MASSACHUSETTS INSTITUTE OF TECHNOLOGY
Department of Electrical Engineering and Computer Science

6.3100: Intro to Modeling and Control—Spring 2025

Disturbance in the Force Postlab - Due April 7th

We suggest writing your solutions in a separate document and then turning it into a pdf (scan it or take pictures if you like to use pencil and paper). And PLEASE INCLUDE YOUR DERIVATIONS!! We have no way of verifying your understanding with just a numerical answer, particularly if there were a minor calculation error.

Problem One

In lab, we had a block diagram for our arm control feedback system, repeated below.

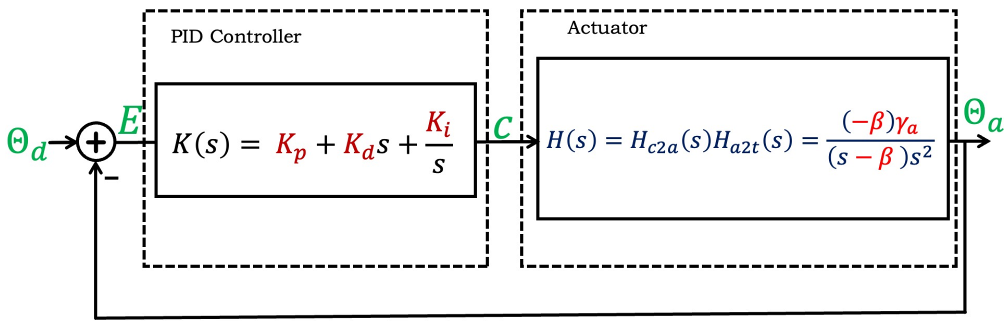

Figure 1: Block Diagram For Arm Feedback System.

We usually refer to $H(s)$ in the above diagram, whose input is the command C and whose output is Θ_a, as the transfer function for the "open-loop" system because there is no looping of the output back to the input. When we add $K(s)$, the controller we design, and then "loop" the output back and subtract it from input Θ_d, we refer to the system as "closed-loop". We then refer to $G(s)$ as the closed-loop transfer function, and it relates Θ_d to Θ_a as in

$$\Theta_a = G(s)\Theta_d.$$

If the input to our system is a complex sinusoid,

$$\theta_d(t) = \Theta_d e^{j\omega t}$$

then in sinusoidal-steady-state, the output of the closed loop system is

$$\theta_a(t) = \Theta_a e^{j\omega t} = G(j\omega)\Theta_d e^{j\omega t}$$

and we refer to $G(j\omega)$ as the "closed-loop" frequency response.

Part A)

If we assume $K_i = 0$, we can write $G(s)$ for the above block-diagrammed system in as a ratio of two polynomials in s,

$$G(s) = \frac{b_1 s + b_0}{s^3 + a_2 s^2 + a_1 s + a_0}.$$

In terms of K_p, K_d, β, and γ_a, what are b_1, b_0, a_2, a_1, and a_0?

Part B)

The a's and b's in the above expression are real, and generally, $G(j\omega)$ is a complex number. The "realness" of the a's and b's means that we known something about $G(-j\omega)$.

The complex conjugate of a complex number $p + qj$, denoted $(p + qj)^*$, is given by $(p + qj)^* = p - qj$. Show by direct calculation that given real a's and b's in the ratio of polynomials representing $G(s)$, then $G(-j\omega) = G(j\omega)^*$ (Hint, if w and v are complex numbers, then $\left(\frac{w}{v}\right)^* = \frac{w^*}{v^*}$).

Part C)

Show that

$$G(j\omega)(j\omega)^2 a_2 + G(j\omega)(j\omega)a_1 + G(j\omega)a_0 - (j\omega)b_1 - b_0 = -G(j\omega)(j\omega)^3.$$

Suppose we rewrite the above equation as

$$G(j\omega)(j\omega)^3 \tilde{a}_3 + G(j\omega)(j\omega)^2 \tilde{a}_2 + G(j\omega)(j\omega)\tilde{a}_1 - (j\omega)\tilde{b}_1 - \tilde{b}_0 = -G(j\omega),$$

Determine an analytic formula for the $\tilde{a}$'s and $\tilde{b}$'s in terms of the a's and b's.

Problem Two

When you used the sweeper in lab, you were measuring the closed-loop transfer function $G(j\omega)$ of the arm control system by applying input sinusoids and measuring the output. HOWEVER, when you took your data, you used a controller with a non-zero integrator gain, as we suggested, but we discovered that adding the integrator makes this postlab problem unnecessarily complicated. So, we are providing data from our arm running the sweeper, but with zero integrator gain. The data file is linked on the postlab page. In this problem, you will fit a transfer function to our data by using (and modifying) our fitter, *labFitter.m* (you can download *labFitter.m* from the postlab webpage).

The function labFitter takes the matrix of measured data we collected in lab, and fits it to a transfer function. Specifically, the fitter finds the a's and b's that provide the best least-squares fit to

$$
\begin{array}{rrr}
G(j\omega_1)(j\omega_1)^2 a_2 + & G(j\omega_1)(j\omega_1)a_1 + G(j\omega_1)a_0 - (j\omega_1)b_1 - b_0 \approx & -G(j\omega_1)(j\omega_1)^3 \\
G(j\omega_2)(j\omega_2)^2 a_2 + & G(j\omega_2)(j\omega_2)a_1 + G(j\omega_2)a_0 - (j\omega_2)b_1 - b_0 \approx & -G(j\omega_2)(j\omega_2)^3 \\
\vdots & \vdots & \vdots \\
G(j\omega_{40})(j\omega_{40})^2 a_2 + & G(j\omega_{40})(j\omega_{40})a_1 + G(j\omega_{40})a_0 - (j\omega_{40})b_1 - b_0 \approx & -G(j\omega_{40})(j\omega_{40})^3 \\
G(j\omega_{41})(j\omega_{41})^2 a_2 + & G(j\omega_{41})(j\omega_{41})a_1 + G(j\omega_{41})a_0 - (j\omega_{41})b_1 - b_0 \approx & -G(j\omega_{41})(j\omega_{41})^3
\end{array}
$$

where the measured frequency index, i, ranges from 1 to 41.

The sweeper saves data for 41 unique frequencies but you should have a data matrix with more than 41 rows, because we took data for two or three sweeps in a row. The matlab script averages the multiple readings for each unique frequency, examine lines 21-32 to see this.

To load our data file, download "data032025.mat" from the postlab page, and in matlab type "load data032025.mat". Then in the matlab command window, you can type

[GsFit,GsSynth] = labFitterMySol(3,1,x,false)

Part A

The script will run and fit a three-pole, one-zero transfer function to synthetic data. The script prints the coefficients in the command window, plots results, and then returns the transfer function for the fit and for the synthetic data. What happens when you change "false" to "true"? That is, what is the fitter doing differently when you type

[GsFit,GsSynth] = labFitterMySol(3,1,x,true)?

Part B

If you examine the output displayed in the command window in Matlab, you will see that the program displays two vectors of results, "abcoeffComplex" and "abcoeffs".

- 1) How do those two vectors relate when you fit synthetic data? When you fit your measured data? Why do you think that is?
- 2) Examine lines near line 76 in the fitter, what is the *Acc* matrix and what is the *Bcc* vector? How might you use them to insure that the a's and b's generated by the least-squares solution are real?
- 3) Modify the labFitter function to use *Acc* and *Bcc* instead of A and B to compute "abcoeffs". Do you see a difference in the fit if you use synthetic data? How about when you use your measured data? Do you get a good fit at high frequency? How about at low frequency?
- 4) The fitter returns the transfer funct "GsFit". Try printing it (just type GsFit at the matlab command window and hit return). what can you say about the poles (natural frequencies) and zeros of the fit? Are they different if you fit synthetic versus measured data?

Part C

For this part, you will be modifying the matlab script so that the script will compute the best least-squares fit to the set of equations

$$\begin{array}{rcl} G(j\omega_1)(j\omega_1)^3 a_3 + & G(j\omega_1)(j\omega_1)^2 a_2 + G(j\omega_1)(j\omega_1) a_1 - (j\omega_1) b_1 - b_0 \approx & -G(j\omega_1) \\ \vdots & \vdots & \vdots \\ G(j\omega_{41})(j\omega_{41})^3 a_3 + & G(j\omega_{41})(j\omega_{41})^2 a_2 + G(j\omega_{41})(j\omega_{41}) a_1 - (j\omega_{41}) b_1 - b_0 \approx & -G(j\omega_{41}). \end{array}$$

Be sure to use the *Acc* matrix and the *Bcc* vector so that your fitter uses both positive and negative frequencies.

- 1) Examine the lines of the matlab function near 70 and near 90 to see hints for how to make the changes. For this problem, please submit the lines you changed.
- 2) Does the fit improve when you fit synthetic data with your new version of the fitter. How about when you fit to your measured data? Do you get a good measured-data fit at high frequency? How about at low frequency?
- 3) The fitter returns the transfer funct "GsFit". Try printing it (just type GsFit at the matlab command window and hit return). What can you say about the poles (natural frequencies) of your fit, and what do they tell you about stability of your control system. Are they different if you fit synthetic versus measured data?
- 4) What are the coefficients of the numerator polynomial of "GsFit" when you fit your measured data? What should they be? What are the natural frequencies of your fit (the roots of the denominator polynomial)? How do they compare to the natural frequencies of the fit to your data for the other two versions of the fitter?