B.Eng. Electrical Engineering
School of Energy

Lab #3 PID Control of the Servo Motor using Time Domain Performance Specifications

Objectives

The objectives of this experiment are:

- to design a P(I)D controller to meet specific time domain performance objectives for a linear 2nd order system
- to test out the controller design on a using the nonlinear, dominant 2nd order model of the real servomotor system
- to implement the controller on the real servomotor system
- to investigate the differences in performance between the model and the real system

You will be designing and implementing the controller for the servomotor that you have worked with in Lab #2. This lab requires 2 lab sessions to complete (6 hours). Although the instructions are not lengthy, please be aware that the lab will take the full 6 hours to complete, and is best done in the lab with the instructor present to answer questions. Consult the Lab3Marking.pdf for full details of the deliverables.

Background

Consider the following PID controller structure, implemented as shown in Figure 1 below:

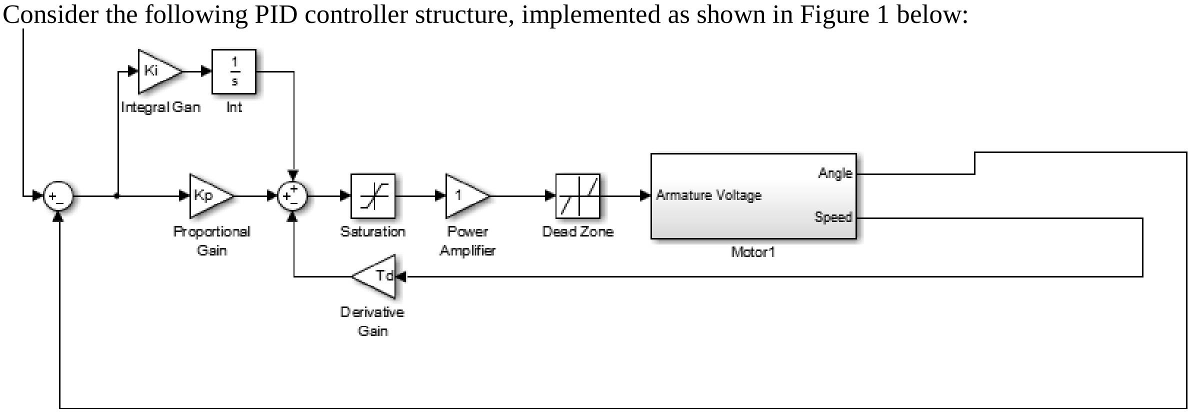


Figure 1 SIMULINK Configuration for PID Controller.

The structure uses 'rate feedback' instead of derivative action, since the motor speed is available for feedback.

In this lab, you are going to take an analytic approach to tuning, where you will position the closed loop system poles to achieve time-domain transient and steady state performance objectives based on 2nd order closed loop system criteria[1]. Namely,

1. Reduce the percent overshoot (PO) of the system closed loop 50^o step response to < 1 degree.
2. Reduce the peak time (t_{peak}) to < 0.1 seconds.
3. Reduce the steady state error ($e_{ss-step}$) of the closed loop step response to < 0.1 degrees.
4. Reduce the steady state error ($e_{ss-ramp}$) of the closed loop ramp response to < 2^o for a ramp slope corresponding to 50^o/sec.

Linear Controller Design

1. Review class notes on pole placement design for dominant 2nd order linear systems in the time domain.
2. For the transient specifications, sketch a pole/zero map of the desired pole locations.
3. For the steady state specifications, determine the required system type, and work out the required gain constraints for a linear second order system. Notice that integral control will not, in theory, be required.
4. Design a P+Rate Feedback Controller to meet your control objectives. To do this, you will have to first approximate your closed loop system with a **linear** 2nd order transfer function model reflecting the structure shown in Figure 1 (i.e., ignore the saturation limits and deadband effects), and work out the closed loop transfer function. Attempt to position the closed loop poles as best as possible to achieve all the performance objectives[2].

Save representative data of the response of your analytic closed loop **linear** transfer function step and ramp responses. The MATLAB functions **stepeval.m** and **rampeval.m**, are available on D2L to help plot and analyse the results. The built-in Matlab function stepinfo is also useful. Please see the marking rubric for details on the requirements for the lab report.

[1] Refer to the Appendix for relationship between time domain performance specifications and 2nd order linear transfer functions.
[2] You might consider writing a MATLAB function that will return the required controller gains K_p, T_d given the desired performance specifications.

Linear Controller Simulation using Linear and Nonlinear Motor Model

In this segment of the experiment you will expand your SIMULINK model for the servo motor that you create in Lab #2 to include the PID Controller structure, as shown in Figure 2[3]:

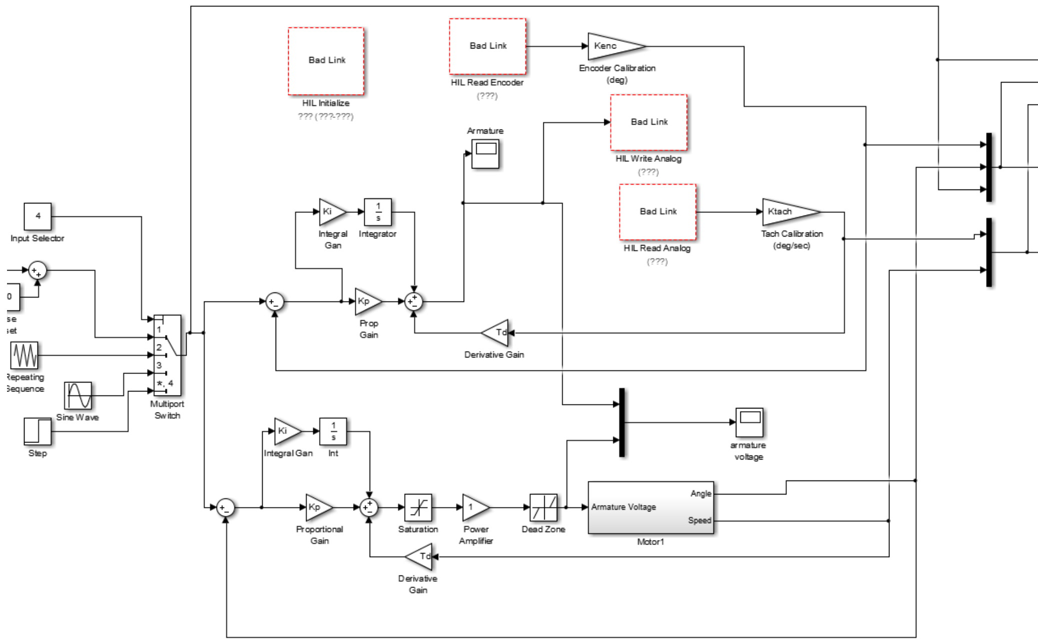


Figure 2 SIMULINK Configuration for Feedback Control.

Implement your design for the PD controller (disable the integral component), and run simulations for the following scenarios:

1. A square wave input of amplitude 50^o. Adjust its period so that the system dynamics are well observed.

2. A triangle wave input with a 50^o/sec slope. Adjust its period so that the system dynamics are well observed.

For each, you should run a simulation first using your **linear** model (i.e., remove the saturation and deadband effects), and then a simulation using your **nonlinear** model. In all situations, monitor the reference signal, the load angle, the load speed, and the armature voltage.

To more easily estimate whether a response meets the performance requirements, you will need to save the response, and replot the results in MATLAB. The MATLAB functions **stepeval.m** and **rampeval.m**, are available on D2L to help plot and analyse the results. The built-in Matlab function stepinfo is also useful. Please see the marking rubric for details on the requirements for the lab report.

Frame out how you would approach retuning of your controller to make your nonlinear model achieve the performance specifications, but **DO NOT MAKE ANY CHANGES TO THE CONTROLLER**

[3] Note: please realize that the input signals now represent reference set points for the load angle, and no longer armature voltage!

Linear Controller Redesign for Real Motor System

In this segment of the experiment you will implement your design on the servomotor and verify its effectiveness.

Implement your design for the PD controller, and run tests on the real motor for the following scenarios:

1. A square wave input of amplitude 50^o. Adjust its period so that the system dynamics are well observed.

2. A triangle wave input with a 50^o/sec slope. Adjust its period so that the system dynamics are well observed.

Compare the simulation using your **linear** model (i.e., remove the saturation and deadband effects), and then a simulation using your **nonlinear** model, to the **real motor**. In all situations, monitor the reference signal, the load angle, the load speed, and the armature voltage.

Are the system specs met for the real motor? If they are, your controller design is complete. If not, retune your controller so that the design achieves the performance objectives when using the actual servomotor. If you think it will help, you are free to include the integral. Follow the plan for redesigning the controller that you previously framed out. The objective is to have the **real motor** meet performance specifications.

Implement your design for the final PD controller, and run tests on the linear simulation, nonlinear simulation and real motor for the following scenarios:

1. A square wave input of amplitude 50^o. Adjust its period so that the system dynamics are well observed.

2. A triangle wave input with a 50^o/sec slope. Adjust its period so that the system dynamics are well observed.

Confirm that the real motor meets performance specification.

Compare the performance of the real motor to that of the simulated linear and nonlinear models. Do you think working with a nonlinear model of the motor is helpful in the design process?

Robustness

In this segment of the lab, you will investigate robustness of your design in situations out of normal operating range.

Implement your final design for the P(I)D controller, and run tests on the real motor for the following scenarios:

1. A square wave input of amplitude 20^o. Adjust its period so that the system dynamics are well observed.

2. A triangle wave input with a 20^o/sec slope. Adjust its period so that the system dynamics are well observed.

Note that this will emphasize the deadband effect. Does the performance suffer as a result?

3. Repeat for the square and triangle wave inputs of 150^o and 150^o/second slope. Note that this will most likely saturate the servomotor, so expect that the performance may be compromised. Is the performance compromised as expected?

4. Save actual motor system responses and tabulated the results compared with the 50^o and 50^o/second slope

Results and Deliverables

Please see the Lab3MarkingRubric.

Appendix

Normalized 2nd order linear transfer function: $G(s) = \frac{\omega_n^2}{s^2+2\zeta\omega_n s+\omega_n^2}$

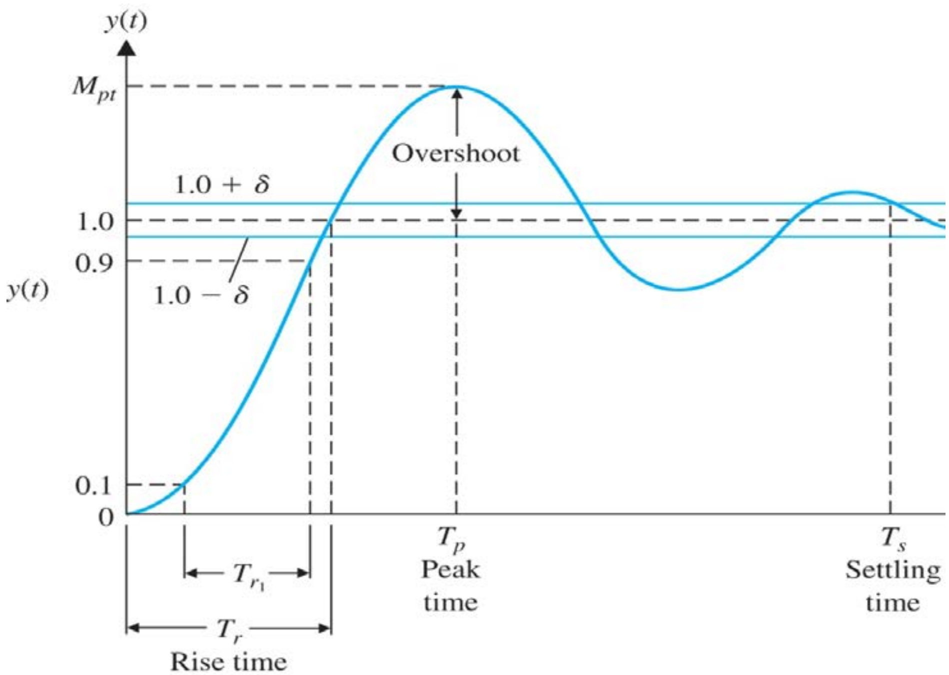
%OS = $100e^{-\zeta\pi/\sqrt{1-\zeta^2}}$

$$T_p = \frac{\pi}{\omega_n\sqrt{1-\zeta^2}}$$

$$M_{p_t} = 1 + e^{-\zeta\pi/\sqrt{1-\zeta^2}}$$

$$T_{s,2\%} = \frac{4}{\zeta\omega_n}$$

$$T_{r_1} \approx \frac{2.16\zeta+0.60}{\omega_n}; 0.3 < \zeta < 0.8$$



Normalized second order linear system unit step response