HOMEWORK #7
DUE 26-FEB-2014

Background: Initially, this is the point in the class where I would send you into the lab to learn how to control a real mechanical experiment (inverted pendulum, three disk set, caged fan, etc.). The problem here is that the hardware has gotten so good, that essentially these now all boil down to playing with Simulink and MATLAB. If I put you on bad hardware, then you spend most of your time debugging the bad hardware (something which I am sure you are all going to do later in your careers anyway). So, instead, I've got a simulink model for you to play with. This is a full, realistic control design exercise—it is open ended, the more you put in, the more you get out.

Attitude Stabilization of a 1-D satellite: Very often in satellite/space ship design, you wind up putting the big heavy things (engines, etc) on one end, and the sensitive sensors on the other end, and because you don't have to support the weight (microgravity environment), you use a slender flexible truss to move them apart. We model the torsion stages as three inertias connected by two torsional springs. To make matters more difficult, the thrusters that can torque the satellite are connected to the aft stage, not to the front where the sensors are. You do get a star tracker on each stage which gives you very high quality rotational attitude information on the fore and aft stages (but not in the middle).

Collocated Control: The rotational dynamics of the entire satellite system, aft thrusters to aft angle, can be represented by the following transfer function (this is the easy one):

$$G_{AFT}(s) = \frac{\Theta_{AFT}(s)}{U(s)} = \frac{K_1(s^2 + 2\zeta_{z1}\omega_{z1}s + \omega_{z1}^2)(s^2 + 2\zeta_{z2}\omega_{z2}s + \omega_{z2}^2)}{s(s + p)(s^2 + 2\zeta_{p1}\omega_{p1}s + \omega_{p1}^2)(s^2 + 2\zeta_{p2}\omega_{p2}s + \omega_{p2}^2)}$$

Where:

	K_1 = 30.20	p = 0.5 rad/s
	ζ_{z1} = 0.026	ω_{z1} = 10.68 rad/s
	ζ_{z2} = 0.018	ω_{z2} = 40.83 rad/s
	ζ_{p1} = 0.020	ω_{p1} = 14.95 rad/s
	ζ_{p2} = 0.018	ω_{p2} = 41.97 rad/s

Now, onto the design of the controller:

1. Design a lead controller, $D_1(s)$, which gives a closed loop bandwidth of 2π rad/sec and a phase margin of at least 55°

2. Sketch a root locus plot vs. the lead gain, K_o, indicate where your closed loop poles are.

3. Sketch your bode plot of the system, indicating cross-over frequency, and phase margins.

4. Go back and check your hand plots using MATLAB (and turn both in).

5. Plot step and impulse responses of the closed loop system (include both the output and the control effort on both plots—that is plot *u* and *ϑ*).

6. Download the simulink model *DiscoveryAft.mdl* from the website, and implement your controller (simply replace the $D_1(s)$ block with your own transfer function). When you simulate the system, the variables ucon1, tcon1, and ycon1 (input, time, and output) will be populated in your workspace as arrays. Simulate your controller on the plant and verify that it matches the step response you found in (5).

7. Convert your compensator, $D_1(s)$, to its discrete equivalent, $D_1(z)$, using a Tustin transformation using a sample rate of 25Hz.

8. Download the simulink model *DisoveryAftDiscrete.mdl* from the website, and implement your digital controller (simply replace $D_1(z)$ with your own transfer function). Compare your time histories for $D_1(s)$ and $D_1(z)$ to see how much it hurt to go digital.

Non-Collocated Control: When you did the control above, the output sensor and the actuator were in the same place. This is called *collocated* control. You should have observed that the high frequency poles tended to have zeros close by that caused the poles to move in favorable (stabilizing) directions when you cranked up the gain. This is typically the case for collocated control. What happens when the sensor and the actuator are not in the same place? This is non-collocated control, and it gets harder. The rotational dynamics of the entire satellite system, aft thrusters to fore angle, can be represented by the following transfer function (this is the hard one):

$$G_{FORE}(s) = \frac{\Theta_{FORE}(s)}{U(s)} = \frac{K_2}{s(s+p)(s^2 + 2\zeta_{p1}\omega_{p1}s + \omega_{p1}^2)(s^2 + 2\zeta_{p2}\omega_{p2}s + \omega_{p2}^2)}$$

Where:
K_2 = 5,746,500
ζ_{p1} = 0.020
ζ_{p2} = 0.018

p = 0.5 rad/s
ω_{p1} = 14.95 rad/s
ω_{p2} = 41.97 rad/s

Now, onto the design of the new controller:

9. Design a simple lead controller, $D_2(s)$, for the new plant, $G_{FORE}(s)$. How high a bandwidth can you get with this kind of compensator? Remember that there is some uncertainty in both the frequency and damping of the poles. Use *DiscoveryFore.mdl* to simulate the system.

10. Convert your compensator, $D_2(s)$ to its discrete equivalent, $D_2(z)$, using the Tustin transformation, again with a 25Hz sample rate. Simulate it using *DiscoveryForeDiscrete.mdl*.