# A MATLAB-Based Digital Signal Processing Laboratory Course

Sanjit K. Mitra
Department of Electrical & Computer Engineering
University of California, Santa Barbara, CA 93106-9560
E-mail: mitra@ece.ucsb.edu

**Abstract**

This paper describes a computer-based DSP laboratory course that supplements a lecture course on the subject. The course consists of a series of laboratory exercises with each exercise containing a number of projects to be carried out on a computer. The programming language used in this course is MATLAB, widely used for high-performance numerical computation and visualization. Each exercise consists of a series of projects with each project followed by a series of questions the student must answer before embarking on the following project. The questions have been designed to ensure that the student understands the basic principles of the topic under investigation or the key features of the program. In some projects, the student is asked to modify the program to perform other experiments or determine the effect of new MATLAB commands. Each exercise also includes in the beginning a section summarizing the materials necessary for a quick review of DSP materials necessary to carry out the projects included in the exercise. Another novel feature of this laboratory is that the report for each exercise is partly written and stored in the computer. The student adds the results of his/her project as he/she continues with the series of the projects and also provides the answers to the questions after each project is completed. This feature permits the student to complete more work in a specified amount of time than that would have been possible without it. The completed laboratory report can also serve as a guide for writing reports in other laboratory courses.

## 1. Introduction

Digital signal processing (DSP) is concerned with the representation of signals as a sequence of numbers and the algorithmic operations carried out on the signals to extract specific information contained in them. In barely 30 years the field of digital signal processing has matured considerably due to the phenomenal growth in both research and applications, and almost every university is now offering at least one or more courses at the upper division and/or first-year graduate level on this subject. With the increasing availability of powerful personal computers and workstations at affordable prices, and the availability of powerful, and easy-to-use software packages specifically designed for signal processing applications, it has become easier to provide the student with a practical environment to verify the concepts and the algorithms learnt in a lecture course.

This paper describes a computer-based DSP laboratory course developed at the University of California, Santa Barbara supplementing a lecture course on the subject. The programming language used in this course is MATLAB, widely used for high-performance numerical computation and visualization.

## 2. The Course Structure

A laboratory course should have three objectives. The first objective should be to provide the student with tools to verify the theory and the algorithms discussed in the lecture part of the course. The second objective should be to go beyond what is being taught in the class and teach the student some of the practical aspects of the subject. The third objective should be to provide the student with real-life design experience for specific practical applications. The laboratory course outlined in this paper meets all of the above three objectives. It provides laboratory verification of concepts and theories, and realistic hands-on design experience thereby supplementing and complementing the lecture course in DSP. In addition, it teaches the student the proper approach to developing the laboratory notebook containing the results of the experiments and their interpretation.

The course also assumes that the student has no background in MATLAB and teaches him/her through tested programs in the first half of the course the basics of this powerful language in solving important problems in signal processing. In the second half of the course the student is asked to write the necessary MATLAB programs to carry out the projects. We believe students learn the intricacies of problem solving with MATLAB faster by using tested, complete programs and later writing simple programs to solve specific problems.

The laboratory course has been designed as a set of exercises with each exercise containing a series of projects. The projects are designed as self-standing experiments. They begin with simple assignments to permit the student get familiar with all the tools being used and gradually get more and more complex. The projects have been designed also to permit the instructor to modify them to suit his/her teaching objectives.

We believe the student performing the experiment must understand the results obtained in each part of the project. To this end, each project is followed by a series of questions which the student must answer before proceeding on to the next project.

There are four specific advantages to a computer-based laboratory course: (1) It provides individualized learning instead of group learning as each student works individually at a single computer. (2) It provides a maximum hands-on interaction. (3) Each laboratory exercise has been designed as a self-guided set of projects permitting the student to work at his/her pace to maximize the learning. (4) As most students now have their own personal computers, they can work on these projects at home at their convenience.

## 3. The Exercises

A list of the exercises already developed are given below. This list also includes the project titles included under each category.

1) **Discrete-Time Signals: Time Domain Representation**

- Generation of Sequences
- Simple Operation on Sequences
- Other Types of Sequences

2) **Discrete-Time Systems: Time-Domain Representation**

- Simulation of Discrete-Time Systems
- Linear Time-Invariant Discrete-Time Systems

3) **Discrete-Time Signals: Frequency-Domain Representations**

- Discrete-Time Fourier Transform
- Discrete Fourier Transform
- z-Transform

4) **Linear Time-Invariant Discrete-Time Systems: Frequency-Domain Representations**

- Transfer Function and Frequency Response
- Types of Transfer Functions
- Stability Test

5) **Digital Processing of Continuous-Time Signals**

- The Sampling Process in the Time-Domain
- Effect of Sampling in the Frequency Domain
- Design of Lowpass Filters
- A/D and D/A Conversions

6) **Digital Filter Structures**

- Realization of FIR Transfer Functions
- Realization of FIR Transfer Functions

7) **Digital Filter Design**

- IIR Filter Design
- FIR Filter Design

8) **Digital Filter Implementation**

- Simulation of IIR Digital Filters
- Simulation of FIR Digital Filters
- Design of Tunable Digital Filters
- DFT Computation
- Function Approximation

9) **Analysis of Finite Word-Length Effects**

- Generation and Quantization of Binary Numbers

- Coefficient Quantization Effects
- A/D Conversion Noise Analysis
- Analysis of Arithmetic Round-Off Errors
- Low-Sensitivity Digital Filters
- Limit Cycles

10) **Multirate Digital Signal Processing**

- Basic Sampling Rate Alteration Devices
- Decimator and Interpolator Design and Implementation
- Design of Filter Banks
- Design of Nyquist Filters

Each laboratory exercise includes at the beginning a summary of the background materials necessary for a quick review by the student.

Appendix A shows a portion of a sample exercise providing a detailed description of one typical project. Also included as Appendix B is a sample section of the partially written laboratory notebook. As indicated earlier, the student performing the exercises record their results and answers at appropriate places in the notebook and turn in the completed notebook section after the completion of the exercise for evaluation by the instructor. This approach has permitted the student to complete more projects in a limited time period than would have been possible otherwise.

## 4. Concluding Remarks

The laboratory exercises have been class tested for about 8 years and have been well-received by over several hundred students who have taken this course. All of the exercises are now available in a book form [1].

## Appendix A

## Sample Portion of An Exercise

### Laboratory Exercise 6

### DIGITAL FILTER STRUCTURES

A structural representation using interconnected basic building blocks is the first step in the hardware or software implementation of an LTI digital filter. The structural representation provides the relations between some pertinent internal variables with the input and the output which in turn provide the keys to the implementation. This exercise considers the development of structural representations of causal IIR and FIR transfer functions in the form of block diagrams.

### 6.1 REALIZATION OF FIR TRANSFER FUNCTIONS

**Project 6.1 Cascade Realization**

The factored form of a causal FIR transfer function $H(z)$ of length M can be determined from its polynomial form representation which can then be utilized to realize $H(z)$ in a cascade form. To this end, a modified form of Program P6_1 which uses the function `zp2sos` can be employed.

```
% Program P6_1
% Conversion of a rational
% transfer function to its
% factored form
%
num = input('Numerator coefficient
vector = ');
den = input('Denominator
coefficient vector = ');
[z,p,k] = tf2zp(num,den);
sos = zp2sos(z,p,k)
```

**Questions**

**Q6.1** Using Program P6_1 develop a cascade realization of the following FIR transfer function:

$$H_1(z) = 2 + 10\,z^{-1} + 23\,z^{-2} + 34\,z^{-3} + 31\,z^{-4} + 16\,z^{-5} + 4\,z^{-6}. \qquad (6.1)$$

Sketch the block-diagram of the cascade realization. Is $H_1(z)$ a linear-phase transfer function?

**Q6.2** Using Program P6_1 develop a cascade realization of the following FIR transfer function:

$$H_2(z) = 6 + 31\,z^{-1} + 74\,z^{-2} + 102\,z^{-3} + 74\,z^{-4} + 31\,z^{-5} + 6z^{-6}. \qquad (6.2)$$

Sketch the block-diagram of the cascade realization. Is $H_2(z)$ a linear-phase transfer function? Develop a cascade realization of $H_2(z)$ with only 4 multipliers. Show the block-diagram of the new cascade structure.

## 6.2 REALIZATION OF IIR TRANSFER FUNCTIONS

### Project 6.2 Cascade and Parallel Realizations

The factored form of a causal IIR transfer function $H(z)$ of order $N$ can be determined from its rational form representation which then can be used to realize $H(z)$ in a cascade form. To this end, Program P6_1 can be employed.

**Questions**

**Q6.3** Using Program P6_1 develop a cascade realization of the following causal IIR transfer function:

$$H_1(z) = \frac{3 + 8z^{-1} + 12z^{-2} + 7z^{-3} + 2z^{-4} - z^{-5}}{16 + 24z^{-1} + 24z^{-2} + 14z^{-3} + 5z^{-4} + z^{-5}}. \quad (6.3)$$

Sketch the block-diagram of the cascade realization.

**Q6.4** Using Program P6_1 develop a cascade realization of the following causal IIR transfer function:

$$H_2(z) = \frac{2 + 10z^{-1} + 23z^{-2} + 34z^{-3} + 31z^{-4} + 16z^{-5} + 4z^{-6}}{36 + 78z^{-1} + 87z^{-2} + 59z^{-3} + 26z^{-4} + 7z^{-5} + z^{-6}}. \quad (6.4)$$

Sketch the block-diagram of the cascade realization.

There are two parallel form realizations of a causal IIR transfer function. Parallel form I is based on its partial-fraction expansion in $z^{-1}$ which can be obtained using MATLAB function `residuez`. Parallel form II is based on the partial-fraction expansion in $z$ which is obtained using the function `residue`. Program P6_2 develops both types of parallel realizations.

```
% Program P6_2
% Parallel Form Realizations
% of an IIR Transfer Function
%
num = input('Numerator coefficient
vector = ');
den = input('Denominator
coefficient vector = ');
[r1,p1,k1] = residuez(num,den);
[r2,p2,k2] = residue(num,den);
disp('Parallel Form I')
disp('Residues are'); disp(r1);
disp('Poles are at'); disp(p1);
disp('Constant value'); disp(k1);
disp('Parallel Form II')
disp('Residues are'); disp(r2);
disp('Poles are at'); disp(p2);
disp('Constant value'); disp(k2);
```

**Questions**

**Q6.5** Using Program P6_2 develop the two different parallel-form realizations of the causal IIR transfer function of Eq. (6.3). Sketch the block diagrams of both realizations.

**Q6.6** Using Program P6_2 develop the two different parallel-form realizations of the causal IIR transfer function of Eq. (6.4). Sketch the block diagrams of both realizations.

# Appendix B

# Sample Portion of the Laboratory Notebook

## Laboratory Exercise 6

## DIGITAL FILTER STRUCTURES

### 6.1 REALIZATION OF FIR TRANSFER FUNCTIONS

### Project 6.1 Cascade Realization

A copy of Program P6_1 is given below:

**Answers**

**Q6.1** By running Program P6_1 with num = [2 10 23 34 31 16 4] and den = [1] we arrive at the following second-order factors:

The block-diagram of the cascade realization obtained from these factors is given below:

$H_1(z)$ is a __________-phase transfer function.

**Q6.2** By running Program P6_1 with num = [6 31 74 102 74 31 6] and den = [1] we arrive at the following second-order factors:

The block-diagram of the cascade realization obtained from these factors is given below:

$H_2(z)$ is a __________-phase transfer function.

The block-diagram of the cascade realization of $H_2(z)$ with only 4 multipliers is shown below:

## 6.2 REALIZATION OF IIR TRANSFER FUNCTIONS

### Answers

**Q6.3** By running Program P6_1 with `num = [3 8 12 7 2 –1]` and `den = [16 24 24 14 5 5]` we arrive at the following second-order factors:

The block-diagram of the cascade realization obtained from these factors is given below:

**Q6.4** By running Program P6_1 with `num = [2 10 23 34 31 16 4]` and `den = [36 78 87 59 26 7 1]` we arrive at the following second-order factors:

The block-diagram of the cascade realization obtained from these factors is given below:

A copy of Program P6_2 is given below:

**Q6.5** By running Program P6_2 with `num = [3 8 12 7 2 –1]` and `den = [16 24 24 14 5 5]` we arrive at the partial-fraction expansion of $H_1(z)$ in $z^{-1}$ given by:

and the partial-fraction expansion of $H_1(z)$ in z given by:

The block-diagram of the parallel-form I realization of $H_1(z)$ is thus as indicated below:

The block-diagram of the parallel-form II realization of $H_1(z)$ is thus as indicated below:

**Q6.6** By running Program P6_2 with `num = [2 10 23 34 31 16 4]` and `den = [36 78 87 59 26 7 1]` we arrive at the partial-fraction expansion of $H_2(z)$ in $z^{-1}$ given by:

and the partial-fraction expansion of $H_2(z)$ in z given by:

The block-diagram of the parallel-form I realization of $H_2(z)$ is thus as indicated below:

The block-diagram of the parallel-form II realization of $H_2(z)$ is thus as indicated below:

## Reference


[1] S. K. Mitra, *Digital Signal Processing Laboratory Using MATLAB*, WCB/McGraw-Hill, Burr Ridge, IL, 1999.