ESE 406/505 & MEAM 513 - SPRING 2011
HOMEWORK #9
DUE 13-Apr-2011 (Monday, 18-Apr-2011 with late pass)

1. Work problem 4.34(c) and 4.35(c) in the textbook.

   Answers: 4.34 $D_3(z) = \dfrac{4.2z - 3.8}{z - 0.6}$ 4.35 $u(k) = 0.6u(k-1) + 4.2e(k) - 3.8e(k-1)$

2. Complete Problem 8.3, which is a fun way to practice working with Z transforms. You are on your own for this one.

3. This problem is an adaptation of problem 8.8 in the text. Read that problem first. Notice that you clearly don't need integral feedback at all to meet the design requirements. It turns out that you can almost exactly meet the specs in the original problem with a purely proportional gain of unity. (You should be sure you can verify that this is true). Let's make the problem more interesting. We will replace the given plant, $G_P(s) = \dfrac{1}{s(s+1)}$, with a 3rd-order plant,

   $$G_P(s) = \frac{1}{s(s+1)(s+5)}.$$

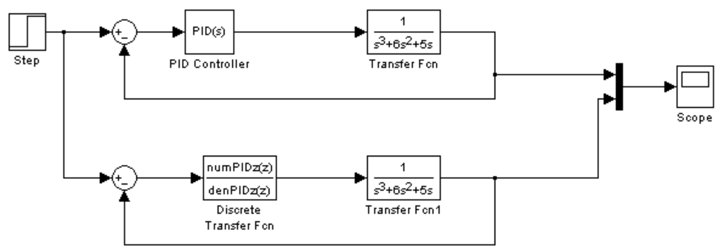
   a. Make a bode plot of $G_P(s)$ and find the frequency at which the phase crosses -180°. Look at the gain of $G_P(s)$ at this frequency to figure out the proportional feedback gain, K_u, required to reach neutral stability, and the expected period, P_u, of the oscillations. Use Simulink to confirm that this gain does indeed result in neutral stability. (Answer: K_u=29.85, P_u=2.8sec)

   b. Now use the Ziegler-Nichols "ultimate sensitivity method", summarized in Table 4.3, to set the gains of a PID controller. Use simulink to examine the closed-loop response to a step input. (Answer: K_P=17.9, K_D=6.3, K_I=12.77. The closed-loop response is somewhat lightly damped.).

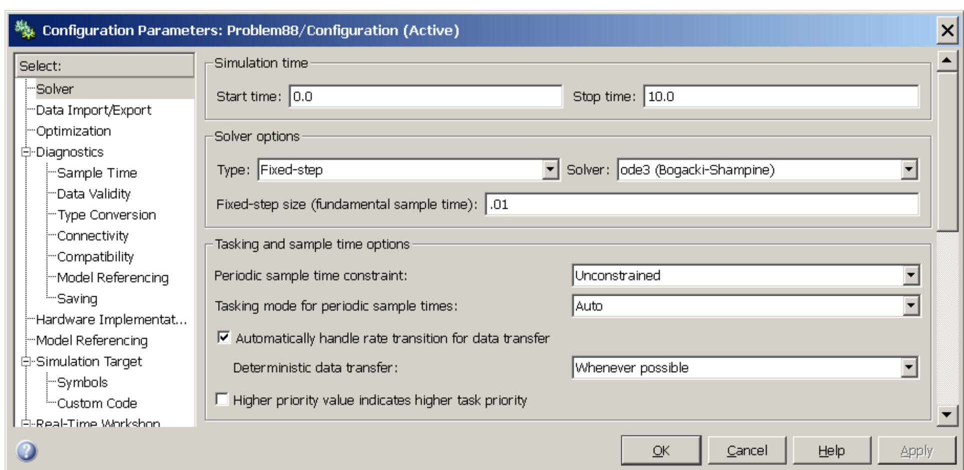
   c. Now make a bode plot of the LOOP transfer function, which is the product of your PID controller and $G_P(s)$. On this plot, find the gain crossover frequency—the frequency where the gain is 0db. Determine how far the phase is above -180° at this frequency. How much pure time delay would have to be added to the loop to make the system unstable? (Answer: 21.9 degrees & 0.22 seconds.) The Ziegler-Nichols compensator is not very good, because the phase is too close to -180° at the crossover frequency. We can do much better if we actually design for good "phase margin", which we will learn to do when we return to Chapter 6 in coming lectures.

   d. Now let's convert to discrete-time. Let's start with a sample time of 0.04 seconds. Use the "Tustin" method to convert the continuous-time PID compensator into a discrete-time compensator (you may use matlab's C2D function to get the discrete-time filter coefficients). What is the discrete-time transfer function of your digital compensator? Use Simulink to compare the closed-loop step responses of the total system with both the analog and discrete compensators.

<u>Hints</u>: Use a discrete-time transfer function from the "discrete" library to put your digital PID compensator in your simulink model. You might use a model that looks like this:



In order to get Simulink to automatically handle the mix of discrete-time and continuous-time blocks in the same model, you have to choose the "Automatically handle rate transition" option under the Configuration Parameters dialog. You should also be sure to set a fixed time step such that the sample rate is an integral multiple of the time step:



(Answers: $\dfrac{332.1z^2 - 627.5z + 296.3}{z^2 - 1}$ & discrete-time response very closely matches analog design.)

e. Finally, increase the sample time to 0.2 seconds (very close to the predicted amount of delay that will result in neutral stability) and again compare step responses of the analog and discrete-time implementations. (Answers: $\dfrac{81.98z^2 - 123z + 46.16}{z^2 - 1}$ & discrete-time response is now unstable.)