ECE-320 Linear Control Systems
Laboratory 7

System Modelling, State Variable Controller Design, and the Real World

Preview In this Lab you will first obtain a second order model of your spring/mass/damper system, design a state variable feedback controller for it, implement the controller on the real system, and then compare the predicted response with the actual response. You will probably make at least two observations during this lab:

- models are not perfect, they are just guides
- real motors have real limits, which can make designing more difficult

Pre-Lab

Print out this lab and **read** it.

Brief Review of State Variable Stuff

Assume we have the plant transfer function

$$G(s) = \frac{k_{clg}}{\frac{1}{\omega_n^2}s^2 + \frac{2\zeta}{\omega_n}s + 1}$$

If the input is $u(t)$ and the output is $x(t)$ we can represent this system with the differential equation

$$\frac{1}{\omega_n^2}\ddot{x}(t) + \frac{2\zeta}{\omega_n}\dot{x}(t) + x(t) = k_{clg}u(t)$$

If we assume $q_1(t) = x(t)$ and $q_2(t) = \dot{x}(t)$, then in terms of these variables we can write a state variable description of the system as

$$\begin{aligned} \frac{d}{dt}\begin{bmatrix} q_1(t) \\ q_2(t) \end{bmatrix} &= \begin{bmatrix} 0 & 1 \\ -\omega_n^2 & -2\zeta\omega_n \end{bmatrix}\begin{bmatrix} q_1(t) \\ q_2(t) \end{bmatrix} + \begin{bmatrix} 0 \\ k_{clg}\omega_n^2 \end{bmatrix} u(t) \\ y(t) &= [1\ 0]\underline{q}(t) \end{aligned}$$

If we are using state variable feedback of the form $u(t) = k_{pf}r(t) - \underline{k}\underline{q}(t)$, then the new form of the system equations is

$$\begin{aligned} \dot{\underline{q}}(t) &= (A - B\underline{k})\underline{q} + Bk_{pf}r(t) \\ y(t) &= C\underline{q}(t) \end{aligned}$$

If the input $r(t)$ is a unit step with amplitude Amp, then in order for $y(t)$ to equal Amp in steady state we must have

$$k_{pf} = -1/C(A - B\underline{k})^{-1}B$$

We need to first **identify** the system:

1. Estimate an initial second order system model using time domain analysis (using either the **log_dec** or **fit** programs).
2. Measure the frequency response (make one measurement at 1Hz, 2Hz, ..., 7 Hz, and at least 4 points near the resonant peak)
3. Use the **fit_bode** command to estimate the gain of the system.
4. Use the **opt_fit_bode** command to fine tune the system model.
5. Determine the closed loop system gain k_{clg}

Designing and Implementing the State variable Controllers

Our general goals for the state variable controllers are as follows:

- produce a position error of less than 0.15
- reach steady state within 1 seconds (the faster the better)
- have as little overshoot as you can manage

Here are some general ideas:

- You need to try and use positive values for k_1 and k_2 ($\underline{k} = [k_1\ k_2]$). The system does not respond very well to negative values. In particular, the steady state values may be off.
- k_1 should be larger than k_2. k_2 is multiplying the derivative, and the estimate of the derivative tends to be noisy.
- Try to keep k_2 less than 0.05 and k_1 less than 1.0.
- A good initial guess is $\underline{k} = [0.1\ 0.01]$
- Be sure that the gains for the second and third carriages are set to zero
- You may need to reset the controller often, such as every time you want to implement a new controller. Click **Utility → Reset Controller**. Only do this **before** you have implemented a controller.
- You may need to rephase the motor. Click **Utility → Rephase Motor**
- Be sure to **Implement** the controller you have designed.
- Try and track a step with an input amplitude of 0.5 to 1 cm.

Direct (Trial and Error) Method

$\boxed{\text{1. Estimate the Gains}}$

Use the program **state_variables_1cart** to guess values for k_1 and k_2. The program will print out the corresponding locations of the closed loop poles and the correct gain k_{pf}, as well as produce a plot of the estimated system response with state variable feedback.

The arguments to this program are:

- the amplitude of the input signal (in cm)
- the system A matrix
- the system B matrix
- the system C matrix
- the feedback gain matrix $\underline{k} = [k_1 \ k_2]$
- the length of time to run the simulation for
- the file name with containing the response of the real system in single quotes. At this point, the filename is just ”

$\boxed{\text{2. Implement the Gains on the ECP System}}$

Once your simulated system has a reasonable response, and probably more importantly, reasonable gains, try running the ECP system with these gains. If the gains are not too large and the system works, save the results to a file. If the system buzzes and doesn't work, go back to step 1 and try again.

$\boxed{\text{3. Comparing the Simulation and the ECP system}}$

Edit the file you saved in part 2 so Matlab can read it. Run the program **state_variables_1cart** again, with the same gains as you used on the system. This time the last argument to the program is the name of the file you saved the response of the system into. You should get a plot containing both the real system and the simulated system. You may want to reduce the final time of the plot so there is not alot of time at steady state showing.

$\boxed{\text{4. Practice Makes Perfect}}$

Try at least three different combinations of gains before you move on the the next method. Be sure to produce a plot for each system, and record the gains and closed loop poles for each system.

Linear Quadratic Regulator Method

1. Estimating the Feedback Gains

Use the Matlab routine **lqr** to estimate the feedback gains k_1 and k_2. The arguments to this routine are

- the A matrix of the system
- the B matrix of the system
- a penalty matrix Q
- a penalty matrix R

The Linear Quadratic Regulator finds the gain $\underline{k}$ to minimize

$$J = \int_0^\infty \left[\underline{x}^T(t)Q\underline{x}(t) + u(t)Ru(t)\right] dt$$

where

$$\begin{aligned}\dot{\underline{x}}(t) &= A\underline{x}(t) + Bu(t) \\ u(t) &= -\underline{k}\underline{x}(t)\end{aligned}$$

In our case Q is a two by two positive definite matrix, and R is a scalar. Since Q is most likely a diagonal matrix, it's easiest to iterate using the following command in Matlab

```
> K = lqr(A,B,diag([q11 q22]),R)
```

where $q11$ and $q22$ are the desired diagonal elements of Q and R is a scalar. In general, as R gets larger (it may have to get very large), the size of the gains goes down. Increasing the value of $q11$ tends to decrease the motion of the cart, while increasing $q22$ tends to limit the velocity of the cart. Iterate on values of Q and R until you think you have something that works.

2. Determining k_{pf}

Again used the program **state_variables_1cart**. The values for $\underline{k}$ have been determined by the lqr routine above. The program will print out the corresponding locations of the closed loop poles and the correct gain k_{pf}, as well as produce a plot of the estimated system response with state variable feedback.

3. Implement the Gains on the ECP System

Once your simulated system has a reasonable response, and probably more importantly, reasonable gains, try running the ECP system with these gains. If the gains are not too large and the system works, save the results to a file. If the system buzzes and doesn't work, go back to step 1 and try again.

4. Comparing the Simulation and the ECP system

Edit the file you saved in part 3 so Matlab can read it. Run the program **state_variables_1cart** again, with the same gains as you used on the system. This time the last argument to the program is the name of the file you saved the response of the system into. You should get a plot containing both the real system and the simulated system. You may want to reduce the final time of the plot so there is not alot of time at steady state showing.

4. Practice Makes Perfect

Try at least three different combinations of gains (corresponding to three different values of Q and R). Be sure to produce a plot for each system, and record the gains and closed loop poles for each system.

Memo

Your memo should compare (briefly) the response of the model and the response of the real system for the different gains you tried. You should have some description of the configuration of the system you were trying to control.

You should include the following items as attachments. Most of these are figures which should have reasonable captions.

- The step response of the time-domain model.
- The initial frequency response of the system.
- The optimized frequency response of the system.
- The data used to determine the closed loop gain.
- The final model of the system.
- The predicted and actual response of the system to each of the different controllers where you guessed the values of $\underline{k}$, and the corresponding closed loop pole locations.
- The predicted and actual response of the system to each of the different controllers where you used the lqr algorithm to determine the values of $\underline{k}$. Also record values of Q and R used and the corresponding closed loop pole locations.