Active Convoy, Inc.
Blacksburg, Virginia 24060
540.231.7385
E-mail: scsouth@vt.edu

Date: November 13, 2018

To: ME 5554 / AOE 5754 / ECE 5754 Students

From: Dr. Steve Southward

Subject: Phase II Development Program (i.e. your Final Project)

Background

The final project will build off of your result from the Midterm Project. Before starting the final project, you must run the Midterm Project and save your open-loop state-space LTI model in a .mat file that can be loaded by your Final Project script. Do NOT reproduce the code from your Midterm Project to generate the open-loop model in the Final Project! Make sure you use a corrected version of the open-loop model if you made any mistake in its formulation from the Midterm Project.

Problem 1. [5 pts.] Load your open-loop LTI state-space model from the Midterm Project. Use the LQR function in Matlab to design a full state feedback gain matrix. The LQR function requires a state weighting matrix $\mathbf{Q}$ and an effort weighting matrix $\mathbf{R}$ to be defined. For the design of $\mathbf{Q}$, you must use “output weighting” instead of “state weighting”. To accomplish this, you must use the following identities to define the equivalent state weighting matrix:

$$\mathbf{y} = \mathbf{C}_{OL}\mathbf{x} \qquad \Rightarrow \qquad \underbrace{\mathbf{y}^T\mathbf{Q}^*\mathbf{y}}_{\substack{\text{output}\\ \text{weighting}}} = \left[\mathbf{x}^T\mathbf{C}_{OL}^T\right]\mathbf{Q}^*\left[\mathbf{C}_{OL}\mathbf{x}\right] = \underbrace{\mathbf{x}^T\mathbf{Q}\mathbf{x}}_{\substack{\text{state}\\ \text{weighting}}}$$

Choose individual weights for each of the outputs (diagonal elements of $\mathbf{Q}^*$) and each of the control signals (diagonal elements of $\mathbf{R}$) to meet the original design requirements from the Midterm Project. In this design, you must attempt to use the maximum allowed control authority on each torque signal while minimizing the settling times for all three outputs.

Once you have designed LQR optimal state feedback gains, construct a closed-loop LTI state-space model.

You must generate only ONE final design!

Problem 2. [5 pts.] Use the INITIAL function in Matlab to simulate 10 seconds of the initial value response for the closed-loop system using the LQR optimal control gains from Problem 1. Use the exact same initial condition from the Midterm Project (not the one in the project document).

Compute the settling times for all three outputs and display this information (either in Matlab, on the plot, or both). You must use one of the built-in Matlab tools such as LSIMINFO or STEPINFO to compute the 5% settling times. All three closed-loop settling times should be below 8 seconds.

Plot a single figure window where all outputs are on a single axis (subplot) at the top of the figure, and all control signals are on a single axis (subplot) at the bottom of the figure. Properly annotate each axes with clearly defined axis labels, grid lines, and legends.

Problem 3. [2 pts.] Show that the open-loop system is Completely Observable. Also determine (and clearly document) which subsets of outputs results in a Completely Observable system. Justify your results.

Problem 4. [3 pts.] Design a Luenberger observer such that the output feedback control system meets the original performance requirements when the closed-loop system is simulated using the same initial condition vector. To demonstrate that your design meets the requirements, you will need to complete Problem 5 below.

Use unique markers to generate a plot of the open-loop poles, the closed-loop LQR state-feedback poles, and the observer poles on a pole diagram (real vs. imag). Use the SGRID function with your plot to draw lines of constant damping and lines of constant frequency. You must also use the AXIS EQUAL command to insure that the aspect ratio is correct between the real and imaginary axes. Properly annotate each axes with clearly defined axis labels, and legends.

Problem 5. [10 pts.] For this problem, you will simulate the complete output feedback controller; however, this simulation must be performed in two stages. For each stage, you will need to construct a separate closed-loop state-space LTI model.

The first stage LTI model will be an augmented system that includes the open loop plant and the observer with no state feedback control. The second stage LTI model will be the same augmented system, but it will include state feedback control. Both LTI models should output $\mathbf{y}$, $\hat{\mathbf{y}}$, and the control vector $\mathbf{u}$ for plotting.

Construct an initial condition vector that uses the IC's from the midterm project for the initial plant states augmented with a zero vector for the initial observer states.

Use the INITIAL function in Matlab to simulate 1 second of the initial value response for the observer-only system. You will need to output y, t, and x from the INITIAL function. Use the state at t=1 second (final state of the stage 1 simulation) as the initial state vector for the second stage simulation.

Use the INITIAL function again to simulate 9 additional seconds of complete output feedback control. You will need to append the solutions from both stages to generate outputs and control signals that can be plotted.

Compute the settling times for all three outputs and display this information (either in Matlab, on the plot, or both). You must use one of the built-in Matlab tools such as LSIMINFO or STEPINFO to compute the 5% settling times. All three closed-loop settling times should be below 8 seconds.

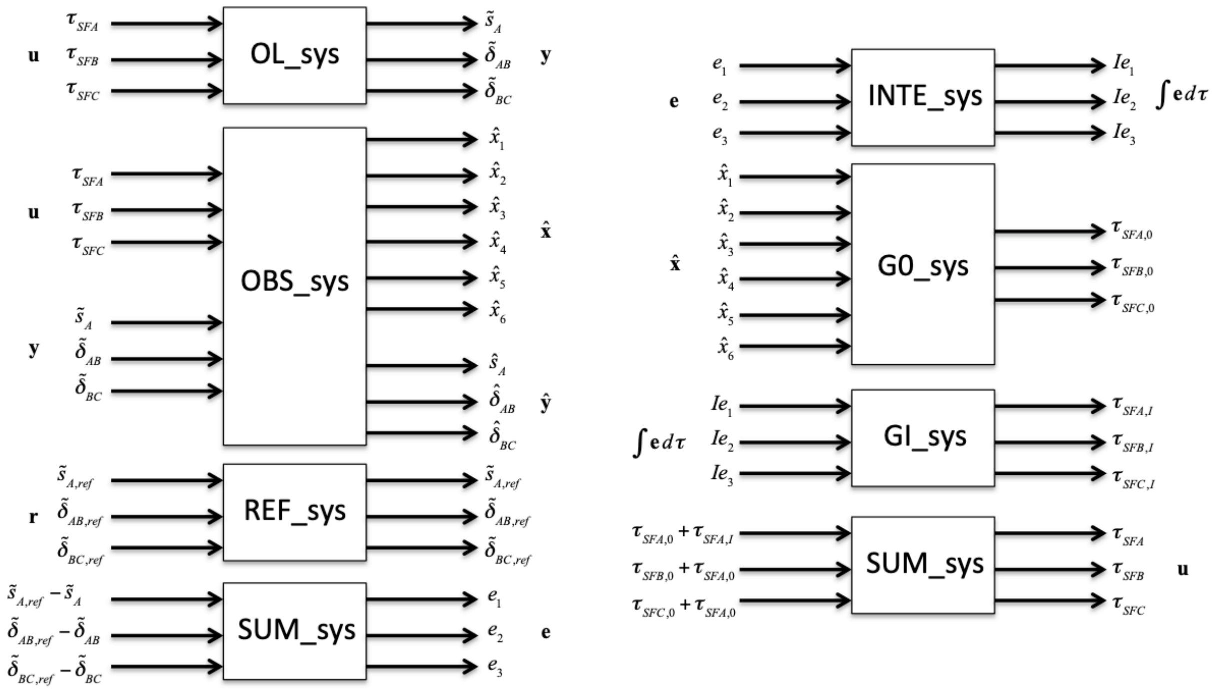
Plot a single figure window where all outputs are on a single axis (subplot) at the top of the figure, and all control signals are on a single axis (subplot) at the bottom of the figure. Properly annotate each axes with clearly defined axis labels, grid lines, and legends. When plotting the outputs, you must plot both the actual outputs $\mathbf{y}$ and the estimated outputs $\hat{\mathbf{y}}$. You must clearly identify the signals since this axis will have six signals plotted on it. Your plot should show that the estimated outputs converge to the actual outputs within the first second of the simulation. Your plot should also show that none of the three control signals exceed the max torque limit.

Problem 6. [10 pts.] Use the LQI function in Matlab to design a gain matrix for state feedback and integral control of the reference inputs. Similar to Problem 1, you will need to choose individual weights for each of the outputs (diagonal elements of $\mathbf{Q}^*$) and each of the control signals (diagonal elements of $\mathbf{R}$), but also each of the individual error terms associated with the integrators.

Once you have designed LQI optimal state feedback gains, separate out the state feedback gains from the integral feedback gains.

Problem 7. [15 pts.] Before simulating the closed-loop response, you must construct a closed-loop LTI object; however, this time you must use the APPEND and CONNECT functions in Matlab to construct the complete augmented output feedback control system. Note that this will NOT be a two-stage design like in Problem 5!

The figure below is provided as a helpful guide for constructing the subsystem LTI objects and setting up the APPEND and CONNECT functions. Hint: Choose whatever order you would like to APPEND these subsystems together, then sequentially number the inputs and sequentially number the outputs.



You will need to construct LTI objects for each of the subsystems above. You already have the OL_sys LTI object. Note that the OBS_sys observer LTI model takes both u and y as inputs, but it must output the estimated state vector (for input to the G0_sys block) and the estimated outputs (for plotting).

To obtain the remaining LTI objects, you can use the following:

```
REF_sys = tf(eye(3));     % Static gain identity matrix
SUM_sys = tf(eye(3));     % Static gain identity matrix
G0_sys = tf(-G0);         % Static gain G0 matrix (negated)
GI_sys = tf(-GI);         % Static gain GI matrix (negated)
```

The integrator LTI object is a diagonal matrix of integrators (1/s). This is compactly constructed by converting the 3x3 identity matrix into a cell array to serve as the matrix of numerator coefficients, and then use [1 0] to represent the polynomial 1*s+0 for all denominators:

```
INTE_sys = tf(num2cell(eye(3)),[1 0]);
```

The only “inputs” to the output feedback control system are the three reference inputs. For the “outputs”, you must choose $\mathbf{y}$, $\hat{\mathbf{y}}$, and $\mathbf{u}$ as outputs. Define a connection matrix that properly connects all of the blocks together (Hint: read the Matlab help for the CONNECT function starting with “Index-based interconnection”).

You may find it valuable to define all of the inputs and outputs of each subsystem LTI object with labels that make physical sense in order to check your final CONNECTed system.

Problem 8. [10 pts.] For this problem, you will simulate the complete output feedback controller using the LTI model from Problem 7.

Construct an initial condition vector that uses the IC's from the midterm project for the initial plant states, augmented with a zero vector for the initial observer states, augmented with a zero vector for the initial integrator states.

Construct a time vector using:

```
Tfinal = 20;    % final sim time (seconds)
Ts = 0.1;       % data sample period (seconds)
T = ts * [0:round(Tfinal/Ts)];      % time vector (seconds)
```

Construct a reference input matrix $\mathbf{r}$ where each reference input is constant over the entire time vector. You will need to construct a matrix for $\mathbf{r}$ that has three columns and one row for each time step.

$$\tilde{s}_{A,ref} = 0.5 \qquad \tilde{\delta}_{AB,ref} = 0.0 \qquad \tilde{\delta}_{BC,ref} = 1.0$$

Use the LSIM function in Matlab to simulate the total response for the output feedback control system due to initial conditions and reference inputs. You will need to output y, and t from the LSIM function.

Plot a single figure window where all outputs are on a single axis (subplot) at the top of the figure, and all control signals are on a single axis (subplot) at the bottom of the figure. Properly annotate each axes with clearly defined axis labels, grid lines, and legends. When plotting the outputs, you must plot the actual outputs $\mathbf{y}$, the estimated outputs $\hat{\mathbf{y}}$, and the desired references $\mathbf{r}$ on the same axes. You must clearly identify the signals since this axis will have nine signals plotted on it. Your plot should show:

1) all estimated outputs converge to the corresponding actual outputs within the first second of the simulation
2) all three actual outputs converge to the desired references within about 10 seconds with zero steady-state error

You are NOT required to meet the max torque requirement for this problem because we are not allowing the observer time to converge before turning control on.