GE420 Laboratory Assignment 11
Control of the Furuta Pendulum Using a Full Order Observer

Goals for this Lab Assignment:

1. Implement a full order observer to estimate the velocity states of the Furuta pendulum

SYS/BIOS Objects Used:

- Any or your choosing

Matlab Functions Used:

- Place.

Prelab:

Read through the entire lab assignment.

Controller Design and Simulation:

In this prelab you will design a full order observer to estimate the velocity states of the Furuta pendulum. Using this observer along with the controller designed in Lab 10, simulate your controller and observer with the given non-linear model of the Furuta pendulum. You must complete the pre-lab and demonstrate it to your TA before proceeding to the implementation exercise.

This lab is designed to give you an introduction to designing and implementing an observer to estimate the unmeasured states of your control system. The Furuta pendulum experiment only has two measured states θ_1 and θ_2. So instead of using our velocity approximations from the previous labs, you will design a full order observer to estimate the velocities given the linear equations of motion for the Furuta about its balancing point. Again the only design specification is to stabilize the Furuta pendulum in the inverted position. You can use the same full state feedback controller that you designed in Lab 10.

The system equations remain the same as in Lab 10 with the exception of C, which now is

$$C = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

indicating that only θ_1 and θ_2 are measured. Using this along with your discrete A and B matrices from Lab 10, use pole-placement to design a full order observer with poles much faster then the closed loop poles of (A-B*K). You will find your observer works best if all the poles are placed to be faster than 0.5 in the z-domain.

The simulation for this experiment should build off the one developed for Lab 10. The simplest way to implement an observer is to use a **Discrete State-Space** block. The block should be set up such that

A_{obs}= [A_d-L*C]

B_{obs}= [B_d L]

C_{obs}= 4x4 Identity matrix (Don't get confused with this C. This is just telling the **Discrete State-Space** block to output all four of the estimated states. This is a different C then the one you used to design the observer L gain).

D_{obs}= 4x3 zeros matrix (to match the dimensions of B_{obs} and C_{obs})

The three inputs to this state space block are the **saturated** control effort, **u**, δx_1, and δx_3. The order of these three inputs is defined by the order of B and L in B_{obs}. Since a state space block in Simulink can only have one input, those two signals will have to be multiplexed into one signal. Note- **Mux** is an abbreviation for a multiplexer. The **Mux** block is found in the **Signal Routing** section of the Simulink library. The output of the state space block is the estimate of the four delta states of the linearized system. **u** is then calculated by multiplying **–K** by these state estimates. A sample Simulink model can be found at N:\labs\GE420\matlab\furutaobs.mdl and is shown in Figure 1.

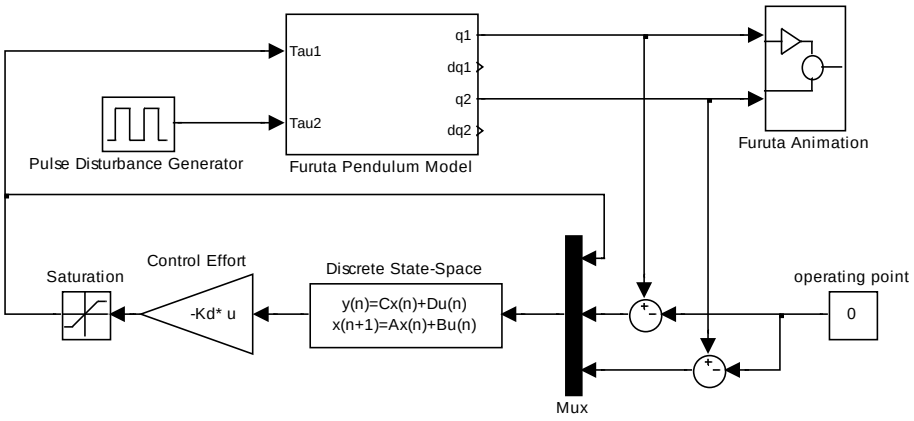

Figure 1: Sample Simulink Model for Pendubot Observer

Laboratory Exercise

Implement the observer and full state feedback controller on the C6713DSK. Use Lab 10's code as a starting point because you will be asked to plot some states in real-time. Remember that the state equations you need to implement in your control code are:

$$\delta\hat{x}_{k+1} = A_d\delta\hat{x}_k + B_d u_k + L(y_k - C\delta\hat{x}_k)$$
$$u_k = -K\delta\hat{x}_k$$

Also leave the equations that you used to approximated velocity in Lab 10 in your code. You will use these approximations as the "actual" velocities to compare to your observed velocity states.

To help you enter your observer matrices in your C code we have written a small M-file, **matrixtoCformat,** to format the data into an array format and then you can cut and paste the arrays into your code.

Lab Check Off:

1. Demo your working Simulink simulation using the non-linear Furuta model.
2. Demo your working full order observer controller.
3. Demo a plot of x4 estimated verses x4 actual (lab 10's approximation of velocity). Notice the convergence of the estimated state when a tap is applied to the link and when a step change is given to link 1.