A MATLAB CAD TOOL FOR THE DESIGN OF PID CONTROLLERS

by

CHAO LIN

Submitted in partial fulfilment of the requirements for the degree of

Master of Science

Department of Electrical and Computer Science

CASE WESTERN RESERVE UNIVERSITY

August, 2014

CASE WESTERN RESERVE UNIVERSITY

SCHOOL OF GRADUATE STUDIES

We hereby approve the thesis/dissertation of

Chao Lin

candidate for the degree of **Master of Science ***.

Committee Chair

Mario Garcia-Sanz

Committee Member

Vira Chankong

Committee Member

Marc Buchner

Date of Defense

5/20/14

*We also certify that written approval has been obtained for any

proprietary material contained therein.

Table of Contents

LIST OF FIGURES

LIST OF TABLE

Nomenclature

e Error between setpoint and the system output
u Control variable
y System output
y_{sp} Setpoint
C Controller
P Plant
G_l Closed-loop system
K Proportional gain
K_p Static process gain
T_i Integration time
T_d Derivative time
l Load disturbance
n Measurement noise
u_b Reset, usually fixed to $(u_{max} + u_{min})/2$
K_u Ultimate gain
T_u Ultimate period
r Radius of the polar coordinate system
$\emptyset$ Angle of the polar coordinate system
k, ki and kd Parameters of the PID controller
e_p Error in proportional part
e_d Error in derivative part
G_{ff} Transfer function from the setpoint y_{sp} to the control signal u

Acknowledgements

I would like to express my appreciation to my advisor professor Mario Garcia-Sanz. Thanks for giving me the opportunity to take this project and thanks for his time, patience and support. I would also thank to my family and my friends for their understanding and encouragement.

A Matlab Platform for The Design and Tuning Methods of PID Controller Based on Four Different Tuning Methods

Abstract

by

CHAO LIN

The PID controller is the most widely used controller in industry. This thesis project explores the combination of the PID controller tuning methodologies and the use of soft computing methodologies in the design of controllers. It emphasizes ease-of-use for control engineers and provides a friendly interface CAD tool for controller designers.

This Matlab CAD toolbox contains the applications of four specific soft-computing techniques to design PID controllers, in order to get an output with better dynamic and static performance.

The application of the four algorithms to the PID controller make it an optimum system output by searching for the best set of solutions for the PID parameters, while the add-on features on the approximation of the plant model, the set point weight and a filter significantly impart the ability of tuning method itself in a process. The project also discusses the advantages and the disadvantages of the methods by comparing them.

Chapter 1: Introduction

1.1 Motivation

The Proportional-Integral-Derivative (PID) controller, is the most popular type of controller in industry today. A great number of control engineers use such control algorithms in their daily work. The controller's uses are so diversified that control engineers must tune PID controllers to meet specific needs.

PID controller tuning methods have been well-developed since the classic PID controller idea was introduced by John G. Ziegler and Nathaniel B. Nichols in 1940s. Pole placement is a straightforward tuning method introduced by Truxal in 1955, while the dominant pole method, which was introduced by Persson in 1992, specifies a few poles to avoid the difficulty of choosing poles for higher order models.

This project attempts to combine the Matlab and the PID controller design theories with a friendly Matlab GUI interface. It also gives a different angle to look at the design problem by comparing the results of the 4 different PID control tuning methods.

1.2 PID controller and its structure

The basic structure of the PID controller is depicted below:

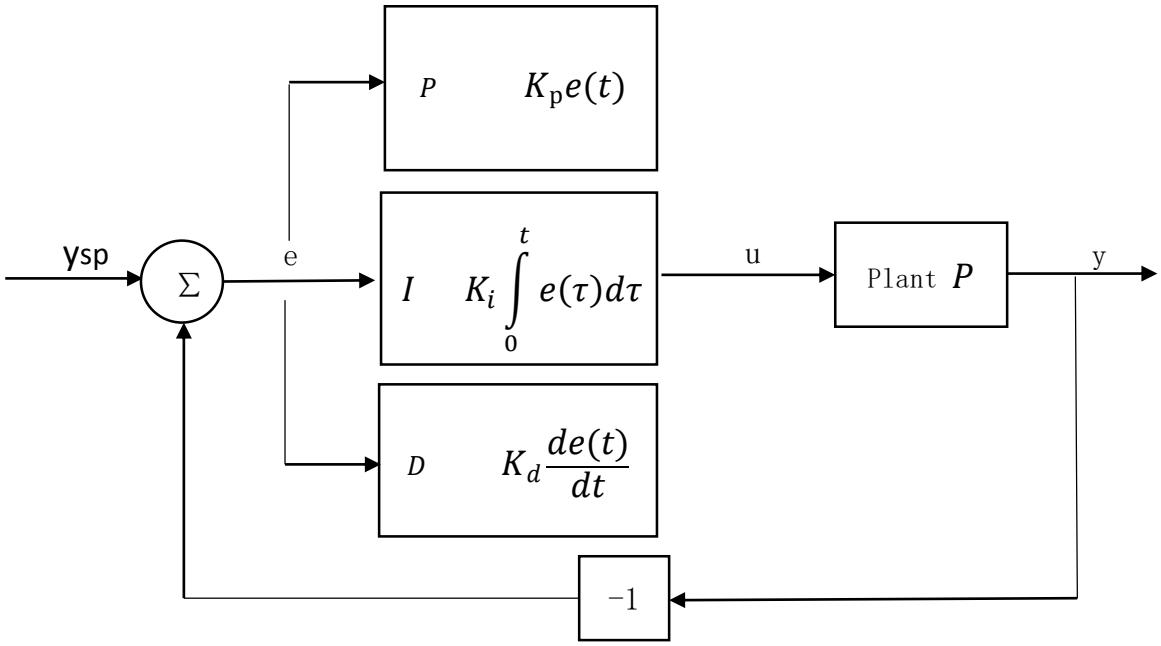

Figure 1 Block diagram of a simple PID feedback system

The equations of this PID is given by:

$$u(t) = K(e(t) + \frac{1}{T_i}\int_0^t e(\tau)d\tau + T_d \frac{de(t)}{dt})$$

Where u is the control variable and e is the error between setpoint and the system output.

1.2.1 Proportional Action

The block diagram of the pure proportional control system is depicted below:

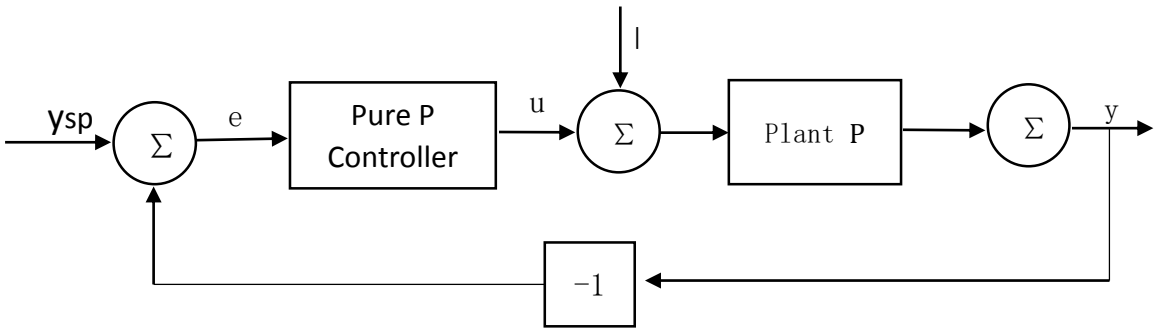

Figure 2 Block diagram of a PID feedback system with pure proportional action

The control equation is reduced to:

$$y = x + n$$

$$x = K_p(u + l)$$

$$u(t) = Ke(t) + u_b$$

Where $u(t)$ is the control signal, K is the proportional gain, K_p is the static process gain, l is a load disturbance, n is the measurement noise, $e(t)$ is the control error ($e = y_{sp} - y$) and u_b is a reset. u_b is usually fixed to $(u_{max} + u_{min})/2$.

Several properties of the pure proportional control can be understood by the following relation according to the above equations:

$$x = \frac{KK_p}{1 + KK_p}(y_{sp} - n) + \frac{KK_p}{1 + KK_p}(l + u_b)$$

First, assuming noise n and reset u_b are zero, the loop gain KK_p then has to be set high in order to make sure the output x is close enough to the setpoint y_{sp} and also

make the output x insensitive to the load disturbance l. However, the loop gain has to be not too large if n is not zero, otherwise the large loop gain will make the system sensitive to the measurement noise n. The situation for the reset u_b is opposite: to ensure it influences the system, the loop gain should be small. Therefore, we have to consider whichever is better depend on different objectives.

Second, $u(t)$ is only equal to u_b when the control error is zero, so that the system output can stay stable and equal to the setpoint y_{sp}. Basically, there will normally be a steady-state error with a pure proportional control.

Third, the above analysis is based on the static model. However, if the process dynamics is considered, the high loop gain will normally be unstable in a closed-loop system. In practice, the maximum loop gain is determined by the process dynamics where the process gain is frequency-dependent in order to avoid the unstable in closed-loop systems.

1.2.2 Integral Action

The purpose of the Integral is to make sure the process output of the plant y equals to the setpoint y_{sp} in steady state. The proportional controller with integral action is given by:

$$u = K(e + \frac{e}{T_i}t)$$

As we discuss above, with a pure proportional controller, there is always a control error in the steady state. Adding an integral portion to a proportional controller will

give additional reset to correct the error. A positive error, which is an error between setpoint and plant output, will give an increasing control signal to the plant and a negative error will give a decreasing control signal to the plant.

The block diagram of the control system, depicted below, shows a proportional controller with an integral action:

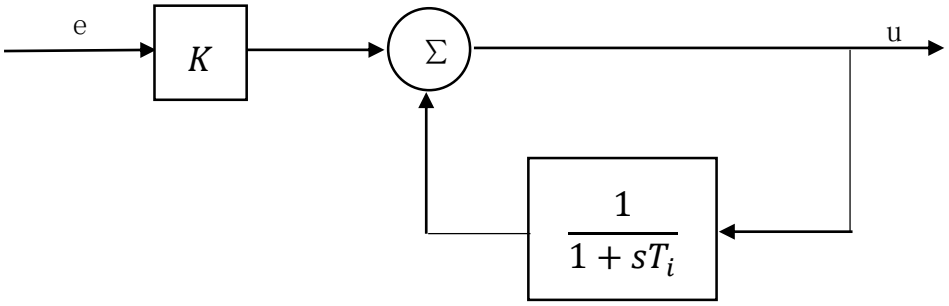

Figure 3 Block diagram of a system with PI controller

The following equations are given from the block diagram:

$$u = Ke + I$$

$$T_i \frac{dI}{dt} + I = u$$

Eventually, we have

$$T_i \frac{dI}{dt} = Ke$$

The integral action is the history on the sum of the error. The features of the system response with PI controller is related to T_i/KK_p approximately. For large values of the integration time T_i, the overshoot of the closed-loop system is larger and the

respond is slower to the error while smaller values of T_i has a faster response to the error. Also the system is more oscillatory with a smaller value of T_i.

1.2.3 Derivative Action

The basic structure of a PD controller is:

$$u(t) = K(e(t) + T_d \frac{de(t)}{dt})$$

Expanse $\mathrm{e(t)} + T_d$ by Taylor series gives:

$$e(t + T_d) \approx e(t) + T_d \frac{de(t)}{dt}$$

The function of the derivative action on a controller is to predict future error and add to the controller in order to improve the stability of a closed-loop system. Assuming there is a change in the control variable, the action of a controller with derivative action may interpret the process output at time T_d ahead, where the prediction is approximately calculated by extrapolating the error by the tangent to the error curve.

1.3 Alternative Representations

There are three different forms of PID controllers.

1.3.1 The Standard Form

Different controllers may have different structures. The controller parameters may also differ if one type of controller is replaced by another type of controller.

This is also a non-interacting form given by the following equation:

$$C(s) = K(1 + \frac{1}{sT_i} + sT_d)$$

The integral time T_i and the derivative part T_d of this form are non-interacting, meaning that the derivative part T_d or any changes in the derivative part T_d does not affect the integral time T_i.

1.3.2 The series form

An interacting form is given by the following equation:

$$C'(s) = K'(1 + \frac{1}{sT'_i})(1 + sT'_d)$$

The integral action T_i and derivative action T_d do interact with each other. A non-interacting controller can represent an interacting controller by:

$$K = K' \frac{T'_i + T'_d}{T'_i}$$

$$T_i = T'_i + T'_d$$

$$T_d = \frac{T'_i T'_d}{T'_i + T'_d}$$

However, an interacting controller can only represent a non-interacting controller only if

$$T_i \geq 4T_d$$

Therefore, the non-interacting controller form is more general.

This thesis uses two algorithm related to the standard form.

1.3.3 The Parallel Form

$$C(s) = k + \frac{k_i}{s} + sk_d$$

The parameters are related to the parameters of the standard form, but the values of the parameters are different:

$$k = K$$

$$k_i = \frac{K}{T_i}$$

$$k_d = KT_d$$

The parallel form is often useful in the analytical calculation of PID controller design, because the parameters appear independently. This is also a ecumenical form because it is easy to obtain pure proportional, integral, or derivative action with number values.

This thesis has two design methods related to this parallel form of PID controller.

1.3.4 The Basic Structure of the Entire System

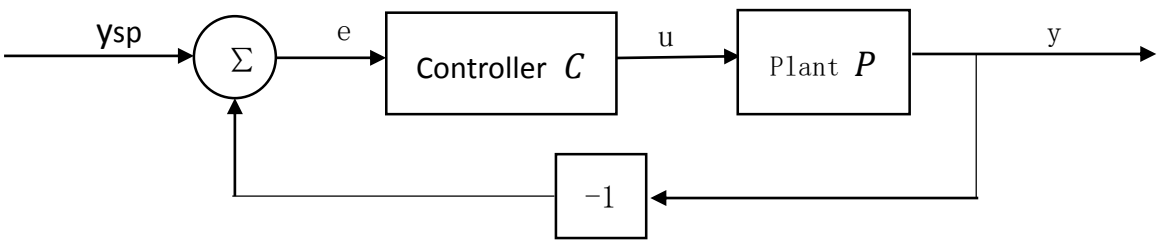

Figure 4 Block diagram of a PID feedback system

The transfer function of this process is as follows:

$$P(s) = \frac{G_l}{1 + G_l}$$

where G_l :

$$G_l(s) = CP$$

A block of diagram of a feedback system is shown in Figure 4. The system has two major components: the process and controller and the feedback loop. The purpose of the feedback system is to control the process variable (y) equal to the desired value (ysp). The control system increases the operation variable when the process variable is less than the desired value disturbance and decreases the operation variable when the process variable is greater than the desired value disturbance. This is achieved by the feedback loop, an explanation of which follows. Assume that the system is in balance and then a disturbance occurs so that the process variable becomes greater than the setpoint which is the desired value disturbance. The error between the setpoint and the process output is then negative, and the controller output decreases, which in turn causes the process output to decrease.

1.4 The Plant Approximation

This thesis is based on the assumption that the transfer function of the process is known.

There are several reasons to simplify the plant model:

1. To simplify the best available model to design a control system to meet certain specifications.
2. To speed up the simulation process. Using a smaller order model can reduce the calculation time.
3. To reduce the control law complexity with only a small change in control system performance by eliminating the unnecessary high orders, which are greater than that which is truly needed in the design validation stage.

In this thesis, the plant transfer functions are approximately reduced to a suitable one pole and two zeros transfer functions. A two-order system catches many processes in the real world, oscillatory systems, and systems with right half-plane zeros. However, it is important to keep in mind that plant reduction approximation error is unavoidable when lower order models approximate the higher models.

For a give causal and rational model P, it can be decomposed to:

$$P(z) = S(z) + U(z)$$

Where S(z) is the stable portion with all poles strictly inside the unit circle and U(z) is the unstable part, only the stable part is approximated.

An eigenvector of the a square matrix S means a non-zero vector v that, when the matrix is multiplied by v, yields a constant multiple of v, the multiplier being commonly denoted by λ:

$$Sv = \lambda v$$

Where S contains all the stable modes of a system.

The eigenvalue λ of each part of the model represents the energy of each part of the plant model. Eliminating all unnecessary states and keeping the larger energy states of the system preserves most of the system's characteristics in terms of stability.

In this thesis, the Matlab function 'balred' is used to implement this idea. 'Balred' first transfers the one zero two poles system into a state-space system, and then decomposes the system into two parts: the stable and unstable portions. This model reduction is based on the hankel singular values of the system in terms of stability. It can achieve a model in which the characteristics of the original system are preserved? States with relatively small eigenvalue values can be safely discarded.

We use an example to illustrate this approximate method as following:

Example 1.1

Plant:

$$P(s) = \frac{s^3 + 4.5s^2 + 5.19s + 1.595}{s^4 + 6s^3 + 11s^2 + 6s + 0.0001}$$

Using balred function in Matlab and the approximated result becomes:

$$P(s) = \frac{0.9911s + 0.5188}{s^2 + 1.952s + 0.0000325}$$

Bode diagram of the plant in the original and approximated plant:

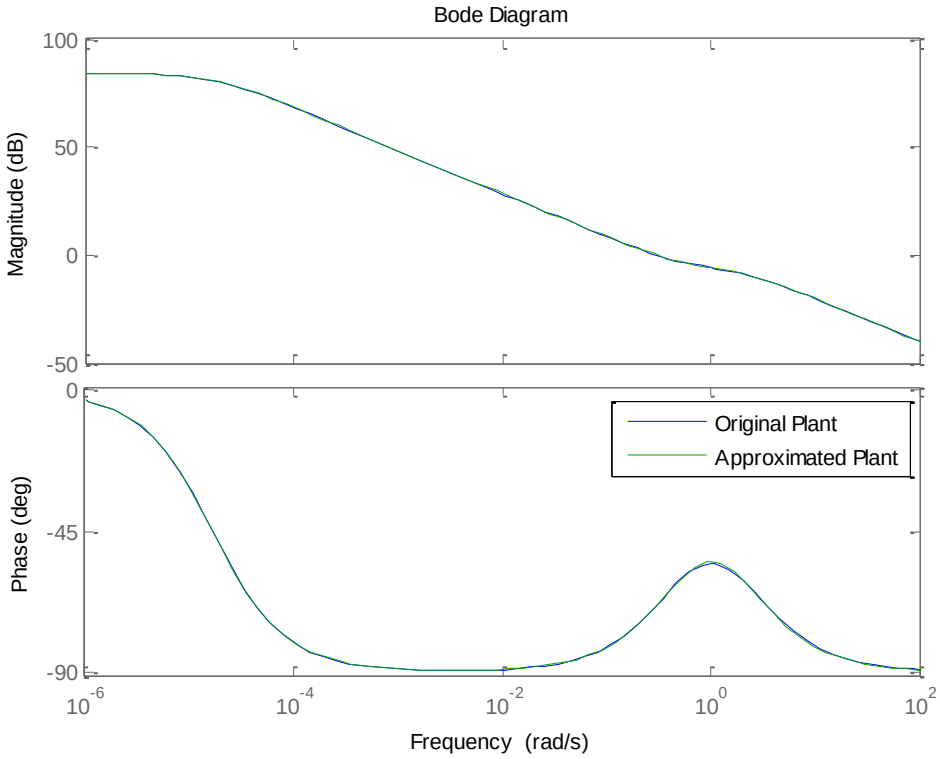

Figure 5 Block diagram of example 1.1

Compare the original and reduced order model. We can see that the original and approximated plants are very similar in terms of the figure view.

The figure below is the hankel singular value of the original and reduced order model.

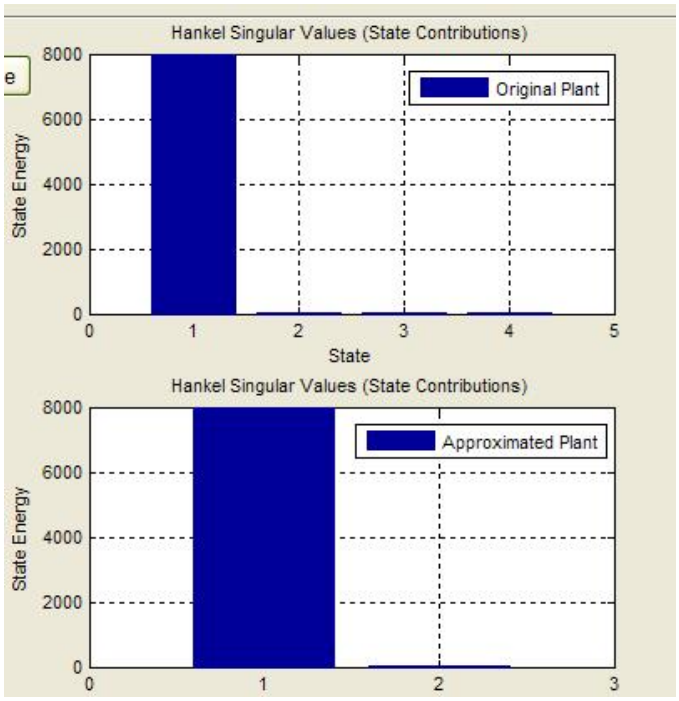

Figure 6 Hankel Sigular Values of example 1.1

The Matlab function ‘balred’ transfers the one zero two poles system into a state-space system, then keeps the two highest energy singular value 1 and 2, and then discards the rest of the singular value. The process is implemented in term of the model stability. Balred is very useful at lease for many of the modeling approximation. The reduced model keeps the majorities’ characteristic of the original model.

1.5 Measurement of the Approximation

Approximated errors are inevitable while using the lower order models represents the higher order models. The performance of model approximation is measured by the corresponding approximation error.

The cost function of the error of the original plant and the approximated plant is given by

$$\mathrm{CF} = |tr_o - tr_a| + |ts_o - ts_a| + |PV_o - PV_a|$$

This is a function based on the step respond of the original plant model and the approximated plant model functions where tr_o and tr_a are the rise time of the step responds of the original and approximated plant, ts_o and ts_a are the settling time and PV_o and PV_a are the Peak value of the step responds.

Chapter 2 the Four Tuning Methods

2.1 Ziegler-Nichols Method

The second design method presented by Ziegler and Nichols is based on a test data of the closed-loop step response of the system. In order to get the data, we first connect a PID controller that sets $T_i = \infty$ and $T_d = 0$ to the process and provide a feedback loop, then increase K until the process oscillate. The gain when this happen is the ultimate gain K_u. And the period of this oscillation is the ultimate period T_u. We then can determine the ultimate gain K_u and the ultimate period T_u by doing such. The closed-loop step response of the system are also characterized by the two parameters K_u and T_u.

The controller parameters can be given in terms of the ultimate gain K_u and the ultimate period T_u as following:

Controller	K	Ti	Td	Tp
P	0.5Ku			
PI	0.4Ku	0.8Tu		1.4Tu
PID	0.6Ku	0.5Tu	0.125Tu	0.85Tu

Table 1 Ziegler-Nichols parameters of PID

The controller is finally given by the standard form of the PID controller:

$$C(s) = K(1 + \frac{1}{sT_i} + sT_d)$$

Or,

$$C(s) = 0.6K_u(1 + \frac{1}{s0.5T_u} + s0.125T_u)$$

2.2 Modified Ziegler-Nichols Method

The modified Ziegler-Nichols method is based on the frequency domain method that position one point of the Nyquist curve to another. With a PID controller, the picked plant point, generally this point is any point form the Nyquist curve, in this case is a specific point $(-1/K_u, 0)$ on the Nyquist curve which is moved by a PID controller to an arbitrary position in the complex plane, so called modified Ziegler-Nichols.

Let the chosen point of the plant P on the Nyquist curve be:

$$A = P(i\omega_0) = r_a e^{i(\pi + \emptyset_a)}$$

Then determine the PID controller move this point to:

$$B = G_l(i\omega_0) = r_b e^{i(\pi+\emptyset_b)}$$

With the PID controller $C(i\omega_0) = r_c e^{i\emptyset_c}$, we obtain:

$$r_b e^{i(\pi+\emptyset_b)} = r_a r_c e^{i(\pi+\emptyset_a+\emptyset_c)}$$

The controller C will be:

$$r_c = \frac{r_b}{r_a}$$

$$\emptyset_c = \emptyset_b - \emptyset_a$$

We then get:

$$K = \frac{r_b \cos(\emptyset_b - \emptyset_a)}{r_a}$$

$$\omega_0 T_d - \frac{1}{\omega_0 T_i} = \tan(\emptyset_b - \emptyset_a)$$

$$T_d = \alpha T_i$$

Where $\alpha = 0.25$,

A Ziegler-Nichols experiment is used to determine the point $(-1/K_u, 0)$ as we discuss above, we have the critical point

$$r_a = -1/K_u$$

$$\emptyset_a = 0$$

The PID controller parameters are then given by:

$$K = K_u r_b cos\emptyset_b$$

$$T_i = \frac{T_u}{\pi}\left(\frac{1 + sin\emptyset_b}{cos\emptyset_b}\right)$$

$$T_d = \frac{T_u}{4\pi}\left(\frac{1 + sin\emptyset_b}{cos\emptyset_b}\right)$$

System with better damping can be achieved by suitable r_b and $\emptyset_b$, which are 0.5 and 0.3491 would be reasonable. A suggestion by Pessen to move the ultimately point to $r_b = 0.41$ and $\emptyset_b = 1.0647$ or $r_b = 0.29$ and $\emptyset_b = 0.8029$ will have better closed-loop system responses.

In this thesis, the ultimately point $r_b = 0.41$ and $\emptyset_b = 1.0647$ is used.

Note that since only one point on the Nyquist curve is moved by the PID controller and fixed in the Nyquist, the closed-loop system can still be changed significantly when the slope of the Nyquist curve is various.

2.3 Pole Placement Method.

The pole placement design method is a method based on the knowledge of the process transfer function.

A process is represented by the second-order model:

$$P = \frac{b_1 s + b_2}{s^2 + a_1 s + a_2}$$

This model has four parameters $b_1\ b_2\ a_1\ a_1$. This one zero and two poles system catches many processes in the real world such as oscillatory systems. We assume that

a parallel PID controller structure with parameters k, ki and kd showed as follow controls the process above:

$$C(s) = k + \frac{k_i}{s} + k_d s$$

A third order characteristic equation of the closed-loop system is then given by:

$$s(s^2 + a_1 s + a_2) + (b_1 s + b_2)(k_d s^2 + ks + k_i) = 0$$

An objective closed-loop characteristic equation of this third-order system should have the following:

$$(s + \alpha\omega_0)(s^2 + 2\zeta\omega_0 s + \omega_0^2) = 0$$

Equating coefficients of equal power in s in above equation and plant model with the PID controller gives the following equations:

$$a_1 + b_2 k_d + b_1 k = (\alpha\omega_0 + 2\zeta\omega_0)(1 + b_1 k_d)$$

$$a_2 + b_2 k + b_1 k_i = (1 + 2\alpha\zeta)\omega_0^2(1 + b_1 k_d)$$

$$b_2 k_i = \alpha\omega_0^3(1 + b_1 k_d)$$

This is a set of linear equations in the controller parameters. The solution is given by

$$k = \frac{a_2 b_2^2 + a_2 b_1 b_2(\alpha + 2\zeta)\omega_0 - (b_2 - a_1 b_1)(b_2(1 + 2\alpha\zeta)\omega_0^2 + \alpha b_1 \omega_0^3)}{b_2^3 - b_1 b_2^2(\alpha + 2\zeta)\omega_0 + b_1^2 b_2(1 + 2\alpha\zeta)\omega_0^2 - \alpha b_1^3 \omega_0^3}$$

$$k_i = \frac{(-a_1 b_1 b_2 + a_2 b_1^2 + b_2^2)\alpha\omega_0^3}{b_2^3 - b_1 b_2^2(\alpha + 2\zeta)\omega_0 + b_1^2 b_2(1 + 2\alpha\zeta)\omega_0^2 - \alpha b_1^3 \omega_0^3}$$

$$k_d = \frac{-a_1 b_2^2 + a_2 b_1 b_2 + b_2^2(\alpha+2\zeta)\omega_0 - b_1 b_2 \omega_0^2(1+2\alpha\zeta) + b_1^2 \alpha \omega_0^3}{b_2^3 - b_1 b_2^2(\alpha+2\zeta)\omega_0 + b_1^2 b_2(1+2\alpha\zeta)\omega_0^2 - \alpha b_1^3 \omega_0^3}$$

These formulas are quite useful in design because, as we discuss above, the one zero two poles plant modeling equation can represent a lot of processes in real world. Incorporating with the second order plant approximation, the pole placement method can even be used to design controllers for higher order system.

2.4 Dominant Pole Design PID controller tuning method.

The pole placement method is attempts to assign all closed-loop poles, while the dominant pole design is to assign a few poles that represent the main characterization. The dominant pole method is first find the quantities $a(\omega_0),\ \ b(\omega_0),\ and\ \varphi(\omega_0)$ to represent the plant, then assign the poles with three parameters under the specifications. And compare the roots of $1 + C(s)P(s) = 0$ and the assigned poles then we will have the PID parameters.

The controller is parameterized as

$$C(s) = k + \frac{k_i}{s} + k_d s$$

Specify three poles of the closed-loop system since the controller has three parameters

We choose them as

$$p_{1,2} = \omega_0 \left(-\zeta_0 \pm i\sqrt{1-\zeta_0^2} \right)$$

$$p_3 = -\alpha_0\omega_0$$

Introduce the quantities $a(\omega_0),\ \ b(\omega_0),\ and\ \varphi(\omega_0)$ defined by

$$P\left(\omega_0 e^{i(\pi-\gamma)}\right) = a(\omega_0)e^{i\varphi(\omega_0)}$$

$$P(-\alpha\omega_0) = -b(\omega_0)$$

The condition that $p_1,\ p_2$ and p_3 and roots of

$$1 + C(s)P(s) = 0$$

Gives the conditions

$$k = -\frac{\alpha_0^2 b(\omega_0)\sin(\gamma+\varphi) + b(\omega_0)\sin(\gamma-\varphi) + \alpha_0 a(\omega_0)\sin 2\gamma}{a(\omega_0)b(\omega_0)(\alpha_0^2 - 2\alpha_0 cos\gamma + 1)sin\gamma}$$

$$k_i = -\alpha_0\omega_0\frac{a(\omega_0)sin\gamma + b(\omega_0)(\sin(\gamma-\varphi) + \alpha_0 sin\varphi)}{a(\omega_0)b(\omega_0)(\alpha_0^2 - 2\alpha_0 cos\gamma + 1)sin\gamma}$$

$$k_d = -\frac{\alpha_0 a(\omega_0)sin\gamma + b(\omega_0)(\alpha_0\sin(\gamma+\varphi) - sin\varphi)}{\omega_0 a(\omega_0)b(\omega_0)(\alpha_0^2 - 2\alpha_0 cos\gamma + 1)sin\gamma}$$

Chapter 3 Setpoint Weighting and Filter

3.1 PID Control Loop Setpoint Weighting

A common form of a control system is shown as in Figure 1.

The control system perform operation on an error that is the difference between the setpoint and the process output. The controller produces a signal by operating on the error, which is then applied to the process. A PID-controller of this form is given by:

$$u(t) = K(e_p + \frac{1}{T_i}\int_0^t e(s)ds + T_d\frac{de_d}{dt})$$

Where the error in the proportional part is,

$$e_p = by_{sp} - y$$

And the error in the derivative part is,

$$e_d = cy_{sp} - y$$

The error in the integral part is,

$$e = y_{sp} - y$$

When introduce b and c into a PID controller, we should have a different PID structure which is explained as following.

A block diagram for a system with PID control is now given by Figure blow:

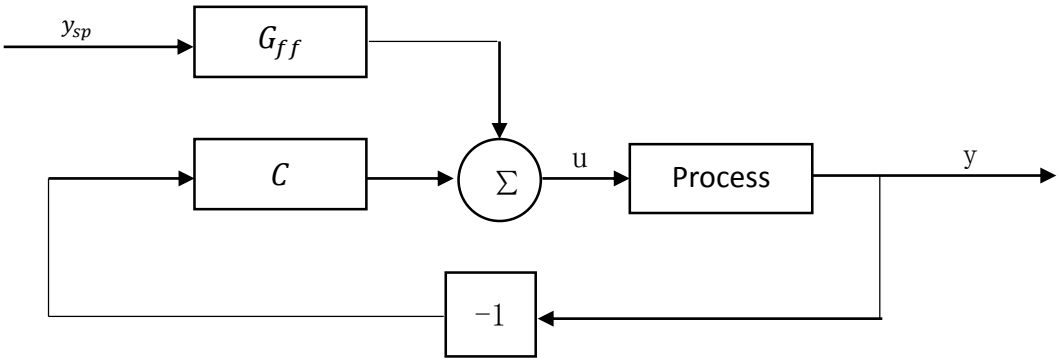

Figure 7 Block diagram for a PID with b and c

Notice that the transfer function from the setpoint y_{sp} to the control signal u is given by

$$G_{ff} = K\left(b + \frac{1}{sT_i} + csT_d\right)$$

And the transfer function from the process variable y to the control variable u is given by

$$C = K\left(1 + \frac{1}{sT_i} + sT_d\right)$$

The closed-loop system is given by

$$G_l = G_{ff}\frac{P}{1 + CP}$$

A control system should have good transient responding with setpoint changes and rejecting load disturbances and noise.

For b = 0, the overshoot changes is smallest and increases when b is increasing. A comparison on the different value of b and how it would affect the system is given by the test example later in the example 1.

The parameter c is normally to be set as zero in order to avoid large transients when a sudden change happened in the setpoint. However, it is acceptable that c is nonzero in a cascade coupling of a secondary controller.

In addition, there are no differences for the values of b and c on how they respond to load disturbances and measurement noise. However, the respond to setpoint change will depend on the value b and c.

3.2 The Filter of the PID controller

If there is high frequency measurement noise, the derivative action may result in trouble. A sinusoidal measurement noise is given by

$$n = a\ sin\omega t$$

Gives the following contribution to the derivative term of the control signal:

$$u_n = KT_d \frac{dn}{dt} = aKT_d\omega\ cos\omega t$$

The derivative term can be changed arbitrary large because that the high frequency measurement noise can change arbitrary which cause the arbitrary changes in the derivative term. The amplitude of the control signal can thus be huge with a noise high frequency (ω). The high frequency gain of the derivative term is then restricted to avoid this difficulty by adding two poles to the derivative term of the PID controller:

$$D = \frac{sk_d}{(1 + \frac{s}{p_1})^2}$$

Chapter 4 Experimental Measurements

Example 1:

One process model is used to test this PID controller platform and the idea how the 4 PID tuning method works.

The process is

$$P = \frac{s + 3}{s^3 + 3s^2 + 3s + 1}$$

Step 1: Menu

In Matlab window input 'Menu' under the platform folder, the following window will appear:

Figure 8 Step 1 'Menu' of example 2

Step 2: Plant Approximation

Click on ‘Plant Approximation’, then the following window appears:

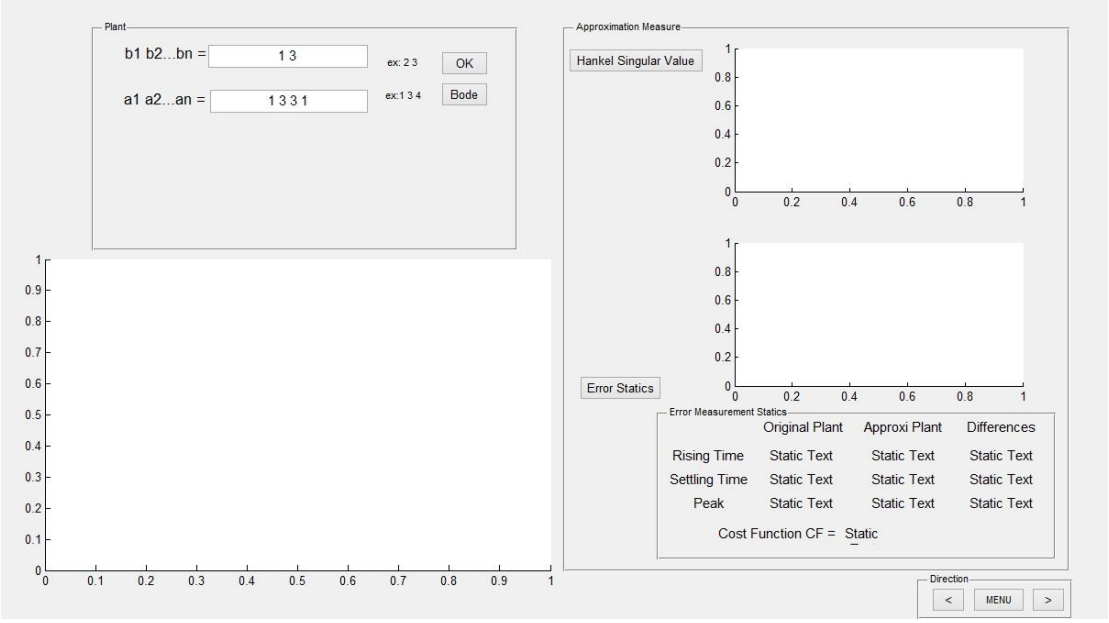

Figure 9 Step 2 'Plant Approximation' of example 2

In the two blank frame, put ‘1’ in the first raw and ‘1 3 3 1’ in the second raw, then click on ‘OK’, ‘Hankel Singular Value’ and the ‘Error Statics’ button, we obtain the following:

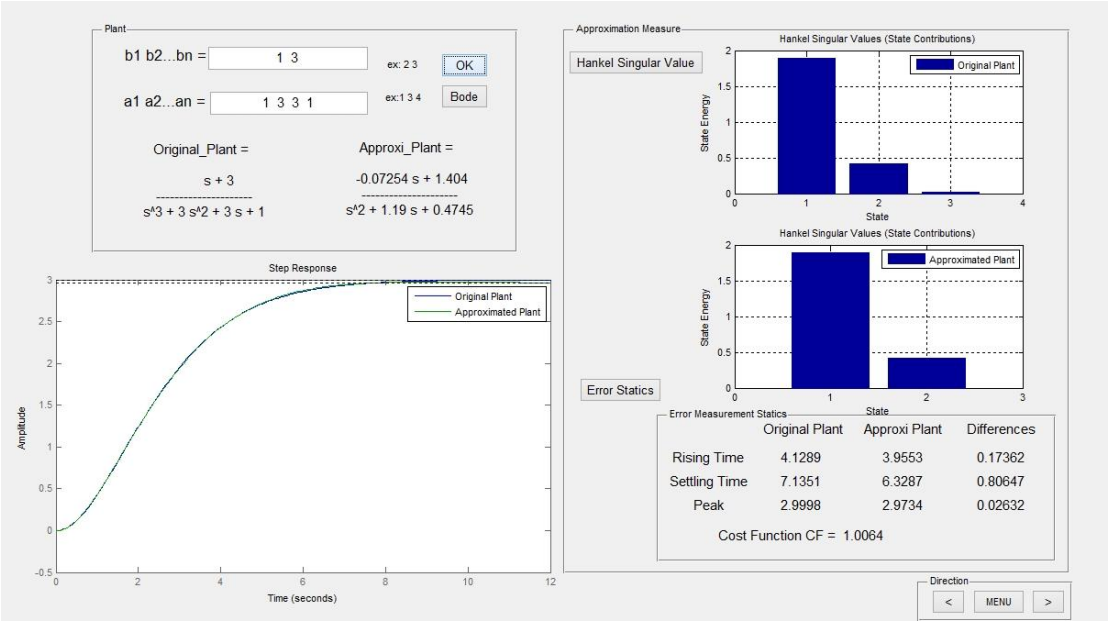

Figure 10 Plant Approximation with plant inputs

The figure has the original and the approximated plant model, the step respond of the two functions, the Hankel Singular Value chart and the error measurement static of the two functions.

The original plant represents the original model of the plant. And the next is the approximated plant which uses the matlab function 'balred' to approximate the original plant based on the hankel singular values of the system. The approximation discards the small eigenvalue values of the original model, so that the reduced model can presents the majorities' characteristic of the original system.

This hankel singular value of the $P = \frac{s+3}{s^3+3s^2+3s+1}$ is also showed in the right side on the figure. 'Balred' discards the relatively small eigenvalue values, which is the number 3 state. The comparison of the step characteristic of the reduced model and the original model is showed in the step figure.

The error statics gives a sense of the number on the error of the two model step characteristic. The cost function gives a total idea how different in the two model.

Click on 'bode', the platform will give you Bode diagram:

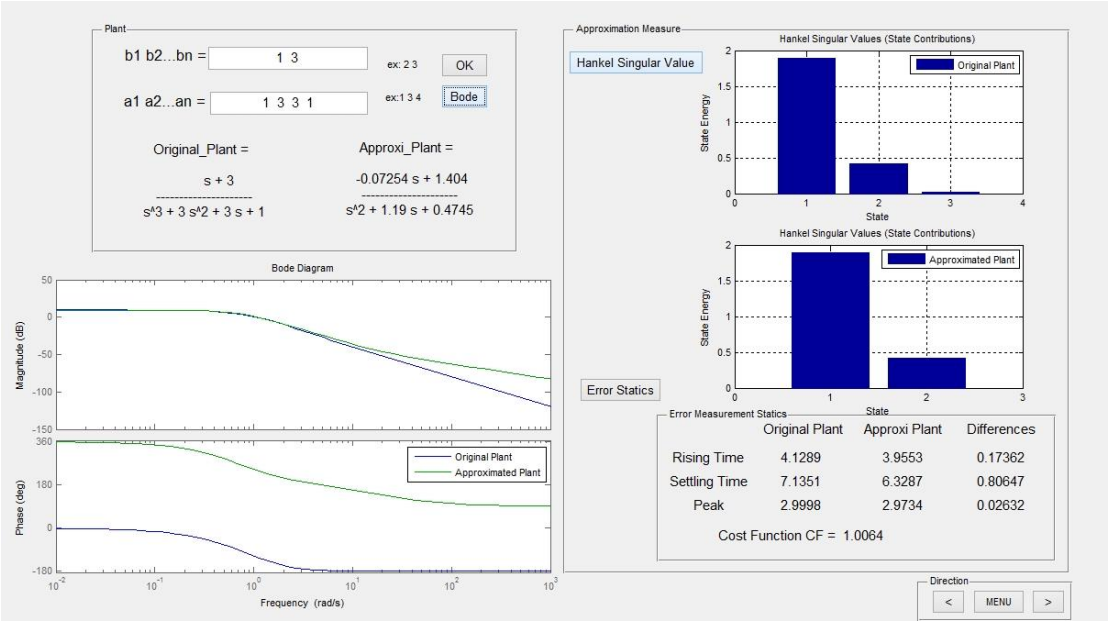

Figure 11 Plant Approximation with bode diagram

Step 3: Objectives

When the functions are ready, next step is the objectives:

The objectives are based on the third order characteristic function:

$$(s + \alpha\omega_0)(s^2 + 2\zeta\omega_0 s + \omega_0^2) = 0$$

Here we choose $\alpha = 0.8, \omega_0 = 1\ and\ \zeta = 0.7$. Then the program gives a step response of rise time 3.27seconds, settling time 5.47 seconds and percent of overshoot 0.38%. The step resonpse figure is also showed below.

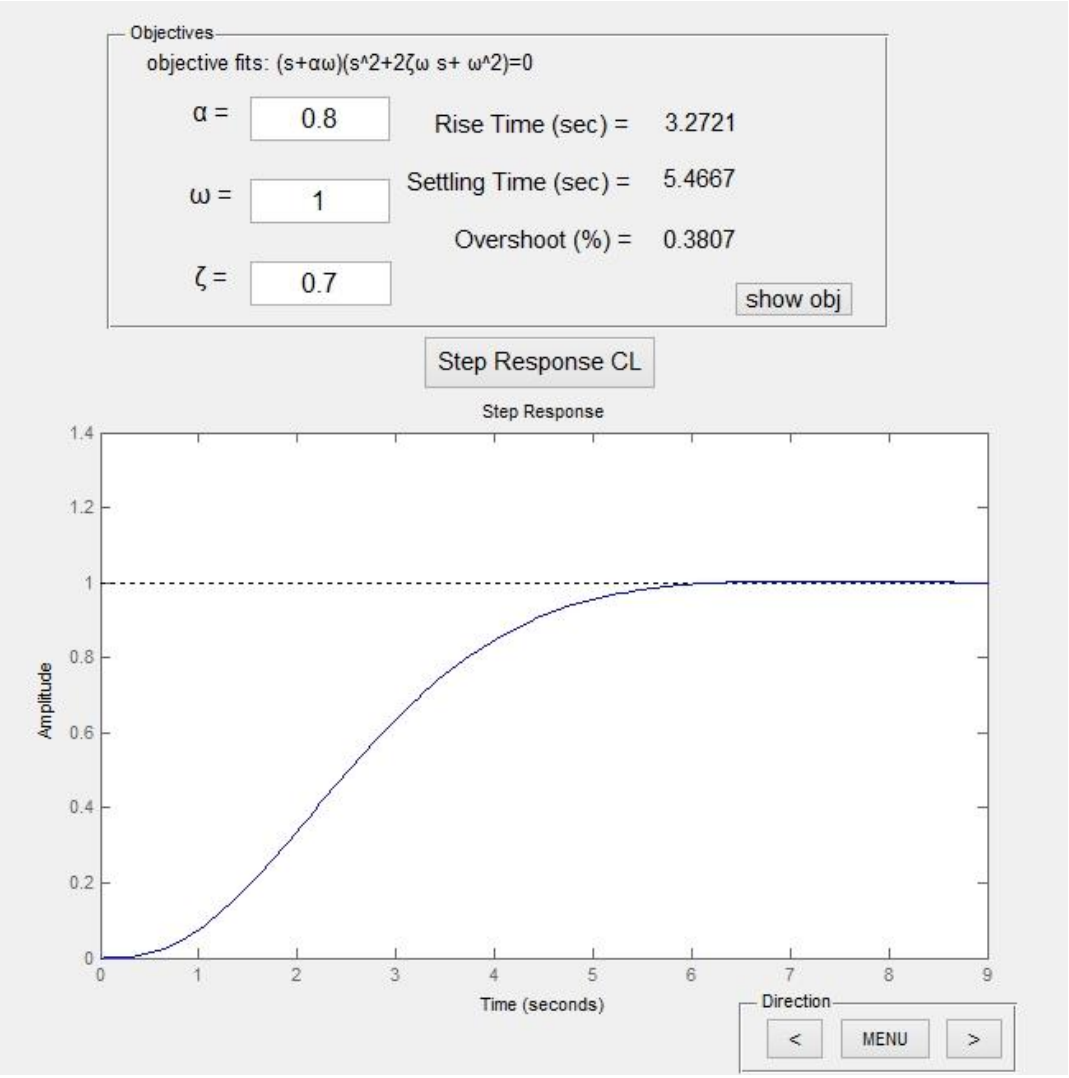

Figure 12 Objectives

Step 4: The Four Controller

In the next step, the controller design process. There are four controller design methods: pole placement, dominant pole, Ziegler Nichols and modified Ziegler Nichols methods.

Ziegler Nichols actually only interact with the ultimate response of the plant model. That means the objectives which is the last step objectives doesn't not have anything to do with this control method. No matter what changes to objectives, there will be no

changes to the Ziegler Nichols method. However, it will be a good reference to compare with other control methods.

Modified Ziegler Nichols is a method based on the frequency domain of the model. In this thesis it pointed the ultimate point in the plant Nyquist to another point. This gives the step response of the feedback system a more freedom and a better step response than the original Ziegler Nichols. But then the step respond will be different depend on different point on the Nyquist curve. And this different point can only be changed in the original source code.

Pole placement is a method based on the mathematical calculation relationship between the product of the plant and controller and the objective function, while dominant pole method is based on the calculation that the needs of the controller of the feedback equals the ideal poles of the feedback system.

They are all followed by the accurate number of rise time, setting time and the percent of overshoot of the step response.

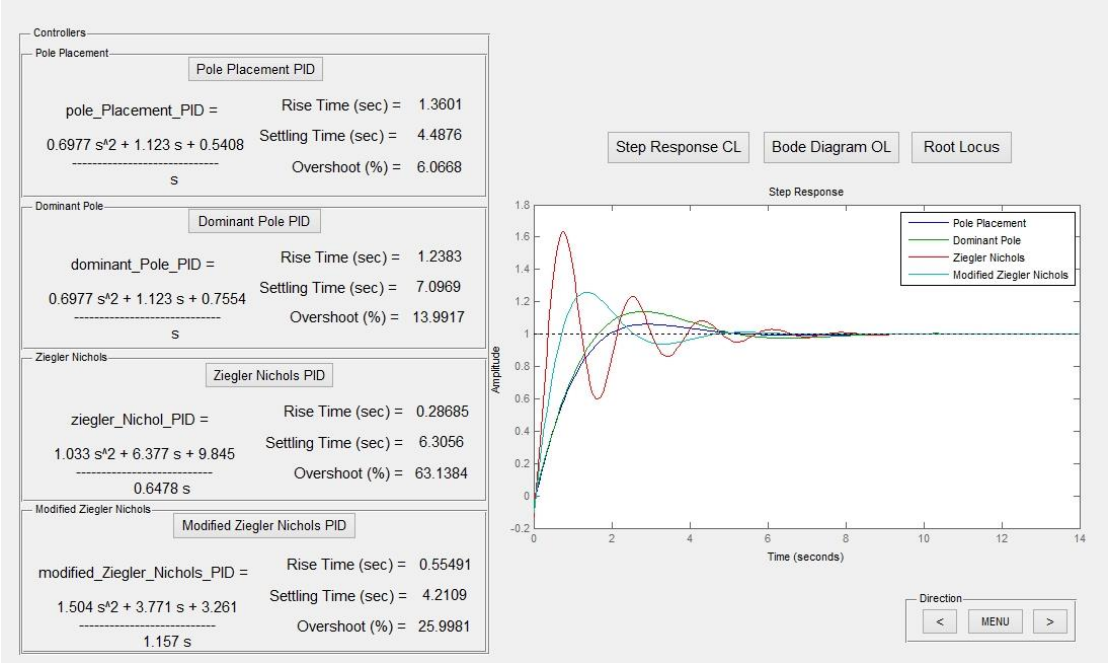

Figure 13 Controller Design with Step Response

The platform also provides Bode diagram and Root Locus, showed as below.

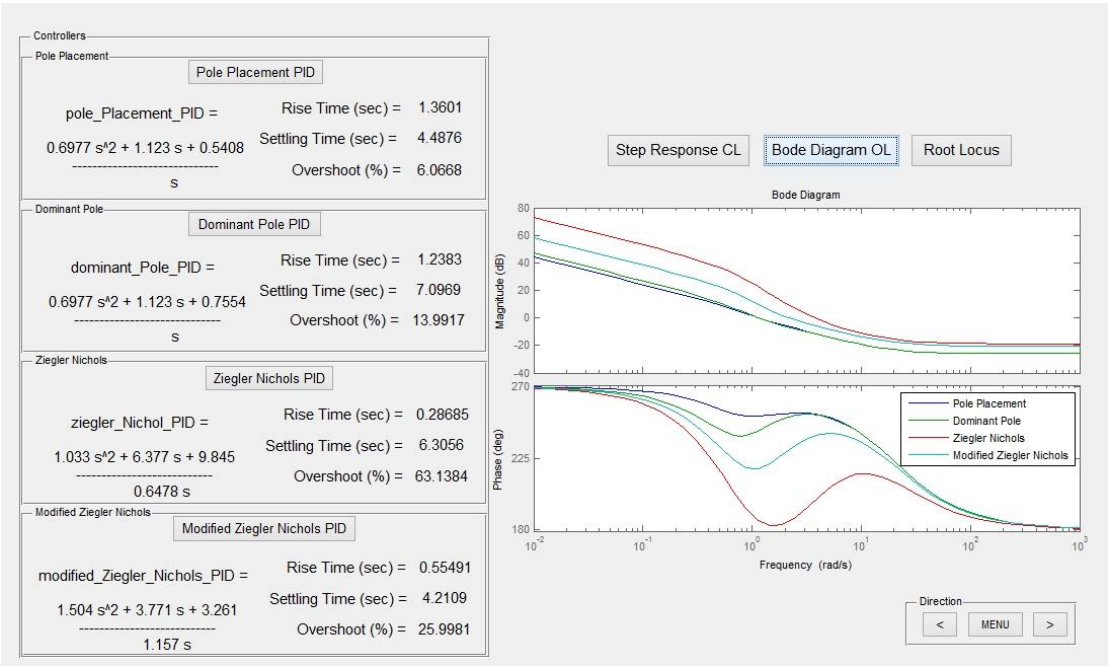

Figure 14 Controller Design with Open Loop Bode Diagram

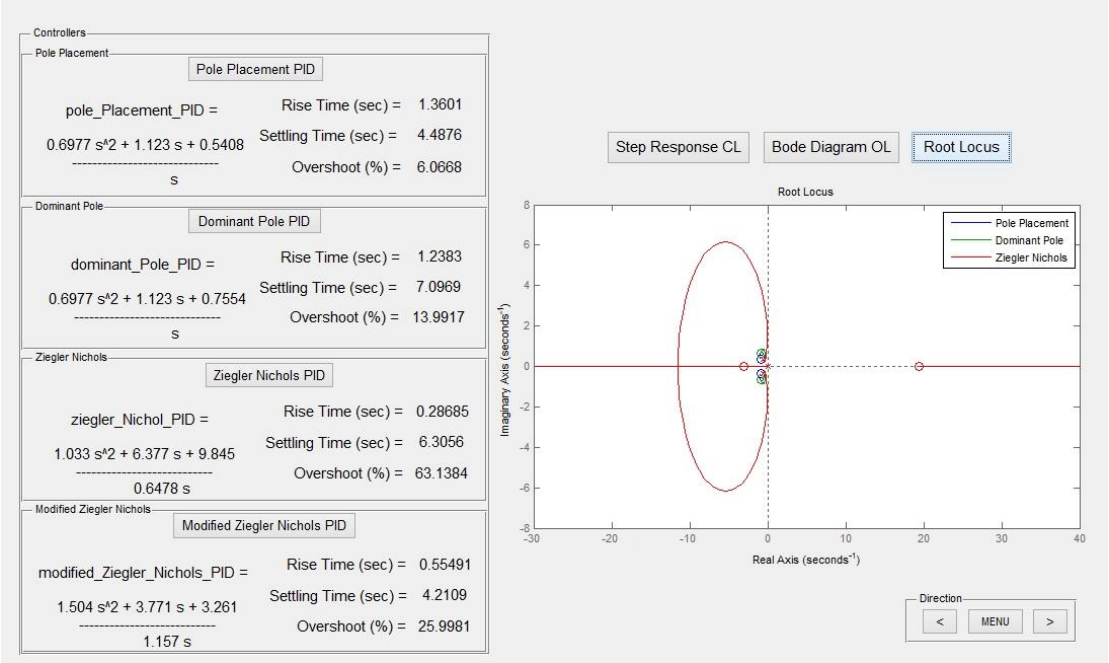

Figure 15 Controller Design with Root Locus

Step 5: Set Point Weighting

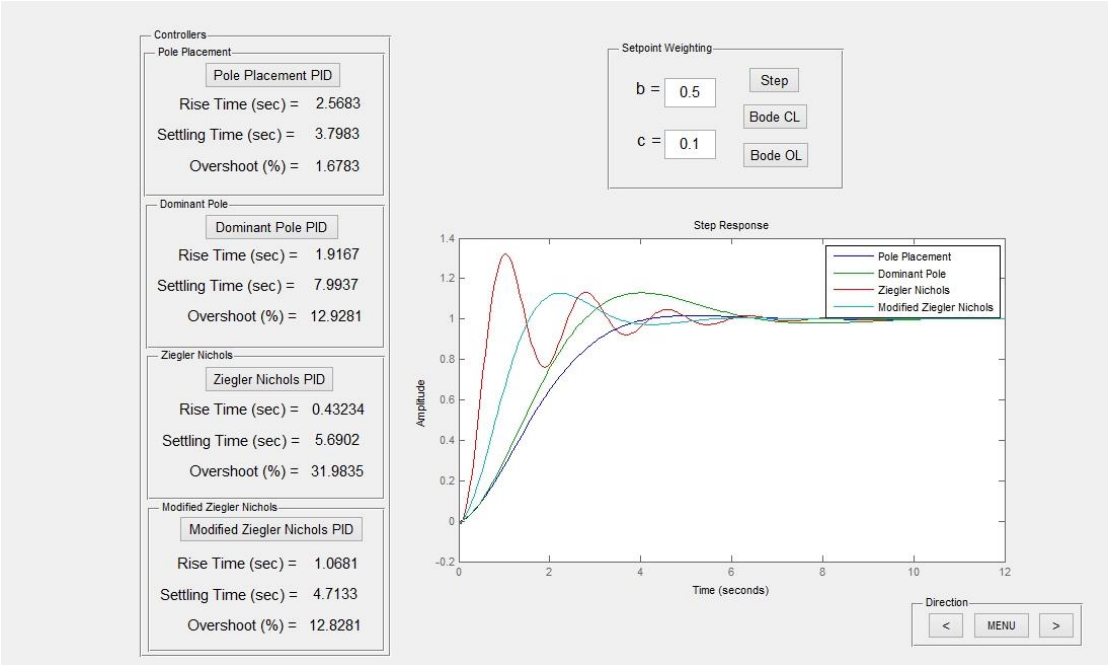

Figure 16 Set Point Weighting with Step Response

The closed-loop system is given by

$$G_l = G_{ff}\frac{P}{1+PC}$$

Where

$$G_{ff} = K\left(b + \frac{1}{sT_i} + csT_d\right)$$

$$C = K\left(1 + \frac{1}{sT_i} + sT_d\right)$$

The control signal is given by

$$u(t) = K(-(by_{sp} - y) + \frac{1}{T_i}\int_0^t (y_{sp} - y)ds + T_d\frac{\mathrm{d}(cy_{sp} - y)}{dt})$$

The b and c are affecting the gain and the derivative part of the controller.

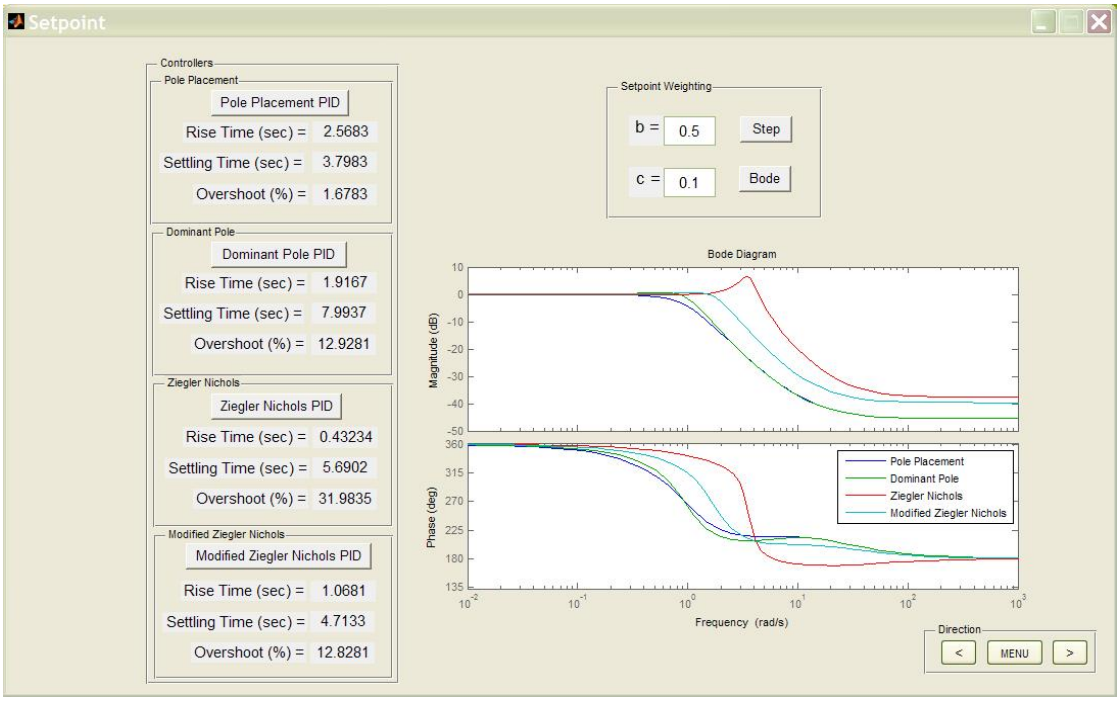

Figure 17 Set Point Weighting with Bode Diagram

A comparison on different value of b

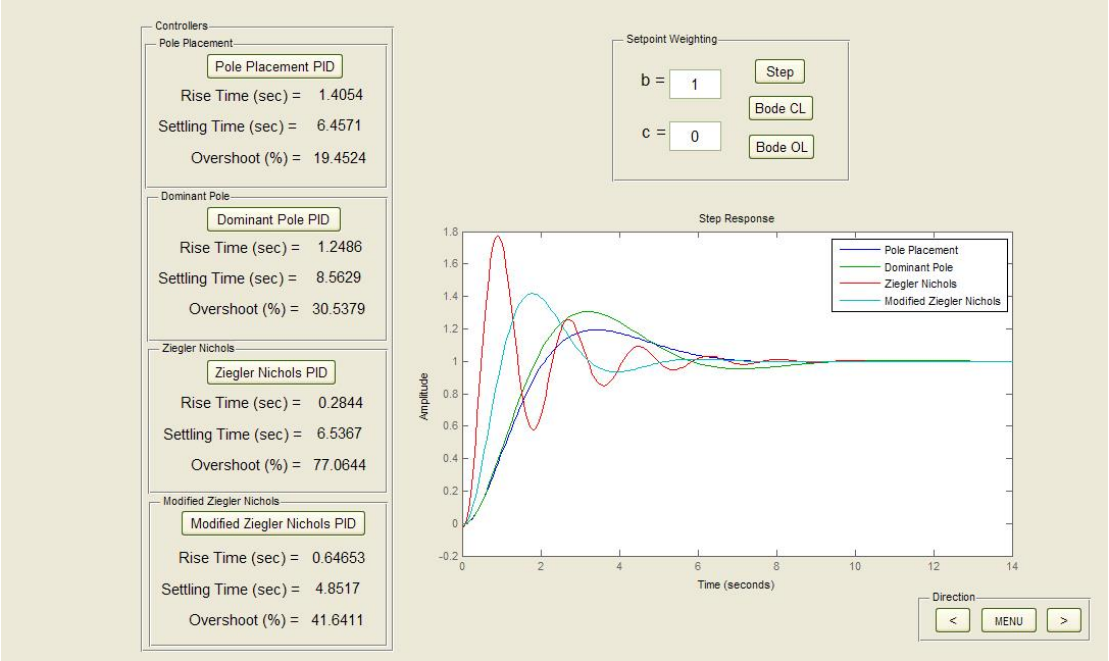

Figure 18 Set Point Weighting with Step Response with b=1 and c=0

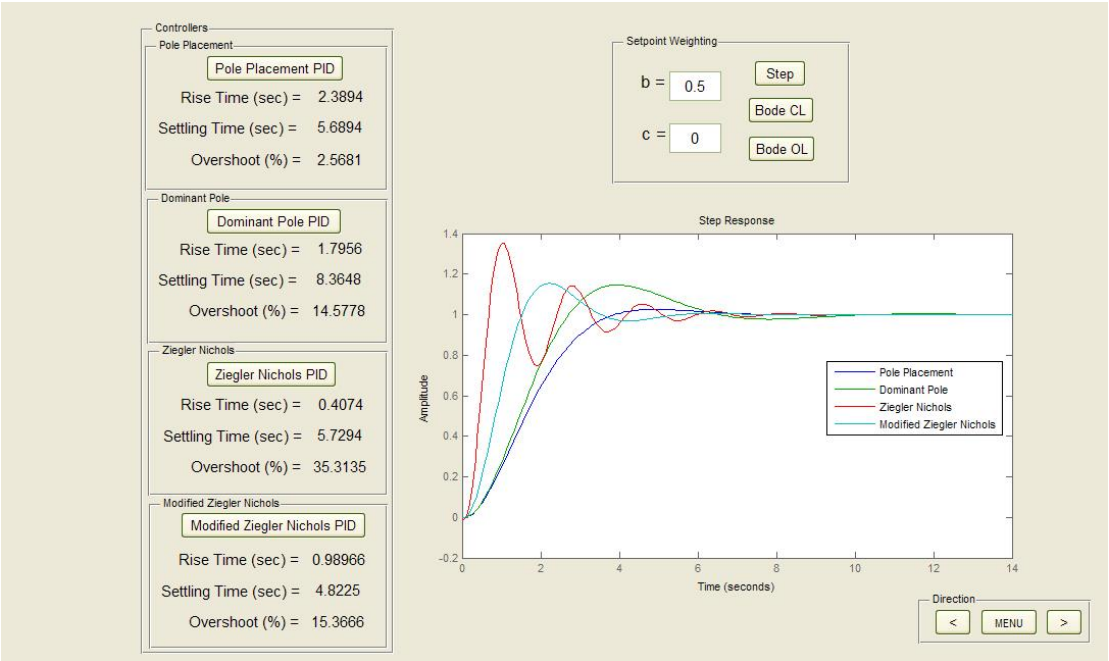

Figure 19 Set Point Weighting with Step Response with b=0.5 and c=0

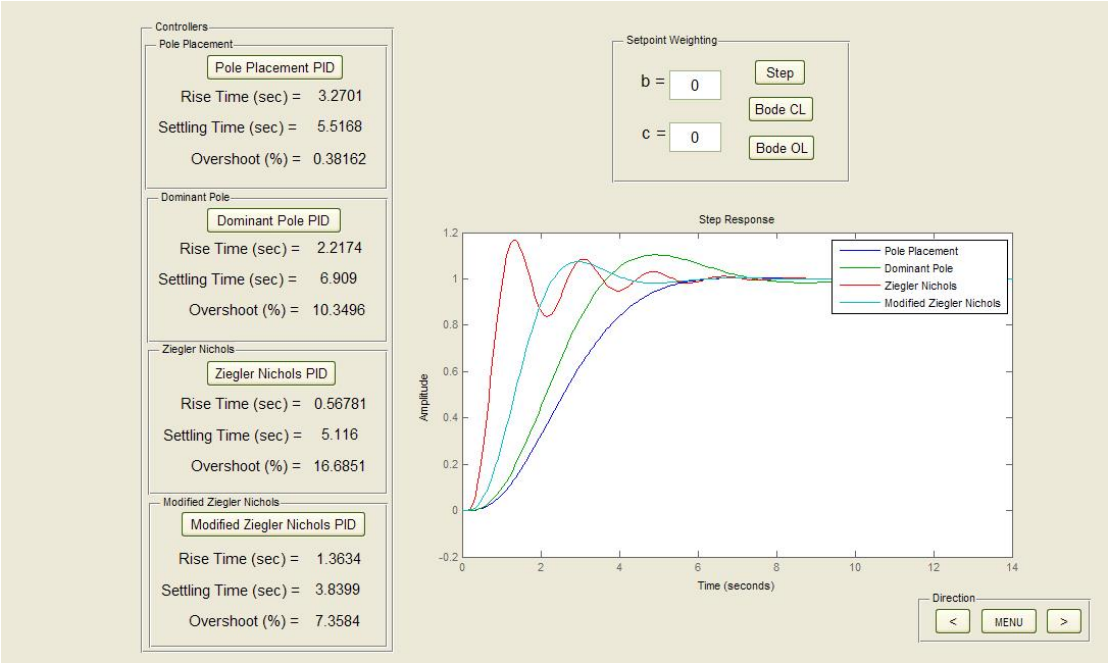

Figure 20 Set Point Weighting with Step Response with b=0 and c=0

A table statics can be obtained from above figures.

	Pole Placement	Dominant Pole	Ziegler Nichols	Modified Ziegler Nichols
b=1, c=0	19.45%	30.54%	77.06%	41.64%
b=0.5, c=0	2.57%	14.58%	35.31%	15.37%
b=0, c=0	0.38%	10.35%	16.69%	7.36%

Table 2 Table of Percentage of Overshoot with Different b

The table shows clearly about the effect to the percent of overshoot with the changing of b. The overshoot for the system is the smallest when b = 0, while the overshoot of the system is the largest when b =1. The percent of overshoot is increasing when the b in the proportional part is increasing.

The c is normally zero in order to avoid large transients when a sudden change happened in the setpoint.

Step 6: Filter

The following figures are the figures after adding the filter in the controller. Later on, there is another example to test how the filter work with a more zeros than a more poles situation.

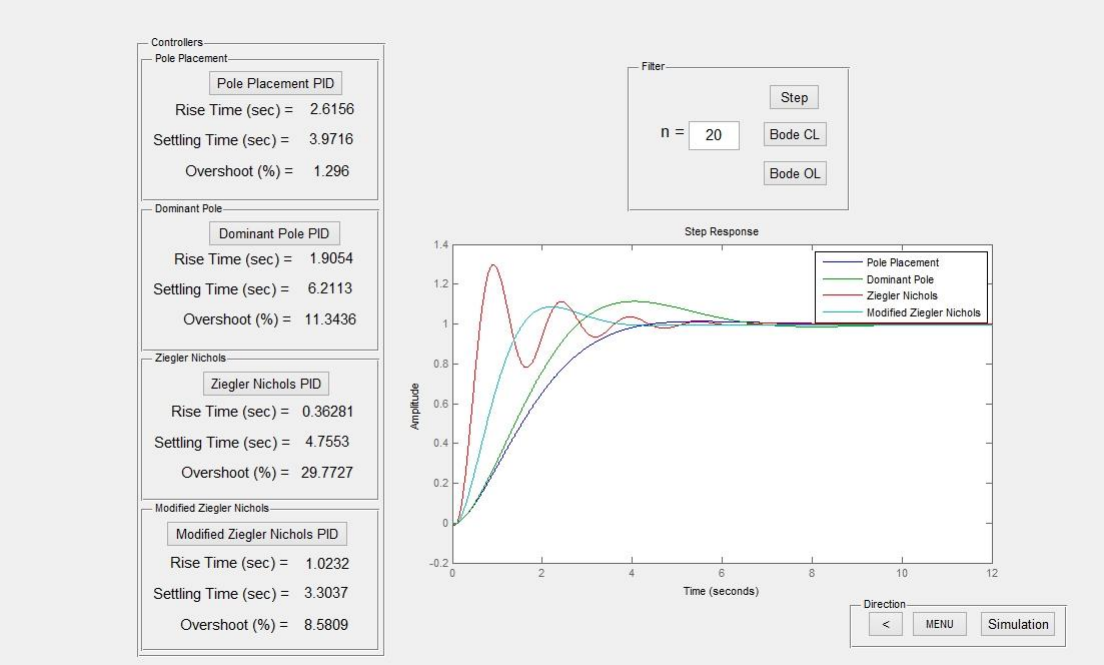

Figure 21 Filter with Step Response

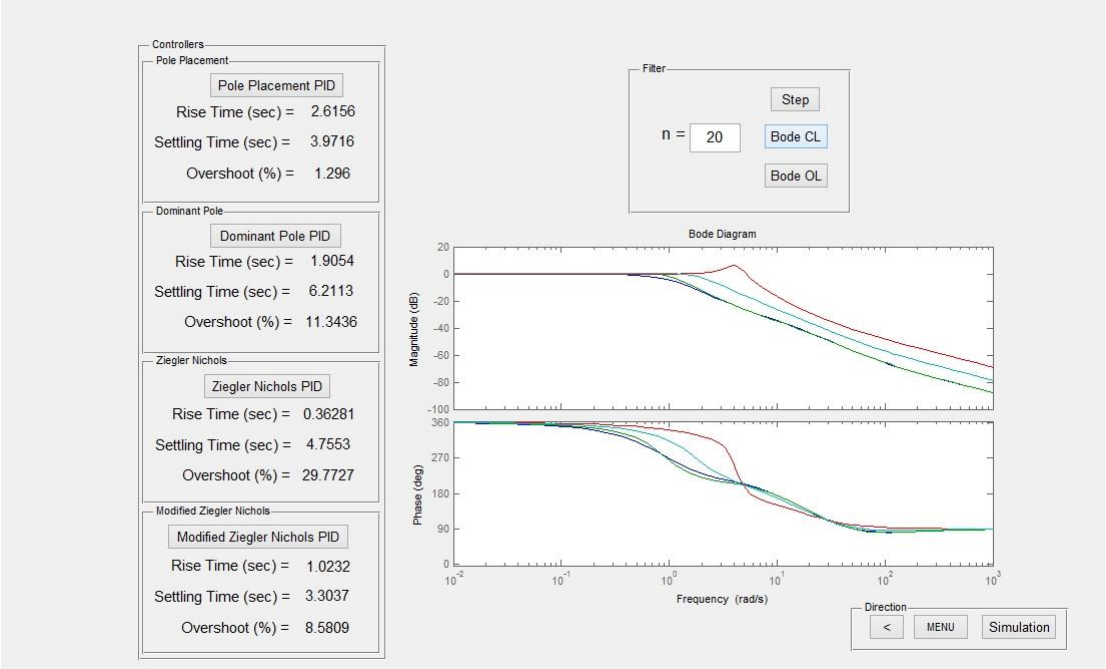

Figure 22 Filter with Closed-loop Bode Diagram

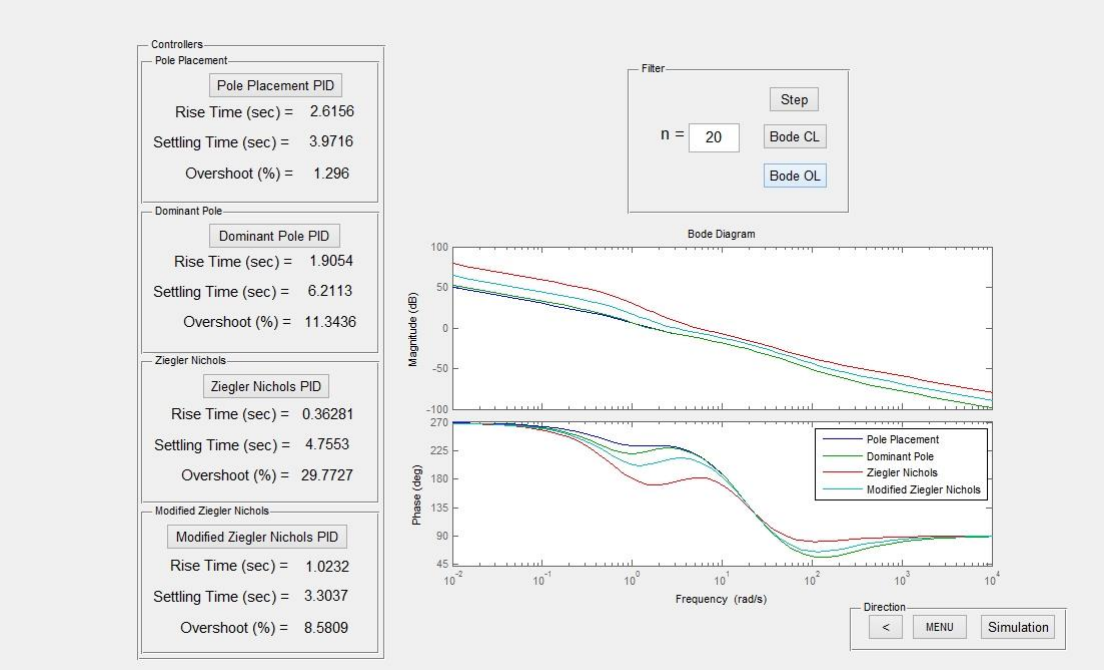

Figure 23 Filter with Open-loop Bode Diagram

Step 7 Simulation

The last step of the design process is the simulation, which is showed as following.

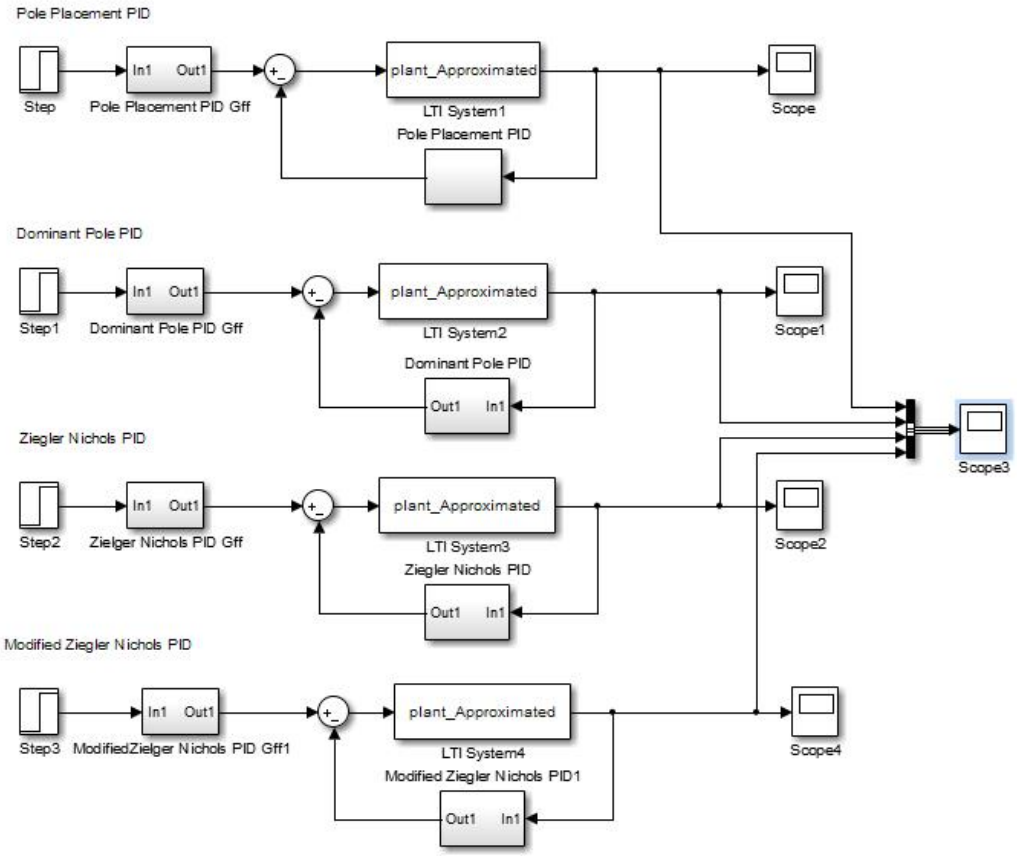

Figure 24 Simulation in example 2

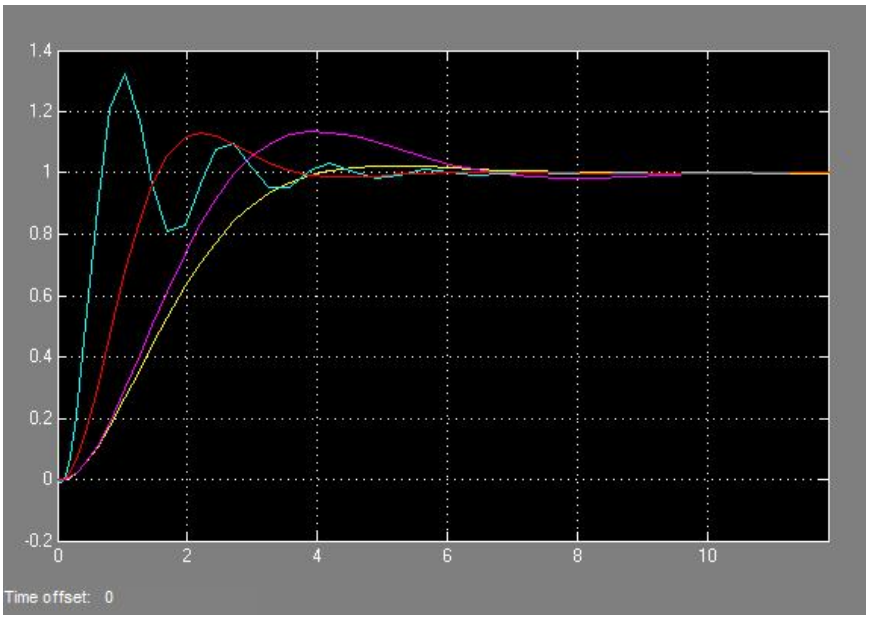

Figure 25 Simulation Result in example 2

Example 2

In order to test the effect of the filter, an example is given by:

$$P = \frac{3s^4 + 3s^3 + 6s^2 + 6s + 3}{s^4 + 3s^3 + 3s^2 + 3s + 1}$$

The steps in design platform is exactly the same as example 1. The step 2 figure is showed as following.

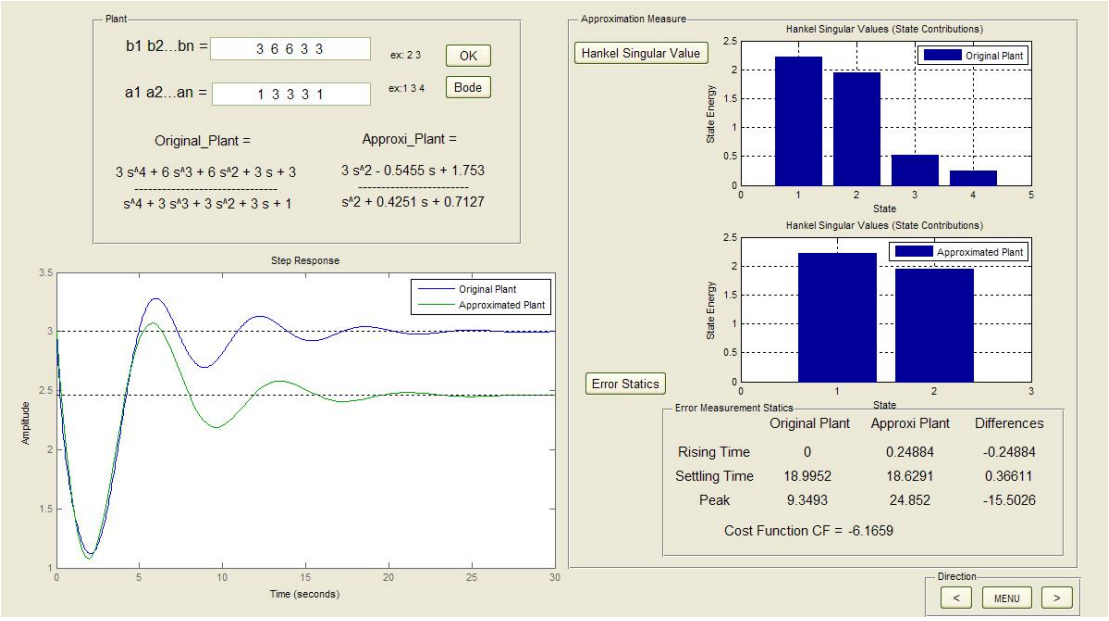

Figure 26 Plant Approximation in example 3

The step 5 setpoint weight after the design process is showed below.

The open loop bode diagram is

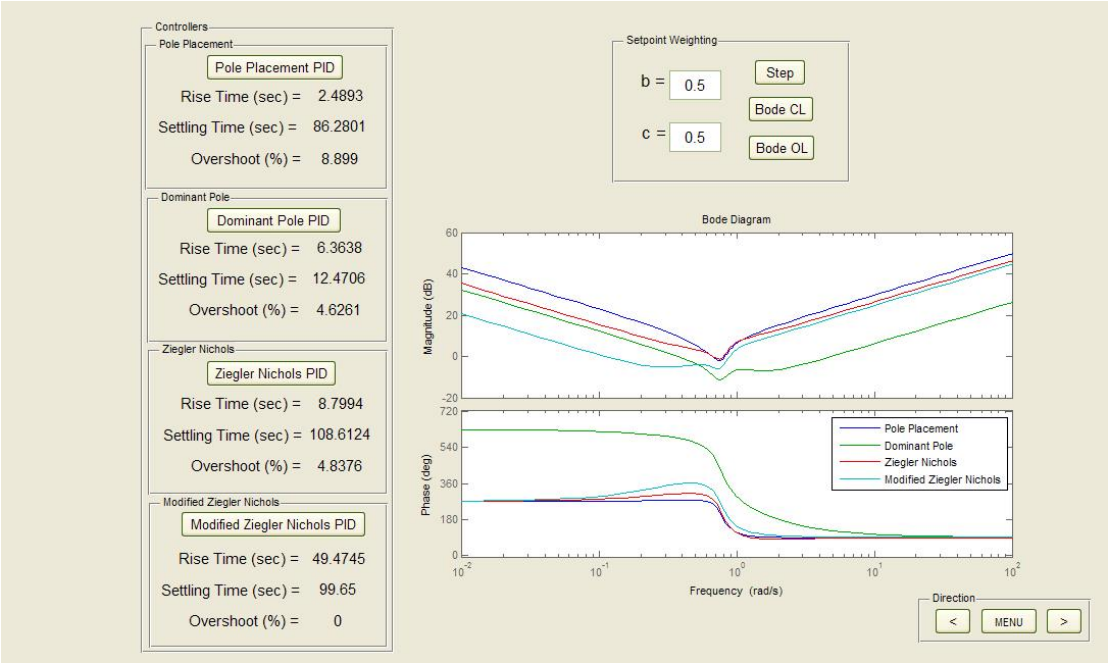

Figure 27 System Open-loop Bode Diagram without Filter

The slope of the open loop bode diagram is positive and increases in high frequency. That means when a high frequency noise is introduced in the system, the system will keep amplifying the high frequency. This can cause the devise damage or the saturation which makes the controller workless. Therefore, it is important to implement the derivative term as

$$D = \frac{sk_d}{(1 + \frac{s}{p_1})^2}$$

The poles added in the derivative term in the controller thus can limited the high frequency noise to avoid the difficulty.

The Bode Diagram of the open loop diagram with the controller in a filter is presented as below.

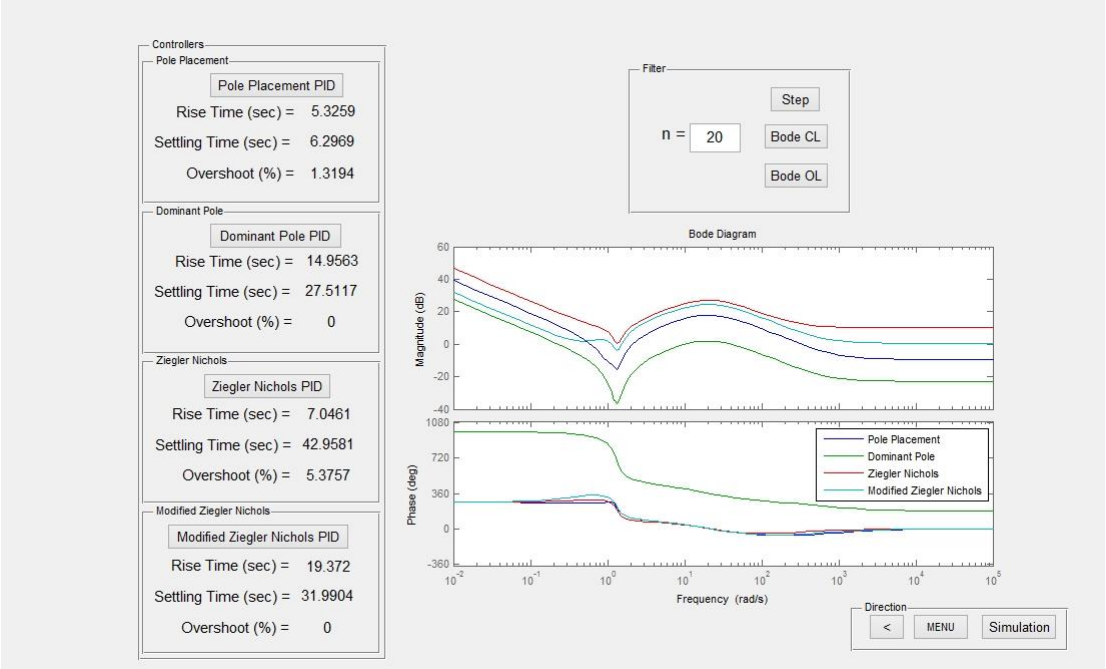

Figure 28 Open-loop Bode Diagram with Filter

The step response of the closed loop system with a PID controller with a filter is showed as following.

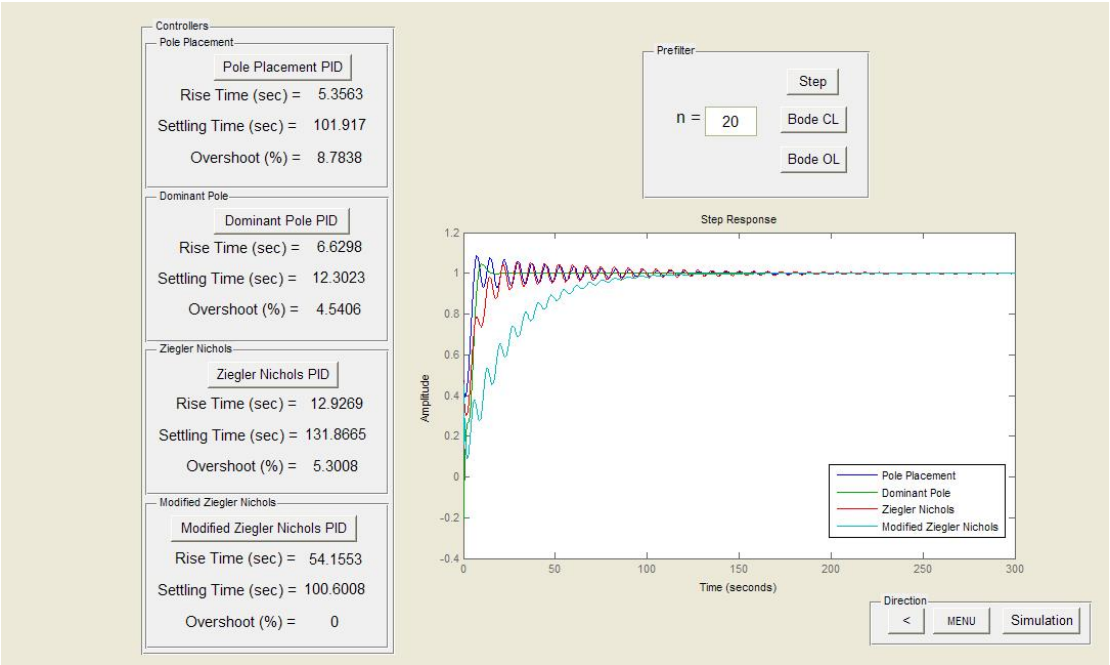

Figure 29 Step Response of the Closed-loop System with Filter

Compare the step response of the process in the platform and the result of the simulation as following.

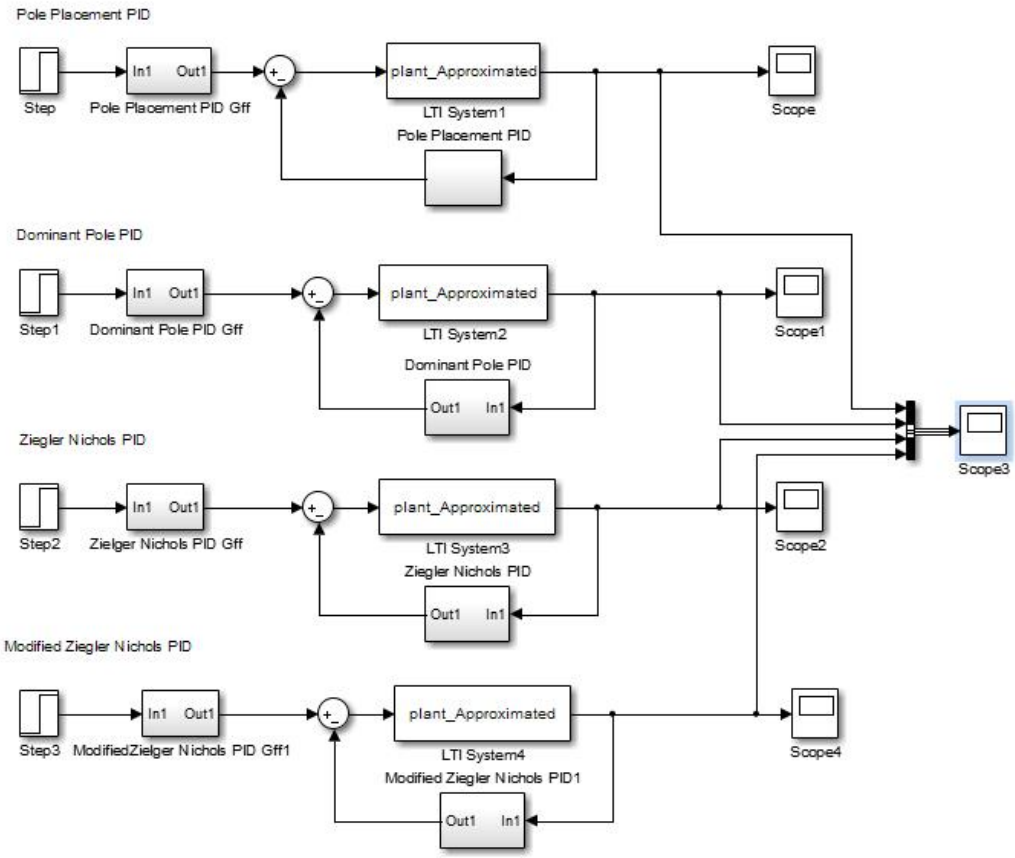

Figure 30 Simulation Structure

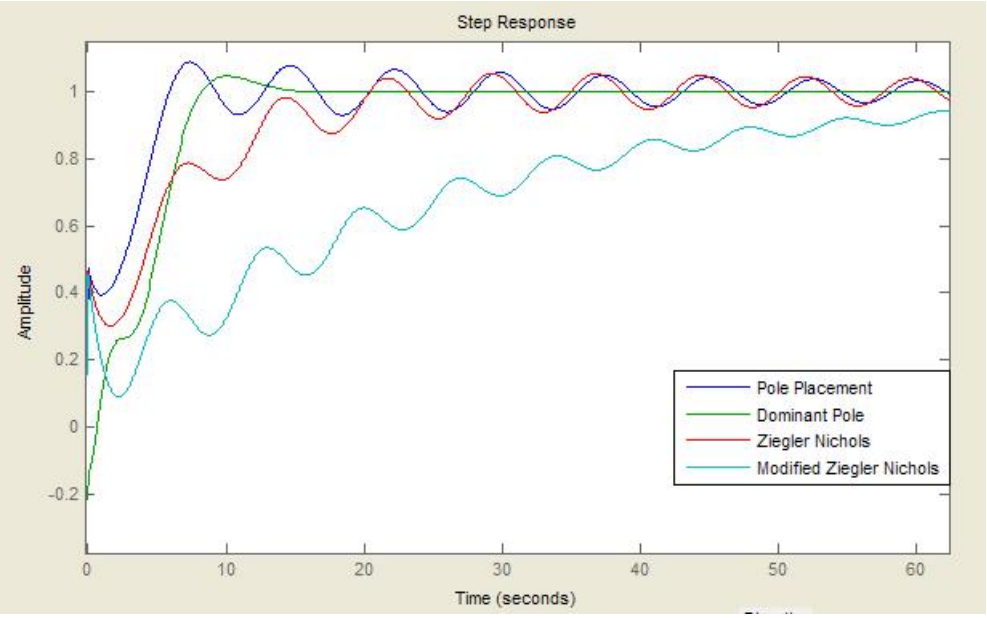

Figure 31 Step Response in the Platform

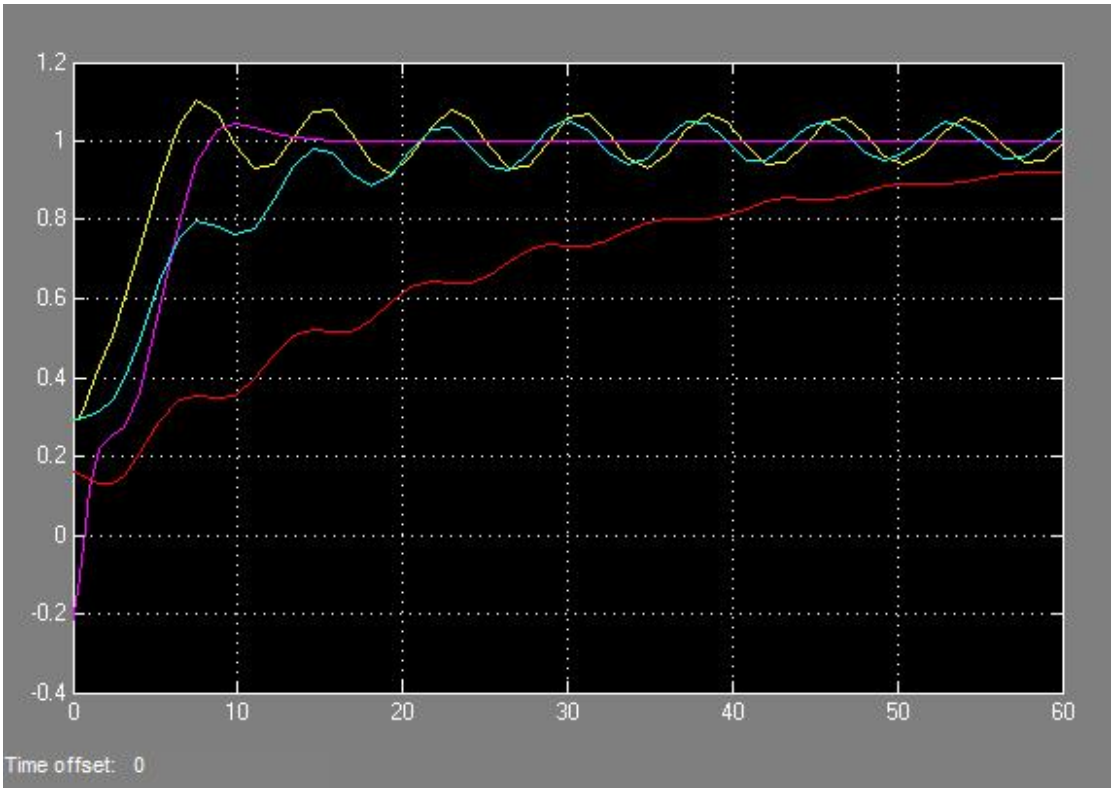

Figure 32 Test Result of the Simulation

The result of the platform and the simulation are the same.

Chapter 5 Summary and Future work

5.1 Summary

As mentioned in the abstract section, the primary objective of this project was to combining the controller tuning methodologies and the using soft computing methodologies to create an easy-to-use interface of PID controller design platform which designs the most widely used controller in industrial within four different tuning methods. In addition, designing a platform for control engineer is kept in mind when designing this Matlab platform. The majority of the project has been fulfilled in the task. The approximation step reduces the high order model to a second order plant model. This step has many benefit to the controller design process. Using a smaller order model can speed up the simulation process and reduce the control law complexity by keeping the major characteristic of the original high order plant.

The platform performs the designing of the four kind's different PID controller based on the information from the approximation model and the requested objective specification. The set point weighting and filter will be designed and added into the PID controller to adjust the controller in order to have a better system responses in terms of rise time, settling time, percent of overshoot, load disturbance and measurement noise. There is another way to test the PID controller which is the simulation. In addition, simulation is a way to add a devices or noise into the system. It is appropriate to have another method to test the PID controller in a different aspect.

5.2 Future work

There are several work can be considered to improve in the future.

1. Plant uncertainties. Plants are sometimes have uncertainties parameters. Add on the plant uncertainties will provide more information about the plant and design a better PID controller based on the plant uncertainties.
2. Design PIDs according to load disturbance or measurement noise. This will give a better controller design to meet the specification such as the load disturbance and the measurement noise.
3. A better measurement of the approximation or even a better error measurement method. One possibility is look at the controllability and the observability of the plant and figure out how much differences are the original plant and the approximated plant according to the controllability and observability.
4. Design which tuning method could work with the plant and which cannot at the beginning of the controller designing platform.
5. More different controller specifications. A good design method should take a number of different specifications into account in a balanced way.
6. Simulation with noise and disturbance. As said above, simulation is great to test the PID controllers. The users can update or add more noise or change to different devices as they want. It is also great to have the simulation initially with noise and disturbances.

7. How the set point weighting will affect the system when load disturbances and the measurement noise are introduced into the system.

References

Astrom, K. J., & Hagglund, T. (1995). *PID Controllers: Theory, Design, and Tuning.* New York: Instrument Society of America.

Blickley, G. J. (1990). Modern control strated with ziegler-nichols tuning. *Control Engineering*, 11-17.

Chien, K., Hrones, J. A., & Reswick, J. B. (1952). On the Automatic Control of Generalized Passive Systems. *Trans. ASME,*, 175-185.

Clarke, D. W. (1984). PID Algorithms and their Computer Implementation. *Trans. Inst. M. C.*, 305-316.

Dahlin, E. B. (1968). Desinging and Tuning Digital Controllers. *Instruments and Control Systems*, 77-83.

Gerry, J. P. (1987). A comparison of PID controller algorithms. *control Engineering*, 102-105.

Gilbert Polonyi, M. J. (1989). PID Controller Tuning Using Standard Form Optimization. *Control Engineering*, 102-106.

Miller, J. A., Lopez, A. M., Smith, C. L., & Murrill, P. W. (1967). A Comparison of controller tuning techniques. *Control Engineering*, 72-75.

Ziegler, J. G., & Nichols, N. B. (1942). Optimum settings for automatic controllers. *Trans. ASME*, 759-768.