Digital Control Systems Lab Practice

Practice 1: Digital PID Controller Design

Adriana Aguirre Alonso
María José Ramírez

2018

CONTENTS

OBJECTIVES

- Learn how to design digital PID controllers directly from a given analog controller.
- Learn how to design digital PID controllers completely in the discrete world using root locus method and SISOTOOL from Matlab ™.
- Design digital PID controllers for the Speed Control System using the methods in this document in order to accomplish, if possible, all of the design requirements.
- Compare the performances of the digital PID controllers designed with different sampling times.
- Compare and contrast the performances of the digital PID controllers designed with two different methods.
- Test all of the digital PID controllers in the Speed Control System and verify if the design requirements are met.

Introduction

Proportional Integral Derivative (PID) controllers are the most used in the control industry. In this practice different digital PID controllers will be designed using SISOTOOL, which is a tool of Matlab^{TM}. The designed controllers will be simulated and tested using SIMULINK, another tool from Matlab ^{TM}.

There are different methods used to design digital controllers but in this case we will use two methods.The first method consists in designing an analog PID controller and then using those parameters to recalculate digital PID parameters. The second method consists in designing the digital PID controller completely in the discrete world using root locus method.
Figure 1.1 shows the architecture for a closed loop control system.The block F represents the filter,

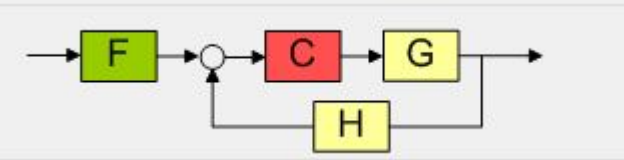

Figure 1.1: System Control Architecture

C is the controller, G is the transfer function of the system and H is the sensor transfer function or gain.Commonly, closed loop control systems have negative feedback.

Three sample times will be used; thus,three different controllers for each method must be designed and tested. The sample times will be selected in order to obtain for the first method the following scenarios: a very similar performance, slight variations in performance and very different performance of the system with digital PID controller compared to the system with analog PID controller. In order to compare both methods, use the same sampling times for the second method.

Before starting to design digital PID controllers for a system, it is assumed that the transfer function of that system is known, and in the case of the first method, that the analog PID controller is also known. You should also know which are the design requirements for the controller.

Theory

The follow equation represent the analog PID in the time domain:

$$m(t) = K[e(t) + \frac{1}{T_i}\int_0^t e(t)dt + T_d\frac{de(t)}{dt}] \tag{2.1}$$

where $e(t)$ is the error of the system and $m(t)$ is the output of the analog PID controller. The Laplace representation of this controller results in the following transfer function:

$$\frac{M(s)}{E(S)} = C_i(s) = \frac{KT_ds^2 + Ks + \frac{K}{T_i}}{s} \tag{2.2}$$

This representation is called ideal PID Controller. Another form of the PID Controller is the parallel representation that has the following transfer function:

$$C_p(s) = \frac{K_ds^2 + K_ps + K_i}{s} \tag{2.3}$$

You can relate the parameters of these controllers from the following Equations:

$$K = K_p; T_i = \frac{K}{K_i}; T_d = K_d/K; \tag{2.4}$$

K_p, K_i and K_d are the analog PID parameters in the parallel form and K, T_i and T_d are the analog PID parameters in the ideal form.

The approximations used to obtain the discrete representation of the analog PID controller are:

$$\int_0^t e(t)dt \approx \sum_{h=1}^{k} e_hT \tag{2.5}$$

$$\frac{de(t)}{dt} \approx \frac{e_k - e_{k-1}}{T} \rightarrow \frac{1}{T}(1 - z^{-1})e(z) \tag{2.6}$$

where T is the sampling time.

Then, the digital PID controller has the following equation:

$$m(kT) = K[e_k + \frac{T}{Ti}\sum_{h=1}^{k} e_h + \frac{T}{Ti}(e_k - e_{k-1})] \tag{2.7}$$

$$M(z) = K[1 + \frac{T}{Ti}\frac{1}{1-z^{-}1} + \frac{Td}{T}1 - z^{-}1]E(z) \tag{2.8}$$

The coefficients of Equation 2.13 can be related with the analog parameters in the ideal form of Equation2.2. The following equations are obtained:

$$K_{pD} = K \tag{2.9}$$

$$K_{iD} = \frac{KT}{T_i} \tag{2.10}$$

$$K_{dD} = \frac{KT_d}{T} \tag{2.11}$$

Then, the digital PID controller has the following transfer function:

$$\frac{M(z)}{E(z)} = G_D(z) = \frac{K_{pD}z^2 - K_{pD}z + K_{iD}z^2 + K_{dD}z^2 - 2K_{dD}z + K_{dD}}{(z-1)z} \tag{2.12}$$

$$G_D(z) = \frac{(K_{pD} + K_{iD} + K_{dD})z^2 - (K_{pD} + 2K_{dD})z + K_{dD}}{(z-1)z} \tag{2.13}$$

Procedure

3.1 First Method: Analog to Digital Design

1. Obtain the analog PID parameters from the analog PID controller. Use as reference Equations 2.3 and 2.4.
2. Using Equations 2.9, 2.10 and 2.11, find the digital PID parameters for each sampling time.
3. Using Equation 2.13, find the transfer function of the digital PID controller for each sampling time.
4. Simulate each digital PID controller and plot the system output for each controller in one figure.
5. Test each digital PID controller on the real system and plot the system output for each controller in one figure.
6. Compare the results obtained after the simulation and implementation.

3.1.1 Example

A system has the following open loop transfer function:

$$G(s) = \frac{11.81}{(s+3.449)(s+3.855)} \tag{3.1}$$

Assume that the analog PID controller for that system is:

$$C(s) = \frac{0.08329s^2 + 0.6076s + 1.105}{s} \tag{3.2}$$

Then, the corresponding analog PID parameters are the following:

- $K_p = 0.6076$
- $K_i = 1.105$

- $K_d = 0.08329$

Now, using Equation 2.4 we obtain the following parameters:

- $K = 0.6076$
- $T_i = 0.5499$
- $T_d = 0.1371$

Then, using Equations 2.9, 2.10 and 2.11 we obtain the digital PID parameters. For this example, we will use a sampling time of 0.0125 [s].

	T= 0.0125 [s]
K_{pD}	0.6076
K_{iD}	0.0138
K_{dD}	6.6632

Table 3.1: Method 1: Digital PID parameters

In Simulink create the structure shown in Figure 3.1 to simulate the system with the designed digital PID controller and the given analog PID controller; obtain the system output for each case. The orange blocks represent the system open loop transfer function $G(s)$, the green block is the analog PID controller $C(s)$, and the light blue blocks represent the digital PID controller $G_D(z)$.

Notice that the zero order hold (ZOH) blocks are used to hold the value of the signal during a sample time because real physical systems need to have continuous input data. Also in this case, we will be considering the system acquisition card delay, represented by 'DAQ delay' block; the delay will be equal to one sampling period.

It is important to double click the 'To Workspace' block to change the save format parameters to 'array' in order to use the simulated data easily.

Plot the time response of the system with the digital PID controller and with the analog PID controller in one figure and compare the results.

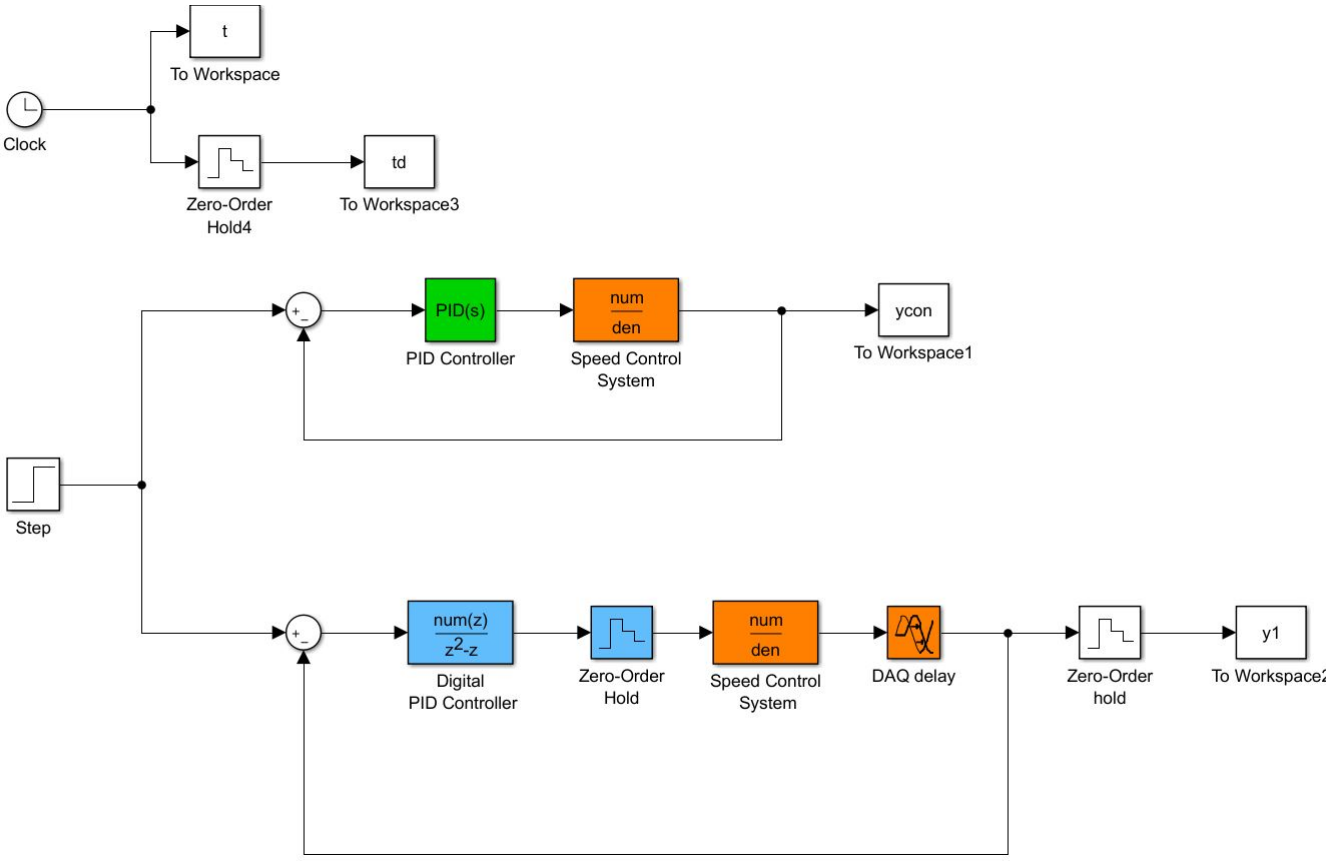

Figure 3.1: Simulink Block Diagram

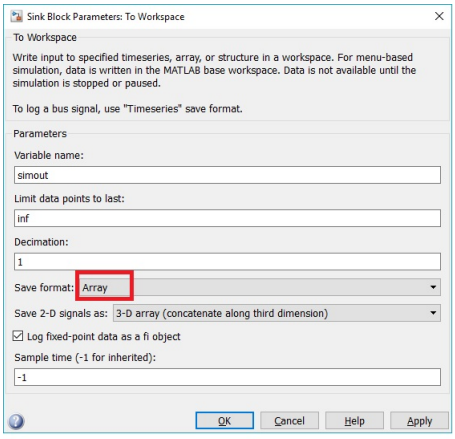

Figure 3.2: 'To Workspace' Configuration

3.2 Second Method: Digital Redesign

1. Obtain the discrete transfer function of the system considering the ZOH and the sampling time. Different sampling times result in different discrete transfer functions of the same system.
2. Consider any additional delays or gains if necessary.
3. Invoke 'sisotool' to design the controller. You should have knowledge of the design requirements before doing the design.
4. In 'sisotool', plot the corresponding contours of the design requirements.
5. Add the known pole or poles to the controller.
6. Add a zero to the controller and adjust its position so that there exists root locus in the intercepting points of the design requirements contours. Depending of the type of controller used(PI,PD,PID) and if needed, add another zero to accomplish this task.
7. Adjust the gain of the controller, so that the closed loop poles are located in the intercepting points of the design requirement contours or that the design requirements are met.
8. Simulate each digital PID controller and plot the system output for each controller in one figure.
9. Test each digital PID controller on the real system and plot the system output for each controller in one figure.
10. Compare the results obtained after the simulation and implementation.

3.2.1 Example

Following the example in Section 3.1.1, using the second method of this practice design a digital PID controller so that the system response to a step input meets the following requirements:

- Steady state error: 0
- 0% Overshoot
- Setting time:4 [s]

The **c2d** function converts a continuous transfer function ('s' domain) to a discrete transfer function ('z' domain). Use the following code:

$$Gz = c2d(G,T,'zoh')$$

where G is the continuous transfer function of the system, T is the sampling time (ie. 0.0125 [s]) and 'zoh' represents the method used for the continuous to discrete transformation.

For designing purposes, we will be considering the acquisition card delay into the discrete transfer function, so:

$$Gdly = tf(1,[1,0],T)$$
$$Gz = series(Gz,Gdly)$$

To start the design of the controller, invoke sisotool:

$$sisotool(Gz)$$

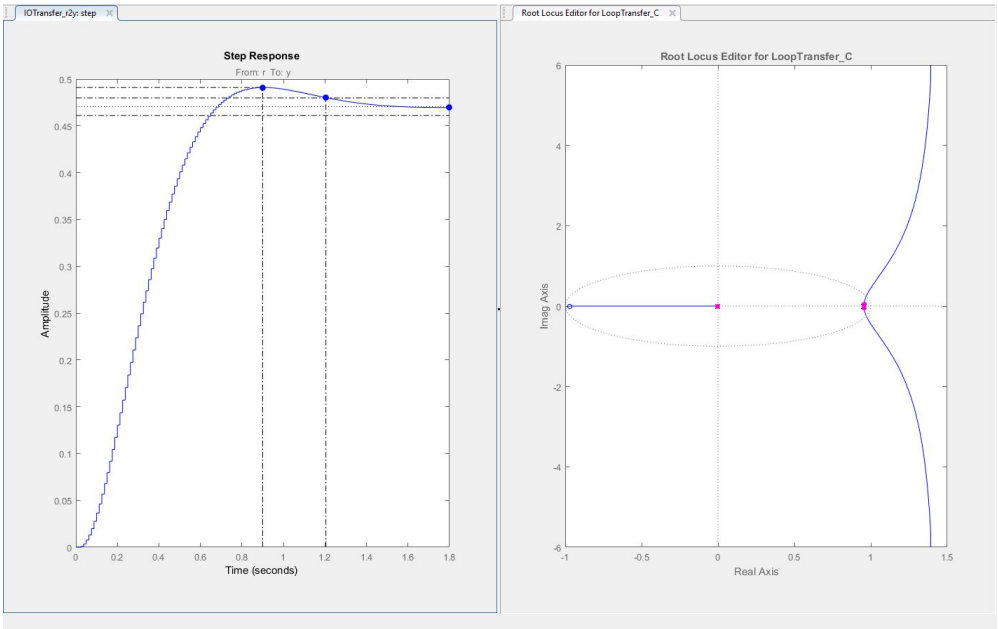

Figure 3.3: Closed Loop Step Response and Root Locus

Figure 3.3 shows the Closed Loop Step Response and Root Locus of the system. The next step is to draw the contours corresponding to each design requirement. The contour for a specific settling time is a circle with radius $e^{-\sigma T}$. For this example, the settling time contour is shown in Figure 3.4.The overshoot contour corresponds to an ellipse-like plot. In this case, the condition of 0% overshoot is satisfied when the dominant closed loop poles are positive, real and less than 1. If this design requirement is also set in sisotool, the contours in Figure 3.5 are obtained.

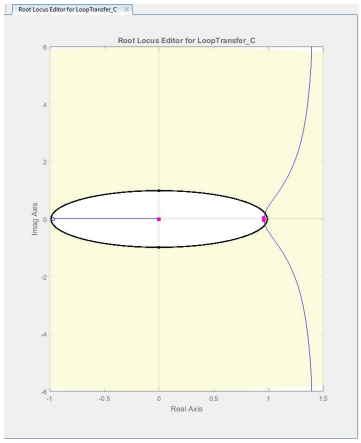

Figure 3.4: Settling Time Requirement Contour

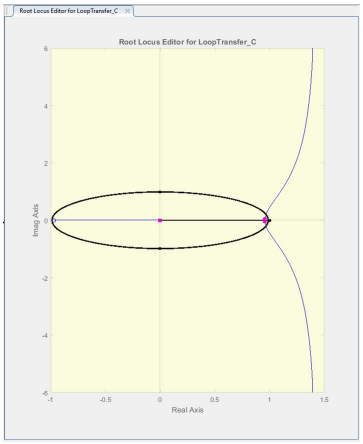

Figure 3.5: Overshoot and Settling Time Requirements Contours

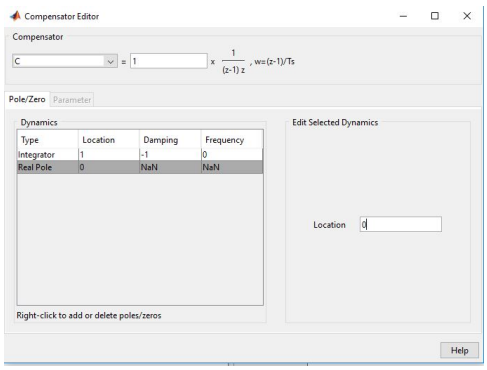

Figure 3.6: Digital PID known poles

Because a digital PID has a pole in zero and a pole in one, add these poles directly to the controller (C) as seen on Figure 3.6.

For this example, the location of the two zeros of the digital PID controller was chosen to be the same location of two of the poles of the discrete system; this technique is used to cancel the effect of unwanted poles. In this case, the zeros were located in 0.9578 and 0.953 as seen in Figure 3.7. Then, the gain of the controller is adjusted so that the dominant pole is located in our zone of interest.

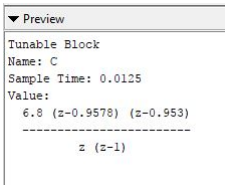

Figure 3.7: Digital PID Controller

The parameters of the digital PID controller are obtained using Equation 2.13 and the resulting controller shown in Figure 3.7. The resulting parameters for this example are summarized in Table 3.2.

	T= 0.0125 [s]
K_{pD}	0.5794
K_{iD}	0.0136
K_{dD}	6.207

Table 3.2: Method 2: Digital PID Parameters

The final root locus and closed loop step response are shown in Figures 3.8 and 3.9 respectively.

Use the block diagram shown on Figure 3.1 to simulate the system with the redesigned digital PID controller and the given analog PID controller; obtain the system output for each case.

Plot the time response of the system with the digital PID controller and with the analog PID controller in one figure and compare the results.

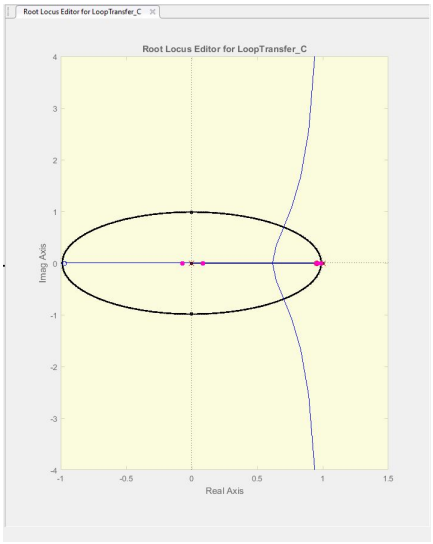

Figure 3.8: Root Locus updated

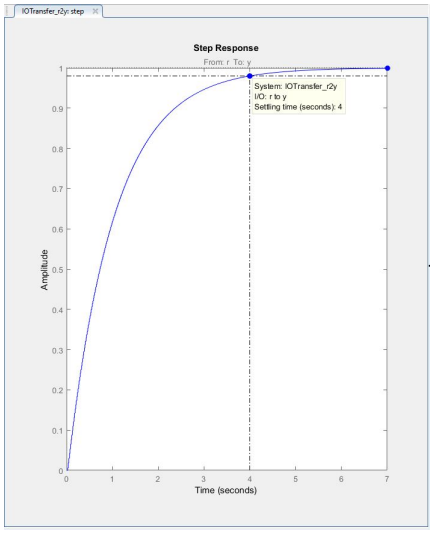

Figure 3.9: Step Response updated

Practice

The Speed Control System transfer function is:

$$G(s) = \frac{11.81}{(S+3.449)(S+3.855)} \tag{4.1}$$

The Design Requirements for this system are:

- Steady state error: 0
- 0% Overshoot
- Setting time: 4[s]

The analog PID controller for the Speed Control System that accomplish the mentioned design requirements is:

$$C(s) = \frac{0.08329s^2 + 0.6076s + 1.105}{s} \tag{4.2}$$

Using both methods explained on the practice, design digital PID controllers for T=0.125 [s] and T=0.75 [s] that accomplish, if possible,the design requirements above mentioned. Give priority to the steady state error, then the overshoot and finally the settling time.

For the first method, fill in table 4.1, in one figure show the simulated system output for each sampling time and on another figure show the real system output for each sampling time.

	T= 0.0125 [s]	T=0.1250 [s]	T=0.75 [s]
K_{pD}	0.6076		
K_{iD}	0.0138		
K_{dD}	6.6632		

Table 4.1: Method 1: PID Parameters

For each sampling time for the second method, show the root locus for each controller, fill in table 4.2, in one figure show the simulated system output for each sampling time and on another figure show the real system output for each sampling time.

	T= 0.0125 [s]	T=0.1250 [s]	T=0.75 [s]
K_{pD}	0.5794		
K_{iD}	0.0136		
K_{dD}	6.207		

Table 4.2: Method 2: PID Parameters

Compare the results for different sampling times in each method and then compare the results obtained with each method. Comment about the differences between the simulated and real output, about the performance of each controller and about whether the design requirements were met or not.