Final Project

For the Final Project, you must develop an output feedback controller and demonstrate regulation of states and outputs for the dynamic system from the Midterm Project. The first step for developing a state feedback or an output feedback controller is to develop a state-space model of the open loop plant, which you completed in the Midterm Project. A model will be provided to you for the final project, or you are welcome to attempt to use your own from the Midterm Project.

Section 1. Before you can develop either a state feedback controller or an output feedback controller, you must first test for controllability and observability. Use the discrete state-space model from the Midterm Project to demonstrate that the open-loop system is completely controllable and completely observable.

(Display your results to the Matlab command window)

Section 2. Use the Matlab **`lqr`** function to design a full state feedback gain matrix for the discrete-time open-loop system. You must design positive definite **`Q`** and **`R`** weighting matrices to meet the following design requirements:

Validation Test Conditions:

- Choose zero initial conditions for your model at t=0; however, you are not allowed to reset the state initial conditions back to zero for any time after t=0
- Hold the excitation constant at **`u1=5`** and **`u2=-5`** for the first 1.0 seconds of the simulation. Note that this first interval of time is required to demonstrate the open-loop response
- At time t = **`1.0`** seconds, apply your LQR full state feedback control law to the model, assuming you have perfect knowledge of your state vector. Do NOT use a state estimator. Do NOT use any built-in simulation functions such as **`lsim`**.
- Terminate your simulation at t = **`8.0`** seconds

Mandatory Performance Requirements:

- The control signal must always be bounded within the saturation limits of $\pm 10V$
- All outputs must always be bounded within the saturation limits of $\pm 10V$
- All outputs must be settled to within $\pm 0.5V$ after t = **`3.0`** seconds

Section 3. Simulate the closed-loop system response of your model with the LQR state feedback control law assuming that you have complete knowledge of the discrete time state vector. Plot the closed-loop time responses up to 8 seconds using the Matlab **`stairs`** function (read the help file). Your single figure must contain two properly formatted and annotated subplots. The upper subplot will include all outputs and the lower subplot will include all control signal. Use a legend to identify specific signals.

(1 Figure only)

Final Project

You will next need to develop an output feedback controller and evaluate its performance on the actual system. To do this, you will need to use the **s19_plant.p** file with a different call structure. The new call structure is given by:

```
[y,u,xhat] = s19_plant(dt_ofc,time);
```

for loop inside the S19_plant loop if we passed in LTI object

where **y** is a $2{\times}N$ matrix of output sensor responses from the actual system, **u** is a $2{\times}N$ matrix of the output control signals, and **xhat** is a $NS{\times}N$ matrix of estimated state responses, where N is the total number of samples of data collected, and NS is the number of states in your open-loop discrete-time model. As in any real physical system, you do not have access to the actual state vector. Using these outputs, you will need to reconstruct the observer output **yhat** for generating the validation comparison plots.

The input variable **dt_ofc** is a discrete-time state-space LTI object that you must generate to represent the complete output feedback controller. This controller must implement the following discrete-time state and output equations:

$$\hat{\mathbf{x}}_{k+1} = \mathbf{A}_{ofc}\hat{\mathbf{x}}_k + \mathbf{B}_{ofc}\begin{bmatrix} \mathbf{u}_k \\ \mathbf{y}_k \end{bmatrix} \qquad \mathbf{u}_k = \mathbf{C}_{ofc}\hat{\mathbf{x}}_k + \mathbf{0}\begin{bmatrix} \mathbf{u}_k \\ \mathbf{y}_k \end{bmatrix}$$

This is the only structure that **s19_plant** will accept so you will need to determine the appropriate **A**, **B**, and **C** matrices for this state-space object (**D** is the zero matrix as shown above). The second input argument, **time**, allows you to define the total duration of the data set as well as the start time for control. Note that you will NOT specify the number of samples N directly! The observer will always begin at t=0, and the observer will always start with zero initial conditions. The dynamic plant will have a random (unknown) initial condition every time you call the function.

To specify both the final time (seconds) and a controller start time (seconds), select:

```
tstart = 1.0;
tfinal = 8.0;
time = [tstart, tfinal];
```

Section 4. For any practical implementation of full-state feedback control, you must estimate the state vector. Design the Kalman state feedback gains using the following Matlab function call:

```
[sys,K] = kalman(my_ss_model,QN,RN,NN,'current');
```

where: **QN** is a 2x2 process noise covariance matrix, **RN** is a 2x2 measurement noise covariance matrix, and **NN** is a 2x2 cross covariance matrix between the process noise and sensor noise. You can estimate the measurement noise covariance matrix using the **COV** function in Matlab (read the help to learn how). The cross covariance noise matrix **NN** can be assumed to be zero. As is common in practice, the process noise covariance matrix is the most difficult to determine. It certainly cannot be chosen to be zero, but it should be positive definite. You may choose **QN = alpha*eye(2)** where **alpha** is a "tuning" gain that you must select by trial and error.

The **KALMAN** function outputs an LTI object **sys** as well as the Kalman feedback gains **K**. You will only use the Kalman feedback gain **K**!

(Display your final Kalman feedback gains to the Matlab command window)

Section 5. Construct the discrete-time state-space output feedback controller LTI object as defined above, then generate the closed-loop response of your output feedback controller by inputting the **dt_ofc** object to the **s19_plant.p** file with the specified time limits.

Plot the closed-loop time responses up to 8.0 seconds using the Matlab **stairs** function. Your single figure must contain three properly formatted and annotated subplots. The upper subplot will include the actual y_1 output, the estimated y_1 output, and the error between the actual and estimated y_1 outputs. The middle subplot will include the actual y_2 output, the estimated y_2 output, and the error between the actual and estimated y_2 outputs. The lower subplot will include the control signals. Your solution must meet the mandatory performance requirements defined in Section 3. This may require you to run the simulation multiple times to obtain "favorable" initial conditions.

(1 Figure only)

RULES:

- You must submit one PDF document that displays your results with clearly labeled sections corresponding to the numbered sections above
- You must also submit one Matlab script (2 document submissions total)
- Your total submission must be less than 2 Mb
- You must not upload the s19_plant.p file that was provided to you
- A portion of your grade will be based on the formatting of all of your figures. All figures must be presented as if they were being submitted to a Journal, so they must be "publication quality"
- Figures are NOT allowed to have shaded backgrounds
- Figures can not be larger than one 8.5"x11" page
- Figures must not be rotated 90 degrees on a page
- All text on all figures must be clearly readable (11 point font minimum)
- All signals and axes within a figure must be properly annotated (legend, axis labels, title, colors, and markers where appropriate)

Final Project

- If a figure contains more than one time-domain plot, each set of axes must have the same x-axis limits for direct comparison
- If a figure contains more than one data set, you must include a legend that clearly identifies each data set with an appropriate label