

System Performance Design

Section 1 -- Background Information

In this lab we will use the Simulink model shown in Figure 1 for the torsion control plant to develop a feedback control for the real system. The feedback control will use position and velocity feedback to achieve the performance specifications given in Table I.

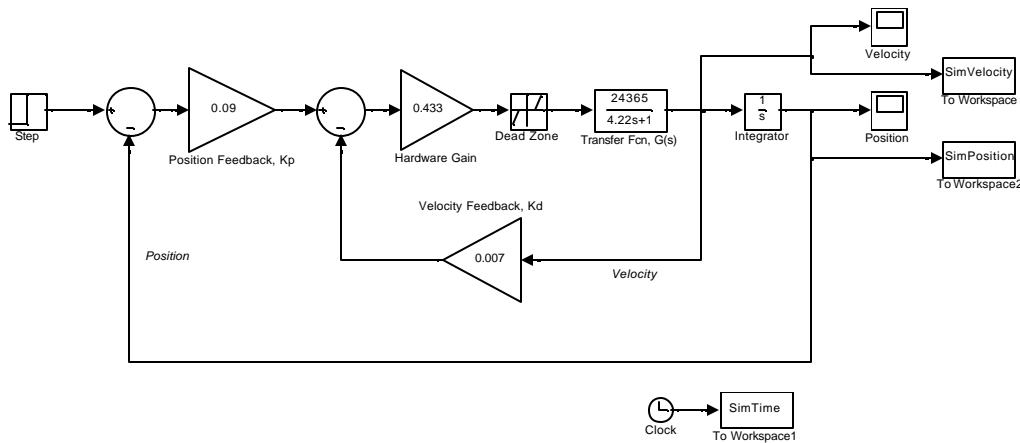


Figure 1. Simulink model of torsion control plant

Table I: Performance Specifications	
Overshoot, %OS	< 10 %
Settling time, T_s	= 0.5 seconds
Position error, e_p	< 5 %

Section 2 – Procedure

2.1 Determine range of gains Kp and Kd to use in the Simulink model

2.1.1 Derive Kp and Kd to meet the desired performance specifications listed in Table I.

2.1.2 Test the gain values for Kp and Kd in your Simulink model. You may need to adjust these gains to achieve the desired system performance. Set the step input amplitude for 100 degrees.

2.2 Verify Performance on the Real System

2.2.1 Using the set of gains found from your Simulink model, establish how well these gains work in the real system.

- a. Setup → User Units, select degrees
- b. Select Command → Trajectory
 - Select Impulse and Unidirectional moves. Click Setup.
 - Select Closed Loop Impulse and set Amplitude (degrees)=100; Pulse width (msec) = 5000; Reps = 1; Dwell Time (msec)= 0;
- c. Select Setup → Control Algorithm.

- Select Continuous, then PI with Velocity Feedback
- Click on Setup Algorithm. Use K_p and K_d from 2.1.2.
- Implement Algorithm, then click OK
- d. Select Data → Setup Data Acquisition
 - Sample Period (servo cycles) = 2
 - Selected Items should be Commanded Position and Encoder 1 Position
 - Click OK
- e. Utility → Zero Position
- f. Select Command → Execute → Run

2.2.2 Adjust K_p and K_d as necessary to achieve the desired performance.

Section 3 -- Report

Submit a report in the standard format for this class. Discuss your observations of the Simulink model performance compared with the real system. Explain any differences observed.

Notes

Neglecting the Dead Zone block, the system of Figure 1 can be redrawn as shown in Figure 2 with $G(s)$ defined by Equation 1. K_m is the motor constant and t_m is the time constant, both of which were derived in a previous lab. K_H is the hardware gain (0.434). Using the simplified model, one verifies that the root locus plot will have a form similar to that shown in Figure 3 with s_1 being the point half way between the origin and the pole p_1 .

$$G(s) = \frac{K_H K_m / t_m}{s(s + p_1)} \tag{Equation 1}$$

with

$$p_1 = \frac{1 + K_d K_H K_m}{t_m}$$

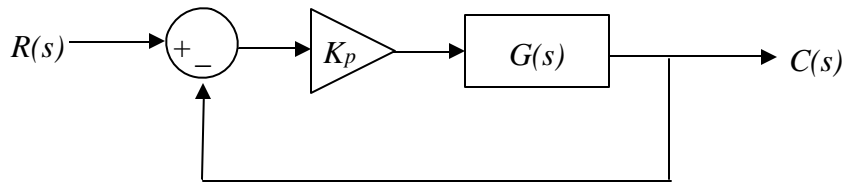


Figure 2. Simplified block diagram

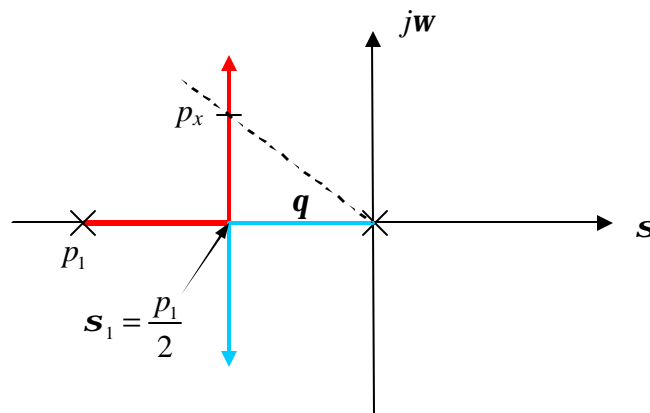


Figure 3. Root locus of the closed-loop control system

The settling time, T_s , is the time for the system response to reach and stay within $\pm 2\%$ of the steady state value. Equation 2 is used to compute T_s from \mathbf{S} . Using this relation with Equation 1, a value for the velocity feedback, K_d , can be determined.

$$T_s = \frac{4.5}{\mathbf{S}} \quad \text{Equation 2}$$

Once a value for K_d has been determined, a value for K_p can be found next. Equation 3 relates the Percent Overshoot, $\%OS$, and the system damping ratio, \mathbf{V} . Use this equation to find the required system damping ratio to achieve the desired $\%OS$.

$$\%OS = e^{-(\zeta/\sqrt{1-\zeta^2})} \times 100 \quad \text{Equation 3}$$

With \mathbf{V} known, it can be used to find the angle \mathbf{q} in Figure 2, where $\cos(\mathbf{q}) = \mathbf{V}$, and also to find the location of the complex pole, p_x , which is the location where the system pole must lie to achieve the desired $\%OS$. Using Matlab's **RLOCUS** function and the knowledge of where p_x must lie in the complex plane, one can determine the corresponding value for K_p that gives the desired $\%OS$.

Regarding the position error, since this system is a Type 1 system, what can be said about the specification requirement for e_p ?