# **Lab #4: Instructions**

# **PID Controllers**

## **Introduction**

There are several different classifications for controllers, based upon the methodology used to generate the signal sent to the actuator circuit that generates a signal that causes the system to adjust its operating point. These include: Proportional (P), Proportional-Integral (PI), Proportional-Derivative (PD), and Proportional-Integral-Derivative (PID). The selection of the type of controller to use in a particular system requires an understanding of how large the error signal is likely to be, how fast the system needs to be readjusted back to the desired operating point when there are any deviations, and whether the system can be driven into an unstable operating point.

PID controllers have been adopted extensively for a wide range of applications. Since the gain, Kp, Ki, and Kd, of proportional, integral, and derivative sections of the controller circuit (shown in Figure 1), the response time required to adjust the system can be optimized while maintaining stable performance within the desired range of operating points.

Block diagram of a PID controller.

*Figure 1: Block diagram of a PID controller.* ***e*** *is the error signal;* ***P*** *is the proportional term;* ***I*** *is the integral term;* ***D*** *is the derivative term.*

The implementation of a PID controller can be done in several different ways using operational amplifiers. The simplest implementation from a conceptual point of view is to design three op amp circuits, one where there is linear gain (i.e., an inverting amplifier) for the proportional controller P, an integrating circuit for the integral controller I, and a differentiator circuit for the derivative controller D. The outputs from each of the individual controllers would be added using a summing amplifier circuit (Fig. 2).

The transfer function of the P+I+D controller and summing stage is:

Example of PID circuit with a summing amplifier at the output.

*Figure 2: Example of PID circuit with a summing amplifier at the output.*

It can be seen that the voltage transfer characteristic can be separated out into parts where

such that:

Several criteria are used when designing a PID controller to obtain a desired response from the control system. The specifications include the peak magnitude of the overshoot, settling time, steady-state error, rise time, and stability of feedback system. The gains for the proportion, integral, and derivative sections of the controller, which is done by “tuning” the control loop, determines how well the controller meets the design criteria. There are several techniques that can be used to tune a PID control when the response of the system to be controlled is difficult to model: guess-and-check, the Ziegler-Nichols method, Cohen-Coon method, and AMIGO tuning rules. The table below shows the effect of manually changing the parameters for step input when the other two are fixed.

*Table1: Effect of increasing a parameter independently for a step input*

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Parameter | Rise Time | Overshoot | Settling Time | Stead-state error | Stability |
| Kp | Decrease | Increase | Small Change | Decrease | Degrade |
| Ki | Decrease | Increase | Increase | Eliminate | Degrade |
| Kd | Minor Change | Decrease | Decrease | No Effect in theory | Improve if Kd small |

**Comments/Advices:**

**- For Bode, Nyquist and Root locus analysis use the OLTF**

**- For Time domain response ‘lsim’ and ‘stepplot’ use CLTF**

**- Remember that OLTF includes the controller, the plant and the feedback transfer functions to assess the characteristic equation in CLTF and therefore its stability**

## **Exercise 1**

Consider the system:

Closed loop system with G(s) the system to control and H(s) the control transfer function.

with:

and

|  |  |
| --- | --- |
| OLTF |  |
| CLTF |  |

1. Sketch the root locus for the system OLTF and insert the annotated plot here indicating the essential characteristics of a root locus (poles/zeros and their values, breakaway point if any, complex axis crossing,..)
2. Draw the Bode magnitude and phase plots for the system OLTF
   1. Enter the command ‘figure’
   2. Use the function bode(G) to plot the log-magnitude and phase plots
   3. Right click in the graph, go down to the menu ‘characteristics’ and then select ‘All stability margins’
   4. Use a datatip to read the values
   5. Insert your annotate Bode plot here
3. Also obtain the Nyquist plot for the same system OLTF
   1. Enter the command ‘figure’
   2. Use the function nyquist(G) to obtain the Nyquist plot
   3. You may notice that the plot has 2 arrows and in the lectures we’ve seen that it only goes from 0 to infinity so how come?
   4. Well Matlab considers negative frequencies
   5. If you right click in the graph, go down to the menu ‘Show’ and untick ‘Negative frequencies’
   6. Right click in the graph, go down to the menu ‘Characteristics and then select ‘All stability margins’
   7. Use a datatip to read the values
   8. Alternatively you can use the Matlab command window *allmargin(TF)*

|  |
| --- |
| GainMargin =  GMFrequency =  PhaseMargin=  PMFrequency=  DelayMargin=  DMFrequency=  Stable/Unstable |

## **Exercise 2**

* Open pidTuner
* Load the open loop transfer function of the plant
* Looking at the step plot of the reference tracking
* Design a step response with an overshoot lower than 5% and a settling time lower than 10s with a PID controller.

|  |  |
| --- | --- |
| Kp |  |
| Ki |  |
| Kd |  |
| Rise Time |  |
| Settling time |  |
| Overshoot |  |
| Peak time |  |
| Gain Margin |  |
| Phase Margin |  |
| Closed-loop stability |  |

Insert your annotated step response in the report here.

* Modify your script to include the transfer function of the PID controller for the closed-loop system.
  + Plot the response of your system to a step input to validate that your model is accurate and matches what you found with pidTuner with stepplot.

|  |  |
| --- | --- |
| Kp |  |
| Ki |  |
| Kd |  |
| Rise Time |  |
| Settling time |  |
| Overshoot |  |
| Peak time |  |
| Gain Margin |  |
| Phase Margin |  |
| Closed-loop stability |  |

* Define a time vector *t* with appropriate sampling.

|  |
| --- |
| t = |

* Define a sinewave using a frequency of your choice u = sin(frequency\*t)

|  |
| --- |
| u = |

* Plot the response of your system to the sinewave you designed using lsim (transfer function, u, t)
* Measure the delay between the input and the output, calculate the corresponding phase

|  |  |  |
| --- | --- | --- |
| Delay |  |  |
| Phase |  |  |

* How do the delay and phase between input and output link with system instability (time to look in the theory in your course notes)

|  |
| --- |
|  |

* Plot the Bode plot of this new system
* Plot the Nyquist plot of this new system
* Determine all the margins

|  |
| --- |
| GainMargin =  GMFrequency =  PhaseMargin=  PMFrequency=  DelayMargin=  DMFrequency=  Stable/Unstable? |

* Redefining your PID controller using PID = tf(num,den,'InputDelay',Delay) and an appropriate frequency make the system unstable

|  |
| --- |
|  |

* Plot the following figures and insert them in your report with appropriate annotations
  + Plot the response of your system to the sinewave you designed using lsim(transfer function, u, t)
  + Plot the bode plot of this new system
  + Plot the nyquist plot of this new system
* State the full theory of the Nyquist criterion

|  |
| --- |
|  |

* Show that this system is unstable by linking theory to what you see
  + Hint: pzmap(pade(OLTF)), pole(pade(OLTF)), zero(pade(OLTF))

|  |
| --- |
|  |

**\*\* Please review the supplementary document ‘lab submission guidelines.pdf’ for instruction on final report submission.**