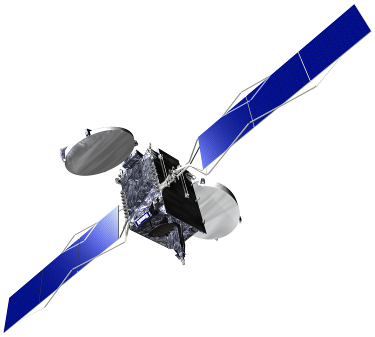
Satellite Recovery Corporation
123 Space Junk Road
Remote Control, Virginia 24520
434/766-6794



Date: April 18, 2013

To: ME 5506 Students

From: Dr. Steve Southward

Subject: **Final Project**

Final Project: Output Feedback Controller Task

In order to begin the Final Project from a common starting point, you must use the 6th order discrete-time state-space model `dt_sys` provided in the file dtplant.mat. Given this discrete-time state-space model of the open-loop MIMO satellite, you can begin Task 2, which is to develop an output feedback controller.

In HW4, you demonstrated that even if the frequency response of a discrete-time model matches the frequency response of a continuous-time plant, this does not guarantee that the time responses will also match. In the first four subtasks below, you will evaluate the time responses of the discrete-time state-space model relative to the actual satellite.

Task 2a. [6 pts] Apply a maximum amplitude single-sample duration pulse to u_1 only, and measure the pulse responses. For this measurement, you must repeatedly call the **`satellite_uplink.p`** function and average the time-domain results to reduce the effects of sensor noise. Determine the duration of sampling to approximately capture the 5% settling time, and also determine a non-excessive number of averages to reasonably reduce the sensor noise. Explain how many averages you chose and why you consider this to be reasonable.

Task 2b. [6 pts] Apply a maximum amplitude single-sample duration pulse to u_2 only, and measure the pulse responses. For this measurement, you must repeatedly call the **`satellite_uplink.p`** function and average the time-domain results to reduce the effects of sensor noise. Use the same duration of sampling and number of averages that you used in Task 2a.

Task 2c. [6 pts] Apply the same pulse inputs, one control input at a time, to the supplied discrete-time state-space model `dt_sys`. No averaging will be necessary for this task because there is no noise, but you must use the same duration of sampling.

Task 2d. [6 pts] Compare the pulse responses on a Matlab plot. On a single figure, create a 3x2 grid of subplots, one subplot for each actuator and sensor pair. In each axes, plot the averaged time domain response and the model response. (1 Figure)

The typical process for developing a state feedback or an output feedback controller is to first develop a simulation study to evaluate the effectiveness of your design. In this simulation study you use your model (`dt_sys`) to represent the actual plant. Once your design has proven successful in this simulation, you can then apply it to the actual plant.

Task 2e. [6 pts] Before you can develop either a state feedback controller or an output feedback controller, you must test for controllability and observability. Use the discrete state-space model `dt_sys` to demonstrate that the open-loop system is completely controllable and completely observable. Display your results to the Matlab command window.

Task 2f. [6 pts] The eigenvalues of the discrete-time state-space open-loop model `dt_sys` indicate that the satellite has two real poles and two pairs of complex conjugate poles representing two flexible modes. Design a set of closed-loop z-domain poles that satisfies the following requirements:

- Do not change the flexible mode natural frequencies
- Increase the damping ratios of both flexible modes to 0.5
- Move the rigid body poles into the LHP such that they have the same magnitude as the first (i.e. lowest frequency) flexible mode

Task 2g. [6 pts] Determine full-state feedback control gains to place the poles designed in Task 2f. Display the full-state feedback control gain matrix to the Matlab command window.

Task 2h. [6 pts] Use the discrete-time state-space model `dt_sys` to simulate the closed-loop system response assuming that you have complete state knowledge. For this simulation, you must choose an initial control signal of `u=[-2.0;3.0]`; and hold this signal constant for the first 1.5 seconds. At `t=1.5` seconds, you must enable full-state feedback until a final time of `tfinal=5.0` seconds. Your discrete-time state-space model should have zero initial conditions. You will **not** be able to use `lsim` for this task, so you will need to construct your own loop through each time step.

Task 2i. [6 pts] On a single figure, create a 2x1 grid of subplots to plot the results of your full-state feedback controller performance from Task 2h. The first subplot should contain all three sensor responses, and the second subplot should contain both control input signals. (1 Figure)

For the final set of subtasks, you will need to develop an output feedback controller and evaluate its performance on the actual satellite. To do this, you will need to use the existing **`satellite_uplink.p`** file with a different call structure. The new call structure is given by:

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

where **`y`** is a $3 \times N$ matrix of sensor responses from the actual satellite, **`u`** is a $2 \times N$ matrix of control signals, and **`xhat`** is a $6 \times N$ matrix of estimated state responses, and N is the total number of samples of data collected. 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`** yourself.

The satellite input variable **`dt_ofc`** is a discrete-time state-space LTI object that you must create 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 the satellite will accept so you will need to determine the appropriate $\mathbf{A}$, $\mathbf{B}$, $\mathbf{C}$, and $\mathbf{D}$ matrices for this state-space object ($\mathbf{D}$ is easy and is already given to you). The second input argument, **`time`**, is optional and 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 satellite itself will have a random (unknown) set of initial conditions as it orbits the earth.

To specify just the final time (seconds), select:

```
time = [tfinal];
```

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

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

Task 2j. [6 pts] Design closed-loop z-domain observer poles by first transforming the closed-loop z-domain state-feedback controller poles to the s-domain, then select an appropriate set of observer poles, and finally transform the s-domain observer poles back to the z-domain. Determine the full state observer feedback control gains to place the observer poles as designed. Display your final observer feedback control gain matrix to the Matlab command window.

Task 2k. [6 pts] Construct a discrete-time state-space LTI object (`dt_ofc`) to implement your complete output feedback controller following the structure requirements defined above. Display your output feedback controller object to the Matlab command window.

Task 2l. [6 pts] Generate the closed-loop response of your output feedback controller by uploading the `dt_ofc` object to the `satellite_uplink.p` file. For this evaluation, you must turn on the controller at t=1.5 seconds and collect data up to t=5 seconds. On a single figure, create a 4x1 grid of subplots to plot the results of your output feedback controller performance. The first three subplots should contain each of the three sensor responses, and fourth subplot should contain both control input signals. In each of the first three subplots, you must plot the actual satellite response y_i, the estimated response $\hat{y}_i$, and the error $\left(y_i - \hat{y}_i\right)$. (1 Figure)

RULES:

- You must submit **only one m-file** that must run on Matlab R2012b without modification, and only using the standard student-distribution toolboxes
- Do not submit the `satellite_uplink.p` file
- Your total submission must be less than 1 Mb
- Clearly label sections of your Matlab code with respect to the Task number (use the supplied m-file template if you want)
- Submit your answers to the questions as Matlab *comment lines* in your m-file
- Do not specify the figure *backgroundcolor* property
- Do not specify the figure *position* property
- Do not specify the *defaultcolororder* property, although you may select specific colors for the curves in your plots
- All signals and axes within a figure must be properly annotated (legend, axis labels, title, colors, and markers where appropriate)
- On axes where outputs y are plotted, set the y-axis limits to [-2.5, 2.5]. The only exception is the pulse response plots, which should be auto-scaled.
- On axes where controls u are plotted, set the y-axis limits to [-10, 10]
- Answer all parts of every question (don't forget parts of multi-part questions)