Flexible Beam

Objectives

The objective of this lab is to learn about the challenges posed by resonances in feedback systems. An intuitive understanding will be gained through the manual control of a flexible beam resembling a large space robotic arm. Control design will be performed in the frequency domain using a lead controller. A notch filter will be incorporated in the feedback loop in order to reduce the excitation of resonances.

Introduction

Systems with lightly-damped, complex poles (resonances), are encountered in many applications. An example is a large robotic arm in space, whose transversal dimensions are made small to reduce weight. The arm will bend and oscillate if moved rapidly. In a computer disk drive, a read/write head is attached to the end of a small, rigid structure which is rotated rapidly to access various tracks. When the head is positioned within fractions of microns, even such a rigid structure behaves like a flexible structure.

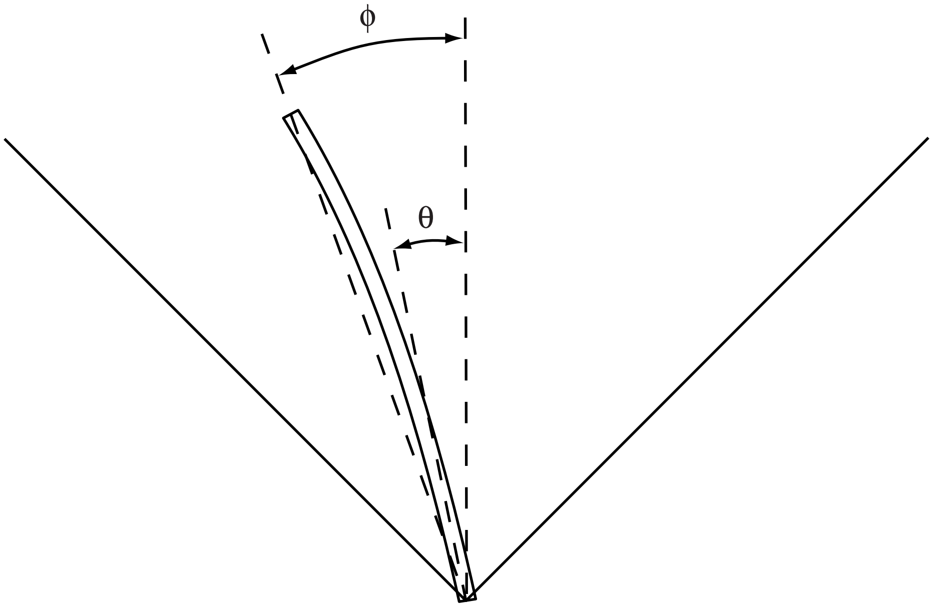

Figure 1: Flexible Beam

The diagram of the flexible beam is shown on Fig. 1. The angle of the beam at the shaft is denoted θ, while ϕ is the angle at the tip. If there was no flexibility, the two angles would be equal. Experimental data was collected on a flexible beam of length $0.4m$. Fig. 2 shows the frequency response that was measured from the motor current to the angular acceleration of the shaft, while Fig. 3 shows the response measured from the motor current to the angular acceleration of the tip. The acceleration at the shaft was obtained by measuring the position with an encoder (and multiplying the frequency response by $-\omega^2$ to obtain the acceleration), while the acceleration at the tip was obtained with an accelerometer. The plots show the experimental data, as well as approximate fits obtained with fourth-order models (as dashed lines).

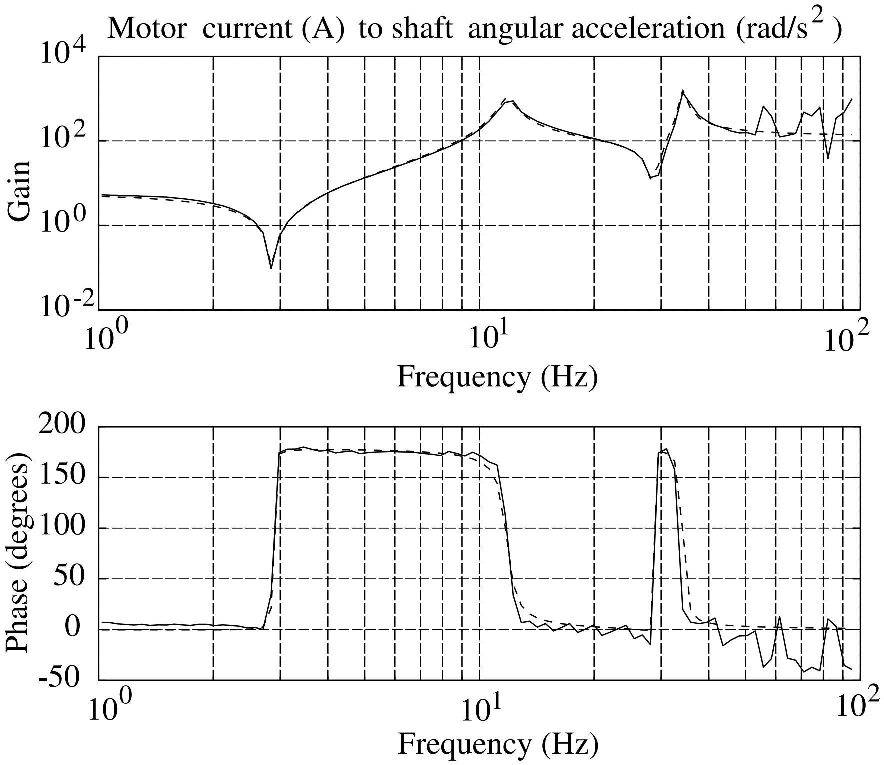

Figure 2: Frequency response of the flexible beam from motor current to shaft angular acceleration

The approximate models shown on the plots are given by

$$\begin{aligned}\frac{s^2\Theta(s)}{I(s)} &= \frac{k_\theta(s-z_1)(s-z_1^*)(s-z_2)(s-z_2^*)}{(s-p_1)(s-p_1^*)(s-p_2)(s-p_2^*)}\\ \frac{s^2\Phi(s)}{I(s)} &= \frac{k_\phi(s-z_3)(s-z_4)(s-z_5)(s-z_6)}{(s-p_1)(s-p_1^*)(s-p_2)(s-p_2^*)}\end{aligned} \tag{1}$$

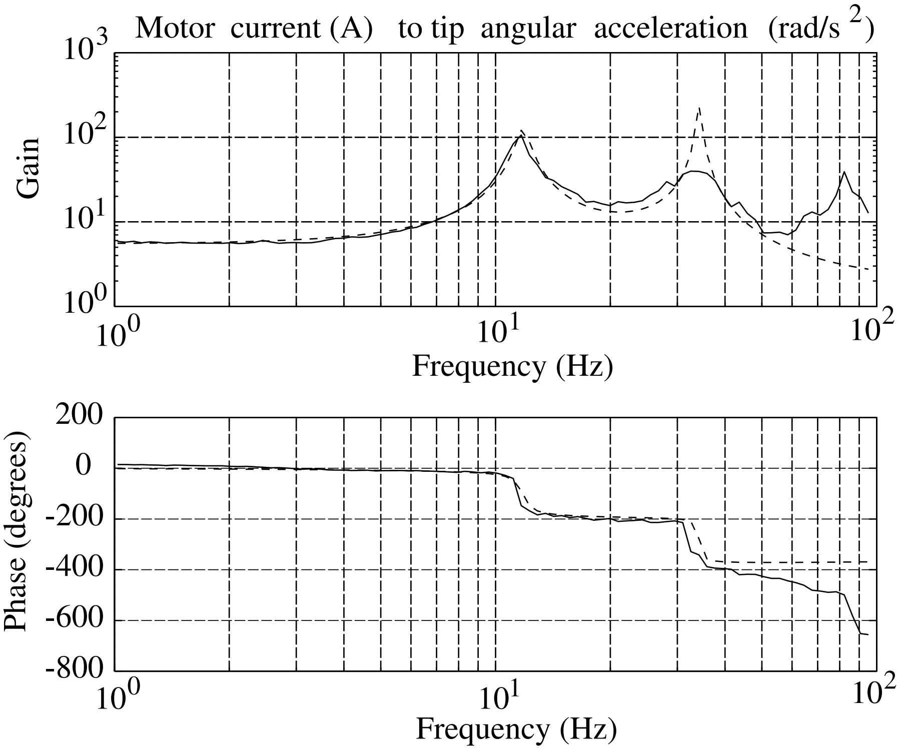

Figure 3: Frequency response of the flexible beam from motor current to tip angular acceleration

where $p_1 = -3 + 74j$, $p_2 = -3 + 215j$, $z_1 = -0.07 + 18j$, $z_2 = -0.07 + 180j$, $z_3 = 100$, $z_4 = -120$, $z_5 = 200$, and $z_6 = -300$ (note that the poles are very lightly damped). The input variable is the current in the motor, measured in A, and the angles θ and ϕ are measured in radians. The constants k_θ and k_ϕ are such that the DC gains of the transfer functions are equal, with

$$k_p = \left(\frac{s^2\Theta(s)}{I(s)}\right)_{s=0} = \left(\frac{s^2\Phi(s)}{I(s)}\right)_{s=0} = 5.5 \tag{2}$$

The equality for θ and ϕ follows from the fact that there is no bending of the beam near zero frequency. For low frequencies, the transfer functions are therefore approximately given by

$$\frac{\Theta(s)}{I(s)} \simeq \frac{\Phi(s)}{I(s)} \simeq \frac{k_p}{s^2} \tag{3}$$

This approximation of the system is the double integrator encountered with the ball and beam. The feedback design is more difficult than for the ball & beam, because of additional poles close to the $j\omega$-axis, and because of zeros close to the $j\omega$-axis and in the right half-plane.

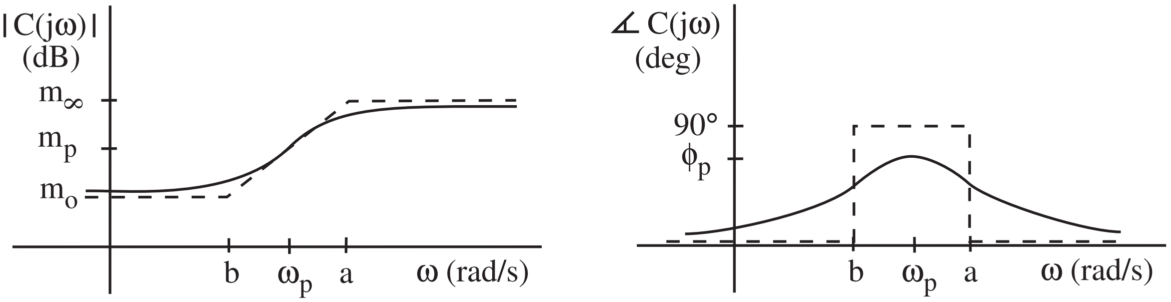

Figure 4: Bode plots of lead controller

Manual control

The simulation file is called *flex.m.* You should play with the simulation and move the beam from the 45° line to the −45° line. You will encounter two difficulties: the $1/s^2$ behavior and the flexibility of the beam. Resonances can be avoided by moving the beam slowly, but performance will be unimpressive. It can be enlightening (and fun) to "excite" the resonances by applying commands in the same frequency range as the flexible modes. The beam will bend to large angles. Once this mechanism is understood, you may return to the task of rapidly moving the beam from side to side without exciting such resonances.

Note that the simulation was implemented differently from the previous ones. The continuous-time model was discretized assuming a sampling period of $200Hz$. Since the program runs at a rate of approximately $20Hz$, the visualization slows the dynamics by a factor of 10. This result is helpful, because the dynamics of the actual system are too fast to be controlled manually.

Lead controller design

The objective is to design a *lead controller*

$$C(s) = \frac{I(s)}{\Phi_{ref}(s) - \Phi(s)} = k_c \frac{(s+b)}{(s+a)} \tag{4}$$

Such a controller was designed for a phase-locked loop in the advanced PLL lab. Here, the design will be performed in the frequency domain. The motor current i is the control signal, and the tip angle ϕ is the output to be regulated, with a reference value ϕ_{ref}.

The Bode plots of the lead controller are shown in Fig. 4. In the notes for the course, the variables are shown tosatisfy the following constraints

$$\begin{aligned} \omega_p &= \sqrt{ab} \\ m_p &= k_c \sqrt{\frac{b}{a}} \end{aligned}$$

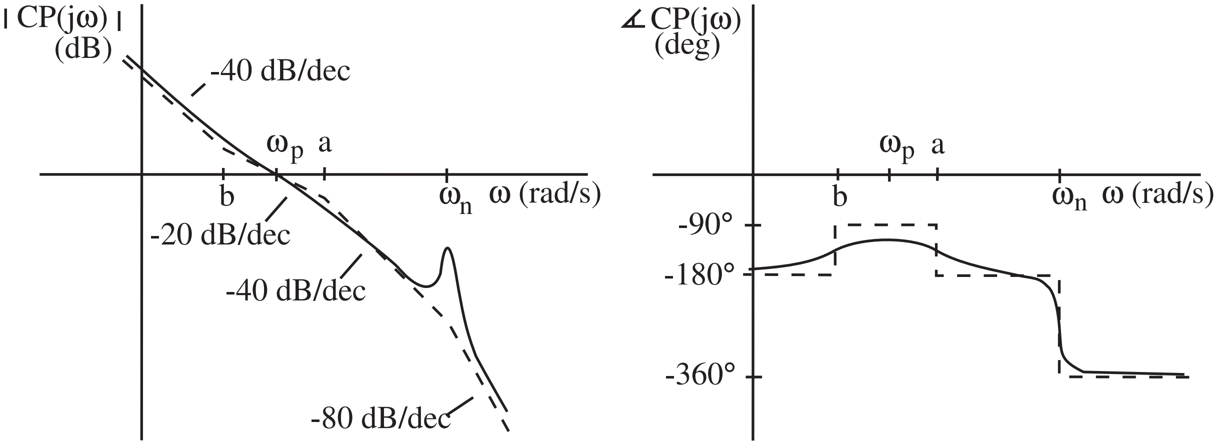

Figure 5: Bode plots of plant and lead controller (only one resonant mode shown)

$$\frac{a}{b} = \frac{1+\sin(\phi_p)}{1-\sin(\phi_p)} \tag{5}$$

To determine the controller parameters, we consider an approximate model of the plant with the two poles at the origin and the first two resonant modes. The tranfer function from i to ϕ is then

$$P(s) = \frac{\Phi(s)}{I(s)} = \frac{k_p}{s^2} \cdot \frac{\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2} \tag{6}$$

where $k_p = 5.5$, $\omega_n = |p_1|$, $\zeta = -\operatorname{Re}(p_1)/\,|p_1|$, and $p_1 = -3 \pm j73$. Fig. 5 shows the Bode plots of the loop transfer function for the approximate plant and the lead compensator. The magnitude plot shows ω_p as the crossover frequency, a condition that is to be enforced by proper choice of the parameters. Assuming that $\omega_n \gg \omega_p$, show using (5) that

$$\begin{aligned} |C(j\omega_p)P(j\omega_p)| &\simeq \frac{k_c k_p}{\omega_p^2}\sqrt{\frac{b}{a}} \\ |C(j\omega_n)P(j\omega_n)| &= \frac{k_c k_p}{2\zeta\omega_n^2} \end{aligned} \tag{7}$$

Then, find the values of the controller parameters k_c, a, and b such that

$$\phi_p = 60^\circ, \quad |C(j\omega_p)P(j\omega_p)| = 1, \quad |C(j\omega_n)P(j\omega_n)| = 0.1 \tag{8}$$

The idea is that the first two conditions will ensure a crossover frequency at ω_p and a phase margin of 60°. The third condition will yield a gain margin of 10, given that the phase crossover frequency will be close to ω_n.

Using the parameters of the lead controller, plot the step response of the closed-loop system (function *step* in Matlab) and the Bode plots with the gain and phase margins (function *margin* in Matlab). *Make sure to use the complete plant transfer function for this step, not the approximate one.* The feedback system may be assembled using the functions *series* and *feedback* in Matlab, if desired. Also compute the locations of the closed-loop poles (function *pole* or *roots* in Matlab).

Automatic control

Design #1

Implement the controller in the simulation with files *flexc.m* and *flexcinit.m*. A discrete-time equivalent of the control system should be computed, as was done in the previous lab. Discretization should be based on the 200Hz sampling frequency, so that visualization will show the system at a rate slowed down by a factor of 10. Plot the responses of i, θ, ϕ, and $\phi - \theta$ to step inputs. Let ϕ_{ref} switch between 45° and −45° every 2 seconds, and record the data for 6 seconds (which will take 60 seconds in the real-time simulation). The visualization should show a slow response, with a large overshoot, due to the low frequency zero at $s = -b$. If you reduce the gain margin by increasing $|C(j\omega_n)P(j\omega_n)|$, you will find that the crossover frequency increases and that the response speeds up, but oscillations are observed due to excitation of the beam's flexible modes.

Design #2

Improve the design by cascading the lead/lag controller with a notch filter and by prefiltering the reference input, so that

$$I(s) = C(s)C_{notch}(s)\left(C_F(s)\Phi_{ref}(s) - \Phi(s)\right) \tag{9}$$

where $C(s)$ is the lead controller considered earlier. Let the prefilter be

$$C_F(s) = \frac{1.2b}{s + 1.2b} \tag{10}$$

where b is the zero of the compensator. Let the notch filter be

$$C_{notch}(s) = \frac{s^2 + \omega_n^2}{s^2 + 2\omega_n s + \omega_n^2} \tag{11}$$

where ω_n is the natural frequency of the first resonant mode. The prefilter will eliminate the overshoot, and the notch filter will allow you to increase the crossover frequency and, as a result, the speed of response.

Re-design the lead controller with $\omega_p = 12$ rad/s, keeping the condition that $\phi_p = 60^\circ$ and the crossover frequency condition

$$\frac{k_c k_p}{\omega_p^2}\sqrt{\frac{b}{a}} = 1 \tag{12}$$

Check the properties of this re-designed controller in combination with the prefilter, notch filter, and plant transfer function: plot the step response, the Bode plots with gain and phase margins, and compute the values of the closed-loop poles. Implement the controller in the simulation, and plot the responses of i, θ, ϕ, and $\phi - \theta$ to step inputs in ϕ_{ref}. Compare the responses to those of the previous controller. Show your code and demonstrate the real-time operation to the TA.

Report at a glance

Be sure to include:

- Description of challenges and strategies for manual control.
- Design and evaluation of the lead controller, with plots of the step response, Bode plots with gain and phase margins, and values of the closed-loop poles. Responses of i, θ, ϕ, and $\phi - \theta$ to step inputs of ϕ_{ref} in the simulation.
- Design and evaluation of the lead controller with notch filter and prefilter, with plots of the step response, Bode plots with gain and phase margins, and values of closed-loop poles. Responses of i, θ, ϕ, and $\phi - \theta$ to step inputs of ϕ_{ref} in the simulation.
- Observations and comments.
- Listing of *flexc.m* and *flexcinit.m*.