ECEN 4638: Lab X.1PI

Rane Brown
Kate Schneider

March 3, 2016

Contents

List of Figures

1 Description

This lab will further explore the Torsional Disc System using Proportional-Integral control, with the goal of designing a PI-based cruise control system that works well for inertia changes of ±10%. The controller is then tested on two different Torsional Disc apparatuses to evaluate the design for robustness. The system setup will be similar to what was used in labX.1P; only the bottom disc of the TDS will be used and the four weights will be set at a radius of 7.5cm.

2 System Model

As in labX.1P, the Torsional Disc system can be modeled as an LTI system, with the below equation

$$J\dot{\omega} + c\omega = k_h u \tag{1}$$

Where $J =$ total system inertia, $\omega =$ velocity rad/sec, $c =$ system drag, $k_h =$ hardware gain, $u =$ reference.

2.1 Calculated Parameters

During labX.1P, each team calculated parameters c and k_h for one torsional disc system based on its experimental data. These parameters are listed in the table below.

System	c	k_h
1	0.0082	0.3577
1	**0.0079**	**0.3598**
2	0.0078	0.15
2	0.0081	0.376
3	0.0110	0.387
3	0.010	0.0084
4	0.0111	0.0084
4	0.0184	0.387

Table 1: Torsional Disc System Parameters

For a radius of 7.5 cm, the calculated total system inertia J is 0.0128 kgm^2.

For this experiment, our cruise control system was designed using the calculated parameters from Torsional Disc System 1 (in bold in the above table), and the PI controller was tested on TDS systems 1 and 4.

2.2 Transfer Functions

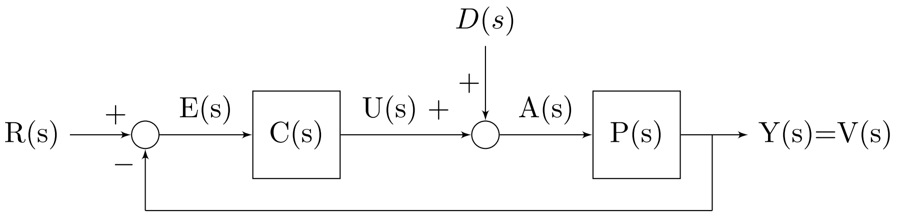

Figure 1: Cruise Control Feedback System

From our LTI model, we see that the system plant is modeled with the below equation.

$$P(s) = \frac{k_h}{Js + c} e^{-T_d s} \tag{2}$$

Where $J =$ total system inertia, $c =$ system drag, $k_h =$ hardware gain, and $T_d =$ time delay. In this lab, we use a PI controller, $C(s) = \frac{k_I}{s} + k_p$.

include info from sheet on these and what they tell us!! Then add plots!! There are multiple transfer functions which are of interest in our system. The first transfer function of interest is the closed-loop transfer function from reference input R to output Y:

$$H_{yr} = \frac{Y}{R} = \frac{CP}{1 + CP} = \frac{kk_I + kk_p s}{Js^2 + (c + kk_p)s + kk_I} \tag{3}$$

From the equation for H_{yr}, we can see that the DC gain of this transfer function is 1, as we would expect because....

Another transfer function of interest is the closed-loop transfer function from reference input $R(s)$ to controller output $U(s)$.

$$H_{ur} = \frac{U}{R} = \frac{C}{1 + CP} = \frac{Jck_I k_p s^3 + Jk_I s^2 + ck_I s}{Js^2 + (c + kk_p)s + kk_I} \tag{4}$$

This transfer function has a DC gain of $H_{ur}(0) = \frac{c}{k}$. This value represents the input that is needed to compensate parameter c in steady state.

We can also look at the transfer function from the disturbance $D(s)$ to the output $Y(s)$, since again, rejecting disturbance input such as hills and bumps is important in a cruise control system.

$$H_{yd} = \frac{Y}{D} = \frac{\frac{k}{J}s}{s^2 + (\frac{c}{J} + \frac{k_p k}{J}s) + \frac{k}{J}k_I} \tag{5}$$

The final closed-loop transfer function of interest is H_{ud}, the transfer function from the disturbance $D(s)$ to the controller output $U(s)$.

$$H_{ud} = \frac{-\frac{k}{J}(k_p s + k_I)}{s^2 + (\frac{c}{J} - \frac{kk_p}{J})s + \frac{kk_I}{J}} \tag{6}$$

We will also examine the open-loop transfer function $L(s) = P(s)C(s)$, which will be useful in determining gain and phase margins of our system to ensure stability.

$$L(s) = P(s)C(s) = \frac{k}{Js + c}(\frac{k_I}{s} + k_p) \tag{7}$$

3 Matlab Analysis

3.1 Time Domain

Before implementing our PI controller on a physical TDS, we can formulate and test the design in Matlab to ensure the system meets certain time and frequency domain specifications. In the time domain, relevant specifications include rise time t_r, settling time t_s, and overshoot M_p. Choosing desired ranges for these parameters allows us to estimate ζ and ω_n for a simple second-order system of the form

$$\frac{K\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2} \tag{8}$$

with the relationships

$$t_r = \frac{1.8}{\omega_n}, \qquad t_s = \frac{4.6}{\zeta\omega_n}, \qquad M_p = e^{\frac{-\pi\zeta}{\sqrt{1-\zeta^2}}} \tag{9}$$

However, when using these estimates, we must keep in mind that our PI controller also introduces an additional zero to the closed-loop transfer function $H_{yr}(s)$ which will affect the time domain response of our system. Most notably this zero will increase M_p and decreasing t_r. Thus, design must aim for a larger ζ, while ω_n can be decreased slightly and still meet rise time specifications.

If, for instance, we choose desired values of $t_r < 1sec$, $t_s < 5sec$, and $M_p < 10\%$, we can estimate suitable ω_n and ζ for $H_{yr}(s)$ by hand calculation, then model the system in Matlab. Iterating on the design in Matlab allows us to fine-tune the controller, which we can then evaluate on the physical system.

discuss/plot all the other transfer functions!!!

The specifications above results in the values $\omega_n =$, $\zeta =$ which gives values $k_I =, k_p =$. This results in the system transfer function $H_{yr}(s) =$. The step response plot for $H_{yr}(s)$ is given below.

add plot here

The additional zero results in a greater overshoot and overall faster response. The plots of the other three transfer functions of interest are included below. add other TF plots here

We can further refine our design in order to get a better time domain response from our system, now that we have observed the effect of the additional zero in Matlab simulations. include one more iteration, with better params which worked for the exp sys.

3.2 Frequency Domain

In addition to time domain specifications, certain aspects of a system's frequency domain response are important to take into consideration to ensure the stability and good performance of that system. These values are the steady-state error, e_{ss}, the phase margin PM, the gain margin GM, and the bandwidth,, which gives us the system's half-power frequency. e_{ss} and ω_{BW} are important for system performance. PM, and GM ensure stability of the system, with $45 < PM < 60$ an ideal region for the phase margin, $GM >$ a good goal for gain margin.

We can plot the frequency response of our first system () and analyze the stability and performance of the system. Note that we have $PM =$, $GM =$, $\omega_{BW} =$, and $e_{ss} =$. discuss what these mean, then do an iteration using freq plots. FINALLY, put all of these together into a system that has good time and freq response characteristics in Matlab. Make sure for this system you plot several other transfer functions as well - use one that may have performed well on the TDS itself - then add in $+-10-\%$ on the J value and discuss how that worked out.

4 Experimental Analysis

4.1 Time Domain

4.2 Frequency Domain

5 PI Controller Design