EE661 Matlab Project #1

State-Space Model Pole Placement

Due date: Friday, November 7, 2014

This assignment is a **Project**, so ***it must be individual work***.

Part 1) Consider the following system, which is in ***modal-canonic form***.

$$A = \begin{bmatrix} -1 & 250 & 0 & 0 & 0 & 0 & 0 \\ -250 & -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 190 & 0 & 0 & 0 \\ 0 & 0 & -190 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & 150 & 0 \\ 0 & 0 & 0 & 0 & -150 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -300 \end{bmatrix}, \quad B = \begin{bmatrix} 21.1 \\ 7.63 \\ 7.65 \\ 2.17 \\ 6.45 \\ 1.52 \\ 25.0 \end{bmatrix}$$

$$C = \begin{bmatrix} 7.63 & 21.1 & 2.17 & 7.65 & 1.52 & 6.45 & 25.0 \end{bmatrix}, \quad D = [0]$$

Note: Even though matrices B and C have similar entries,
they are **not** simply related by the transpose operation.

The three vibrational (resonant) modes of the system are all very lightly damped (*i.e.*, $\zeta < 0.1$).

Using Ackermann's formula (controllable-canonic method shown in class), find the state-feedback gain vector K that will place the complex closed-loop poles along the lines of constant $\zeta = 0.2$, while maintaining the damped frequencies of the three modes (*i.e.*, imaginary parts of 150, 190 and 250 radians per second). The closed-loop location of the seventh pole, which is at $s = -300$ in the open loop, can be left the same as in open-loop, or it can be moved slightly further left, if you prefer.

For this problem assume that you have access to the states of the system, and not just the output, so that it is possible to do state-feedback. Later, we will see how to build a *state-observer* to estimate the states.

Show that the closed-loop poles are exactly what they were intended to be.

Simulate the closed-loop homogeneous response, given that the initial value of the state is non-zero,

$$x(0^-) = [\, \mathbf{-1.94} \quad \mathbf{-1.65} \quad \mathbf{-0.78} \quad \mathbf{0.54} \quad \mathbf{-0.72} \quad \mathbf{0.31} \quad \mathbf{-0.87} \,]^T.$$

Use a **simulation resolution** of 1 millisecond, or smaller. Demonstrate the improved transient decay time, compared to open-loop (in other words, simulate both the open-loop and the closed-loop cases).

(*continued on next page*)

Part 2) Consider the following system, which is in ***modal-canonic form***.

$$A = \begin{bmatrix} -1 & 250 & 0 & 0 & 0 & 0 & 0 \\ -250 & -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 190 & 0 & 0 & 0 \\ 0 & 0 & -190 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & 150 & 0 \\ 0 & 0 & 0 & 0 & -150 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -300 \end{bmatrix}, \quad B = \begin{bmatrix} 21.1 & 10.2 \\ 7.63 & 12.7 \\ 7.65 & 4.21 \\ 2.17 & 8.32 \\ 6.45 & 0.85 \\ 1.52 & 5.13 \\ 25.0 & 21.3 \end{bmatrix}$$

$$C = \begin{bmatrix} 7.63 & 21.1 & 2.17 & 7.65 & 1.52 & 6.45 & 25.0 \end{bmatrix}, \quad D = \begin{bmatrix} 0 & 0 \end{bmatrix}$$

This system is almost the same as in Part 1. The only difference is that there are two inputs (*i.e.*, matrix *B* now has two columns). Otherwise, all matrices are the same, and even the first column of *B* is unchanged.

Using MIMO state-feedback gain design (the null-space method shown in class, also found in Brogan §13.4), find the state-feedback gain vector ***K*** that will place the complex closed-loop poles along the line of constant ζ = 0.2, while maintaining the damped frequencies of the three modes (*i.e.*, 150, 190 and 250 radians per second). These are the same locations as in Part 1. As before, the closed-loop location of the seventh pole, which is at *s* = -300 in the open loop, can be the same as in open-loop, or it can be moved slightly further left. The gain matrix you design must have nonzero values in both rows (*i.e.*, nonzero feedback must be entering both inputs).

For this problem, again assume that you have access to all states of the system, and not just the output.

Show that the closed-loop eigenvalues are, in fact, what they were intended to be.

Finally, compare the 2-norms of the two state-feedback gain matrices, from Part 1 and Part 2.

What to submit:

For each part, submit a Matlab script (m-file) that, when executed line-by-line, will perform the required computations to obtain the state-feedback gain matrix.

Also in the script, include Matlab commands that perform any additional required tasks, like closed-loop simulations, analyses of the gain matrices, analyses of eigenvalues, creation of any necessary plots (with labels and titles), *etc*.

The two m-files (one for each part) should be submitted by e-mail as attachments.

Submit hardcopy of written answers to questions that are not easily included in an m-file.