_{\min,n} = \bar{y}^{-1}\, u_n^{\mathrm{T}} P u_n\,.$$

Put $c_{\max,n} = c_{\max,n-1}$.

4. For $j_n = 2$ ($u_n = \hat{u}_n$), prove if

$$\bar{y}^{-1} u_n^{\mathrm{T}} P u_n \ge c_{\max,n-1}\,.$$

If yes then $c_{\max,n} = c_{\max,n-1}$. If not, determine new $c_{\max,n}$

$$c_{\max,n} = \bar{y}^{-1}\, u_n^{\mathrm{T}} P u_n\,.$$

Put $c_{\min,n} = c_{\min,n-1}$, $\bar{\Delta}_c(n) = [c_{\min,n}, c_{\max,n})$.

5. Choose randomly $c_n$ from $\bar{\Delta}_c(n)$ and put $\bar{c} = c_{n-1}$ in (11.31).

6. Choose randomly $u_n$ from $D_u(c_n)$. □

The example may be easily extended for the case when $D_u(c)$ is a domain closed by a hypersurface $F(u) = c$ for one-dimensional $c$ and a given function $F$. The simulations showed the significant influence of the shape of $D_u(c)$ and the probability distributions $f_c(c)$, $f_u(u \mid c)$, on the convergence of the learning process and the quality of the decisions.

# 12 Intelligent and Complex Control Systems

## 12.1 Introduction to Artificial Intelligence

One of the important functions which may be performed by a computer is replacing or supporting a human being in solving difficult problems for which a knowledge and an intelligence is required. The computerization of problem solving which requires not only a determined knowledge but also a proper reasoning process is often informally and imprecisely called an *artificial intelligence*. Its products are intelligent computer systems of different kinds, including intelligent control systems [10, 101]. The artificial intelligence is usually understood as a set of problems, methods, tools and techniques connected with the computerization mentioned above.

There exists a great variety of definitions and meanings of the terms *artificial intelligence* and intelligent systems, very frequently used nowadays. Independently of the different names, concrete precisely defined problems and methods formulated under these names are important. A set of names, problems, methods and practical products in this area has been changing during several decades of its development. Similarly, we can speak about changes of a role of a computer (more precisely speaking of a software system implemented in a suitable device) as a tool supporting a human being in his mental works: from a tool for computing via a tool for problem solving according to a program precisely developed by a designer and a tool for information storing and processing to a role of computer expert, i.e. a tool for a knowledge-based problem solving with the application of reasoning (see corresponding remarks in [18, 20]).

At present, the following problems and methods are usually included in the artificial intelligence area:

1. Computerization of logical operations; in particular, algorithmization and computerization of inference processes.
2. Decision making under uncertainty defined in different ways, with the use of a prediction, solution searching algorithms and a formulation of multi-variant solutions.

3. Application of so called *genetic algorithms* to the computerization of a solution searching for a class of optimization problems.
4. Algorithmization and computerization of learning processes.
5. Application of so called *neural networks* (or *artificial neuron networks*) as a universal tool for problem solving with learning in a wide class of computational problems.
6. Knowledge-based problem solving in so called *expert systems* [18, 92].

The concepts and areas mentioned above are interconnected in different ways and all of them may concern a single individual intelligent system, e.g. a system in which a knowledge-based decision making under uncertainty, with the application of a learning process can occur.

The problems, methods and algorithms concerning the control with knowledge representations, with non-probabilistic descriptions of the uncertainty and with the application of learning processes described in the previous chapters could be considered under the name *intelligent control systems*. Their location in the previous chapters was reasoned by a determined composition of the book. In this chapter we shall present shortly problems of the control with a logical knowledge representation and the application of neural networks in control systems, i.e. the topics perhaps most typical for intelligent control systems. These topics could be presented earlier as well: the control with the logical knowledge representation in Chap. 6 and neural networks in Chap. 11 in connection with the adaptation and learning processes.

The short description of the intelligent systems in the first part of this chapter may be treated only as a part of a very wide area concerned with intelligent and expert control systems. The second part of this chapter is devoted to selected problems of complex control systems with traditional models and with knowledge representations.

## 12.2 Logical Knowledge Representation

In Chapter 6 we considered a plant described by a relation $R(u, y)$ with the input vector $u \in U = R^p$ and the output vector $y \in Y = R^l$. For such a plant the input property $\varphi_u(u) = \text{“}u \in D_u\text{”}$ and the output property $\varphi_y(y) = \text{“}y \in D_y\text{”}$ have been introduced and the analysis and decision making problems have been considered. It has been assumed that the relation $R$ as well as the properties $\varphi_u$, $\varphi_y$ or the sets $D_u$, $D_y$ are described with the help of a set of equalities and (or) inequalities concerning the components of the vectors $u$ and $y$, and auxiliary additional variables (components of the

vector $w \in W$) used in the knowledge representation. It has been also mentioned that for such a knowledge representation, solving of both problems may be very difficult, not possible to be presented in the form of a universal algorithm and may require individual approaches. We shall now consider a plant described by a relation which may be presented with the help of logical formulas composed of so called *simple formulas* and logical operations: *or*, *and*, *not*, *if ... then*. The simple formulas are elementary properties concerning the variables $u$, $y$, $w$. In the simple formulas the logical operations mentioned above and quantifiers may occur as well, but in the knowledge representation the simple formulas are treated as undivided units, "blocks" used in building the knowledge description.

Let us introduce the following notation:

1. $\alpha_{uj}(u)$ – simple formula (i.e. simple property) concerning $u$, $j = 1, 2, ..., n_1$, e.g. $\alpha_{u1}(u) =$ "$u^{\mathrm{T}} u \le 2$ ".
2. $\alpha_{wr}(u, w, y)$ – simple formula concerning $u$, $w$ and $y$, $r = 1, 2, ..., n_2$.
3. $\alpha_{ys}(y)$ – simple formula concerning $y$, $s = 1, 2, ..., n_3$.
4. $\alpha_u = (\alpha_{u1}, \alpha_{u2}, ..., \alpha_{un_1})$ – subsequence of simple formulas concerning $u$.
5. $\alpha_w = (\alpha_{w1}, \alpha_{w2}, ..., \alpha_{wn_2})$ – subsequence of simple formulas concerning $u$, $w$ and $y$.
6. $\alpha_y = (\alpha_{y1}, \alpha_{y2}, ..., \alpha_{yn_3})$ – subsequence of simple formulas concerning $y$.
7. $\alpha(u, w, y) \stackrel{\Delta}{=} (\alpha_1, \alpha_2, ..., \alpha_n) = (\alpha_u, \alpha_w, \alpha_y)$ – sequence of all simple formulas in the knowledge representation, $n = n_1 + n_2 + n_3$.
8. $F_i(\alpha)$ – the $i$-th fact given by an expert. It is a logic formula composed of the subsequence of $\alpha$ and the logic operations: $\vee$ – *or*, $\wedge$ – *and*, $\neg$ – *not*, $\rightarrow$ – *if ... then*, $i = 1, 2, ..., k$.

For example $F_1 = \alpha_1 \wedge \alpha_2 \rightarrow \alpha_4$, $F_2 = \alpha_3 \vee \alpha_2$ where $\alpha_1 =$ "$u^{\mathrm{T}} u \le 2$ ", $\alpha_2 =$ "the temperature is low or $y^{\mathrm{T}} y \le 3$ ", $\alpha_3 =$ "$y^{\mathrm{T}} y > w^{\mathrm{T}} w$", $\alpha_4 =$ "$y^{\mathrm{T}} y = 4$ ".

9. $F(\alpha) = F_1(\alpha) \wedge F_2(\alpha) \wedge ... \wedge F_k(\alpha)$.
10. $F_u(\alpha_u)$ – input property, i.e. the logic formula using $\alpha_u$.
11. $F_y(\alpha_y)$ – output property.
12. $a_m \in \{0, 1\}$ – logic value of the simple property $\alpha_m$, $m = 1, 2, ..., n$.

13. $a = (a_1, a_2, ..., a_n)$ – zero-one sequence of the logic values.

14. $a_u(u)$, $a_w(u, w, y)$, $a_y(y)$ – zero-one subsequences of the logic values corresponding to $\alpha_u(u)$, $\alpha_w(u, w, y)$, $\alpha_y(y)$.

15. $F(a)$ – the logic value of $F(\alpha)$.

We used a uniform term *facts* for all formulas $F_i$. In the literature, $F_i$ in the form of implications are often called *production rules*. All facts given by an expert are assumed to be true, i.e. $F(a) = 1$.

The description

$$< \alpha, F(\alpha) > \stackrel{\Delta}{=} \text{KP}$$

may be called a *logical knowledge representation* of the plant. For illustration purposes let us consider a very simple example:

$u = (u^{(1)}, u^{(2)})$, $y = (y^{(1)}, y^{(2)})$, $w \in \mathrm{R}^1$,

$\alpha_{u1} =$ "$u^{(1)} + u^{(2)} > 0$ ", $\alpha_{u2} =$ "$u^{(2)} > 2$ ",

$\alpha_{y1} =$ "$y^{(2)} < y^{(1)}$ ", $\alpha_{y2} =$ "$y^{(1)} + y^{(2)} = 4$ ",

$\alpha_{w1} =$ "$u^{(1)} - 2w + y^{(2)} < 0$ ", $\alpha_{w2} =$ "$u^{(2)} > y^{(1)}$",

$F_1 = \alpha_{u1} \wedge \alpha_{w1} \to \alpha_{y1} \vee \neg\alpha_{w2}$, $F_2 = (\alpha_{u2} \wedge \alpha_{w2}) \vee (\alpha_{y2} \wedge \neg\alpha_{u1})$,

$F_u = \alpha_{u1} \vee \alpha_{u2}$, $F_y = \neg\alpha_{y2}$.

After substituting $a$ in the place of $\alpha$, the forms $F_i(a)$, $F(a) = F(a_u, a_y, a_w)$ are *algebraic expressions* in two-valued logic algebra. The expressions $F(a)$ have the same form as the formulas $F(\alpha)$, e.g.

$$F_1(a_{u1}, a_{w1}, a_{w2}, a_{y1}) = a_{u1} \wedge a_{w1} \to a_{y1} \vee \neg a_{w2}.$$

The logic formulas $F_i(\alpha)$, $F_u(\alpha_u)$ and $F_y(\alpha_y)$ are special forms of the relations introduced in Sects. 6.1 and 6.2. Now the relation (6.3) has the form

$$R_i(u, w, y) = \{(u, w, y) \in U \times W \times Y : \; F_i[a(u, w, y)] = 1\}, \qquad (12.1)$$

$$i = 1, 2, ..., k.$$

For example, if in one-dimensional case $F_1(\alpha_1, \alpha_2) = \alpha_1 \to \alpha_2$ and $\alpha_1(u, y) =$ "$u^2 + y^2 \geq 4$", $\alpha_2(u, y) =$ "$y < 2u$", then such a fact defines a set of points lying inside the circle $u^2 + y^2 < 4$ or below the line $y = 2u$. Then an expert says that in our plant may appear only the pairs $(u, y)$ belonging to this set. The logical knowledge representation is then a specific form of a

relational knowledge representation, consisting of the relations $R_i$ which after elimination of $w$ may be reduced to one relation

$$R(u,y) = \{(u,y) \in U \times Y: \bigvee_{w \in W} [(u, w, y) \in R_1(u, w, y) \cap \ldots \cap R_k(u, w, y)]\}$$

or

$$R(u,y) = \{(u,y) \in U \times Y: \bigvee_{w \in W} [F(a) = 1]\}.$$

The input and output properties may be expressed as follows:

$$u \in D_u, \qquad y \in D_y$$

where

$$D_u = \{u \in U: \ F_u[a_u(u)] = 1\}, \tag{12.2}$$

$$D_y = \{y \in Y: \ F_y[a_y(y)] = 1\}. \tag{12.3}$$

For example, let

$$\alpha_1 = \alpha_{u1} = \text{“}u_2 > 2u_1\text{”}, \qquad \alpha_2 = \alpha_{u2} = \text{“}u_2 \geq 4u_1\text{”},$$

$$\alpha_3 = \alpha_{u3} = \text{“}u_1 < u_2 < 3u_1\text{”}$$

and

$$F_u = (\alpha_{u1} \wedge \neg \alpha_{u2}) \vee \alpha_{u3}.$$

Then

$$F_u = \text{“}u_1 < u_2 < 4u_1\text{”}.$$

The description with $F(a)$, $F_u(a_u)$, $F_y(a_y)$ may be called a *description on the logical level*. The expressions $F(a)$, $F_u(a_u)$ and $F_y(a_y)$ describe *logical structures* of the plant, the input property and the output property, respectively. The description at the logical level is independent of the particular meaning of the simple formulas. In other words, it is common for the different plants with different practical descriptions but the same logical structures. At the logical level our plant may be considered as a relational plant with input $a_u$ (a vector with $n_1$ zero-one components) and output $a_y$ (a vector with $n_3$ zero-one components), described by the relation

$$F(a_u, a_w, a_y) = 1 \tag{12.4}$$

(Fig. 12.1). The input and output properties for this plant corresponding to the properties $u \in D_u$ and $y \in D_y$ for the plant with input $u$ and output $y$ are as follows:

$$a_u \in S_u \subset \overline{S}_u, \qquad a_y \in S_y \subset \overline{S}_y$$

where $\overline{S}_u$, $\overline{S}_y$ are the sets of all zero-one sequences $a_u$, $a_y$, respectively, and

$$S_u = \{a_u \in \overline{S}_u : F_u(a_u) = 1\}, \qquad S_y = \{a_y \in \overline{S}_y : F_y(a_y) = 1\}. \tag{12.5}$$

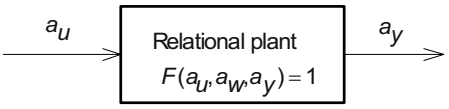


**Fig. 12.1.** Plant at logical level

## 12.3 Analysis and Decision Making Problems

The analysis and decision making problems for the relational plant described by the logical knowledge representation are analogous to those for the relational plant in Sect. 6.2. The analysis problem consists in finding the output property for the given input property and the decision problem is an inverse problem consisting in finding the input property (the decision) for the required output property.

**Analysis problem:** For the given facts $F(\alpha)$ and input property $F_u(\alpha_u)$ find the best property $F_y(\alpha_y)$ such that the implication

$$F_u(\alpha_u) \to F_y(\alpha_y) \tag{12.6}$$

is satisfied.

If it is satisfied for $F_{y1}$ and $F_{y2}$, and $F_{y1} \to F_{y2}$, then $F_{y1}$ is better than $F_{y2}$. The property $F_y$ is then the best if it implies any other property for which the implication (12.6) is satisfied. The best property $F_y$ corresponds to the smallest set $D_y$ in the formulation presented in Sect. 6.2.

**Decision problem:** For the given $F(\alpha)$ and $F_y(\alpha_y)$ (the property required by a user) find the best property $F_u(\alpha_u)$ such that the implication (12.6) is satisfied.

If it is satisfied for $F_{u1}$ and $F_{u2}$, and $F_{u2} \to F_{u1}$, then $F_{u1}$ is better than $F_{u2}$. The property $F_u$ is then the best if it is implied by any other property for which the implication (12.6) is satisfied. The best property $F_u$ corresponds to the largest set $D_u$ in the formulation presented in Sect. 6.2.

**Remark 12.1.** The solution of our problem may not exist. In the case of the analysis it means that there is a contradiction between the property $F_u(\alpha_u)$ and the facts $F(\alpha_u, \alpha_w, \alpha_y)$, i.e. the sequence $a_u$ such that $F_u(a_u) \wedge F(a_u, a_w, a_y) = 1$ does not exist. In the case of the decision making it means that the requirement $F_y$ is too strong. The existence of the solution will be explained in the next section. □

**Remark 12.2.** Our problems are formulated and will be solved on the logical level. Consequently they depend on the logical structures (the forms of $F$ and $F_y$ or $F_u$) but do not depend on the meaning of the simple formulas. The knowledge representation KP and the problem formulations may be extended for different variables, objects and sets (not particularly the sets of real number vectors) used in the description of the knowledge. For instance, in the example in the previous section we may have the following simple formulas in the text given by an expert:

$\alpha_{u1}$ = "operation $O_1$ is executed after operation $O_2$",

$\alpha_{u2}$ = "temperature is low",

$\alpha_{w1}$ = "pressure is high",

$\alpha_{w2}$ = "humidity is low",

$\alpha_{y1}$ = "state S occurs",

$\alpha_{y2}$ = "quality of product is sufficient".

Then the facts $F_1$ and $F_2$ in this example mean:

$F_1$ = "If operation $O_1$ is executed after operation $O_2$ and pressure is high then state S occurs or humidity is not low".

$F_2$ = "Temperature is low and humidity is low or quality is sufficient and operation $O_1$ is not executed after operation $O_2$". □

**Remark 12.3.** The possibilities of forming the input and output properties

are restricted. Now the sets $D_u$ and $D_y$ may be determined by the logic formulas $F_u(\alpha_u)$ and $F_y(\alpha_y)$ using the simple formulas $\alpha_u$ and $\alpha_y$ from the sequence of the simple formulas $\alpha$ used in the knowledge representation. □

## 12.4 Logic-algebraic Method

The solutions of the analysis and decision problems formulated in Sect. 12.3 may be obtained by using the so-called *logic-algebraic method* [18, 19, 23, 24, 25, 52]. Let us denote by $A$ the set of all zero-one sequences $a = (a_u, a_w, a_y)$. A set $S_a \subset A$ and a formula $F(\alpha)$ are said to be equivalent if $a \in S_a \leftrightarrow F(a) = 1$, i.e. $S_a = \{a \in A: \ F(a) = 1\}$. In this case one may say that $F(a)$ is determined by $S_a$.

For example, if $a = (a_1, a_2, a_3)$ and $S_a = \{(1,1,0), (1,0,1)\}$, then

$$F(\alpha) = (\alpha_1 \wedge \alpha_2 \wedge \neg\alpha_3) \vee (\alpha_1 \wedge \neg\alpha_2 \wedge \alpha_3).$$

The sets $S_u$ and $S_y$ in (12.5) are equivalent to $F_u(\alpha_u)$ and $F_y(\alpha_y)$, respectively. Let us denote by $R(a_u, a_w, a_y)$ the set equivalent to $F(a_u, a_w, a_y)$. Then

$$R(a_u, a_w, a_y) = \{(a_u, a_w, a_y): F(a_u, a_w, a_y) = 1\}. \tag{12.7}$$

Consequently, the analysis problem is reduced to the equivalent problem for the relational plant with the input $a_u$ and the output $a_y$, described by the relation (12.7).

**Equivalent analysis problem:** For the given relation (12.7) and the given set $S_u$ one should determine the smallest set $S_y$ satisfying the implication $a_u \in S_u \rightarrow a_y \in S_y$, i.e. the implication $F_u \rightarrow F_y$.

Taking into account the equivalent formulation stated above, it is easy to show that the analysis problem is reduced to solving the following algebraic equation:

$$\widetilde{F}(a_u, a_w, a_y) = 1 \tag{12.8}$$

with respect to $a_y$, where

$$\widetilde{F}(a_u, a_w, a_y) = F_u(a_u) \wedge F(a_u, a_w, a_y).$$

Now $F(a_u, a_w, a_y)$, $F_u(a_u)$ and $F_y(a_y)$ are algebraic expressions in two-valued logic algebra. If $S_y$ is the set of all solutions then $F_y$ is determined by $S_y$, i.e. $a_y \in S_y \leftrightarrow F_y(a_y) = 1$. For example, if $a_y = (a_{y1}, a_{y2}, a_{y3})$ and $S_y = \{(1,1,0),(0,1,0)\}$ then $F_y(\alpha_y) = (\alpha_{y1} \wedge \alpha_{y2} \wedge \neg\alpha_{y3}) \vee (\neg\alpha_{y1} \wedge \alpha_{y2} \wedge \neg\alpha_{y3})$.

The set of solutions $S_y$ (i.e. the set of all sequences $a_y$ for which there exists $\tilde{a} = (a_u, a_w)$ such that the equality (12.8) is satisfied) determines the output property $F_y(\alpha_y)$ to be defined. Let us order all sequences $a = (\tilde{a}, a_y)$ and denote by $a^{(i)}$ the $i$-th sequence. The block scheme of the *algorithm generating the solution* $S_y$ is illustrated in Fig. 12.2. The result $S_y$ should be presented in the form $F_y(\alpha_y)$ and transformed to the simplest form by using the known rules of logic-algebra.

Let us consider now the decision making problem and introduce two sets of the algebraic equations

$$\left.\begin{aligned} F(a_u, a_w, a_y) &= 1 \\ F_y(a_y) &= 1 \end{aligned}\right\}, \qquad \left.\begin{aligned} F(a_u, a_w, a_y) &= 1 \\ F_y(a_y) &= 0 \end{aligned}\right\}. \tag{12.9}$$

Denote by $S_{u1}$ and $S_{u2}$ the sets of all solutions with respect to $a_u$ of the first and the second set of equations, respectively.

**Theorem 12.1.** The solution $F_u$ of the decision making problem is determined by the set $S_u = S_{u1} - S_{u2}$.

*Proof*: If $F_y = 1$ then $a_u \in S_{u1}$; hence $a_u \in S_u$. If $F_y = 0$ then $a_u \in S_{u2}$; hence $a_u \notin S_u$. Consequently, $a_u \in S_u \leftrightarrow F_y = 1$ and $S_u$ is the largest set of elements $a_u$ such that the implication

$$(a_u \in S_u) \wedge F(a) \rightarrow F_y(a_y)$$

is satisfied. □

It is worth noting that the solution $D_u$ of the decision problem for the plant described by a relation $R(u, y)$, considered in Chap. 6 can be presented as follows:

$$D_u = \{u \in U: \ (u \in D_{u1}) \wedge (u \notin D_{u2})\} = D_{u1} - D_{u2} \tag{12.10}$$

where

$$\left.\begin{aligned} D_{u1} &= \{u \in U : \bigvee_{y \in D_y} [(u,y) \in R]\}, \\ D_{u2} &= \{u \in U : \bigvee_{y \in Y - D_y} [(u,y) \in R]\}. \end{aligned}\right\} \tag{12.11}$$

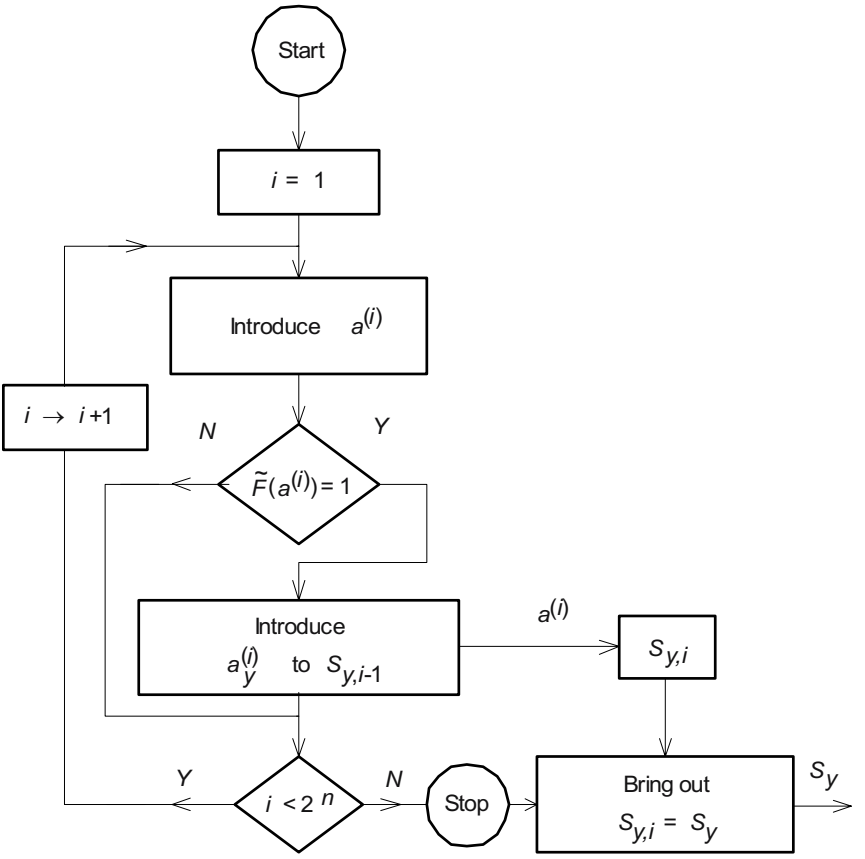


**Fig. 12.2.** Block scheme of algorithm generating the solution of analysis problem
Y – Yes, N – No

The sets $S_u$, $S_{u1}$, $S_{u2}$ introduced here correspond to the sets $D_u$ in (12.10) and $D_{u1}$, $D_{u2}$ in (12.11), respectively. Thus, the determination of the formula $F_u$ is reduced to the generation of the sets of solutions of the equa-

tions (12.9). The *algorithm generating the decision* $F_u$ is in a certain sense analogous to the former algorithm in the analysis problem and consists in the following: if the successive $a^{(i)}$ satisfies the first set in (12.9) then its part $a_u^{(i)}$ is included into $S_{u1,i-1}$ and if $a^{(i)}$ satisfies the second set in (12.9) then $a_u^{(i)}$ is included into $S_{u2,i-1}$ (Fig. 12.3). The result $S_u$ should be presented in the form $F_u(\alpha_u)$ and simplified if necessary.

The generation of the set $S_y$ in the analysis problem requires the testing of all sequences $a = (a_u, a_w, a_y)$ and the execution time may be very long for the large size of the problem. The similar computational difficulties may be connected with the solution of the decision problem. The generation of $S_y$ (and consequently, the solution $F_y$) may be much easier when the following decomposition is applied:

$$F_u \wedge F = \overline{F}_1(\overline{a}_0, \overline{a}_1) \wedge \overline{F}_2(\overline{a}_1, \overline{a}_2) \wedge \ldots \wedge \overline{F}_N(\overline{a}_{N-1}, \overline{a}_N) \qquad (12.12)$$

where $\overline{a}_0 = a_y$, $\overline{F}_1$ is the conjunction of all facts from $\widetilde{F}$ containing the variables from $\overline{a}_0$, $\overline{a}_1$ is the sequence of all other variables in $\overline{F}_1$, $\overline{F}_2$ is the conjunction of all facts containing the variables from $\overline{a}_1$, $\overline{a}_2$ is the sequence of all other variables in $\overline{F}_2$ etc. As a result of the decomposition the following *recursive procedure* may be applied to obtain $\overline{S}_0 = S_y$:

$$\overline{S}_{m-1} = \{\overline{a}_{m-1} \in S_{m-1} : \bigvee_{\overline{a}_m \in \overline{S}_m} [\overline{F}_m(\overline{a}_{m-1}, \overline{a}_m) = 1]\} \qquad (12.13)$$

where $S_m$ is the set of all $\overline{a}_m$, $m = N, N-1, \ldots, 1$, $\overline{S}_N = S_N$.

In such a way, the testing of all sequences $a = (a_u, a_w, a_y)$ for $k$ facts can be replaced by the testing of the significantly shorter sequences $(\overline{a}_{m-1}, \overline{a}_m)$ for $N$ successive subsets of the facts, and in these subsets the number of the facts to be tested may be much smaller than the number of all possible facts. Additional decreasing of the execution time in comparison with the direct method (without a decomposition) can be obtained by the application of a parallel processing in a multiprocessor system.

The recursive procedure (12.13) has two interesting interpretations.

**A. System analysis interpretation**

Let us consider the cascade of relational elements (Fig. 12.4) with input $\overline{a}_m$, output $\overline{a}_{m-1}$ (zero-one sequences), described by the relations

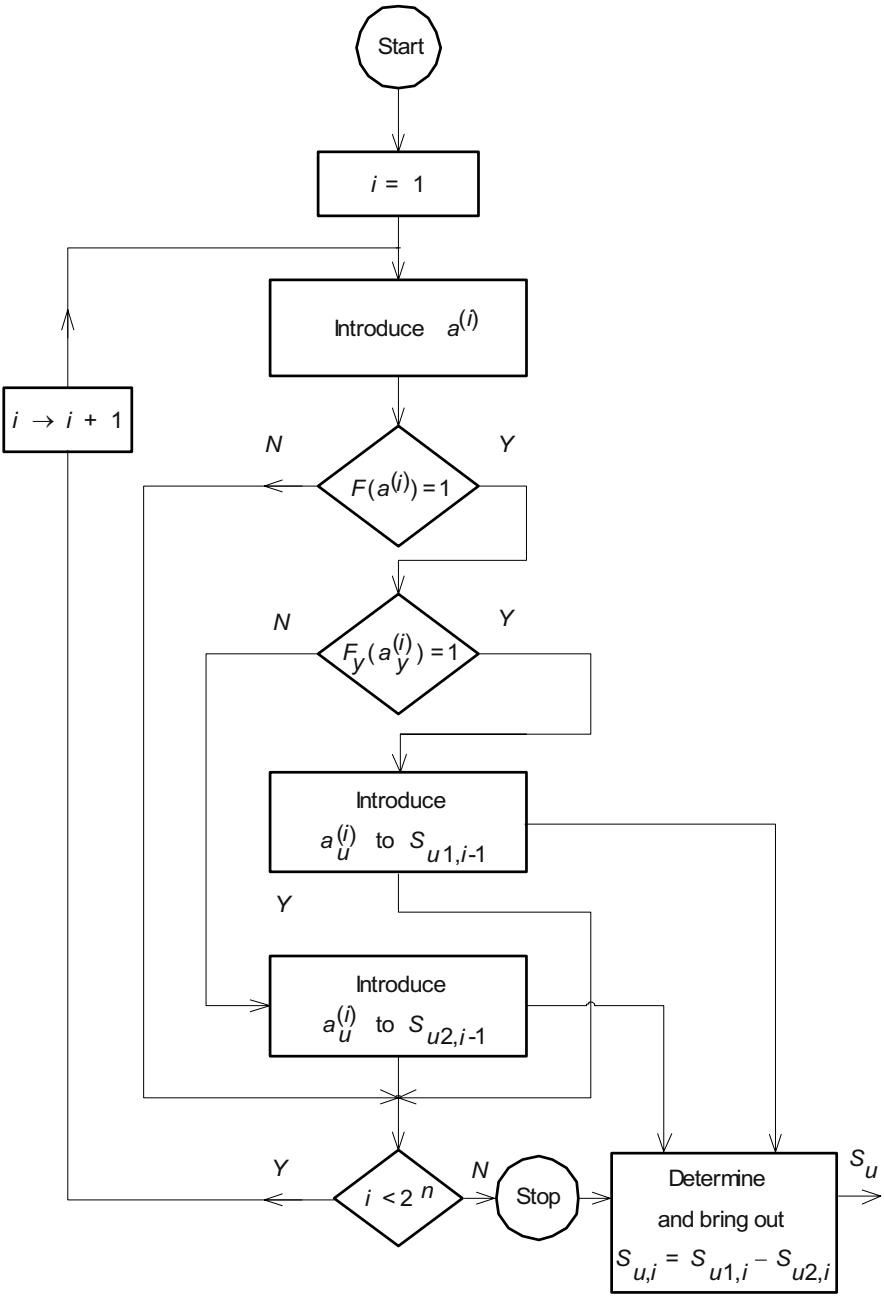


**Fig. 12.3.** Block scheme of algorithm generating the decision, N – No, Y – Yes

$\overline{F}_m(\overline{a}_{m-1}, \overline{a}_m) = 1$ $(m = N, N-1, ..., 1)$. Then $\overline{S}_{m-1}$ is the set of all possible outputs from the element $\overline{F}_m$ and $\overline{S}_0$ is the set of all possible outputs from the whole cascade.

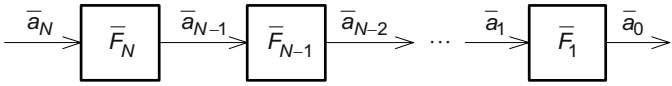


**Fig. 12.4.** Relational system

**B. Deductive reasoning interpretation**

The set $\overline{S}_{m-1}$ may be considered as the set of all elementary conclusions from $\overline{F}_N \wedge ... \wedge \overline{F}_m$, and $\overline{S}_0$ is the set of all elementary conclusions from the facts $F_u \wedge F$.

A similar approach may be applied to the decision problem. To determine $S_{u1}$ and $S_{u2}$ we may use the recursive procedure (12.13) with $F$ in (12.12) instead of $F_u \wedge F$ and with $\overline{a}_0 = (a_u, a_y)$. After the generation of $\overline{S}_0$ from (12.13) one can determine $S_{u1}$ and $S_{u2}$ in the following way:

$$S_{u1} = \{a_u : \bigvee_{a_y \in S_y} [(a_u, a_y) \in \overline{S}_0]\},$$

$$S_{u2} = \{a_u : \bigvee_{a_y \in \overline{S}_y - S_y} [(a_u, a_y) \in \overline{S}_0]\}$$

where

$$S_y = \{a_y \in \overline{S}_y : F_y(a_y) = 1\}$$

and $\overline{S}_y$ is the set of all $a_y$.

The *algorithm generating the decision* $F_u(\alpha_u)$ is then as follows:

1. Generation of the set $\overline{S}_0$ using the recursive procedure.
2. Testing of all sequences $(a_u, a_y)$ in $\overline{S}_0$: if in the successive $(a_u, a_y)$ the part $a_y$ satisfies the equation $F_y(a_y) = 1$ then $a_u$ should be included into $S_{u1}$, otherwise $a_u$ should be included into $S_{u2}$.
3. Determination of $S_u = S_{u1} - S_{u2}$ and $F_u$.

The main idea of the *logic-algebraic method* presented in this section for the generation of the solutions consists in replacing the individual rea-

soning concepts based on inference rules by unified algebraic procedures based on the rules in two-valued logic algebra. The results may be considered as a unification and generalization of the different particular reasoning algorithms for a class of systems with the logical knowledge representation for which the logic-algebraic method has been developed. The logic-algebraic method can be applied to the design of complex knowledge-based computer systems [52].

**Example 12.1** (analysis). The facts $\widetilde{F}$ are the following:
$F_1 = (\alpha_3 \vee \neg\alpha_1) \to \alpha_4$, $F_2 = (\neg\alpha_1 \wedge \alpha_7) \vee \neg\alpha_3$, $F_3 = (\alpha_9 \wedge \alpha_1) \to \alpha_2$,
$F_4 = (\alpha_4 \wedge \neg\alpha_7) \vee \alpha_5$, $F_5 = \alpha_6 \to (\alpha_4 \wedge \alpha_8)$, $F_6 = \alpha_2 \to (\neg\alpha_4 \wedge \alpha_6)$,
$F_7 = (\alpha_3 \wedge \alpha_2) \vee \alpha_{10}$, $\alpha_y = (\alpha_9, \alpha_{10})$.

It is not important which simple formulas from $\alpha_1 - \alpha_8$ are $\alpha_u$ and which fact from the set $\{F_1, F_2, F_4, F_5, F_6\}$ (not containing $\alpha_y$) is the input property. It is easy to see that
$\overline{F}_1(\overline{a}_0, \overline{a}_1) = F_3(a_1, a_2, a_9) \wedge F_7(a_2, a_3, a_{10})$, $\overline{a}_1 = (a_1, a_2, a_3)$,
$\overline{F}_2(\overline{a}_1, \overline{a}_2) = F_1(a_1, a_3, a_4) \wedge F_2(a_1, a_3, a_7) \wedge F_6(a_2, a_4, a_6)$,
$\overline{a}_2 = (a_4, a_6, a_7)$,
$\overline{F}_3(\overline{a}_2, \overline{a}_3) = F_4(a_4, a_5, a_7) \wedge F_5(a_4, a_6, a_8)$, $\overline{a}_3 = (a_5, a_8)$.

In our case $N = 3$, $S_N = \{(1,1),(1,0),(0,1),(0,0)\}$. According to (12.13) one should put successively the elements of $S_N$ into $\overline{F}_3$ and determine all 0-1 sequences $(a_4, a_6, a_7)$ such that $\overline{F}_3 = 1$. These are the elements of $\overline{S}_2$. In a similar way one determines $\overline{S}_1$ and finally $\overline{S}_0 = \{(0,1),(1,1)\}$. Then $F_y = (\neg\alpha_9 \wedge \alpha_{10}) \vee (\alpha_9 \wedge \alpha_{10}) = \alpha_{10}$. □

**Example 12.2** (decision making). The facts $F$ in the knowledge representation KP are the following:
$F_1 = \alpha_1 \wedge (\alpha_4 \vee \neg\alpha_6)$, $F_2 = (\alpha_2 \wedge \alpha_4) \to \alpha_6$, $F_3 = \neg\alpha_4 \vee \neg\alpha_3 \vee \alpha_5$,
$F_4 = \alpha_4 \wedge (\alpha_3 \vee \neg\alpha_5)$, $F_5 = (\alpha_4 \wedge \neg\alpha_2) \to \alpha_7$, $\alpha_u = (\alpha_1, \alpha_2)$,
$\alpha_y = (\alpha_6, \alpha_7)$.

Now $\overline{a}_0 = (a_u, a_y) = (a_1, a_2, a_6, a_7)$, $\overline{F}_1 = F_1 \wedge F_2 \wedge F_5$, $\overline{F}_2 = F_3 \wedge F_4$, $\overline{a}_1 = a_4$, $\overline{a}_2 = (a_3, a_5)$.

Using (12.13) (two steps for $m = 2,1$) we obtain $\overline{S}_0 = \{(1,1,1,1), (1,1,1,0), (1,0,1,1), (1,0,0,1)\}$. We can consider the differ-

ent cases of $F_y(\alpha_6, \alpha_7)$. It is easy to see that for $F_y = \alpha_6 \vee \alpha_7$ we have $S_y = \{(1,1),(1,0),(0,1)\}$, $S_{u1} = \{(1,1),(1,0)\}$, $S_{u2}$ is an empty set, $S_u = S_{u1}$ and $F_u = (\alpha_1 \wedge \alpha_2) \vee (\alpha_1 \wedge \neg\alpha_2) = \alpha_1$. If $F_y = \alpha_6$ then $F_u = \alpha_1 \wedge \alpha_2$, if $F_y = \alpha_7$ then $F_u = \alpha_1 \wedge \neg\alpha_2$, if $F_y = \alpha_6 \wedge \alpha_7$ then $S_{u1} = S_{u2}$, $S_u$ is an empty set and the solution $F_u$ does not exist.

The formulas $\alpha$ and the facts may have different practical sense. For instance, in the second example $u, y, c \in R^1$ and: $\alpha_1$ = "$u \le 3c$", $\alpha_2$ = "$u^2 + c^2 \le 1$", $\alpha_3$ = " pressure is high ", $\alpha_4$ = " humidity is low ", $\alpha_5$ = " temperature is less than $u + y + c$", $\alpha_6$ = "$y^2 + (c - 0.5)^2 \le 0.25$ ", $\alpha_7$ = "$-c \le y \le c$" for a given parameter $c$. For example, the fact $F_2$ means that: "if $u^2 + c^2 \le 1$ and humidity is low then $y^2 + (c - 0.5)^2 \le 0.25$", the fact $F_3$ means that: "humidity is not low or pressure is not high or temperature is less than $u + y + c$ ". The required output property $F_y = \alpha_6$ is obtained if $F_u = \alpha_1 \wedge \alpha_2$, i.e. if $u \le 3c$ and $u^2 + c^2 \le 1$. □

## 12.5 Neural Networks

Let us return to the concept of an adjusted reference model presented in Sect. 11.2. In that section the identification consisting in the adapting of the model $\bar{y} = \Phi(u, c)$ to the plant represented by the sequence of the measurement results

$$(u_1, y_1), (u_2, y_2), ..., (u_n, y_n) \tag{12.14}$$

has been considered. The adapting has consisted in successive changes of the values of the vector parameter $c \in C$ on the basis of current measurement results. The parametric problem of the *approximation* of the sequence (12.14) by the function $\Phi$ with a given form, consisting in a proper choice of the value $c$, may have also other practical applications. The function $\Phi$ may denote an approximate computation algorithm and the values $y_i$ in the sequence (12.14) may denote correct (exact) computation results for the given $u_i$. In particular, the function $\Phi$ may denote a proposed form of a static decision algorithm calculating decisions $\bar{y}$ for the

given $u$, with unknown parameters $c$. In this case $y_i$ may denote a sequence of decisions optimal in a certain sense for the given $u_i$. The source of the data (12.14) may have different interpretations. In the case of the identification they are the results of a real plant observations. In the case of the computation algorithm (decision algorithm) $y_i$ may denote the results obtained by applying an exact algorithm if it exists or the correct decisions given by a decision maker for the known $u_i$. The choice of proper values $c$ in the first case means the best approximation of the exact algorithm by the approximate computation algorithm $\Phi$, and in the second case – means the best approximation of the decision maker, i.e. the identification of the decision maker. Independently of different terms and interpretations for various concepts in this area, we can speak generally about an *approximation problem* and about the convergence of an approximation process in the case of recursive algorithms determining the changes of $c$. One can speak about a unified common approach to investigations of recursive algorithms of control, identification, recognition, adaptation, learning, adjustment etc. – as approximation algorithms based on uniform approximation methods [8, 103] (see remarks in Sect. 10.1).

In the further considerations $\Phi$ will be called an *approximation algorithm*, the sequence (12.14) – a *learning sequence*, a source of the results $y_i$ for the given $u_i$ – a *trainer*, and a process of changing $c$ – a *learning process*, as in the second part of Chap. 11. Denote by $\varphi(\bar{y}, y) \triangleq v$ an index of the approximation quality for the fixed $y$, which is to be minimized. Then the changes of $c$ mean an extremal control (a static optimization) for the plant with the input $c$, the output $v$ and the model

$$v = \varphi(\bar{y}, y) = \varphi[\Phi(u, c), y] \triangleq G(c, u, y).$$

The gradient algorithm in a closed-loop system for the fixed $u$ and $y$, or the *learning algorithm* is as follows

$$c_{n+1} = c_n - \gamma w_n \tag{12.15}$$

where

$$w_n = \operatorname*{grad}_c G(c,u,y)\Big|_{c=c_n} = \operatorname*{J}_c[\Phi(u,c)]\Big|_{c=c_n} \cdot \operatorname*{grad}_{\bar{y}} \varphi(\bar{y}, y)\,. \tag{12.16}$$

In the formula (12.16) $J$ denotes Jacobian matrix for the set of functions $\Phi$, i.e. the functions $y^{(i)} = \Phi^{(i)}(u, c)$ for components of the vector $y$ (see Sect. 4.5). In particular, for $\varphi(\bar{y}, y) = (\bar{y} - y)^{\mathrm{T}}(\bar{y} - y)$ we obtain $\operatorname{grad} \varphi(\bar{y}, y) = 2(\bar{y} - y)$. In comparison with the algorithm used in

Sect. 10.8, the simplest version with the single coefficient $\gamma$ instead of the matrix $K$ has been presented here. Because of the reasons mentioned in Sect. 10.8, the algorithm with trial steps is usually applied, and for assuring the convergence despite variation of the points $(u, y)$, i.e. when $(u_n, y_n)$ is put in the place of $(u, y)$ in (12.15) and (12.16) – the stochastic approximation algorithm with a decreasing coefficient $\gamma_n$ can be applied.

Let us consider a complex executor of the algorithm $\Phi$, consisting of two connected parts: $x = \Phi_1(u, d)$ and $\bar{y} = \Phi_2(x, e)$. In other words, the extremal control plant with the input $c = (d, e)$ is a cascade connection of the two parts. Then the task of the minimization

$$\min_c G(c, u, y) = \min_{d,e} \varphi\{\Phi_2[\Phi_1(u, d), e], y\}$$

may be decomposed into two stages. In the first stage, as a result of minimization

$$\min_e \varphi[\Phi_2(x, e), y] \triangleq v_2(x, y)$$

we obtain the relationship $e = \Psi_2(x, y)$. In the second stage, as a result of minimization

$$\min_d v_2[\Phi_1(u, d), y] \triangleq v_1(u, y)$$

we obtain the relationship $d = \Psi_1(u, y)$. Such an approach may be generalized for any number of units in a cascade connection. As a result of the decomposition a multistage minimization starting from the end, i.e. from the unit with the output $\bar{y}$ may be applied. It is an old and widely known way of the decomposition for multistage decision problems (see dynamic programming in Sect. 4.3). We shall return to this idea in Sect. 12.8 when the control of a complex plant with the cascade structure will be described. For the approximation problem and the complex approximation algorithm with a cascade structure considered here, such a decomposition means a learning process consisting in adjusting of the units successively from the end.

A neural network announced in the title of this section is a system performing the complex algorithm $\Phi$ with a cascade structure, in which separate units have a specific form, convenient for building and using. This form appeared for the first time several decades ago as so called *classifying network* in classification and pattern recognition problems. Consider an object represented by a vector of features $u \in U$, which should be classified to one of two classes. The classification (or recognition) problem in this case may consist in the determination of a hypersurface (a surface in a three-dimensional case) dividing the space $U$ into two parts. In the sim-

plest case it is a discriminating hyperplane $w^{\mathrm{T}}u + b = 0$ where $w$ is a vector of coefficients and $b$ is a one-dimensional parameter. The classification algorithm is then as follows:

$$\bar{y} = \mathrm{sign}(w^{\mathrm{T}}u + b) \tag{12.17}$$

where $\bar{y} = +1$ and $\bar{y} = -1$ denote the first and the second class, respectively. The algorithm should be completed by the statement to which class an object with $u$ lying on the discriminating hyperplane should be classified. The components of the vector $w$ are called *weight coefficients* (or shortly – *weights*) and the parameter $b$ is called a *threshold*. In order to increase the classification abilities one can use a complex system composed of the elements (12.17), i.e. with separate linear classifiers.

The algorithm may be generalized by introducing the form

$$\bar{y} = f(w^{\mathrm{T}}u + b) \tag{12.18}$$

where $f$ is a function which assigns real numbers $\bar{y}$ to real numbers $w^{\mathrm{T}}u + b$. The approximation algorithm $\Phi$ with a cascade structure may be composed of the elements (12.18) in such a way that each part (called a layer) inside this cascade contains a set of the elements (12.18) the inputs of which are outputs of the elements from the former layer and the outputs of which are inputs of the elements from the next layer. The components of the vector $u$ are the inputs of the first layer and the components of the vector $\bar{y}$ are the outputs of the last layer. A specific terminology is used here: the element (12.18) is called an *artificial neuron* (or shortly – a *neuron*), the algorithm $\Phi$ is called a *neuron-like algorithm*, and the multilayer system described above, i.e. the executor of the algorithm $\Phi$ is called a *neural network* (or *artificial neuron network*, or shortly – *neuron network*). Most often the term "executor of the algorithm" is used instead of the "algorithm" itself as it has occurred already many times in the book (e.g. a controller and a control algorithm) and as it has occurred here in the case of the neuron. The above terminology is justified by some associations with a set of neurons in a living organism. The learning process consists here in a proper changing of the vector $c$ the components of which are all the weights and the thresholds in the all neurons. The method of parameters $c$ adjustment in successive layers starting from the end is called here a *back propagation* method. It is worth noting that similar recursive procedures starting from end and concerning other problems have similar names, e.g. an inference algorithm starting from the end and used for solving problems based on a logical knowledge representation is sometimes called a *backward procedure*.

A simple example of a neural network with three layers in which neu-

rons are marked by triangles is illustrated in Fig. 12.5.

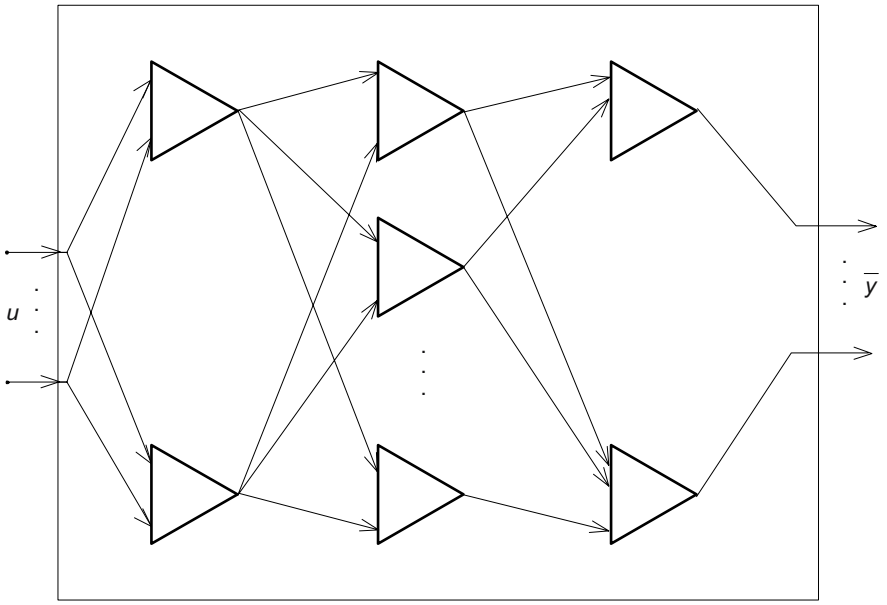


**Fig. 12.5.** Example of neural network

A network with a great number of layers and neurons has the following abilities and advantages important for the application in control systems:
1. It can approximate with a high degree of accuracy various algorithms from a wide class of computation algorithms (including decision algorithms) via an adjustment (adapting) to a concrete algorithm by a learning process.
2. It enables to replace an analytical determination of an exact problem solving algorithm by solutions given by a trainer and imitates the trainer's acting as a result of a learning process. In this case an exact algorithm may not exist.
3. It can be computerized not only by a software but also by a hardware as a net of artificial neurons, i.e. as a *neurocomputer* in which the execution of a program is replaced by a signal processing in the net.

In the next section a brief review of different possible applications of neural networks in control systems will be presented. Wide descriptions of neural networks and their applications may be found in the books [61, 87, 97, 106].

## 12.6 Applications of Neural Networks in Control Systems

Numerous concepts concerning the neural networks applications in control systems may most often be reduced to several basic ideas [21, 87] which we shall characterize in this section. They do not require a precise theoretical analysis and their abilities in specific cases are usually investigated by simulations. It follows from the main idea and a character of an approach applied here that a control algorithm or its modifications are not determined analytically by using a given description of the plant but are assumed in the form of a universal neuron-like algorithm. The quality of the control for the given plant description cannot be investigated analytically either, because of a very complicated form of the neuron-like algorithm $\Phi$. From this point of view the situation is similar to that described for a fuzzy controller. The basic ideas will be explained here for a static neuron-like algorithm considered in Sect. 12.5, although it is possible to build and use neural networks with memory, convenient for dynamical systems modelling. The explanation will consist in the specification of a role of the neuron-like algorithm (its place in the control system) and data used in a learning algorithm.

### 12.6.1 Neural Network as a Controller

The neuron-like algorithm may be applied as a control algorithm. In this version the input of the neural network is the state of the plant $x$ (in a closed-loop system) or a disturbance acting on the plant (in an open-loop system) and the output of the network is the control decision $\overline{u}$ put at the input of the plant. Then, the neuron-like algorithm takes a role of a control algorithm with a given form in which undetermined parameters $c$ (the weights and thresholds) should be adapted to the plant in an adaptation process (see Sect. 11.4) called a learning process in this case. A simple example of such algorithm in a closed-loop system, i.e. of a neuron-like controller has been given in Sect. 5.5. Three basic ways of learning are used:

1. Learning based on an evaluation of a quality index.
2. Learning with an exact algorithm as a generator of the learning sequence.
3. Learning with an external trainer (an expert) as a generator of the learning sequence.

The ways **1** and **3** are analogous to the corresponding concepts described for learning systems in Chap. 11. The way **1** is illustrated by Fig. 12.6 where NA denotes the neuron-like algorithm, L is the learning algorithm and W denotes the determination of the quality index. It is a typical con-

cept of an adaptation via the adjustment of controller parameters, described in Chap. 11 and consisting in changing the parameters $c$ based on a periodical evaluation of the quality index $Q$.

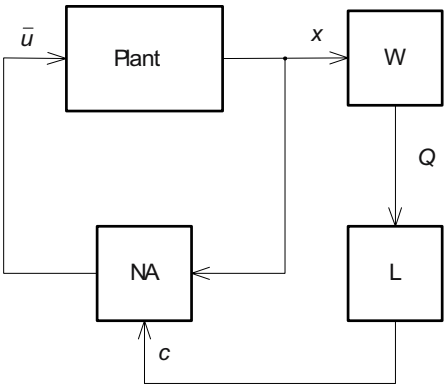


**Fig. 12.6.** Control system with neuron-like algorithm and learning based on quality evaluation

The ways **2** and **3** are illustrated in Fig. 12.7 where LSG denotes Learning Sequence Generator.

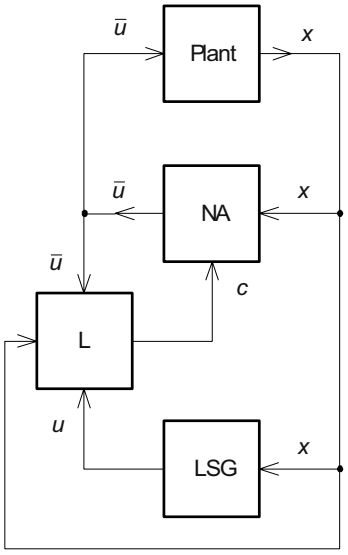


**Fig. 12.7.** Learning with a generator of learning sequence

In the case **2**, the learning sequence generator is an executor of an exact control algorithm. It means that for the given plant description the exact

control algorithm has been determined but it is too complicated to be used for a steady control and it is used during a learning process only. After finishing this process the neural network acts approximately as the exact algorithm.

Finally, in the case **3** the decisions $\bar{u}$ for varying data $x$ are given by an expert taking a role of a trainer. Unlike in the prescriptive approach described formerly for fuzzy controllers, now the expert does not give a description of the knowledge on how to control but he himself proposes concrete control decisions. This is an essential difference between the neuron-like and fuzzy controllers.

### 12.6.2 Neural Network in Adaptive System

A neural network may be a part of an adaptator in an adaptive control systems with the adaptation via identification (see Sects. 11.2 and 11.3). One can consider two cases of such an application of the neural network:
1. Neural network as a plant model used in the identification process.
2. Neural network as an approximate designing algorithm.

Ad 1. As a learning sequence, a sequence of the measured input and output values of the plant is used here to successive changes of the values of weights and thresholds in the network. The application of this concept in the adaptive control system is rather limited. It follows from the fact that in this case the plant model is not presented in an analytical form with the parameters $c$ the current values of which obtained as a result of the identification might be put into the designing algorithm $a = H(c)$ considered in Chap. 11. It is however possible to use the neural network as a model replacing a real plant during the process of adaptation via adjustment of controller parameters. Then, during the adjusting process the basic controller would not control a real plant but its model in the form of the neural network, and would be adapted to this model. It is worth noting that such a concept would be rather complicated, as it contains two learning processes: the adaptation of the network to a real plant and the adaptation of the basic controller to the network.

Ad 2. In this case the neural network takes a role of a neuron-like algorithm approximating the exact designing algorithm $a = H(c)$, i.e. in the successive adaptation periods $c_m$ is determined as a result of the current identification by a traditional identifier and is put at the input of the neural network which finds the approximate values $\bar{a}_m$ of the parameters in the basic control algorithm. In this case three ways of the learning may be indicated as in Sect. 12.6.1: the learning based on the evaluation of the qual-

ity index and the learning with a generator of the learning sequence (Fig. 12.8) in the form of an exact algorithm or in the form of a trainer-expert. The second way means that the exact designing algorithm $a = H(c)$ has been determined but is too complicated to a steady use and it is used during the learning process only. The third way means that in the learning process for the successive values $c_m$ an expert gives the values $a_m$ of the basic control algorithm, accepted as correct values.

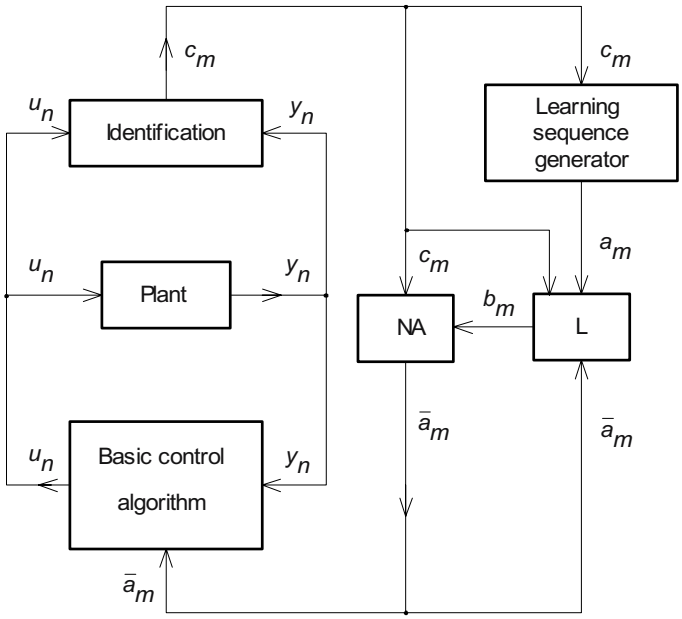


**Fig. 12.8.** Scheme of adaptive control system with neuron-like algorithm NA; $b_m$ – varying parameters of algorithm NA

## 12.7 Decomposition and Two-level Control

Usually a *decomposition of a problem* means its division (or partition) into interconnected simpler problems. Most often a solution of a problem obtained via the decomposition is an approximate solution only in comparison with the exact solution obtained by a direct approach. Different kinds of a decomposition occurred already several times in our considerations. For example, the problem of the determination of a control decision for a plant with unknown parameter, considered in Sect. 7.3, has been decom-

posed into two problems: an estimation of the unknown parameter and the determination of the decision for the known parameter. The determination of a multistage decision process by using the dynamic programming method presented in Sect. 4.3 has meant a *decomposition with respect to time*, i.e. a division of the problem into the determination of decisions in the successive stages. The main idea of the adaptive control has consisted in a partition of the control algorithm into two parts: the basic algorithm and the algorithm improving the performance of the basic algorithm. Finally, in this chapter we have applied a decomposition of a set of facts in the logical knowledge representation, which has given a recursive procedure for solving analysis and decision making problems. The decomposition is not always reasoned by computational aspects, which has already been mentioned in the connection with an adaptive control and which will be mentioned in this chapter as well.

We shall now present shortly a decomposition of the static optimization problem (or the extremal control problem). This decomposition leads to a two-level control or the partition of the control algorithm into two partial algorithms executed at two levels of the control. A decomposition of the partial algorithms into subalgorithms leads to a multi-level control system. Let us note that the *decomposition of the determination* of a control algorithm may lead to the *decomposition of the control*, i.e. of executing of the partial algorithms. The separate execution of the partial algorithms may be reasoned not only by the limited abilities of the executors but also by reliability aspects.

Let us consider the static multi-input plant $y = \Phi(u, z)$ in which the control decision $u$ and the external disturbance $z$ are vectors and $y$ is a single value which is to be minimized. A direct solution gives the control algorithm in an open-loop system

$$u^* = \arg \min_u \Phi(u, z) \stackrel{\Delta}{=} \Psi(z). \tag{12.19}$$

Let us divide the vector $u$ into two subvectors $u_1, u_2$ and consider two optimization problems

$$u_1^* = \arg \min_{u_1} \Phi(u_1, u_2, z) \stackrel{\Delta}{=} H_1(u_2, z), \tag{12.20}$$

$$u_2^* = \arg \min_{u_2} \Phi[H_1(u_2, z), u_2, z] \stackrel{\Delta}{=} H_2(z). \tag{12.21}$$

Substituting (12.21) into (12.20) we obtain the control algorithm $u^* = \Psi(z)$. The decomposition of the problem (12.19) into the problems (12.20) and (12.21) does not have to be a computational operation only but may mean the decomposition of the algorithm $\Psi$ into two partial algorithms $H_1$ and

$H_2$, executed in a two-level control system (Fig. 12.9), in which after introducing $z$, an executor at the upper level finds $u_2$ according to the algorithm $H_2$ and next, an executor at the lower level finds $u_1$. The decisions $u_1$ and $u_2$ can be transformed to the input of the plant by the both executors, respectively as it is shown in Fig. 12.9 or by the lower level executor if only this executor directly controls the plant. In the case of incomplete knowledge of the plant the algorithms $H_1$ and $H_2$ cannot be determined in advance by a designer but the values $u_1^*$ and $u_2^*$ are found *step by step* in the successive approximation process with the participation of the plant, i.e. in the process of the extremal control in a closed-loop system (see Sect. 10.8).

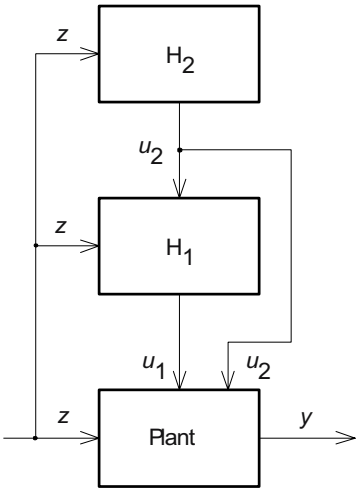


**Fig. 12.9.** Simple scheme of two-level control system

Then in one step of the successive approximation procedure at the upper level, the whole control process determining the value $u_1$ for the given current value $u_2$ is executed at the lower level. Of course, the value $z$ has to be constant during the searching of the extremum ($u_1^*$ and $u_2^*$). The problem becomes more complicated if there is a constraint $u \in D_u$ which cannot be divided into two independent constraints for $u_1$ and $u_2$. Possible trials of a decomposition in this case give a control differing from the control (12.19).

The presented concept has a practical sense (except the computational aspect) mainly when the upper level operations are slow in comparison

with fast operations of the lower level, or when the plant has a complex structure. Assume that $z = (z_1, z_2)$ and the period of changing $z_2$ (upper level period) is a multiple of the period of changing $z_1$ (lower level period). Denote by $z_{2,m}$ the value $z_2$ in the $m$-th upper period and by $z_{1,mn}$ the value $z_1$ in the $n$-th lower period inside the $m$-th upper period; $n = 1, 2, ..., N$ where $N$ is a number of the lower periods in one upper period. The control from the upper level will be slow (exactly speaking, constant during one upper period) in the following cases:

1. $u_2^*$ does not depend on $z_1$. Then

$$u_{1,mn} = H_1(u_{2,m}, z_{1,mn}, z_{2,m}), \qquad u_{2,m} = H_2(z_{2,m}).$$

2. In the algorithm $H_2$ we put $z_{1,m1}$ in the place of $z_{1,mn}$, i.e.

$$u_{2,m} = H_2(z_{1,m1}, z_{2,m}),$$

that is the changes of $z_{1,mn}$ at the upper level are not taken into account.

3.

$$u_{2,m} = H_2(\tilde{z}_{1,m}, z_{2,m})$$

where $\tilde{z}_{1,m}$ is the mean value of the previous observations. The modification of this approach may consist in the minimization by the upper level of the mean value of the quality index $y_{mn}$.

Let us now assume that the plant $y = \Phi(u, z)$ is a complex system consisting of $M+K$ ordered units. The outputs $y_1, y_2, ..., y_M$ of the units 1, 2, ..., $M$ are the outputs of the system as a whole, i.e. $y = (y_1, y_2, ..., y_M)$, and the outputs of the other units are the inputs of at least one of the units in the first part. The units of the first part may be called *partial* or *local plants*. The other units present the connections between the local plants. Let us denote the inputs of the local plants by $u_1 = (u_{11}, u_{12}, ..., u_{1M})$, and the inputs of the other units by $u_2 = (u_{21}, u_{22}, ..., u_{2K})$. The local plants are described by the relationships

$$y_i = \Phi_i(u_{1i}, u_2, z), \qquad i = 1, 2, ..., M.$$

Of course, not all components of the vector $u_2$ and (or) not all components of the vector $z$ must occur in a single local plant. For the complex plant we introduce a global quality index $\bar{y} = F(y_1, y_2, ..., y_M)$ which is a function of the local indexes $y_1, y_2, ..., y_M$ and which should be minimized. It may be e.g. the sum of the local indexes with weight coefficients. The plant

with the input $u = (u_1, u_2)$ and the output $\bar{y}$, described by the equation

$$\bar{y} = F[\Phi_1(u_{11}, u_2, z), ..., \Phi_M(u_{1M}, u_2, z)] \triangleq \Phi(u_1, u_2, z) \qquad (12.22)$$

is called a *global plant*. For this plant, the concept of the decomposition described above may be applied, that is, according to the notations introduced here, one may accept $u_1$ as the lower level decisions and $u_2$ as the upper level decisions. Then, according to (12.20) and (12.21)

$$u_{1i}^* = \arg \min_{u_{1i}} \Phi_i(u_{1i}, u_2, z) = H_i(u_2, z), \qquad i = 1, 2, ..., M, \qquad (12.23)$$

$$u_2^* = \arg \min_{u_2} F[H_1(u_2, z), H_2(u_2, z), ..., H_M(u_2, z)] = H_2(z). \qquad (12.24)$$

The formula (12.23) denotes a *local optimization* or the optimization of the local plants at the lower level for the fixed value $u_2$, and the formula (12.24) denotes a *global optimization* at the upper level. A block scheme of the two-level control system is presented in Fig. 12.10 where for simplicity, interconnections between the plants are not marked. The variables $u_2$ are sometimes called *coordinating variables* and the control at the upper level is called a *coordination* of the local controls. The coordinating variables may also occur in the constraints for $u$ if it is not possible to divide this constraint into independent constraints for the individual local plants. The decomposition becomes much more complicated when it is not possible to divide the control decisions $u$ into two parts $u_1$ and $u_2$ described above. Eventual trials of a decomposition lead then to solutions which are sometimes called approximate although in many cases it is difficult to evaluate an effect of this approximation. Various problems, cases and methods concerned with a decomposition, multi-level control and related problems of complex control systems are described in [64].

Additionally, let us pay attention to three problems:

1. Sometimes the control decisions for individual plants in a complex plant mean amounts of resources or sizes of tasks assigned to these plants (see Chapter 13), and the sum of these resources or the sizes of tasks is fixed. Then one can apply so called *aggregation*, i.e. the partition of the set of plants into subsets treated from the upper level as units (as a whole). Then the upper level control consists in the distribution of resources or tasks among the subsets and the lower level control consists in the distribution of the quantities obtained form the upper level among individual plants in the subset.

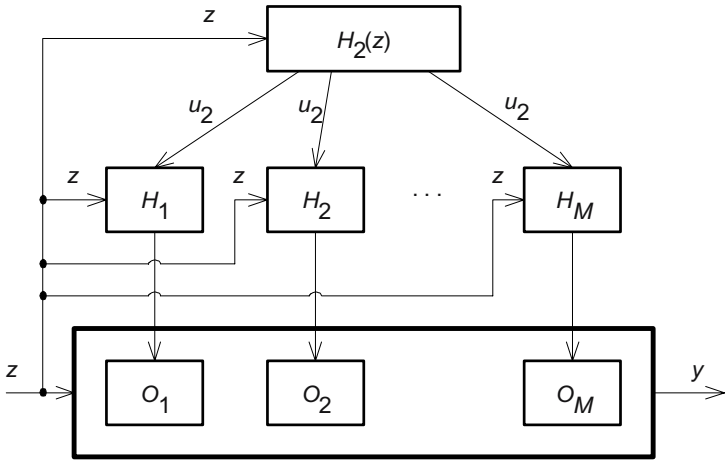


**Fig. 12.10.** Two-level control system with local plants

2. The period (interval) in which the optimal decisions $u = \Psi(z)$ are determined and executed may be long, e.g. it may be one cycle in a multi-cycle production process. If at the beginning of the cycle the decisions $u$ are set indirectly with the help of some variables $v$ (e.g. $u$ is a temperature set by a voltage $v$ in an electric heater) then the value $u$ may change during the cycle because of varying disturbances. Then it is better to set $u$ as a required value of a controller (e.g. a temperature controller) which during a cycle will change $v$ in order to stabilize $u$. Consequently, we have two-level control system with a long period of the static optimization at the upper level and a short period of the control (or with a continuous control) at the lower level. The outputs of the upper level control algorithm (i.e. of the optimizer), or the components of the vector $u$ are required values of local controllers installed in the process considered. The decomposition has here a natural character, i.e. the lower problem of the control and the upper problem of the static optimization are solved independently.
3. For an uncertain complex plant one can apply the approaches described for a single uncertain plant, in particular one can apply an adaptation via identification. In this case a decomposition of the identification is important. The concept of so called *global identification* of a complex plant with a given structure has proved to be useful in this case [14, 16].

## 12.8 Control of Complex Plant with Cascade Structure

The control of a complex plant with determined local plants and corresponding local decision algorithms does not always have to be a two-level control. Let us consider a static plant with a cascade structure (i.e. a series connection of local plants) presented in Fig. 12.11 where the local plants are described by the relationships

$$y_i = \Phi_i(u_i, y_{i-1}, z_i), \qquad i = 2, 3, ..., M, \tag{12.25}$$

$$y_1 = \Phi_1(u_1, z_1).$$

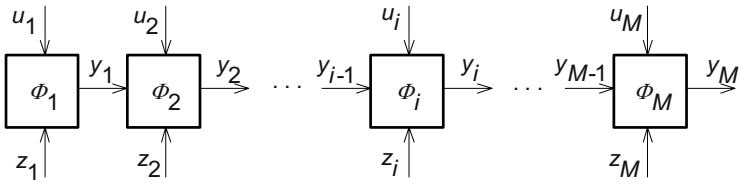


**Fig. 12.11.** Plant with cascade structure under consideration

In general, the all variables occurring here are vectors. This is a case of a complex production process, frequently met in practice, in which a product obtained in the $i$-th unit is used as a raw material in the next unit. The quality index $Q_M$ of the complex plant may concern $y_M$ only, i.e. $Q_M = \varphi(y_M)$, or may be the sum of local quality indexes

$$Q_M = \sum_{i=1}^{M} \varphi_i(y_i).$$

The quality evaluation may also contain the control decision $u$ and the disturbance $z$. Then $Q_M = \varphi(y_M, u_M, z_M)$ or

$$Q_M = \sum_{i=1}^{M} \varphi_i(y_i, u_i, z_i). \tag{12.26}$$

**Decision making (control) problem:** For the given $\Phi_i$, $z_i$, $\varphi_i$ ($i = 1, 2, ..., M$) or $\varphi$ one should find the control decisions $u_1, u_2, ..., u_M$.

Let us consider the quality index in the form (12.26). Substituting (12.25) yields

$$Q_M = \sum_{i=1}^{M} \varphi_i[\Phi_i(u_i, y_{i-1}, z_i), u_i, z_i] = \sum_{i=0}^{M-1} g_i(u_{i+1}, y_i, z_{i+1})$$

where

$$g_i(u_{i+1}, y_i, z_{i+1}) = \varphi_{i+1}[\Phi_{i+1}(u_{i+1}, y_i, z_{i+1}), u_{i+1}, z_{i+1}].$$

Let us note that this is a multistage decision problem similar to that in the case of an optimal control for a discrete dynamical plant considered in Sect. 4.3. The index $i$ of a successive local plant corresponds to the successive moment $n$ for the dynamical plant and, in a way similar to that presented in Sect. 4.3, the problem solving may be decomposed and performed successively starting from the end, i.e. one can apply a recursive procedure called *dynamic programming* in Sect. 4.3.

Let us introduce the notation

$$V_{M-i}(y_i, z_{i+1}, z_{i+2}, ..., z_M) = \min_{u_i, u_{i+1}, ..., u_M} \sum_{j=i}^{M} g_j(u_{j+1}, y_j, z_{j+1}).$$

For the last plant we determine

$$u_M^* = \arg \min_{u_M} g_{M-1}(u_M, y_{M-1}, z_M) \stackrel{\Delta}{=} \Psi_M(y_{M-1}, z_M)$$

and

$$V_1(y_{M-1}, z_M) = g_{M-1}[\Psi_M(y_{M-1}, z_M), y_{M-1}, z_M].$$

For $i = M-1$

$$V_2(y_{M-2}, z_{M-1}, z_M) = \min_{u_{M-1}, u_M} \{g_{M-2}(u_{M-1}, y_{M-2}, z_{M-1}) + g_{M-1}(u_M, y_{M-1}, z_M)\} = \min_{u_{M-1}} \{g_{M-2}(u_{M-1}, y_{M-2}, z_{M-1}) + V_1[\Phi_{M-1}(u_{M-1}, y_{M-2}, z_{M-1}), z_M]\}.$$

As a result of the minimization we obtain

$$u_{M-1}^* \stackrel{\Delta}{=} \Psi_{M-1}(y_{M-2}, z_{M-1}, z_M)$$

and

$$V_2(y_{M-2}, z_{M-1}, z_M) = V_2[\Phi_{M-2}(u_{M-2}, y_{M-3}, z_{M-2}), z_{M-1}, z_M].$$

Continuing this procedure we determine successively

$$u_i^* = \Psi_i(y_{i-1}, z_i, z_{i+1}, ..., z_M) \qquad i = M-2, ..., 2,$$

$$u_1^* = \Psi_1(z_1, z_2, ..., z_M).$$

The control is then executed by local controllers with the control algorithms $\Psi_i$. At their inputs, except the disturbances the value of the output

from the preceding plant is put as well (Fig. 12.12). The control is executed successively in time, starting from the beginning, i.e. the decisions $u_1, u_2, ..., u_M$ are successively determined and executed.

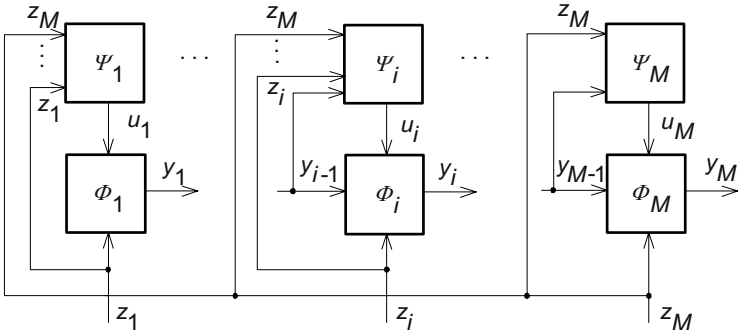


**Fig. 12.12.** Control system for the plant with cascade structure

For the quality index $Q_M = \varphi(y_M, u_M, z_M)$ we apply the analogous recursive procedure. Let us introduce the notation

$$V_{M-i}(y_i, z_{i+1}, ..., z_M) = \min_{u_i, .., u_M} \varphi(y_M, u_M, z_M).$$

For the last plant we determine

$$u_M^* = \arg \min_{u_M} \varphi[\Phi_M(u_M, y_{M-1}, z_M), u_M, z_M] \stackrel{\Delta}{=} \Psi_1(y_{M-1}, z_M)$$

and

$$V_1(y_{M-1}, z_M) = \varphi\{\Phi_M[\Psi_1(y_{M-1}, z_M), y_{M-1}, z_M], u_M, z_M\}.$$

For $i = M-1$

$$V_2(y_{M-2}, z_{M-1}, z_M) = \min_{u_{M-1}} V_1[\Phi_{M-1}(u_{M-1}, y_{M-2}, z_{M-1}), z_M].$$

As a result of the minimization we obtain

$$u_{M-1}^* = \Psi_{M-1}(y_{M-2}, z_{M-1}, z_M).$$

Continuing this procedure we determine the all local control algorithms

$$u_i^* = \Psi_i(y_{i-1}, z_i, z_{i+1}, ..., z_M), \qquad i = M-1, M-2, ..., 2,$$

$$u_1^* = \Psi_1(z_1, z_2, ..., z_M).$$

## 12.9 Control of Plant with Two-level Knowledge Representation

Components of an uncertain complex control plant may be described in the form of a knowledge representation differing from traditional functional models (see Sect. 6.1). Then the analysis and decision making problems based on a complex or *distributed knowledge* arise. The description of the distributed knowledge contains knowledge representations of separate parts and the description of the systems structure. To the determination of control decisions one may apply a direct approach or a decomposition leading to separate control algorithms for particular parts. The decomposition may also concern the process of the knowledge validation and updating, described in Chap. 11. To characterize the problems occurring in this area, let us consider a plant with the two-level structure (Fig. 12.13) described in the form of a relational knowledge representation [28, 33, 40]. In a production system, products of the lower level units may be raw materials for the upper level unit. The components of the vectors $y$, $y_1$, $y_2$, ..., $y_M$ may denote features of the products at the upper and the lower level, respectively, and $u_1$, ..., $u_M$ may denote features of raw materials used at the lower level, which can be chosen as control decisions. The components of the vectors $z$, $z_1$, ..., $z_M$ denote other variables which for the fixed inputs can influence the set of possible outputs in the separate units.

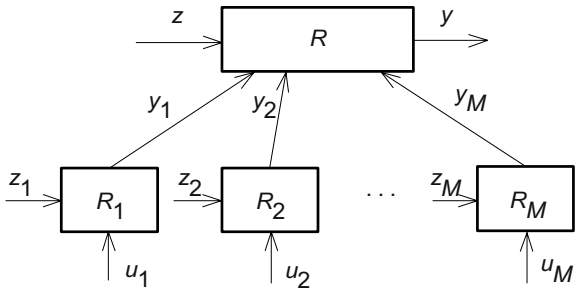


**Fig. 12.13.** Complex plant with two-level knowledge representation

The knowledge representations for the upper and the lower level are the following

$$R(y, y_1, y_2, ..., y_M; z) \subset Y \times Y_1 \times Y_2 \times ... \times Y_M \times Z, \tag{12.27}$$

$$R_i(u_i, y_i, z_i) \subset U_i \times Y_i \times Z_i, \qquad i = 1, 2, ..., M \tag{12.28}$$

where $U, Z$ and $Y$ denote the respective vector spaces. The decision problem for the complex plant as a whole can be formulated in such a way as in Chap. 6.

**Decision making (control) problem:** For the given $R$; $R_1$, ..., $R_M$; $z$; $z_1$, ..., $z_M$ and the set $D_y \subset Y$ determining a user's requirement one should find the largest set $D_u \subset U$ such that the implication

$$u \in D_u \rightarrow y \in D_y$$

is satisfied.

To apply a direct approach to the problem solving it is needed to determine the relation $\hat{R}(u, y, \bar{z})$ where

$$u = (u_1, ..., u_M) \in U, \quad U = U_1 \times U_2 \times ... \times U_M,$$

$$\bar{z} = (z_1, ..., z_M; z) \in \bar{Z}, \quad \bar{Z} = Z_1 \times Z_2 \times ... \times Z_M \times Z.$$

The relation $\hat{R}(u, y, \bar{z})$ can be obtained via the elimination of the variables $y_1, ..., y_M$:

$$\hat{R}(u, y, \bar{z}) = \{(u, y, \bar{z}) \in U \times Y \times \bar{Z} : \bigvee_{\bar{y} \in \bar{Y}} [(y, \bar{y}; \bar{z}) \in R(y, \bar{y}; z)]$$

$$\wedge \bigwedge_{i \in \overline{1, M}} (u_i, y_i, z_i) \in R_i(u_i, y_i, z_i)\}$$

where

$$\bar{y} = (y_1, y_2, ..., y_M) \in \bar{Y}.$$

Then, according to (6.19) and (6.18)

$$D_u = \{u \in U: \ D_y(u, \bar{z}) \subseteq D_y\} \stackrel{\Delta}{=} \bar{R}(\bar{z}, u) \tag{12.29}$$

where

$$D_y(u, \bar{z}) = \{ y \in Y: (u, y, \bar{z}) \in \hat{R}(u, y, \bar{z}) \}.$$

The relation $\bar{R}(\bar{z}, u)$ is a relational control algorithm for our plant in an open-loop control system.

If the determination of $\hat{R}$ and (or) a direct execution of the control according to the algorithm $\bar{R}$ are too difficult, one can apply a decomposition of our problem into two levels, and at the lower level – into particular

plants of this level. Then, for the upper level we determine the largest set $\overline{D}_y \subset \overline{Y}$ for which the implication

$$\overline{y} \in \overline{D}_y \rightarrow y \in D_y$$

is satisfied, i.e. we solve the known decision problem for the plant at the upper level, described by the relation (12.27). Then we approximately decompose the set $\overline{D}_y$ into the separate sets $D_{yi} \subset Y_i$ for the particular variables $y_i$ or, more precisely speaking, we determine the sets $D_{y1}$, ..., $D_{yM}$ in such a way that the set $D_{y1} \times D_{y2} \times ... \times D_{yM} \stackrel{\Delta}{=} \widetilde{D}_y$ is the largest set satisfying the relation

$$\widetilde{D}_y = D_{y1} \times ... \times D_{yM} \subseteq \overline{D}_y .$$

Of course, the determination of the sets $D_{yi}$ so that the set $\widetilde{D}_y$ satisfies the above condition is in general not unique. The second stage concerns the lower level and consists in the determination for the particular plants of the largest sets $D_{ui} \subset U_i$ satisfying the implication

$$u_i \in D_{ui} \rightarrow y_i \in D_{yi} ,$$

by using the relation (12.28).

Using the known way presented by the formulas (6.19) and (6.18) we find

$$D_{ui} = \{u_i \in U_i : \; D_{yi}(u_i, z_i) \subseteq D_{yi}\} \stackrel{\Delta}{=} \overline{R}_i(z_i, u_i) \tag{12.30}$$

where

$$D_{yi}(u_i, z_i) = \{y_i \in Y_i : \; (u_i, y_i, z_i) \in R_i(u_i, y_i, z_i)\}.$$

In such a way we obtain the local relational control algorithms (12.30) for the particular plants at the lower level. It is worth noting that the substitution of the algorithms (12.29) by the algorithms (12.30) for $i = 1, ..., M$ gives in general a worse effect in the sense that

$$D_{u1} \times D_{u2} \times ... \times D_{uM} \subset D_u$$

is in general smaller than the set (12.29), i.e. as a consequence of the decomposition we may obtain a smaller set of possible decisions.

The presented approach can be also applied to the relations with unknown parameters considered as values of random variables or uncertain variables, to the logical knowledge representation and to the learning process in which the knowledge validation and updating can concern the

knowledge of the plant as a whole or the distributed knowledge [28, 40, 53].

Finally, let us note that the control of a complex plant may concern the plant with components described by different forms of a knowledge representation, including traditional mathematical models. Then problems of a uniform approach to the analysis and control with a possible decomposition for plants with a *hybrid knowledge representation* arise.

# 13 Control of Operation Systems

Now we shall present basic problems and control algorithms for a specific control plant, namely for an operation system or a complex of operations. It may be a complex of technical operations in a production system (in a manufacturing process) or a complex of computational operations in a computer system. Control problems for the operation systems are concerned with a design of control systems for so called discrete manufacturing processes considered as a complex of technical operations (e.g. control of an assembly process) and with a control of multicomputer or multiprocessor systems. The problems of the control of operation systems have also an important role in a design of management systems, in particular, for a *project management*.

## 13.1 General Characteristic

By a *complex of operations* (or *operation system*) in a wide sense of the word we call a complex system composed of *operations*, i.e. some activities which are characterized by an initial moment, a final moment and a duration time. In "input-output" system, a system structure is determined by the interconnections between the components (the parts) of the system, i.e. outputs of some parts are inputs of others. In a complex of operations, a system structure is defined by time interrelations and successions of the moments, i.e. some operations cannot start until the other ones are finished. The structure of the complex may be described in the form of a graph, in which branches denote the operations and nodes denote the starting and finishing moments. For example the graph presented in Fig. 13.1 means that the operations $O_3$ and $O_4$ may start after finishing the operation $O_1$, and the operation $O_5$ may start after finishing the operations $O_2$ and $O_3$. The node $W_1$ denotes the starting moment and the node $W_2$ – the finishing moment of the whole complex.

The operation usually consists in performing (executing) a determined *task* which requires certain *resources*. Among them one can distinguish basic resources called *executors* (e.g. tools needed to perform technical

operations) and additional resources (e.g. fuel or additional materials necessary to perform a technical operation). Roughly and generally speaking a control of the complex of operations consists in distributed in time assigning of resources to tasks or on the contrary, satisfying determined requirements, in particular – optimizing a quality index, e.g. minimizing the execution time of the complex or the amount of resources used for the execution of the complex, with different kinds of constraints. Most often we consider the following control functions, interconnected in one system:
1. Control of tasks and resources distribution (or *allocation*).
2. Control of a succession of operations (scheduling problems).
3. Control of a service of tasks.
4. Control of the execution process for individual operations.

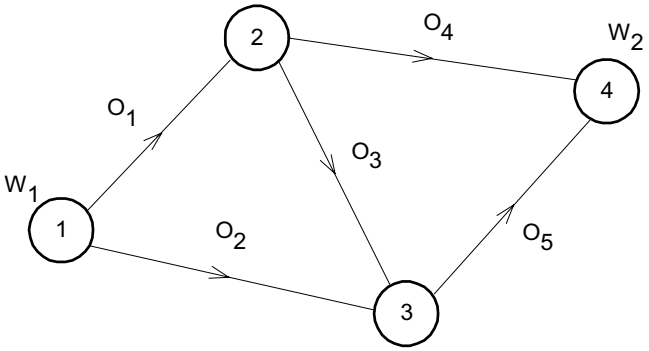


**Fig. 13.1.** Example of complex operation structure

In the control of operation system problems, the operation is considered as a whole from the upper level of the control, e.g. it is one cycle of a production process. In point 4 we take into account that during an execution of one operation the control is required, e.g. the temperature control during one cycle of a production process.

Generally formulated control problems for a complex of operations may have various practical applications depending on a practical character of a real complex. For a complex of production or service operations with a technical character, they are problems concerning a control of so called *discrete manufacturing processes* and (or) of service operations, e.g. the control of an assembly process, of processing of elements in a production of tools and machines, control of a transport, a diagnostic process, storing of materials etc. In the case of computational and decision operations they are problems concerning the control of performing complex computation and decision algorithms by a computer system, that is problems connected with the design of computer *operating systems*.

From the general control theory point of view, the complexes of operations are specific control plants in which the relationship between the output $y$ and the input $u$ (or the plant model) may have a special form, and in which specific control goals and constraints can occur. Most often the operations and complexes of operations may be treated as static plants. Sometimes, however, in a sequence of operations (e.g. in the assembly process considered in Sect. 13.6), relations between the successive stages in the sequence can occur and consequently, the whole complex should be considered as a dynamical plant. It is worth reminding that speaking about a control we have in mind a decision making which may be a single operation or may be distributed in time.

The methods and algorithms of the operation systems control are based on general methods and algorithms developed in the area called *operational researches*, strictly connected with a control and management sciences. To operational researches and their application to control and management a wide literature is devoted (e.g. [4, 105]). In some cases presented in the next sections, the considerations will be limited to problem formulations only, in such a form to which known and widely described algorithms can be applied. For example, the algorithm for solving classical transportation problem and the algorithms in integer programming and in related problems reduced to this programming.

## 13.2 Control of Task Distribution

This section is concerned with the control of the complex of parallel operations containing unknown parameters in the relational knowledge representation. The complex of parallel operations is considered here as a specific control plant. The control consists in a proper distribution of the given size of a task taking into account the execution time of the whole complex. It may mean the distribution of raw material in the case of a manufacturing process or a load distribution in a group of parallel computers. In the deterministic case where the operations are described by functions determining the relationship between the execution time and the size of the task, the optimization problem consisting in the determination of the distribution that minimizes the execution time of the complex may be formulated and solved (see e.g. [13]).

In order to formulate and solve the task distribution or allocation problem it is necessary to introduce models of individual operations. For our purpose it is sufficient to handle static models formulating the relationship

between the execution time and the size of the task assigned to the operation

$$T_i = \varphi_i(u_i, c_i), \qquad i = 1, 2, ..., k \tag{13.1}$$

where $T_i$ denotes the execution time of the *i*-th operation, $u_i$ is the size of the task assigned, $c_i$ is a vector of parameters and $k$ is a number of operations in the complex. Of course, all the names one can apply to the operation or to the executor of the operation, speaking about the model of the executor, the execution time of a task assigned to the executor etc. The functions (13.1) are increasing functions of $u_i$ and $\varphi_i(0, c_i) = 0$. Under the assumption that all operations start in the same time, the moment of finishing the whole complex is the moment of finishing the last operation. Then the execution time of the complex

$$T = \max_i T_i = \max\{\varphi_1(u_1, c_1), \varphi_2(u_2, c_2), ..., \varphi_k(u_k, c_k)\} \triangleq \Phi(u, c) \tag{13.2}$$

where $u^{\mathrm{T}} = [u_1 \;\; u_2 \;\; ... \;\; u_k] \in \overline{U}$ is a control vector, $T = y$ is the control plant output and $c = [c_1 \;\; c_2 \;\; ... \;\; c_k]$ is the matrix of parameters. The set $\overline{U} \subset R^k$ is determined by the constraints

$$\bigwedge_i u_i \geq 0, \qquad \sum_{i=1}^{k} u_i = U \tag{13.3}$$

where $U$ is the total size of the task to be distributed among the operations. The parameters $c = z$ may be also treated as disturbances which can be measured and which can change in successive periods or cycles of the control in which the successive global task to be distributed appears. The complex of parallel operations may be considered as a specific decision (control) plant (Fig. 13.2) with the input vector $u$ and a single output $y = T$, described by the function $\Phi(u, c)$ determined according to (13.2). The problem of the time optimal distribution considered here is then the problem of static optimization to which all methods described in former chapters can be applied, regardless the specific terms and the form of the plant model. However, the special form of the model (13.2) and the constraints (13.3) require additional considerations concerning this problem. Let us assume that $U$ may be divided in any way to $u_1, ..., u_k$, i.e. $u_i \in R^1$. Then we speak about the problem with a *continuous* division of the total

size $U$, distinguishing it from a *discrete* division where $U$ and $u_i$ denote numbers of some elementary tasks.

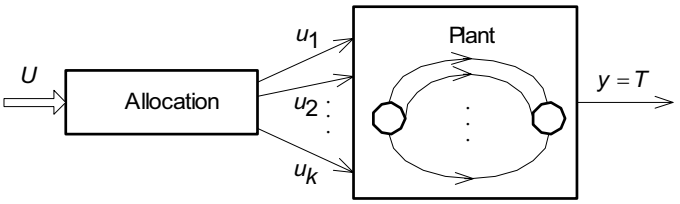


**Fig. 13.2.** Complex of parallel operations as a decision plant

**Decision making (control) problem:** For the given functions $\varphi_i$, parameters $c_i$ ($i \in \overline{1,k}$) and the value $U$ one should determine the distribution $u^*$ minimizing the execution time (13.2) subject to constraints (13.3).

It is easy to show that if all operations start at the same time then $u^*$ satisfies the following equation:

$$T_1(u_1) = T_2(u_2) = \ldots = T_k(u_k) = T_{\min}.$$

That is why the optimal distribution or allocation problem in this case is often called a *load balancing* problem. From the property of $u^*$ mentioned above, the algorithm of the optimal distribution determination follows:
1. From the equations (13.1) one should determine the relationships

$$u_i = \varphi_i^{-1}(T_i, c_i). \tag{13.4}$$

2. One should solve the equation

$$\sum_{i=1}^{k} \varphi_i^{-1}(T, c_i) = U \tag{13.5}$$

with respect to $T$.

3. One should substitute the solution of this equation $T \triangleq T_{\min}$ into (13.4) and in such a way find the optimal distribution

$$u_i^* = \varphi_i^{-1}(T_{\min}, c_i) \triangleq \Psi_i(c). \tag{13.6}$$

Consider as an example the following models of the operations:

$$T_i = \frac{u_i^{\alpha}}{c_i}, \qquad \alpha > 0, \quad c_i > 0. \tag{13.7}$$

According to (13.4)

$$u_i = (c_i T_i)^{\frac{1}{\alpha}}.$$

Then the equation (13.5) takes the form

$$T^{\frac{1}{\alpha}} \sum_{i=1}^{k} c_i^{\frac{1}{\alpha}} = U.$$

From this equation we obtain

$$T_{\min} = \frac{U^{\alpha}}{(\sum_{i=1}^{k} c_i^{\frac{1}{\alpha}})^{\alpha}}$$

and

$$u_i^* = c_i^{\frac{1}{\alpha}} \; U \; (\sum_{i=1}^{k} c_i^{\frac{1}{\alpha}})^{-1} = \Psi_i(c), \qquad i = 1, ..., k. \tag{13.8}$$

The set of relationships (13.8) may be presented in a unified form

$$u^* = \Psi(c).$$

In particular, for the linear models $(\alpha = 1)$, the optimal distribution is as follows:

$$u_i^* = c_i \; U \; (\sum_{i=1}^{k} c_i)^{-1}.$$

For more complicated models (13.1) it is not possible to obtain a result in analytical form and for finding the optimal distribution suitable numerical methods should be used. As it was already said, the determination of the decision $u$ may be repeated in successive cycles. In these cycles the values $c_i$ may change because they depend on features of the global tasks which appear at the beginning of the cycle (e.g. on properties and parameters of a raw material which influence the time of the material processing in a production process) and on executors productivity which may change in the successive cycles as well. If by $\bar{c}_n = (c_{1n}, c_{2n}, ..., c_{kn})$ we denote a vector of parameters $c_i$ in the $n$-th cycle then the determination of the con-

trol decisions will consist in finding the distribution $u_n^{\mathrm{T}} = [u_{1n}\ u_{2n}\ \ldots\ u_{kn}]$ minimizing the time $T$ of this cycle for the given $\bar{c}_n$. In more complicated cases the values $\bar{c}_n$ must be determined by using the measurement results of the respective variables characterizing the global task and the executors (Fig. 13.3). These variables or directly the parameters $\bar{c}_n$ take a role of disturbances in an open-loop control system. For the models (13.7), the control algorithm $\Psi_i(c)$ is defined by the formula (13.8).

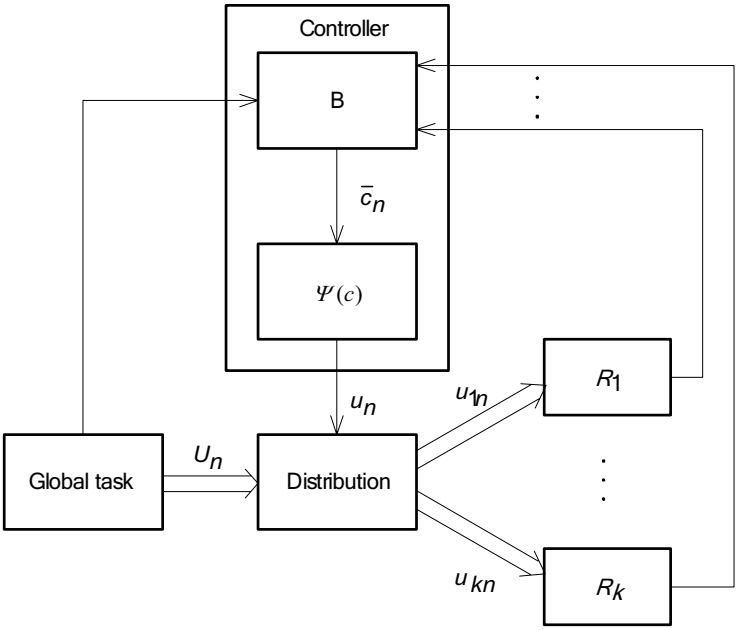


**Fig. 13.3.** Control system for task distribution, B – block determining $\bar{c}_n$

For the complex of operations as a specific control plant with the input $u$ and the output $y = T$, other problems and algorithms described in previous chapters can be considered. If $c = z$ is a random disturbance with a known probability distribution, one can state and formulate a static optimization problem considered in Chap. 7 and consisting in the determination of the distribution $u^*$ minimizing the expected value of the execution time $T$, that is

$$Q = \mathrm{E}(\underline{T}) = \int_Z \max_i \varphi_i(u_i, z_i) f_z(z) dz \qquad (13.9)$$

where $z = (z_1, z_2, ..., z_k)$. Assume that the random variables $\underline{z}_i$ are independent for different $i$. Then it is more convenient to handle probability distributions in the form of the distribution function

$$F_i(z_i) = P(\underline{z}_i \le z_i) \tag{13.10}$$

where $P$ denotes the probability that the random variable $\underline{z}_i$ takes a value not greater than $z_i$. Knowing (13.10) and the relationship $\underline{T}_i = \varphi_i(u_i, \underline{z}_i)$, one can determine the distribution function $F_{iT}(\lambda)$ for the random variable $\underline{T}_i$, i.e.

$$F_{iT}(\lambda) = P(\underline{T}_i < \lambda).$$

Consequently, the formula (13.9) becomes

$$Q = \mathrm{E}(\max \underline{T}_i) = \int_0^\infty \lambda d(\prod_{i=1}^{k} F_{iT}(\lambda)).$$

The relationship $Q(u_1, u_2, ..., u_k)$ can be obtained in an analytical form for special probability distributions only. It will be shown in an example of an analogous problem for the resource distribution, which will be presented in the next section.

If the parameters of the control plant $c_i$ are unknown, the control in a closed-loop system or extremal control presented in Chap. 10 can be applied. Then the convergence condition for the minimum searching process (i.e. the convergence of $u_n$ to $u^*$) should be determined and satisfied. Since the function $\Phi$ (13.2) describing the plant is not differentiable with respect to $u$, the gradient method is useless and in practice the method of trial steps, that is a step by step correction of the distribution $u_n$ based on the evaluation of current execution time in the $n$-th period should be applied.

Distribution or allocation problems, both for the task distribution described in this section and for resource allocation which will be presented in the next section, can be much more complicated, connected with a decomposition and multilevel control, and concerned with an adaptive control using the methods and algorithms of so called *global identification* of the complex of operations [13, 14, 15, 17, 57, 59].

Finally, let us consider shortly a case of a *discrete distribution* and assume that the global task is a set of $N$ identical undividable elementary tasks with the size $u_e$. For example, it may be a set of identical portions of a raw material to be distributed among devices processing the material or a

set of identical elementary programs to be distributed among computers [46]. Now $U = N \cdot u_e$ and $u_i = n_i \cdot u_e$ where $n_i$ denotes the number of elementary tasks assigned to the $i$-th executor. If the number of all elementary tasks is large and $u_e$ is small then one can solve the problem for a continuous distribution, i.e. find $u_i^* \in R^1$ as it has been presented above, and then find $n_i$ such that $n_i \cdot u_e$ is the nearest to the value $u_i^*$. If such an approximation is not acceptable, the problem from the beginning should be treated as a discrete one, as it will be presented in Sect. 13.4.

## 13.3 Control of Resource Distribution

The problem of resource distribution or allocation in a complex of parallel operations is analogous to that of task distribution considered in the previous section. Let $u_i$ in (13.1) denotes the amount of a resource assigned to the $i$-th operation (or to the $i$-th executor). We assume that $\varphi_i$ are decreasing functions of $u_i$ and $\varphi_i(u_i, c_i) \to \infty$ for $u_i \to 0$. The decision making or control problem is the same as in Sect. 13.2, i.e. one should find the distribution $u_1^*, ..., u_k^*$ minimizing the execution time (13.2) subject to constraints (13.3). As in the former case, when the optimal distribution is applied then $T_1 = T_2 = ... = T_k = T_{\min}$. The algorithm generating the optimal allocation is the same as in Sect. 13.2, i.e. it consists of steps in which the formulas (13.4), (13.5), (13.6) occur.

As an example let us consider the following models of the operations, analogous to the models (13.7):

$$T_i = \frac{c_i}{u_i^\alpha}, \qquad \alpha > 0, \quad c_i > 0.$$

According to (13.4)

$$u_i = \left(\frac{c_i}{T_i}\right)^{\frac{1}{\alpha}}. \tag{13.11}$$

Solving the equation (13.5) we obtain

$$T_{\min} = \frac{\left(\sum_{i=1}^{k} c_i^{\frac{1}{\alpha}}\right)^\alpha}{U^\alpha}$$

and after substituting $T_i = T_{\min}$ into (13.11) we have the optimal distribution algorithm

$$u_i^* = c_i^{\frac{1}{\alpha}} U \left(\sum_{i=1}^{k} c_i^{\frac{1}{\alpha}}\right)^{-1} = \Psi_i(c).$$

The time-optimal resource distribution is more complicated for more complex structures of the complex of operations (differing from the parallel structure), e.g. such as presented in Fig.13.1. Under the assumption that the same kind of a resource is assigned to each operation, one should now determine the values of $u_i^*$ for each operation so as to minimize the execution time of the whole complex and that the sum of $u_i$ for all operations executed at the same time is equal to $U$. The considerations concerning this problem are limited here to a basic idea and a sketch of a procedure giving the optimal distribution.

Let us consider a structure described by a graph which has one initial node (i.e. there are no branches ending in this node) and one final node (i.e. there are no branches starting from this node). Assume that this is a graph without cycles, plane and connective, which means that for each pair of nodes there exists a path connecting these nodes. Let us order the nodes in such a way that the later node has the greater index, e.g. as it is shown in Fig. 13.1. The execution time of the whole complex may be divided into separate intervals by the moments corresponding to the nodes. Then one can introduce the following sets:

1. $Q_j$ – the set of indexes of the operations which may be executed in the $j$-th interval, $j = 1, 2, ..., m$. In the example in Fig. 13.1 we have three intervals: from 1 to 2 ($j = 1$), from 2 to 3 ($j = 2$), and from 3 to 4 ($j = 3$); $Q_1 = \{1, 2\}$, $Q_2 = \{2, 3, 4\}$, $Q_3 = \{4, 5\}$.

2. $P_i$ – the set of indexes of the intervals in which the $i$-th operation may be executed. In the example $P_1 = \{1\}$, $P_2 = \{1, 2\}$, $P_3 = \{2\}$, $P_4 = \{2, 3\}$, $P_5 = \{3\}$.

Let $u_i$ be a flow intensity of a resource supplied to the $i$-th operation (e.g. energy, fuel, material, a financial resource etc.) which is assumed to be constant during the execution of the operation. The constraint concerning $u_i$ is then as follows:

$$\bigwedge_{j \in \overline{1,m}} \left(\sum_{i \in Q_j} u_i = U\right) \tag{13.12}$$

which means that the sum of $u_i$ for all operations executed in the *j*-th interval should be equal to the total value *U*. In order to find $u^*$ it would be necessary to determine the model of the plant, i.e.

$$T = \Phi(u_1, u_2, ..., u_k) \tag{13.13}$$

where *k* denotes the number of operations in the complex and for simplicity the parameters *c* are omitted, and then the function (13.13) should be minimized with the constraint (13.12). Unfortunately, for the given structure and the given functions $\varphi_i(u_i)$, the function (13.13) cannot be presented in an analytical form, by a formula convenient for the minimization. There exist however algorithms enabling to obtain the value *T* for the given values $u_i$ and consequently, for the given $T_i$. They are algorithms determining so called *critical path* [105], i.e. such a path connecting the initial and final nodes for which the sum of the times $T_i$ for the operation on this path is the greatest. This sum is equal to the execution time *T* of the whole complex. The general procedure for the determination of the optimal resource distribution consists in finding successive approximations of the solution. For the given approximation of the optimal allocation $u^*$ one determines the critical path and the corresponding time *T*, and on this basis one corrects the approximation and finds the next one. The successive approximation process can be performed in a closed-loop control system where the control computer using the current values $T_i$ determines the next approximation of the optimal distribution.

The problem is modified if a transfer of resources between operations during their execution is possible and acceptable. Denote by $x_i(t)$ one-dimensional operation state at the moment *t* and introduce operation models determining the relationship between the velocity of the state changes and the value $u_i$:

$$\dot{x}_i = \Phi_i(u_i).$$

Let $x_i^*$ denote so called *operation size*, which means that the operation is finished when the state $x_i^*$ is reached. If $u_i$ is constant and $x_i(0) = 0$ then

$$T_i = \frac{x_i^*}{\Phi_i(u_i)} \triangleq \varphi_i(u_i, x_i^*). \tag{13.14}$$

Let us divide $x_i^*$ into parts $x_{ij}$ executed in the separate intervals of the set $P_i$. Then the complex described can be considered as a sequence of sub-

complexes with a parallel structure, executed in the separate time intervals. In the $j$-th interval the parts of operations belonging to the set $Q_j$ are executed. Consequently, our problem may be decomposed into two subproblems:

1. The optimization of the subcomplexes of parallel operations in the successive time intervals using the formulas (13.14) with constraints (13.12). As a result we obtain the functions $\bar{u}_j(\bar{x}_j)$ and $\bar{T}_{j\min}(\bar{x}_j)$ where $\bar{u}_j$ denotes an optimal distribution in the $j$-th subcomplex, $\bar{x}_j$ is a set of $x_{ij}$ for all $i \in Q_j$, $\bar{T}_{j\min}(\bar{x}_j)$ is the minimal execution time of the $j$-th subcomplex.

2. Minimization of $T$ with respect to all $x_{ij}$, i.e. finding $\bar{x}_j^*$ $(j = 1, ..., m)$ which minimize

$$T = \sum_{j=1}^{m} \bar{T}_{j\min}(\bar{x}_j)$$

and satisfy the constraints

$$\bigwedge_{i \in \overline{1,k}} \left( \sum_{j \in Q_i} x_{ij} = x_i^* \right)$$

which means that the whole size of each operation will be executed in the successive intervals. Referring to Sect. 12.7 one can say that this is another example of decomposition which in this case consists in the division of the execution time of the whole complex into time intervals and consequently, in the decomposition of the complex into a sequence of subcomplexes. The variables $x_{ij}$ take here a role of coordinating variables.

If $c = z$ is a random disturbance with a known probability distribution then we may formulate the problem of the determination of a resource distribution minimizing the expected value of $T$ [17, 57]. For the complex of parallel operations the problem solving is analogous to that for the case of a task distribution. Knowing the distribution function (13.10) one should determine $Q$ (13.9), i.e.

$$Q = \mathrm{E}(\max \underline{T}_i) = \int_0^{\infty} \lambda d\left(\prod_{i=1}^{k} F_{iT}(\lambda)\right) = Q(u_1, u_2, ..., u_k). \tag{13.15}$$

The determination of the function (13.15) and its minimization subject to the constraints (13.3) may be connected with great computational difficulties and it is possible to obtain an analytical solution in simple cases only.

**Example 13.1.** Let us determine the resource distribution optimal in a probabilistic sense for two operations with the models

$$T_1 = \frac{z_1}{u_1}, \qquad T_2 = \frac{2z_2}{u_2}$$

where the random variables $\underline{z}_1$ and $\underline{z}_2$ have the same exponential probability density

$$f_z(z) = ae^{-az}$$

for $z > 0$ and $f_z(z) = 0$ for $z \le 0$. According to (13.15), after substituting $u_2 = U - u_1$ and some transformations we obtain

$$Q = \frac{U^2 + 3u_1^2}{au_1(U - u_1)(U + u_1)}.$$

Equating to zero the derivative of $Q$ with respect to $u_1$ gives the equation

$$3u_1^4 + 6U^2u_1^2 - U^4 = 0.$$

A unique root of this equation satisfying the constraint $0 < u_1 < U$ is the following

$$u_1^* = U\sqrt{\frac{2}{3}\sqrt{3} - 1}.$$

Then

$$u_2^* = U - u_1^* = U(1 - \sqrt{\frac{2}{3}\sqrt{3} - 1}).$$ □

## 13.4 Control of Assignment and Scheduling

In the problem of a discrete task distribution considered at the end of Sect. 13.2, the size $u_i$ of the task assigned to the *i*-th executor has been reduced to a number of elementary tasks, which can be obtained by rounding off the result of a continuous distribution. If the total number of elementary tasks is not large, such an approximation may be not acceptable and the problem from the beginning should be considered as a discrete one. Then one should find the numbers of elementary tasks $n_1, n_2, ..., n_k$ minimizing

the execution time of the global task

$$T = \max \{n_1 \tau_1, n_2 \tau_2, ..., n_k \tau_k\}$$

and satisfying the constraint $n_1 + n_2 + ... + n_k = N$ where $N$ is a total number of elementary tasks to be distributed, $n_i$ is a number of elementary tasks assigned to the $i$-th executor and $\tau_i$ is an execution time of an elementary task by the $i$-th executor. The problem of minimization of a function with respect to variables which may take only integer values is called an *integer programming*, and in the case when the variables may have only the values 0 or 1 – a *zero-one programming*. Many problems from the area of operational researches are reduced to so called *combinatorial problems* in which one should choose a decision from a finite set of possible decisions. Consequently, a combinatorial problem is reduced to the integer programming for which suitable algorithms (such as branch and bound algorithm and its different modifications) and corresponding computer programs have been developed. They may be found in the literature cited in Sect. 13.1 and will not be presented here. The next considerations will be limited to formulations of basic assignment and scheduling problems for which the known algorithms may be applied.

Let us return to the problem of task distribution among executors and consider a set of tasks $\mathcal{Z} = \{Z_1, Z_2, ..., Z_M\}$. Unlike the case mentioned formerly, the tasks do not have to be identical. That is why in this case we do not speak about a distribution of the tasks among the executors but about an *assignment* of the determined tasks to the executors from the set $\mathcal{R} = \{R_1, R_2, ..., R_k\}$. For the fixed ordering of the tasks and the executors it is more convenient to handle the set of task indexes $J = \{1, 2, ..., M\}$ and the set of executor indexes $I = \{1, 2, ..., k\}$. If the execution time is assumed as a quality index (as in former considerations), the problem consists in a partition of the set $J$ into separate subsets assigned to the particular executors and the optimal partition is a partition minimizing the execution time for the whole set of the tasks. Denote by $\tau_{ij}$ the execution time of the $j$-th task by the $i$-th executor and introduce the numbers $c_{ij} \in \{0,1\}$ determining the assignment, i.e. $c_{ij} = 1$ if the $j$-th task is assigned to the $i$-th executor and $c_{ij} = 0$ otherwise. The matrix

$$[\tau_{ij}]_{\substack{i=1,...,k \\ j=1,...,M}} \triangleq \tau \tag{13.16}$$

is a matrix of data necessary for the problem solution and the zero-one matrix

$$[c_{ij}]_{\substack{i=1,\ldots,k \\ j=1,\ldots,M}} \triangleq P$$

denotes a decision to be determined. Using the terms introduced in the previous sections we can say that the entries of this matrix are the components of the control vector $u$. The working time of the $i$-th executor, i.e. the time needed for the execution of all tasks assigned to this executor is the following

$$T_i = \sum_{j=1}^{M} \tau_{ij}\, c_{ij}\,, \qquad i \in \overline{1, k}\,. \tag{13.17}$$

**Decision making (assignment) problem:** For the given matrix $\tau$ one should find the matrix $P$ minimizing

$$T = \max\,\{T_1, T_2, \ldots, T_k\} \tag{13.18}$$

subject to constraints

$$c_{ij} \in \{0,1\} \quad \text{for every } (i,j)$$

and

$$\bigwedge_{j \in \overline{1,M}} \left(\sum_{i=1}^{k} c_{ij} = 1\right). \tag{13.19}$$

The constraint (13.19) means that each task should be assigned to one and only one executor. The above problem may be also called a *problem of task assignment control*. The determination and realization of the decisions may be repeated in successive cycles of the operations performing. At the $n$-th period (i.e. in the $n$-th cycle of the task execution) a controlling computer finds the assignment matrix $P_n$ on the basis of the data $\tau_n$ introduced, and in practice – finds a set of the pairs $(i, j)$ for which $c_{ij} = 1$. The control is more complicated if the execution times for the successive sets of tasks are not given directly but should be calculated by using the given functions $\tau_{ij}(z)$ where $z$ is a parameter of disturbances, i.e. variables influencing the execution time $\tau_{ij}$. For example, they may be sizes or properties (features) of the $j$-th element which is to be an object of a manufacturing operation, or it may be number evaluating a complexity of an operation influencing the execution time in a case of a project management. Then the control program consists in the calculation of the matrix $\tau_n$ on the basis of the data $z_n$ introduced and next the determination of the matrix $P_n$. For an uncertain plant, the control can be performed in a closed-loop system in

which for the fixed $z$ a successive approximation process for the determination of $P_n$ on the basis of the evaluation of the execution time $T_n$ is realized, as it has been mentioned in Sect. 13.2 in the cases of a task distribution.

Now let us consider the case with different kinds of the executors, i.e. assume that for each task $j$, a set $I_j \subset I$ of the indexes of executors able to execute this task is determined. In such a way, the sets $J_i \subset J$ of the indexes of tasks which may be executed by the $i$-th executor ($i$ = 1, ..., $k$) are determined as well. Then into the formula (13.18) one should put

$$T_i = \sum_{j \in J_i} \tau_{ij}\, c_{ij}\,, \qquad i \in \overline{1, k} \tag{13.20}$$

i.e. the summing is performed for all $j \in J_i$. Instead of the constraint (13.19) we have now

$$\bigwedge_{j \in \overline{1,M}} \left( \sum_{i \in I_j} c_{ij} = 1 \right). \tag{13.21}$$

The formulas (13.17) and (13.19) are special cases of the formulas (13.20) and (13.21), respectively.

Essential complications occur when we should take into account some succession constraints for the task executions, formulated with the help of a set of pairs $(j, l) \in J \times J$ where the pair $(j, l)$ means that the $l$-th task should be executed after the $j$-th task. Then the decision problem under consideration is often called a *task scheduling problem*. Its solution means the determination of the partition of the set $J$ into the subsets $\bar{J}_i$ assigned to the $i$-th executor and the determination of the succession of the execution for the tasks from the set $\bar{J}_i$ in each executor, with taking into account the succession constraints. To describe these constraints, it is convenient to introduce starting moments for individual tasks. Consequently, the finishing moments obtained by adding the execution time to the starting moments are determined as well. The succession of the task execution is defined by a sequence of task indexes in particular sets $\bar{J}_i$, i.e. by so called *permutation*. This is one-to-one mapping $m = F_i(j)$ which to each $j \in \bar{J}_i$ assigns $m \in \bar{J}_i$ where $m$ determines the succession, e.g. if $F(4) = 2$ then the task with the index 4 will be executed as the second in a sequence.

**Decision making (scheduling) problem:** For the given matrix $\tau$ and the set of pairs $(j, l) \in J \times J$ defining the succession constraints one should find

the decision in the form of

$$< \bar{J}_i , F_i > \quad \text{for} \quad i = 1, 2, ..., k.$$

Of course, the sets $\bar{J}_i$ can be presented in the form of a zero-one matrix $P$, as previously. One should note that there is an interconnection between the determination of $\bar{J}_i$ and $F_i$ which cannot be performed independently.

The problem of task scheduling control will be presented more precisely for an important case when the task means an execution of a certain operation on a certain object. Let us assume that we have a set of objects (elements) $\{S_1, S_2, ..., S_N\}$ and for each of them a sequence of operations $O_1$, $O_2$, ..., $O_k$ should be executed successively by the executors $R_1, R_2, ..., R_k$. It means that the object is entering the executor $R_1$ which performs the operation $O_1$, then is leaving the executor $R_1$ and entering the executor $R_2$ which performs the operation $O_2$ etc. Quite often in this case we use a term servicing or processing channel instead of an executor and we say that the object enters the servicing channel $R_i$ where the operation $O_i$ is performed. The problem consists in the following: For the known execution times of all operations for each object, one should find the succession of putting the objects at the input of the sequence of the servicing channels, minimizing the execution time of the all operations for the all objects.

It is convenient to identify the tasks in the set $Z$ by two indexes: $z_{ij}$ denotes the $i$-th operation executed on the $j$-th object; $i = 1, ..., k$; $j = 1, ..., N$. The succession constraint means that for each $j$ the tasks $z_{ij}$ form a sequence of tasks executed in the succession $z_{1j}, z_{2j}, ..., z_{kj}$. The entries $\tau_{ij}$ in the matrix (13.16) denote the execution times of the $i$-th operation for the $j$-th object. The succession of putting the objects at the input of the servicing channel sequence is determined by a permutation $m = F(j)$, as in the former considerations, e.g. if $F(4) = 2$ then the object $S_4$ will be put as the second in the succession. The decision making problem can be then stated as follows: For the given matrix $\tau$ one should find the permutation $F$ for which the execution time of all operations is the smallest.

For the given permutation $F$ let us introduce the notation $S_j = \bar{S}_m$ (i.e. this permutation defines a succession of the objects $\bar{S}_1, \bar{S}_2, ..., \bar{S}_N$) and $\tau_{ij} = \bar{\tau}_{im}$. Denote by $t_{im}$ the starting moment of the $i$-th operation for the $m$-th object (i.e. for the object $\bar{S}_m$) or the moment of putting the $m$-th object into the $i$-th executor. The global execution time is then

$$T = t_{kN} + \bar{\tau}_{kN}$$

and since $\bar{\tau}_{kN}$ is given, the problem is reduced to the minimization of $t_{kN}$. Let us note that two-index sequence $t_{im}$ for $i = 1, 2, ..., k$ and $m = 1, 2, ..., N$ is described by the following difference equation:

$$t_{im} = \max \{t_{i-1,m} + \bar{\tau}_{i-1,m}, t_{i,m-1} + \bar{\tau}_{i,m-1}\}. \tag{13.22}$$

It means that the moment $t_{im}$ does not have to precede the finishing moment of the operation for the $m$-th object in the previous executor, equal to $t_{i-1,m} + \bar{\tau}_{i-1,m}$ and the finishing moment of the operation for the ($m$–1)-th object in the $i$-th executor equal to $t_{i,m-1} + \bar{\tau}_{i,m-1}$. The initial conditions for the difference equation (13.22) are as follows:

$$\left.\begin{aligned} t_{1,1} = 0, \qquad t_{1,m} &= \sum_{l=1}^{m-1} \bar{\tau}_{1,l} \quad \text{for } m > 1, \\ t_{i,1} &= \sum_{l=1}^{i-1} \bar{\tau}_{l,1} \quad \text{for } i > 1. \end{aligned}\right\} \tag{13.23}$$

The problem under consideration is then reduced to the determination of such a succession $F$ which minimizes the solution $t_{kN}$ of the equation (13.22) with the initial conditions (13.23). The determination of $F$ can be presented as a zero-one problem by introducing the matrix $P = [c_{js}]$ with $N$ rows and $N$ columns, in which $c_{js} = 1$ if the object $S_s$ is introduced into the sequence of the executors directly after the object $S_j$, and $c_{js} = 0$ otherwise. It is easy to note that there is one and only one unit in each row and in each column of the matrix $P$. Choosing the matrix $P$ is subject to constraints

$$\bigwedge_j \left(\sum_{s=1}^{N} c_{sj} = 1\right), \qquad \bigwedge_s \left(\sum_{j=1}^{N} c_{sj} = 1\right).$$

A block scheme of the system controlling the scheduling in a successive cycle is illustrated in Fig. 13.4. At the beginning of the $n$-th cycle, the block $B_1$ determines the execution times $\tau_n$ using the measured values of the parameters characterizing the set of objects $S_n$ which has appeared in this cycle for an execution and the current values of the executors parameters which can influence the values of the entries of the matrix $\tau_n$. Then the

block $B_2$, using a suitable algorithm of a zero-one programming for the scheduling problem under consideration, determines the decision $P_n$ defining a succession of the objects entering the sequence of the executors.

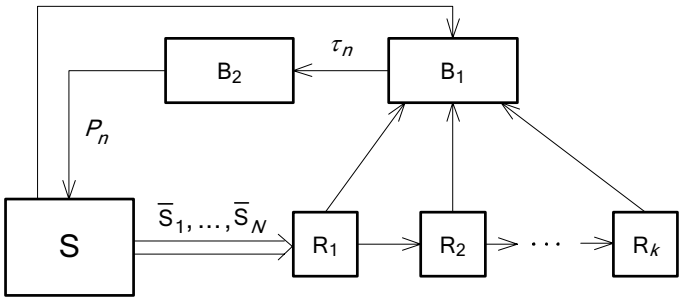


**Fig. 13.4.** Control system for scheduling in a case under consideration

The case under consideration may have different practical interpretations. In a production process $O_1$, $O_2$, ..., $O_k$ may denote successive production operations performed on objects forming the set $S$, e.g. assembly operations. Then $R_i$ is a sequence of devices or processing channels performing these operations. In the case of a project management $O_1$, $O_2$, ..., $O_k$ may denote successively performed operations in a complex project, e.g. $S$ may be a set of designs (of the same kind) to be elaborated, in which the same sequence of operations is defined, and $R_i$ denotes a designer or a designing group performing only one from these operations. In a computer system, $S$ may denote a set of computation tasks (programs) divided into successively executed partial tasks and $R_i$ are processors in a multiprocessor system with a parallel processing. If each processor can execute each partial task then the execution time can be additionally decreased by putting a successive partial task to one of the processor free at the moment.

The problems described above have been concerned with the control of a succession of operations. The control of a service of tasks mentioned in Sect. 13.1 will occur when the set $S$ is not given in advance but the objects are arriving at the input of the system (most often at random moments) and are waiting in a waiting line. The control decisions may consist in defining an index of the objects which should be taken from the queue at the moment when one of the servicing channels is free and ready to admit the object. Related situations occur in so called *admission control* in computer networks (see e.g. [99]). Generally speaking they are problems considered in an area of queuing and mass service systems which are not described

here. Their importance in the control theory and engineering (especially oriented for the control of computer systems) shows deep and essential connections between the control science and other related areas such as operational researches.

## 13.5 Control of Allocation in Systems with Transport

The control of a complex of operations is getting complicated when different functions should be taken into account simultaneously, e.g. the control of an allocation (task and/or resource distribution) and a service or the control of a task scheduling and a traffic of executors and/or of objects of technical operations, under the assumption that executors or objects may remove in so called *flexible manufacturing system* [42, 74]. From this wide area let us take for the consideration a case of the allocation control with taking into account a transport of raw materials and products in a production system (Fig. 13.5).

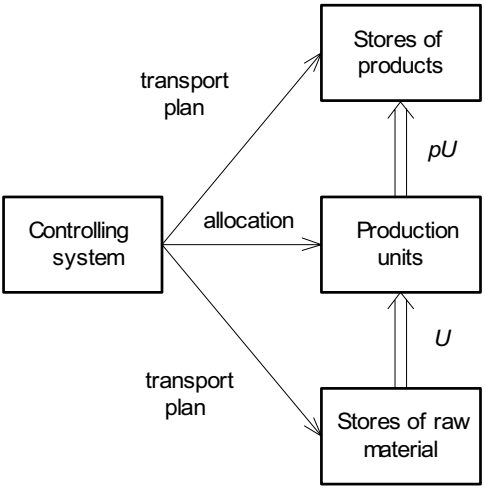


**Fig. 13.5.** Simplified scheme of the system under consideration

The system consists of $k$ production units working in parallel and producing the same kind of a product, raw material stores from which the raw material is transported to the production units, and product stores to which the product is transported from the production units. The controlling system is performing three functions: the control of the raw material transport, of the raw material distribution among the units and of the product trans-

port. As a quality index in one production cycle one may introduce the sum of three components [54, 58]

$$Q = Q_1 + Q_2 + Q_3$$

where $Q_1$ denotes the cost of the raw material transport, $Q_2$ is the cost proportional to the production time in one cycle, and $Q_3$ is the cost of the product transport. The value of a production time (and consequently, the value $Q_2$) depends on the distribution of the total amount of the raw material $U$ among the units, and if the transport was not taken into account we would determine the optimal distribution $u_1, ..., u_k$ as it has been presented in Sect. 13.2. The minimization of $Q_1$ means the determination of such amounts of the raw material transported from the individual stores to the individual units, i.e. such a transport plan that the global transportation cost is minimal under the assumption that the sum of the raw materials which can be taken from the stores is equal to $U$. Similarly, the minimization of $Q_3$ means the determination of such amounts of the product transported from the individual units to the individual stores that the global transportation cost is minimal. The separate minimization of $Q_1$ and $Q_3$ is called a *transportation problem* for which there exist suitable algorithms and computer programs [105]. However, the minimization of the global cost $Q$ could not be decomposed into independent minimization of $Q_1$, $Q_2$, $Q_3$ because these problems are interconnected: $Q_2$ depends on the values $u_1, u_2, ..., u_k$ which at the same time are data for both transportation problems. One should then find directly (without a decomposition) the decisions $u_1^*, u_2^*, ..., u_k^*$ and the transport plans for the raw material and the product – minimizing the global quality index $Q$. It may be obtained by applying a suitable computation algorithm or by realizing successive approximate control decisions in a closed-loop control system.

The problem is much simpler in the case of one raw material store and one product store, and is reduced to the determination of the optimal distribution $u^*$ taking into account the transport cost. Let us introduce the following notation concerning one cycle containing the transport of the raw material, the distribution of the raw material among the units, obtaining the product and the transport of the product to the stores:

$T_i$ – the production time in the $i$-th unit,

$u_i$ – the amount of the raw material assigned to the $i$-th unit,

$v_i$ – the amount of the product obtained from the $i$-th unit,

$\bar{c}_i$ – the unit cost of the raw material transport,

$\hat{c}_i$ – the unit cost of the product transport.

Assume that

$$\left.\begin{array}{ll} T_i = \varphi_i(u_i), & i = 1,2,...,k, \\ Q_1 = \sum_{i=1}^{k} \bar{c}_i u_i, & Q_3 = \sum_{i=1}^{k} \hat{c}_i v_i, \\ v_i = d_i u_i, & Q_2 = p \cdot T, \end{array}\right\} \tag{13.24}$$

where $T = \max\{T_1, T_2, \ldots, T_k\}$ is the production time of the global amount of the product. Then the problem consists in finding the values $u_1^*, u_2^*, \ldots, u_k^*$ minimizing the cost

$$Q(u_1, u_2, \ldots, u_k) = Q_1 + Q_2 + Q_3$$

subject to constraints (13.3). Hence, it is a task allocation problem taking into account the cost of the transport. Introducing this cost may significantly change the distribution optimal with respect to $Q_2$ or $T$ only. For example, when only $T$ is taken into account, the value $u_i$ for a unit with a great productivity will be great in comparison with the values of $u$ for other units, however taking into account $Q_1$ may cause a significant decrease in $u_i$ if the transport of the raw material to this unit is expensive.

According to (13.24)

$$Q(u_1, u_2, ..., u_k) = p \cdot \max_i [\varphi_i(u)] + \sum_{i=1}^{k} c_i u_i \tag{13.25}$$

where $c_i = \bar{c}_i + d_i \hat{c}_i$. The optimal distribution $u_1^*, u_2^*, ..., u_k^*$ minimizing the function (13.25) with the constraints (13.3) may be obtained in an analytical form in simple cases only. Let us consider two production units and linear relationships between $T_i$ and $u_i$:

$$T_1 = k_1 u_1, \qquad T_2 = k_2 u_2.$$

The optimal distribution minimizing $T = \max\{T_1, T_2\}$ is the following (see (13.8) for $\alpha = 1$):

$$\bar{u}_1 = \frac{k_2 U}{k_1 + k_2}, \qquad \bar{u}_2 = \frac{k_1 U}{k_1 + k_2}.$$

Substituting $u_2 = U - u_1$ to the function (13.25) yields

$$Q = \max\{k_1 u_1, k_2(U-u_1)\} + c_1 u_1 + c_2(U-u_1) \triangleq F(u_1).$$

An analysis of this function leads to the following rules for the determination of $u_1^*$:

1. If $F(u_1)$ is an increasing function for $u_1 < \bar{u}_1$ then $u_1^* = 0$.
2. If $F(u_1)$ is a decreasing function for $u_1 < \bar{u}_1$ and is an increasing function for $u_1 > \bar{u}_1$ then $u_1^* = \bar{u}_1$.
3. If $F(u_1)$ is a decreasing function for $u_1 > \bar{u}_1$ then $u^* = U$.

From these rules we obtain the following *algorithm of the optimal distribution* taking into account the transport cost, in which the notation $c_1 - c_2 = c$ has been introduced:

1. If $c - k_2 > 0$ then $u_1^* = 0,\ u_2^* = U$.
2. If $c - k_2 < 0$ and $c + k_1 > 0$ then $u_1^* = \bar{u}_1,\ u_2^* = \bar{u}_2$.
3. If $c + k_1 < 0$ then $u_1^* = U,\ u_2^* = 0$.
4. If $c - k_2 = 0$ then $u_1^*$ is any value from the interval $[0,\ \bar{u}_1]$.

The block scheme of the control algorithm is presented in Fig. 13.6.

Until now we have assumed that three parts of the cycle (raw material transport, production and product transport) are separated in time, i.e. the distribution is executed after obtaining the global amount of the raw material from the stores and the production in each unit starts at the same time. These operations, however, can be organized in the other way: the production in a unit starts immediately after receiving the raw material and the product is taken from the unit directly after finishing the production process. Then, the whole system can be treated as a complex of $k$ parallel operations, each of them consisting of the raw material transport, the production and the product transport. Consequently, the problem is reduced to the time-optimal task distribution considered in Sect. 13.2, i.e. to the determination of the decisions $u_1^*, u_2^*, ..., u_k^*$ minimizing

$$T = \max\{\bar{T}_1, \bar{T}_2, ..., \bar{T}_k\}$$

where $\bar{T}_i$ denotes the sum of production time and the time of both transportations, i.e.

$$\bar{T}_i = \varphi(u_i) + c_i u_i$$

where $c_i = \bar{c}_i + d_i \hat{c}_i$ as formerly, and $c_i,\ \hat{c}_i$ denote the transport times for a unit of the raw material and the product, respectively, under the assump-

tion that the transport time is proportional to an amount of a material. In particular, for the linear models $T_i = k_i u_i$, according to (13.8) for $\alpha = 1$ the control algorithm is as follows:

$$u_i^* = e_i^{-1} \cdot U(\sum_{i=1}^{k} e_i^{-1})^{-1}, \qquad i = 1, 2, ..., k$$

where $e_i = k_i + c_i$.

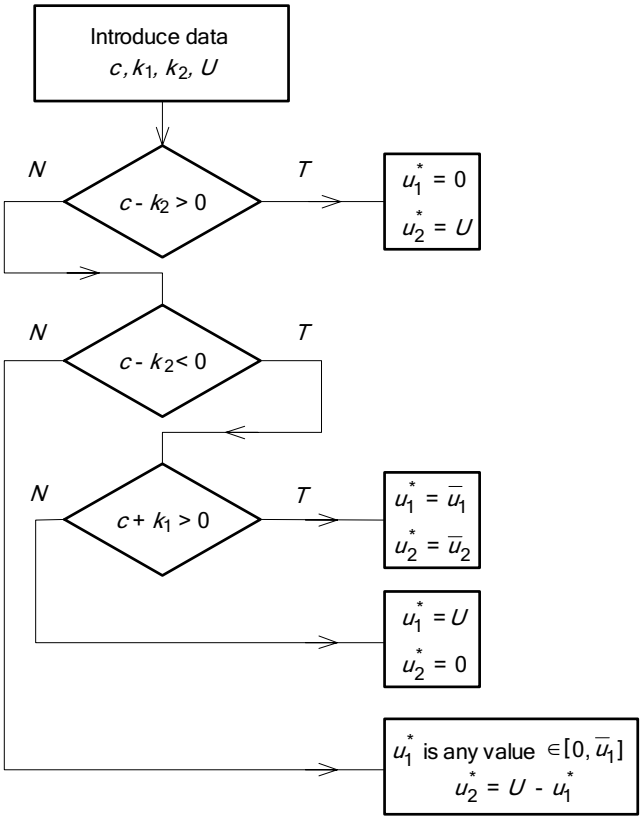


**Fig. 13.6.** Scheme of the control algorithm in one cycle

## 13.6 Control of an Assembly Process

In a traditional assembly process, the sequence of assembly operations executed in successive stages is given in advance, and the control consists

only in a controlling of technical actions necessary to perform assembly operations. An essential decision making problem arises when in successive stages the assembly operations are chosen currently, based on a recognition of the current state of an assembly object. If these operations are executed by an intelligent assembly robot, its basic functions at one stage are the following: the recognition of the object state based on the results of observations, the choice of a corresponding operation from a set of operations given for this stage, and the execution of this operation. The decision problem described here and consisting in a choice of a sequence of operations in the assembly process may be considered as a specific example of a multistage decision process described in Sect. 4.3 in connection with dynamic programming application [41, 52].

Let us introduce the following notations:

$S_n = \{S_{n1}, S_{n2}, ..., S_{nk_n}\}$ – the set of possible object states at the $n$-th stage,

$s_n \in \{1, 2, ..., k_n\}$ – the set of indexes indicating the possible states at the $n$-th stage,

$O_n = \{O_{n1}, O_{n2}, ..., O_{nm_n}\}$ – the set of possible operations at the $n$-th stage,

$l_n \in \{1, 2, ..., m_n\}$ – the set of indexes indicating the possible operations at the $n$-th stage,

$y_n$ – the vector of the observation (measurement) results at the $n$-th stage.

The components of the vector $y_n$ denote features of the assembly object, i.e. variables characterizing the current effect of the assembly process and used to the evaluation of the assembly quality. They may be e.g. dimensions or sizes evaluating the precision, accuracy or tolerance in the placement and fastening of elements. Assume that the recognitions of the state $s_n$ on the basis of $y_n$ are correct and performed according to a known recognition algorithm

$$s_n = G_n(y_n).$$

Assume also that the relationship between the state $s_n$, the next state $s_{n+1}$ and the current operation

$$s_{n+1} = f_n(s_n, l_n)$$

is known. This relationship is given in the form of table (matrix) where the index $s_{n+1}$ is written in the row with index $s_n$ and the column with index $l_n$. Let us introduce a performance index $\varphi_n(s_n)$ evaluating the state $s_n$

with the help of a non-negative number $\varphi_n(s_n)$ so that this number is as less as the current effect of the assembly is better.

**Assembly control problem:** For the given sets $s_n$, $l_n$ and the functions $G_n, f_n$ for every $n = 0, 1, ..., N$, one should determine the control algorithm in a closed-loop system $l_n = \overline{\Psi_n}(y_n)$, which minimizes the global quality index evaluating the whole $N$-stage assembly process:

$$Q_N = \sum_{n=1}^{N} \varphi_n(s_n).$$

In order to apply dynamic programming in a way similar to that presented in Sect. 4.3, let us introduce the notation:

$$\varphi_n(s_n) = \varphi_n[f_n(s_{n-1}, l_{n-1})] \stackrel{\Delta}{=} g_n(s_{n-1}, l_{n-1})$$

$$Q_N = \sum_{n=1}^{N} g_n(s_{n-1}, l_{n-1}) = \sum_{n=0}^{N-1} g_n(s_n, l_n) \tag{13.26}$$

$$V_{N-n}(s_n) = \min_{l_n, l_{n+1}, ..., l_N} \sum_{i=n}^{N-1} g_n(s_n, l_n).$$

The $g_n$ is presented in the form of a table containing the number $v_n \stackrel{\Delta}{=} g_n(s_n, l_n)$ in the row with index $s_n$ and the column with index $l_n$. For $n = N - 1$ we minimize the last component of the sum (13.26) with respect to $l_n$ and as a result we obtain $l_{N-1} = \Psi_{N-1}(s_{N-1})$, i.e. the relationship between the last optimal decision and the state $s_{N-1}$. Let us note that the minimization of $g_{N-1}$ with respect to $l_{N-1}$ is an integer programming problem. It should be solved for all $s_{N-1} \in \{1, 2, ..., k_{N-1}\}$. Since usually the numbers of possible operations and states are not great for each $s_{N-1}$ it is possible to determine $g_{N-1}$ for each $l_{N-1}$ and to choose $l_{N-1}$ for which $g_{N-1}$ is minimal. As a result, the relationship $l_{N-1} = \Psi_{N-1}(s_{N-1})$ is obtained in the form of a table giving for each state $s_{N-1}$ the corresponding decision $l_{N-1}$. Next, one should determine the relationship

$$V_1(s_{N-1}) = V_1[f_{N-2}(s_{N-2}, l_{N-2})]$$

in the form of a table containing the number $V_1$ in the row with index

$s_{N-2}$ and the column with index $l_{N-2}$. This table is obtained in such a way that for the pair $(s_{N-2}, l_{N-2})$ we take $s_{N-1}$ from the table $f_{N-2}$, $l_{N-1}$ from the table $\Psi_{N-1}$ and $v_{N-1}$ from the table $g_{N-1}$. At the next step we find the relationship $l_{N-2} = \Psi_{N-2}(s_{N-2})$ by minimization

$$\min_{l_{N-2}} \{g_{N-2}(s_{N-2}, l_{N-2}) + V_1[f_{N-2}(s_{N-2}, l_{N-2})]\}.$$

Continuing this procedure we finally obtain the table $l_0 = \Psi_0(s_0)$. The algorithm for the determination of $\Psi_n$ can be presented in the form of the following recursive procedure:

$$V_{N-n}(s_n) = \min_{l_n} \{g_n(s_n, l_n) + V_{N-n-1}[f_n(s_n, l_n)]\},$$

$n = N - 1, N - 2, ..., 0;\ V_0 = 0.$

As a result we obtain the relationships

$$l_n = \Psi_n(s_n), \qquad n = 0, 1, ..., N - 1,$$

and after substituting the formula $s_n = G_n(y_n)$, we have the relationships

$$l_n = \Psi_n[G_n(y_n)] = \overline{\Psi}_n(y_n)$$

determining the control algorithm in a closed-loop system.

The algorithm for the determination of the tables $\Psi_n$ may be called *an algorithm for a control system design*. It can be written as follows:

1. Using the tables $g_n$ and $V_{N-n-1}$ taken from the memory, determine the table $\Psi_n$ in the following way: for each index $s_n = 1, 2, ..., k_n$ find the index $l_n$ for which the sum of numbers corresponding to the pair of indexes $(s_n, l_n)$ is the smallest.
2. Using the tables $f_{n-1}$ and $g_n$ taken from the memory and the table $\Psi_n$ which has been determined, find the table $V_{N-n}$.
3. Introduce the tables $\Psi_n$ and $V_{N-n}$ into the memory.

The way of finding the tables $\Psi_n$ and $V_{N-n}$ has been described above for $n = N - 1$. In the data base of a computer executing the above algorithm (Fig. 13.7) one should put the tables $f_n$ and the tables $g_n$ obtained from $f_n$ for the given functions $\varphi_n$ defining the way of the performance index calculation. The determined tables should be put into the data base of a controlling computer.

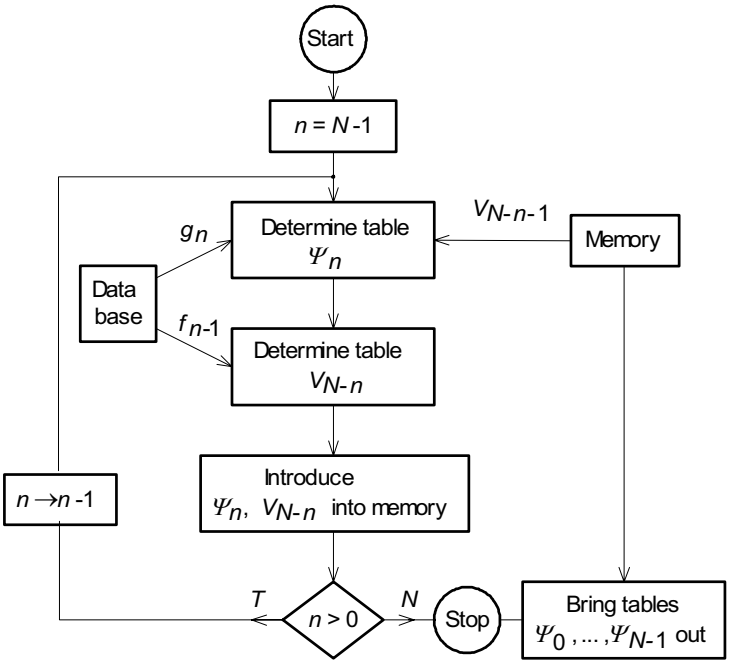


**Fig. 13.7.** Simplified block scheme of algorithm for control system design

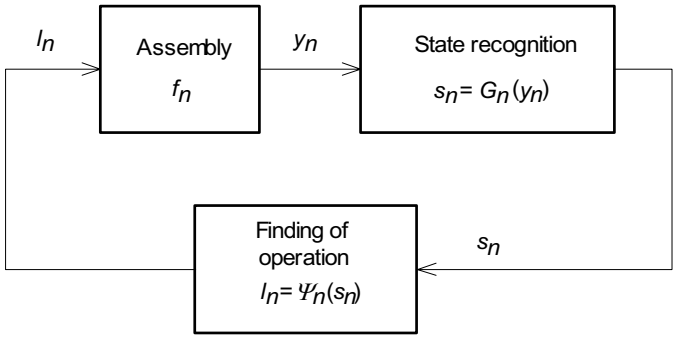


**Fig. 13.8.** Scheme of control system for assembly process under consideration

The algorithm of a real-time control (Fig. 13.8) is the following:

1. Introduce the measurement results $y_n$.
2. Determine the index of a current state $s_n = G_n(y_n)$.

3. From the table $\Psi_n$ read over the index of an assembly operation $l_n$ and bring it out for the execution.

## 13.7 Application of Relational Description and Uncertain Variables

For uncertain complexes of operations one can apply the methods described in the former chapters for control of uncertain plants, in particular – methods and algorithms based on a relational knowledge representation, on descriptions using uncertain and fuzzy variables and on the knowledge validation and updating in a learning process [30, 34, 42, 46, 48, 49, 52]. Now we shall describe shortly the distribution problems based on a relational knowledge representation in the form of a set of inequalities, and on a description with uncertain variables.

Consider an uncertain complex of operations described by a set of inequalities

$$T_i \le \varphi_i(u_i) \tag{13.27}$$

where $\varphi_i$ is a known function, increasing in the case of tasks and decreasing in the case of resources. The inequalities (13.27) together with the function $T = \max\{T_1, T_2, ..., T_k\}$ form a relational knowledge representation $R(u, y)$ in our case. For a user's requirement $T \le \alpha$ or $T \in [0, \alpha]$ one can formulate the decision problem in a way described in Sect. 6.3.

**Decision problem:** For the given $\varphi_i$ $(i = 1,2,...,k)$ and $\alpha$ one should find the largest set $D_u \subset \overline{U}$ such that the implication

$$u \in D_u \to T \in [0, \alpha]$$

is satisfied, where the set $\overline{U}$ is determined by the constraints (13.3).

Then

$$D_u = \{u \in \overline{U} : D_T(u) \subseteq [0, \alpha]\} \tag{13.28}$$

where $D_T(u)$ is the set of possible values $T$ for the fixed $u$, i.e. $D_T(u)$ is determined by the inequality

$$T \le \max_i \varphi_i(u_i) .$$

Consequently,

$$D_u = \{u \in \overline{U}: \bigwedge_{i \in \overline{1,k}} [\varphi_i(u_i) < \alpha]\}. \tag{13.29}$$

For $\varphi(u_i) = c_i u_i$

$$D_u = \{u \in R^k : (\sum_{i=1}^{k} u_i = U) \wedge \bigwedge_{i \in \overline{1,k}} (u_i \geq 0) \wedge (c_i u_i \leq \alpha)\}.$$

It is easy to note that the solution exists (i.e. $D_u$ is not an empty set) if $\alpha \geq T^*$ where $T^*$ is the minimal execution time obtained for the optimal allocation $u^*$. In this case $u^* \in D_u$. For example, if $k = 2$ and

$$\alpha \geq \frac{c_1 c_2 U}{c_1 + c_2} = T^*$$

then $D_u$ is determined by equality $u_2 = U - u_1$ and inequality $\beta \leq u_1 \leq \delta$ where

$$\beta = \frac{\alpha}{c_1}, \qquad \delta = U - \frac{\alpha}{c_2}.$$

Let us consider a complex of parallel operations described by the inequalities

$$T_i \leq \varphi_i(u_i, x_i), \qquad i = 1,2,...,k \tag{13.30}$$

where $u_i$ is the size of a task assigned to the *i*-th operation, $x_i \in R^1$ is a parameter and $\varphi_i$ is a known increasing function of $u_i$. The parameter $x_i$ is unknown and is assumed to be a value of an uncertain variable $\overline{x}_i$ described by a certainty distribution $h_{xi}(x_i)$ given by an expert. Now the relational knowledge representation, consisting of (13.30) and the relationship $T = \max(T_1, T_2, ..., T_k)$, is completed by the functions $h_{xi}(x_i)$. We assume that $\overline{x}_1, \overline{x}_2, ..., \overline{x}_k$ are independent uncertain variables, i.e.

$$h_x(x) = \min_i h_{xi}(x_i)$$

where $x = (x_1, x_2, ..., x_k)$. The largest set of decisions $D_u(x)$ depends on *x* and is determined by (13.29) with $\varphi_i(u_i, x_i)$ in place of $\varphi_i(u_i)$. The description of the complex is analogous for the resource allocation problem. Then $u_i$ is the amount of a resource assigned to the *i*-th operation, $\varphi_i$ is a decreasing function of $u_i$ and *U* denotes the total amount of the resource

to be distributed.

According to the general formulation of the decision problem presented in Sect. 8.2, the allocation problem may be formulated as an optimization problem consisting in finding the optimal allocation $u^*$ that maximizes the certainty index of the soft property: "$u$ approximately belongs to $D_u(\bar{x})$" or "the set of possible values $T$ approximately belongs to $[0,\alpha]$" (i.e. belongs to $[0,\alpha]$ for an approximate value of $\bar{x}$).

**Optimal allocation (decision) problem:** For the given $\varphi_i$, $h_{xi}$ $(i \in \overline{1,k})$, $U$ and $\alpha$ find

$$u^* = \arg\max_{u \in \overline{U}} v(u)$$

where

$$v(u) = v\{D_T(u;\bar{x}) \,\tilde{\subseteq}\, [0,\alpha]\} = v(T(u,\bar{x}) \,\tilde{\leq}\, \alpha). \tag{13.31}$$

The soft property "$D_T(u;\bar{x}) \,\tilde{\subset}\, [0,\alpha]$" is denoted here by "$T(u,\bar{x}) \,\tilde{\leq}\, \alpha$", and $D_T(u;x)$ denotes the set of possible values $T$ for the fixed $u$, determined by the inequality

$$T \leq \max_i \varphi_i(u_i, x_i).$$

According to (13.31)

$$v(u) = v\{[T_1(u_1,\bar{x}_1) \,\tilde{\leq}\, \alpha] \wedge [T_2(u_2,\bar{x}_2) \,\tilde{\leq}\, \alpha] \wedge \ldots \wedge [T_k(u_k,\bar{x}_k) \,\tilde{\leq}\, \alpha]\}.$$

Then

$$u^* = \arg\max_{u \in \overline{U}} \min_i v_i(u_i) \tag{13.32}$$

where

$$v_i(u_i) = v[T_i(u_i,\bar{x}_i) \,\tilde{\leq}\, \alpha] = v[\varphi_i(u_i,\bar{x}_i) \,\tilde{\leq}\, \alpha] = v[\bar{x}_i \,\tilde{\in}\, D_{xi}(u_i)],$$

$$D_{xi}(u_i) = \{x_i \in R^1 : \varphi_i(u_i, x_i) \leq \alpha\}.$$

Finally

$$v_i(u_i) = \max_{x_i \in D_{xi}(u_i)} h_{xi}(x_i) \tag{13.33}$$

and

$$u^* = \arg\max_{u \in \overline{U}} \min_i \max_{x_i \in D_{xi}(u_i)} h_{xi}(x_i). \tag{13.34}$$

The value $v_i(u_i)$ denotes the certainty index that in the $i$-th operation an

approximate value of the execution time is less than $\alpha$. The procedure of finding the optimal allocation $u^*$ is then the following:

1. To determine $v_i(u_i)$ using (13.33).

2. To determine $u^*$ according to (13.32), subject to constraints (13.3).

Assume that $\varphi_i(u_i, x_i)$ is an increasing function of $x_i$. Then the set $D_{xi}(u_i)$ is determined by the inequality $x_i \le \hat{x}_i(u_i)$ where $\hat{x}_i(u)$ is the solution of the equation

$$\varphi_i(u_i, x_i) = \alpha \tag{13.35}$$

and

$$v_i(u_i) = \max_{x_i \le \hat{x}_i(u_i)} h_{xi}(x_i) . \tag{13.36}$$

In many cases an expert gives the value $x_i^*$ and the interval of the approximate values of $\bar{x}_i$: $x_i^* - d_i \le x_i \le x_i^* + d_i$. Then we assume that $h_{xi}(x_i)$ has a triangular form presented in Fig. 13.9 where $d_i \le x_i^*$. Let us consider the relation (13.30) in the form $T_i \le x_i u_i$ where $x_i > 0$ and $u_i$ denotes the size of a task. In this case, using (13.36) it is easy to obtain the following formula for the functions $v_i(u_i)$:

$$v_i(u_i) = \begin{cases} 1 & \text{for} \quad u_i \le \dfrac{\alpha}{x_i^*} \\ \dfrac{1}{d_i}\left(\dfrac{\alpha}{u_i} - x_i^*\right) + 1 & \text{for} \quad \dfrac{\alpha}{x_i^*} \le u_i \le \dfrac{\alpha}{x_i^* - d_i} \\ 0 & \text{for} \quad u_i \ge \dfrac{\alpha}{x_i^* - d_i} . \end{cases} \tag{13.37}$$

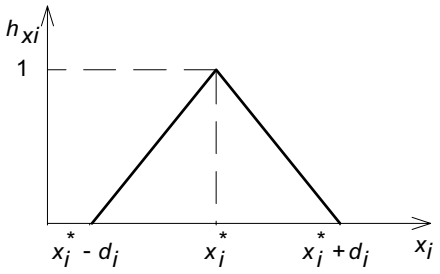


**Fig. 13.9.** Example of certainty distribution

For the relations $T_i \le x_i u_i^{-1}$ where $u_i$ denotes the size of a resource, the functions $v_i(u_i)$ have an analogous form, with $u_i^{-1}$ in place of $u_i$:

$$v_i(u_i) = \begin{cases} 0 & \text{for} \quad u_i \le \dfrac{x_i^* - d_i}{\alpha} \\ \dfrac{1}{d_i}(\alpha u_i - x_i^*) + 1 & \text{for} \quad \dfrac{x_i^* - d_i}{\alpha} \le u_i \le \dfrac{x_i^*}{\alpha} \\ 1 & \text{for} \quad u_i \ge \dfrac{x_i^*}{\alpha}. \end{cases} \tag{13.38}$$

**Example 13.2.** Let us consider the task allocation for two operations. In the maximization problem (13.32) the decision $u_1^*$ may be found by solving the equation $v_1(u_1) = v_2(U - u_1)$ and $u_2^* = U - u_1^*$. Using (13.37), we obtain the following result:

1. For

$$\alpha \le \frac{U(x_1^* - d_1)(x_2^* - d_2)}{x_1^* - d_1 + x_2^* - d_2} \tag{13.39}$$

$v(u) = 0$ for any $u_1$.

2. For

$$\frac{U(x_1^* - d_1)(x_2^* - d_2)}{x_1^* - d_1 + x_2^* - d_2} \le \alpha \le \frac{U x_1^* x_2^*}{x_1^* + x_2^*} \tag{13.40}$$

$u_1^*$ is a root of the equation

$$\frac{1}{d_1}\left(\frac{\alpha}{u_1} - x_1^*\right) = \frac{1}{d_2}\left(\frac{\alpha}{U - u_1} - x_2^*\right)$$

satisfying the condition

$$\frac{\alpha}{x_1^*} \le u_1^* \le \frac{\alpha}{x_1^* - d_1},$$

and $v(u^*) = v_1(u_1^*)$.

3. For

$$\alpha \ge \frac{U x_1^* x_2^*}{x_1^* + x_2^*} \tag{13.41}$$

$v(u^*) = 1$ for any $u_1$ satisfying the condition

$$U - \frac{\alpha}{x_2^*} \le u_1 \le \frac{\alpha}{x_1^*}.$$

For example, if $U = 2$, $\alpha = 2$, $x_1^* = 2$, $x_2^* = 3$, $d_1 = d_2 = 1$ then using (13.40) yields $u_1^* = 1.25$, $u_2^* = 0.75$, $v(u^*) = 0.6$.

The result is simpler under the assumption

$$\frac{x_1^*}{d_1} = \frac{x_2^*}{d_2} \stackrel{\Delta}{=} \gamma. \tag{13.42}$$

Then in the case (13.42)

$$u_1^* = \frac{Ux_2^*}{x_1^* + x_2^*}, \qquad u_2^* = \frac{Ux_1^*}{x_1^* + x_2^*},$$

$$v(u^*) = v_1(u_1^*) = \frac{1}{d_1}\left(\frac{\alpha}{u_1^*} - x_1^*\right) + 1 = \gamma\left[\frac{\alpha(x_1^* + x_2^*)}{U x_1^* x_2^*} - 1\right] + 1. \tag{13.43}$$

The formula (13.43) shows that $v(u^*)$ is a linear function of the parameter $\gamma$ characterizing the expert's uncertainty. □

The determination of the control decision $u^*$ may be difficult for $k > 2$ because of great computational difficulties. To decrease these difficulties we can apply the decomposition of the complex into two subcomplexes and consequently obtain a two-level control system (Fig. 13.10). This approach is based on the idea of decomposition and two-level control presented for the deterministic case [13]. At the upper level the value $U$ is divided into $U_1$ and $U_2$ assigned to the first and the second subcomplex, respectively, and at the lower level the allocation $u^{(1)}$, $u^{(2)}$ for the subcomplexes is determined. Let us introduce the following notation:

$n, m$ – the number of operations in the first and the second complex, respectively, $n + m = k$,

$T^{(1)}$, $T^{(2)}$ – the execution times in the subcomplexes, i.e.

$$T^{(1)} = \max(T_1, T_2, ..., T_n), \qquad T^{(2)} = \max(T_{n+1}, T_{n+2}, ..., T_{n+m}),$$

$u^{(1)}$, $u^{(2)}$ – the allocations in the subcomplexes, i.e.

$$u^{(1)} = (u_1, ..., u_n), \qquad u^{(2)} = (u_{n+1}, ..., u_{n+m}).$$

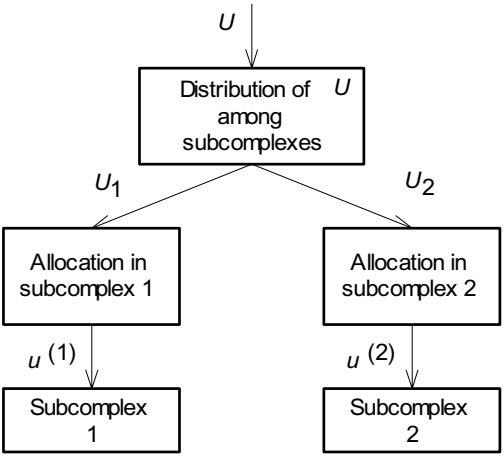


**Fig. 13.10.** Two-level control system

The procedure of the determination of $u^*$ is then the following:

1. To determine the allocation $u^{(1)*}(U_1)$, $u^{(2)*}(U_2)$ and the certainty indexes $v^{(1)*}(U_1)$, $v^{(2)*}(U_2)$ in the same way as $u^*$, $v^*$ in the first part of this section where the direct approach has been applied, with $U_1$ and $U_2$ in place of *U*.

2. To determine $U_1^*$, $U_2^*$ via the maximization of

$$v(T \tilde{\leq} \alpha) = v[(T^{(1)} \tilde{\leq} \alpha) \wedge (T^{(2)} \tilde{\leq} \alpha)] \stackrel{\Delta}{=} v(U_1, U_2).$$

Then

$$(U_1^*, U_2^*) = \arg \max_{U_1, U_2} \min\{v^{(1)*}(U_1), v^{(2)*}(U_2)\}$$

with the constraints: $U_{1,2} \geq 0$, $U_1 + U_2 = U$.

3. To find the values of $u^{(1)*}$, $u^{(2)*}$ and $v^*$ putting $U_1^*$ and $U_2^*$ into the results $u^{(1)*}(U_1)$, $u^{(2)*}(U_2)$ obtained in point 1 and into $v(U_1, U_2)$ in point 2.

## 13.8 Application of Neural Network [21]

Let us pay attention to the possibility of using a neuron-like algorithm as an approximate control algorithm for the task distribution in a complex of parallel operations described in Sect. 13.2. For the operation models (13.7), the exact control algorithm determining the distribution $u$ on the basis of the measurement result $z = c$ is given by the formula (13.8). For more complicated operation models (13.1) this algorithm can be presented in the form of a computational procedure. The simplest neuron-like algorithm for the execution of these computations has the form of one-layer net with single neurons for the individual operations in the complex (Fig. 13.11). The algorithm of a single neuron with the input $s_{11}, s_{12}, ..., s_{1,k-1}$ is here the following:

$$y_i = | w_{i1}s_{i1} + w_{i2}s_{i2} + ... + w_{i,k-1}s_{i,k-1} + 1|, \qquad i = 1, 2, ..., k \qquad (13.44)$$

where

$$s_{i1} = \frac{c_1}{c_i}, \; s_{i2} = \frac{c_2}{c_i}, \; ..., \; s_{i,i-1} = \frac{c_{i-1}}{c_i}, \; s_{ii} = \frac{c_{i+1}}{c_i}, ..., \; s_{i,k-1} = \frac{c_k}{c_i}. \qquad (13.45)$$

The elements $e$ marked in the scheme of the control system are blocks determining the decisions $u_i = y_i^{-1}$. The form (13.44) is reasoned by the form of the algorithm (13.8) obtained for the model (13.7) which may be an acceptable approximation of the dependency of $T_i$ upon $u_i$ for real operations. It is easy to note that for the positive values $w_{i,1}, w_{i,2}, ..., w_{i,k-1}$ the dependency of $u_i = y_i^{-1}$ upon $c_1, c_2, ..., c_k$ is identical with the exact algorithm (13.8) for $\alpha = 1$. The block B denoted in the figure determines the values $s_{i,1}, s_{i,2}, ..., s_{i,k-1}$ according to (13.45). It is worth noting that to apply correctly the neuron-like algorithm in our case, it is necessary to take into account the form of a control plant model and consequently to process initially the values $c_1, c_2, ..., c_k$ introduced in the control algorithm and to process the outputs of the neurons by the blocks $e$.

For changing the weights in the neuron-like algorithm presented, one can apply a learning process and concepts of learning mentioned in Sect. 12.6. In particular an exact algorithm (if it exists) may be used as a generator of a learning sequence, or an adjustment process based on the current evaluation of the execution time of whole complex $T$ may be applied. In order to improve the accuracy of the approximation one may apply multi-layer networks, having in mind that it may result in the increas-

ing of the time of learning. The neuron-like algorithms may be also used in other control systems with a control plant being a complex of operations, e.g. in a resource distribution system.

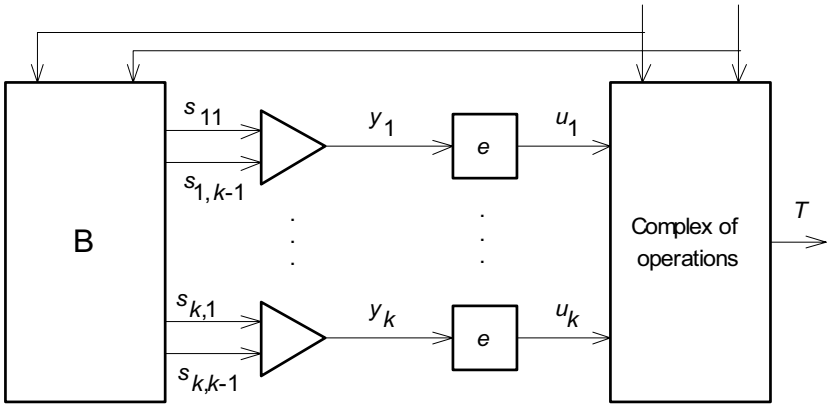


**Fig. 13.11.** An example of control system with one-layer neural network

# Conclusions

The book has presented a uniform description of basic control problems, methods and algorithms for different cases concerning the information on the control plant and for different ways of obtaining and using the knowledge of the plant during the control process. The division of the subject into five parts presented in Sect. 1.6 is more clear and understandable after reading the whole text. **Part two** containing Chaps. 3, 4 and 5 has been devoted to deterministic control problems, i.e. to considerations for deterministic plants, under the assumption that the plant description in the form of a function for the static case and in the form of a difference (differential) equation (or an equivalent form) for the dynamical case – is exactly known. **Part three** consisting of Chaps. 6, 7, 8 and 9 has been concerned with different formal descriptions of *a priori* uncertainty and with control problems formulated adequately to the model of the uncertainty taken into consideration. The uncertainty has been caused by the non-deterministic behaviour of the plant and (or) the uncertainty of an expert formulating the knowledge of the plant. In Chap. 6 we have considered the plant described in the form of a relational knowledge representation, and in Chaps. 7, 8, 9 we have presented the control problems for the plants with the description of the uncertainty using random, uncertain and fuzzy variables, respectively. Analogies and relations between formulations and solutions of the decision problems based on different descriptions of the uncertainty have been indicated, and different interpretations of the models and results in spite of their formal similarities have been discussed. It is worth noting that different cases of the uncertainty may occur in a single system. In particular, it may be so called *second order uncertainty*, that is the uncertainty concerning parameters occurring in a basic model of the uncertainty. Such cases have been presented in Chaps. 7 and 8 where the relational knowledge representation with unknown parameters has been considered.

**Part four** consisting of Chaps. 10 and 11 has been concerned with the uncertain control systems as well, but unlike the considerations in **Part tree**, we have described the control problems with using the information obtained in a closed-loop system. Two concepts have been considered: using the evaluations of the control effects to the direct determination of the next control decisions (Chap. 10) and to the improvement of the basic con-

trol algorithm in the adaptation and learning processes (Chap. 11). In both cases, the convergence problem for the approximation process executed in a closed-loop control system is of a significant importance. In the case of the first concept, it is the stability problem, widely developed and described in the literature.

**Part five** containing Chaps. 12 and 13 has been devoted to selected problems important in practical situations, namely the applications of selected artificial intelligence methods (neural networks and logical formulas processing), complex control systems and the control of complexes of operations treated as specific control plants consisting of interconnected activities.

The uniform and compact description of various control problems and methods presented in the book may be considered as a basis for studying more advanced and specific problems which may be put into the framework given in the book (as it was said in Sect. 1.6). Precise formulas and algorithms presented for particular cases can be directly used for the development of corresponding computer programs or may be useful in developing control algorithms and programs in more complicated situations occurring in the design of computer control and management systems, and more generally – the design of decision support systems.

Characterizing present and future directions and perspectives of the modern control theory one should take into account the following aspects:

1. Various technical and non-technical plants and processes can be considered as subjects of decision making (control and management). They are not only continuous technological processes but also so called flexible manufacturing systems and processes, complex systems, organizations and projects as management plants, and computer systems as specific decision plants. In many cases like these, even advanced methods and results of the traditional control theory are proved to be useless. They can be applied with good results e.g. to the control of continuous chemical reactors, but they cannot be used e.g. to the transport or traffic control, to the control of an assembly process or the control of computational process and data processing in a multiprocessor computer system.
2. In many practical situations, traditional mathematical models of control plants are proved to be insufficient or useless and it is necessary to develop algorithms for solving difficult problems based on different forms of the plant knowledge formulated by experts. Quite often this is incomplete and imprecise knowledge, sometimes not presented in the form of formulas and numbers but in the form of a linguistic description of facts and rules.
3. As a rule, modern control and management systems may be considered as specific information systems containing computers for determining the decisions or for supporting the decision making process. They are often so

called *expert systems* in which the computer-expert solves a problem on the basis of a knowledge representation, using a reasoning procedure [18, 92].

4. Nowadays, distinguishing between decision problems for control and management is not necessary and not possible. The same concerns distinguishing between information science and control science which may be treated as interconnected disciplines based on two fundamental areas: system engineering and knowledge engineering.

Taking into account the above remarks, one should indicate the following interconnected directions of the future development of the modern control theory treated as a part of the information and computer science dealing with foundations of the design of computer decision systems for the needs of control and management (see remarks in Sect. 1.5):

1. Problems of uncertain control systems using different forms of knowledge representations.
2. Developing artificial intelligence methods and the design of intelligent control systems with the application of a prediction, the algorithmization of reasoning, logic formulas processing, learning processes, neural networks, evolutionary programming etc.
3. Applications of advanced operational research methods to the control of complex manufacturing and computational operations, and to the management of complex organizations and projects.
4. Developing formal foundations of a diagnosis and reliability of control systems.
5. Problems of complex control systems, in particular – hybrid systems with various forms of knowledge representations for different parts of the complex plants and consequently, with different formulations of partial control problems adequate to the forms of the knowledge representation and to the models of uncertainty.
6. Developing the knowledge exploration and discovering problems and the concepts of decision making based on so called *distributed knowledge* (see [28, 40, 53]), oriented towards the needs of control and management systems.

There are different important and interesting technical problems connected with the design, building and exploitation of computer control and management systems. They exceed the framework of the control theory and have not been presented in this book.

# Appendix

## Operational Transforms

An *operational transform* or *Laplace transform* of the function $x(t)$ determined for $t \geq 0$ is defined as a function of a complex variable $s$, assigned to the function $x(t)$ according to the following formula:

$$X(s) = \int_0^\infty x(t)\, e^{-st}\, dt \triangleq \mathcal{L}\{x(t)\} .$$

Such a mapping is called an operational transformation (Laplace transformation), the transformation of the function $X(s)$ into the function $x(t)$ is called an inverse operational transformation and its result $x(t)$ – an inverse Laplace transform of $X(s)$. The most important properties of this transformation are the following:

1. For any functions $x_1(t)$ and $x_2(t)$ for which there exist Laplace transforms, and for any real numbers $c_1$ and $c_2$

$$\mathcal{L}\{c_1 x_1(t) + c_2 x_2(t)\} = c_1 X_1(s) + c_2 X_2(s)$$

where $X_1(s)$ and $X_2(s)$ are Laplace transforms of the functions $x_1(t)$ and $x_2(t)$, respectively.

2. For any function $x(t)$ for which there exists Laplace transform $X(s)$

$$\mathcal{L}\{\dot{x}(t)\} = sX(s) - x(0) .$$

Under the condition $x(0) = 0$, differentiating of the function $x(t)$ corresponds to multiplying $X(s)$ by $s$. That is why the variable $s$ is called a *differential operator*.

A linear differential equation for one-dimensional functions $x(t)$ and $y(t)$

$$x^{(m)}(t) + a_{m-1}x^{(m-1)}(t) + ... + a_1\dot{x}(t) + a_0 x(t) = b_l y^{(l)}(t) + b_{l-1}y^{(l-1)}(t) + ... + b_1\dot{y}(t) + b_0 y(t) \tag{D1}$$

in the operational transforms "language" takes the form of an algebraic equation. For zero initial conditions

$$s^m X(s) + a_{m-1}s^{m-1}X(s) + ... + a_1 sX(s) + a_0 X(s) = b_l s^l Y(s) + b_{l-1}s^{l-1}Y(s) + ... + b_1 sY(s) + b_0 Y(s)$$

and

$$X(s) = \frac{b_l s^l + b_{l-1}s^{l-1} + ... + b_1 s + b_0}{s^m + a_{m-1}s^{m-1} + ... + a_1 s + a_0} Y(s). \tag{D2}$$

To solve the differential equation (D1) one should determine $X(s)$ according to the formula (D2) and then determine or read over from suitable tables the inverse transform $x(t)$. If $X(s)$ is a rational function of $s$ (i.e. is a ratio of two polynomials) then it is convenient to present $X(s)$ as a sum of so called partial fractions and read from the tables the inverse transforms for the separate components. The list of the Laplace transforms for the most frequently used functions is presented in Table D1.

Procedures of handling the operational transformations in solving differential equations and in related problems are called an *operational calculus*.

**Table D1**

| $x(t)$ | $X(s)$ |
|---|---|
| $\delta(t)$ | $1$ |
| $\mathbf{1}(t)$ | $\frac{1}{s}$ |
| $t$ | $\frac{1}{s^2}$ |
| $\frac{t^{k-1}}{(k-1)!}$ | $\frac{1}{s^k}$ |
| $e^{-\alpha t}$ | $\frac{1}{s+\alpha}$ |
| $1-e^{-\alpha t}$ | $\frac{\alpha}{s(s+\alpha)}$ |

| | |
|---|---|
| $t^{k-1}e^{\alpha t}$ | $\dfrac{k-1}{(s-\alpha)^k}$ |
| $\sin\omega t$ | $\dfrac{\omega}{s^2+\omega^2}$ |
| $\cos\omega t$ | $\dfrac{s}{s^2+\omega^2}$ |
| $e^{-\alpha t}\sin\omega t$ | $\dfrac{\omega}{(s+\alpha)^2+\omega^2}$ |
| $e^{-\alpha t}\cos\omega t$ | $\dfrac{s+\alpha}{(s+\alpha)^2+\omega^2}$ |

A *discrete operational transform* or *Z-transform* of the discrete function $x_n$ determined for $n \geq 0$ is defined as a function of a complex variable $z$, assigned to the function $x_n$ according to the following formula

$$X(z) = \sum_{n=0}^{\infty} x_n z^{-n} \triangleq T_Z\{x_n\}.$$

Such a mapping is called a discrete operational transformation or *Z*-transformation, the transformation of the function $X(z)$ into the function $x_n$ is called an inverse discrete operational transformation and its result $x_n$ – an inverse *Z*-transform. The most important properties of this transformation are the following:

1. For any functions $x_{1n}$ and $x_{2n}$ for which there exist *Z*-transforms, and for any real numbers $c_1$ and $c_2$

$$T_Z\{c_1 x_{1n} + c_2 x_{2n}\} = c_1 X_1(z) + c_2 X_2(z)$$

where $X_1(z)$ and $X_2(z)$ are *Z*-transforms of the functions $x_{1n}$ and $x_{2n}$, respectively.

2. For any function $x_n$ for which there exists *Z*-transform $X(z)$

$$T_Z\{x_{n+1}\} = zX(z) - zx_0.$$

Under the condition $x_0 = 0$, shifting of the function $x_n$ corresponds to multiplying $X(z)$ by $z$. That is why the variable $z$ is called a *shift operator*.

A linear difference equation for one-dimensional functions $x_n$ and $y_n$

$$x_{n+m} + a_{m-1}x_{n+m-1} + ... + a_1 x_{n+1} + a_0 x_n = b_l y_{n+l} + b_{l-1}y_{n+l-1} + ... + b_1 y_{n+1} + b_0 y_n \quad \text{(D3)}$$

in the operational transforms "language" takes the form of an algebraic equation. For zero initial conditions

$$z^m X(z) + a_{m-1}z^{m-1}X(z) + ... + a_1 zX(z) + a_0 X(z) = b_l z^l Y(z) + b_{l-1}z^{l-1}Y(z) + ... + b_1 zY(z) + b_0 Y(z)$$

and

$$X(z) = \frac{b_l z^l + b_{l-1}z^{l-1} + ... + b_1 z + b_0}{z^m + a_{m-1}z^{m-1} + ... + a_1 z + a_0} Y(z) . \quad \text{(D4)}$$

To solve the difference equation (D3) one should determine $X(z)$ according to the formula (D4) and then determine or read from the suitable tables the inverse transform $x_n$. If $X(z)$ is a rational function of $z$ then, after presenting it in the form of a sum of partial fractions, we obtain $x_n$ as a sum of inverse transforms for the partial fractions, read from the tables. The list of $Z$-transforms for the most frequently used functions is presented in Table D2.

**Table D2**

| $x_n$ | $X(z)$ |
|---|---|
| $\mathbf{1}(n)$ | $\frac{z}{z-1}$ |
| $n$ | $\frac{z}{(z-1)^2}$ |
| $n^2$ | $\frac{z^2+z}{(z-1)^3}$ |
| $\alpha^n$ | $\frac{z}{z-\alpha}$ |
| $n\alpha^n$ | $\frac{z}{(z-\alpha)^2}$ |
| $\sin \omega n$ | $\frac{z \sin \omega}{z^2 - 2z\cos\omega + 1}$ |

| | |
|---|---|
| $\cos \omega n$ | $\dfrac{z^2 - z\cos\omega}{z^2 - 2z\cos\omega + 1}$ |
| $\alpha^n \sin \omega n$ | $\dfrac{z\alpha \sin\omega}{z^2 - 2z\alpha\cos\omega + \alpha^2}$ |
| $\alpha^n \cos \omega n$ | $\dfrac{z^2 - z\alpha\cos\omega}{z^2 - 2z\alpha\cos\omega + \alpha^2}$ |

# References


1. Åström KJ, Wittenmark B (1989) Adaptive control. Addison-Wesley, Reading
2. Athans M, Falb PL (1966) Optimal control. Mc Graw Hill, Cambridge
3. Bellman R (1957) Dynamic programming. Princeton, New Jersey
4. Blazewicz J, Ecker KH, Pesch E, Schmidt G, Weglarz J (1996) Scheduling computer and manufacturing processes. Springer, Berlin Heidelberg
5. Brdys MA, Malinowski K (eds) (1994) Computer aided control system design methods, tools and related topics. World Scientific, Singapore New Jersey London
6. Brogan WL (1991) Modern control theory. Prentice-Hall, Englewood Cliffs
7. Bubnicki Z (1964) On the stability condition of nonlinear sampled-data systems. IEEE Trans on AC 9:272–274
8. Bubnicki Z (1966) Convergence of aproximation processes in discrete systems. Technical University Press, Wroclaw (in Polish)
9. Bubnicki Z (1967) Automatic identification of linear static plant by degressive feedback and digital computer. Electronics Letters 3
10. Bubnicki Z (1967) On the convergence condition in discrete optimisation systems. Automation Remote Control (Automatika i Telemechanika) 10:115–123
11. Bubnicki Z (1968) Automatic identification of nonlinear static plant by degressive feedback and digital computer. Electronics Letters 4
12. Bubnicki Z (1968) On the linear conjecture in the deterministic and stochastic stability of discrete systems. IEEE Trans on AC 13:199–200
13. Bubnicki Z (1978) Two-level optimization and control of the complex of operations. In: Proc of VII World IFAC Congress, vol 2. Pergamon Press, Oxford
14. Bubnicki Z (1980) Identification of control plants. Elsevier, New York Oxford Amsterdam
15. Bubnicki Z (1984) Optimal models of complex operation systems. In: Proc of VI Int Congress of Cybernetics and Systems. AFCET, Paris

16. Bubnicki Z (1985) Global modelling and identification of complex systems. In: Proc of the 7th IFAC/IFORS Symposium on Identification and Process Parameter Estimation. Pergamon Press, York Oxford

17. Bubnicki Z (1987) Time-optimal control of large scale operation systems with the use of globally optimal models. Systems Science 13

18. Bubnicki Z (1990) Introduction to expert systems. Polish Scientific Publishers, Warsaw (in Polish)

19. Bubnicki Z (1992) Decomposition of a system described by logical model. In: Trappl R (ed) Cybernetics and systems research, vol 1. World Scientific, Singapore, pp 121–128

20. Bubnicki Z (1993) Fundations of management information systems. Technical University Press, Wroclaw (in Polish)

21. Bubnicki Z (1996) Adaptive control of operation systems using neural networks. In: Proc of the 11th Int Conference on Systems Engineering, Las Vegas, pp 945–950

22. Bubnicki Z (1997) Logic-algebraic method for a class of dynamical knowledge-based systems. In: Sydow A (ed) Proc of the 15th IMACS World Congress on Scientific Computation, Modelling and Applied Mathematics, vol 4. Wissenschaft und Technik Verlag, Berlin, pp 101–106

23. Bubnicki Z (1997) Logic-algebraic method for a class of knowledge based systems. In: Pichler F, Moreno-Diaz R (eds) Computer aided systems theory. Lecture Notes in Computer Science, vol 1333. Springer, Berlin, pp 420–428

24. Bubnicki Z (1998) Logic-algebraic method for knowledge-based relation systems. Systems Analysis Modelling Simulation 33:21–35

25. Bubnicki Z (1999) Learning processes and logic-algebraic method in knowledge-based control systems. In: Tzafestas SG, Schmidt G (eds) Progress in system and robot analysis and control design. Lecture Notes in Control and Information Sciences, vol 243. Springer, London, pp 183–194

26. Bubnicki Z (1999) Uncertain variables and learning algorithms in knowledge-based control systems. Artificial Life and Robotics 3:155–159

27. Bubnicki Z (2000) General approach to stability and stabilization for a class of uncertain discrete non-linear systems. Int Journal of Control, 73:1298–1306

28. Bubnicki Z (2000) Knowledge validation and updating in a class of uncertain distributed knowledge systems. In: Shi Z, Faltings B, Musen M (eds) Proc of the 16th IFIP World Computer Congress, vol "Intelligent information processing". Publishing House of Electronics Industry, Beijing, pp 516–523

29. Bubnicki Z (2000) Learning algorithms for a class of knowledge-based systems with dynamical knowledge representation. Systems Science 26:15–

27

30. Bubnicki Z (2000) Learning control system for a class of production operations with parametric uncertainties. In: Groumpos PG, Tzes AP (eds) Preprints of IFAC Symposium on Manufacturing, Modeling, Management and Control, Patras, pp 228–233
31. Bubnicki Z (2000) Learning processes in a class of knowledge-based systems. Kybernetes 29:1016–1028
32. Bubnicki Z (2000) Uncertain variables in the computer aided analysis of uncertain systems. In: Pichler F, Moreno-Diaz R, Kopacek P (eds) Computer aided systems theory. Lecture Notes in Computer Science, vol 1798. Springer, Berlin, pp 528–542
33. Bubnicki Z (2001) Application of uncertain variables and logics to complex intelligent systems. In: Proc of the 6th International Symposium on Artificial Life and Robotics, Tokyo, vol 1, pp 220–223
34. Bubnicki Z (2001) Application of uncertain variables to control for a class of production operations with parametric uncertainties. In: Preprints of IFAC Workshop on Manufacturing, Modelling, Management and Control, Prague, pp 29–34
35. Bubnicki Z (2001) Uncertain variables – a new tool for analysis and design of knowledge-based control systems. In: Hamza MH (ed) Modelling, identification and control, vol II. Acta Press, Zurich, pp 928–930
36. Bubnicki Z (2001) Uncertain variables and their application to decision making. IEEE Trans on SMC, Part A: Systems and Humans 31:587–592
37. Bubnicki Z (2001) Uncertain variables and their applications for a class of uncertain systems. Int Journal of Systems Science 32:651–659
38. Bubnicki Z (2002) A probabilistic approach to stability and stabilization of uncertain discrete systems. In: Proc of the 5th IFAC Symposium "Nonlinear Control Systems", vol 2. Elsevier Science, Oxford, pp 1071–76
39. Bubnicki Z (2002) A unified approach to descriptive and prescriptive concepts in uncertain decision systems. Systems Analysis Modelling Simulation 42:331–342
40. Bubnicki Z (2002) Application of uncertain variables to decision making in a class of distributed computer systems. In: Proc of the 17th IFIP World Computer Congress, vol "Intelligent information processing". Kluwer Academic Publishers, Norwell MA, pp 261–264
41. Bubnicki Z (2002) Application of uncertain variables for a class of intelligent knowledge-based assembly systems. In: Proc of the 7th Int Symposium on Artificial Life and Robotics, Oita, Japan, pp 98–101
42. Bubnicki Z (2002) Learning process in a class of computer integrated manu-

facturing systems with parametric uncertainties. Journal of Intelligent Manufacturing 13:409–415

43. Bubnicki Z (2002) Uncertain logics, variables and systems. Springer, Berlin London New York
44. Bubnicki Z (2002) Uncertain variables and their applications for control systems. Kybernetes 31:1260–1273
45. Bubnicki Z (2002) Uncertain variables and their applications to stability analysis and stabilization of uncertain discrete systems. In: Proc of the 5th Portuguese Conference on Automatic Control CONTROLO 2002, Aveiro, Portugal, pp 629–634
46. Bubnicki Z (2003) Application of uncertain variables and learning algorithm to task allocation in multiprocessor systems. In: Proc of the 8th Int Symposium on Artificial Life and Robotics, Oita, Japan, vol 1, pp I-19–I-22
47. Bubnicki Z (2003) Application of uncertain variables in a class of control systems with uncertain and random parameters. In: Proc of the European Control Conference, Cambridge, UK
48. Bubnicki Z (2003) Application of uncertain variables to a project management under uncertainty. Systems Science 29:65–79
49. Bubnicki Z (2003) Application of uncertain variables to task and resource distribution in complex computer systems. In: Moreno-Diaz R, Pichler F (eds) Computer aided systems theory. Lecture Notes in Computer Science, vol 2809. Springer, Berlin Heidelberg, pp 38–49
50. Bubnicki Z (2003) Soft variables as a generalization of uncertain, random and fuzzy variables. In: Neural networks and soft computing. Advances in Soft Computing. Physica-Verlag, Heidelberg New York, pp 4–17
51. Bubnicki Z (2003) Stability and stabilization of discrete systems with random and uncertain parameters. In: Proc of the 15th IFAC World Congress, vol E "Robust control". Pergamon, Oxford, pp 193–198
52. Bubnicki Z (2004) Analysis and decision making in uncertain systems. Springer, Berlin London New York
53. Bubnicki Z (2004) Application of uncertain variables and learning algorithms in a class of distributed knowledge systems. In: Proc of the 18th IFIP World Computer Congress, vol "The Symposium on Professional Practice in AI". Laboratoire d'Automatique et d'Analyse des Systemes, Toulouse, pp 111–120
54. Bubnicki Z (2004) Uncertain variables and learning process in an intelligent transportation system with production units. In: Preprints of the 5th IFAC/EURON Symposium on Intelligent Autonomous Vehicles IAV2004, Lisbon
55. Bubnicki Z (2004) Uncertain variables and systems – new problems and re-

sults. In: Artificial intelligence and soft computing – ICAISC 2004. Lecture Notes in Artificial Intelligence, vol 3070. Springer, Berlin, pp 17–29

56. Bubnicki Z (2004) Uncertain variables and their applications in knowledge-based decision systems. In: Proc of the 2nd Int IEEE Conference "Intelligent Systems", Varna, Bulgaria, vol 1, pp 13–18
57. Bubnicki Z, Lebrun A (1980) Stochastic approach to two-level optimization in the complex of operations. Lecture Notes in Control and Information Sciences, vol 23. Springer, Berlin
58. Bubnicki Z, Wolkowinski K (1998) On the allocation problem in a production system with transport of materials. Systems Science 24:88–89
59. Bubnicki Z, Staroswiecki M, Lebrun A (1981) Optimal planning of production under uncertain operating conditions. In: Proc of VIII World IFAC Congress, vol 9. Pergamon Press, Oxford
60. Corne D, Dorigo M, Glover F (eds) (1999) New ideas of optimization. Advanced Topics in Computer Science. Mc Graw-Hill, London
61. De Wilde P (1997) Neural network models: Theory and projects. Springer, London
62. Driankov D, Hellendoorn H, Reinfrank M (1993) An introduction to fuzzy control. Springer, Berlin
63. Ellis G (2000) Control system design guide. Academic Press, London New York
64. Findeisen W, Bailey FN, Brdys K, Malinowski K, Tatjewski P, Wozniak A (1980) Control and coordination in hierarchical systems. Wiley, London
65. Franklin GF, Powell JD, Workman ML (1998) Digital control of dynamic systems. Addison-Wesley, Reading
66. Goodwin GC, Sin KS (1984) Adaptive filtering. Prediction and control. Prentice-Hall, New York
67. Goodwin GC, Graeb SF, Salgado ME (2001) Control system design. Prentice-Hall, New Jersey
68. Gorecki H (1989) Analysis and synthesis of time delay systems. PWN – Wiley, Warsaw – Chichester
69. Gorecki H (ed) (1993) Advances in optimization and stability of dynamical Systems. AGH, Cracow
70. Gupta MM, Sinha NK (eds) (1996) Intelligent control systems. Theory and applications. IEEE Press, New York
71. Gupta S (2002) Elements of control systems. Prentice-Hall, New York
72. Hostetter GH (2002) Design of feedback control systems. Oxford University

Press, Oxford

73. Isidori A (1999) Nonlinear control systems. Springer, London
74. Jozefczyk J (2001) Decision making in complexes of operations. Technical University Press, Wroclaw (in Polish)
75. Kacprzyk J (1997) Multistage fuzzy control. Wiley, Chichester
76. Kaczorek T (1993) Linear control systems. Wiley, New York
77. Kaczorek T (2001) Positive 1-D and 2-D systems. Springer, Berlin
78. Kaufman H, Barkana I, Sobel K (1998) Direct adaptive control algorithms. Springer, New York
79. Khalil HK (2002) Nonlinear systems. Prentice-Hall, New Jersey
80. Klamka J (1991) Controllability of dynamical systems. Kluwer Academic Publishers, Dordrecht
81. Klir GJ, Folger TA (1988) Fuzzy sets, uncertainty, and information. Prentice-Hall, Englewood Cliffs New York
82. Klir GJ, Yuan B (1995) Fuzzy sets and fuzzy logic. Prentice-Hall, New Jersey
83. Kokotovic PV (1991) Foundations of adaptive control. Springer, Berlin
84. Lewis L, Syrmos VL (1995) Optimal control. Wiley, New York
85. Lewis PH, Yang C (1997) Basic control systems engineering. Prentice-Hall, New Jersey
86. Liu B (2004) Uncertainty theory. Springer, Berlin New York
87. Liu GP (2001) Nonlinear identification and control: A neural network approach. Springer, Berlin New York
88. Lyshevski SE (2001) Control systems theory with engineering applications. Birkhauser, Boston
89. Michalski RS, Tecuci G (eds) (1993) Machine learning: A multistrategy approach, vol 4. Morgan Kaufman, San Mateo
90. Naidu DS (2003) Optimal control systems. CRC Press, Boca Raton
91. Niederlinski A, Moscinski J, Ogonowski Z (1995) Adaptive control. Polish Scientific Publishers, Warsaw (in Polish)
92. Nikolopoulos C (1997) Expert systems. Marcel Dekker, New York
93. Nise NS (2004) Control systems engineering. Wiley, New York
94. Ogata K (1995) Discrete-time control systems. Prentice-Hall, Englewood Cliffs
95. Pedrycz W, Gomide F (1998) Fuzzy sets. MIT Press, Cambridge MA

96. Rutkowski L (2004) New soft computing techniques for system modelling, pattern classification and image processing. Springer, Berlin

97. Schalkoff RJ (1997) Artificial neural networks. Mc Graw Hill, Cambridge

98. Shinners SM (1998) Modern control system theory and design. Wiley, New York

99. Siwek L (2000) Control system of calls admission in computer network using fuzzy approach. In: Proc of the 16th IMACS World Congress, Lausanne

100. Solodownikow WW (1960) Statistical dynamics of linear control systems. Fizmatgiz, Moscow

101. Sriram RD (1997) Intelligent systems for engineering. Springer, London

102. Tsypkin JZ (1963) Sampled-data systems theory. Fizmatgiz, Moscow (in Russian)

103. Tsypkin JZ (1970) Fundations of learning systems theory. Nauka, Moscow (in Russian)

104. Vinter RB (2000) Optimal control. Birkhauser, Boston

105. Wagner HM (1975) Principles of operations research. Prentice Hall, Englewood Cliffs

106. Zurada JM (1992) Artificial neural networks. West Publishers, St. Paul MM

# Index