=EEE 481 Homework 1

Problem 1.

Suppose that we measure a signal 0-5V with a 16-bit A/D.

1. What is the maximum error?
2. What is the percent maximum error for V in [3-5](V)?
3. Assuming that the clock used in the A/D conversion is 1MHz, find the maximum conversion time for a successive approximation converter.
4. Suppose that the measurement is corrupted by 4-bit white noise, uniformly distributed. What is the variance of the noise in V and what would be the variance if we perform averaging of N values, where N = 8,16,32.

1. For min/max values at the end of the range, the A/D will have 2^n distinct values dividing the interval. Thus, the resolution is $\frac{5-0}{2^n}$ =0.076 mV. For a truncating A/D, the maximum error is the same. 0.076 mV, and for a rounding A/D the maximum error is ½ LSB=0.3815 mV.
2. The relative error (%), in the worst case, is the maximum error divided by the value of the voltage. This achieves its maximum at the lower end of the interval.

   For a truncating A/D, % error=$\frac{0.076*10^{-3}*100}{3}$=0.00254 %

   For a rounding A/D, % error=$\frac{0.3815*10^{-3}*100}{3}$=0.01271 %
3. A successive approximation converter will use roughly 1 clock cycle per bit (since DAC's are much faster than that) and with a bisection algorithm that requires n-steps, so the conversion time is 16 μsec.
4. Considering continuous system and then sampling out, variance of noise= $\frac{(\frac{5}{2^{12}})^2}{3}$ =4.967e-7 V. Because of independence, the outputs when averaged is given by Var_avg=$\frac{4.967e-7}{N}$. N=8, Var_avg=6.208e-8 V, N=16, Var_avg=3.104e-8 V, N=32, Var_avg=1.552e-8 V

Problem 2.

Consider the system $y(k+1) - \frac{1}{32}y(k) = \frac{1}{4}x(k) + \frac{1}{2}x(k-1)$, where the multiplications and the addition are quantized to 1/128. Use simulation to assess the mean, worst-case amplitude, and variance of the error due to quantization (compared to non-quantized operation). Apply various inputs x(k), e.g., random, sinusoid, quantized to 1/128. Compare your results with the theoretical bounds computed from the corresponding transfer functions. (Noise q maps through a DT system H to output y.)

1. $\mathrm{mean(y[k])\ =\ H(1)mean(q[k])}$
2. $\mathrm{max|y(k)|} \le (\sum|\mathrm{h(k)}|)\mathrm{max|q(k)|}, (\mathrm{h(k)\ =\ Z^{-1}\{H(z)\}})$
3. $\mathrm{var\{y(k)\}} \le \left|\mathrm{H}\left(\mathrm{e}^{\mathrm{j}\Omega}\right)\right|_2^2 \mathrm{var\{q(k)\}}$
4. $\mathrm{var\{y\}} \simeq \mathrm{RMS^2\{y\}} \le \mathrm{max^2}\, |\mathrm{H(e^{j\Omega})}|\, \mathrm{RMS^2\{q\}}$

For a round-off quantization, whose mean is 0 LSB, max(|n|) = ½ LSB = 0.005 and var = 1/3(½ LSB)^2 =8.33e-6. RMS(n) = (var{n})$^{1/2}$ = 0.0029.

Realizing the transfer function in terms of delays of the output and input (as shown in the figure below) There are 4 quantization blocks, each one contributing ½ LSB uncorrelated noise to the same summation node. The transfer function from each one is $G(z) = \frac{1}{z-\frac{1}{8}}$, for which, $G(1) = 1.143, \sum|g(k)| =$

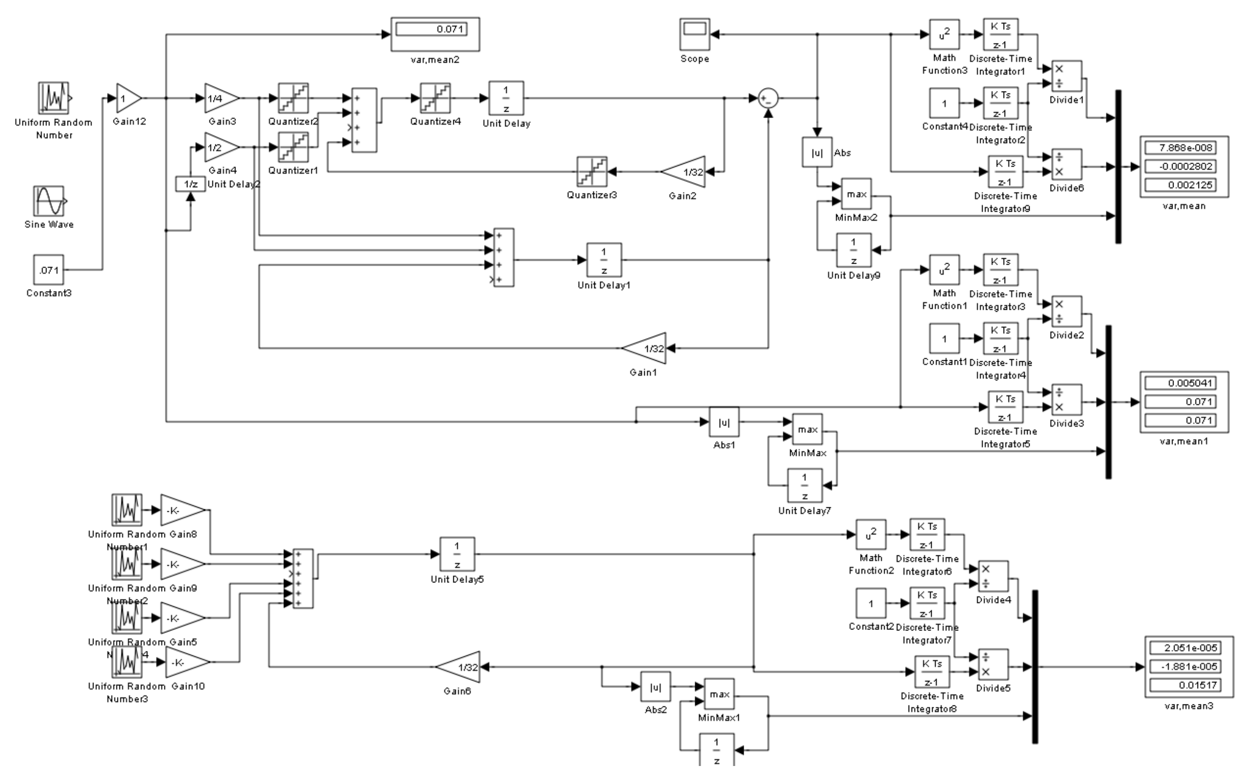
$1.143, \max\left|G\left(e^{j\Omega}\right)\right| = 1.143, \left|\left|G\right|\right|_2 = 1.008$. (The equality of G(1), ||g||, ||G|| is only a coincidence of the simple form of the transfer function. Evaluating the above estimates (with x_n denoting the error due to quantization)

1. $\mathrm{mean}(x_n) = G(1)\mathrm{mean}(n) = 1.143 * 4 * 0 = 0$ (deterministic noise 1.143*4*0.005 = 0.0229)
2. $\max|x_n(k)| \leq (\sum|g(k)|)\max|n(k)|, (g = Z^{-1}\{G\}) = 1.143 * 4 * 0.005 = 0.0229$
3. $\mathrm{var}\{x_n(k)\} \leq 4 * \left|G\left(e^{j\Omega}\right)\right|_2^2 \mathrm{var}\{n(k)\} = 4 * 1.008^2 * 8.33e-6 = 3.39e-5$
4. $\mathrm{var}\{x\} \simeq \mathrm{RMS}^2\{x\} \leq (4 * \max|G(e^{j\Omega})|\,\mathrm{RMS}\{n\})^2 \leq (4 * 1.143 * 0.0029)^2 = 1.76e-4$

Next, we simulate the quantized system, the ideal system, and the system with the noise model of quantizations and tabulate the results as follows:.

	Rand[-1,1]	Const.=0.072	Rand noise model	Theoretical Estimate
Var	2.49e-5	12.16e-5	3.39e-5	3.39e-5 [RMS: 17.6e-5]
Mean	2.68e-5	-0.0067	7.6e-6e-5	0 [deterministic 0.023]
Max	0.0156	0.0085	0.021	0.023 [deterministic 0.023]

Notice that the stochastic variance estimate (using the 2-norm of G) is closer to the observed variance and that the random noise model is fairly representative of the actual errors (for this selection of external inputs). The conservative variance estimate using the RMS deterministic bound (in brackets) is much higher, while the estimate of the maximum amplitude is only conservative by 50%. (This is also because of the specific properties of the system for which sum(|g(k)|) = max|G(e^jw)|.) The deterministic estimates become more accurate for deterministic inputs that expose the worst case. Here a constant 0.072 produced RMS error that was larger than the stochastic estimate. Also note that for the simulation of the random noise model the random number generators must be initialized with different and appropriate seeds so that they produce uncorrelated outputs.



Problem 3.
In a laboratory data acquisition application we would like to use the Diamond MM board to sample 15 signals and transmit the results to a nearby computer over the RS-232 serial port. How fast can we sample under reasonable assumptions.

The MM has a 12-bit A/D so, without special compression, it will use 2 bytes per channel. According to the problem, a total of 2*15=30 Bytes per sample time, or 300 Bits (assuming one start, one stop, 8-data; other valid protocols are also accepted). Selecting standard rate as 115200 bps, and assuming that the distance of transmission is fairly short, sampling frequency comes out to be 384 Hz (or 2.6 msec) << 2 KHz (maximum sampling rate achieved using MATLAB programming).

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%5%%%%%%%%%%%%

EEE 481 Homework 2

Problem 1.
Compute the z-transforms of the following sequences (here u(.) denotes the unit step)

$$u(k+2), \quad \sin\left(\frac{\pi}{10}k-1\right), \qquad \{1-e^{-2(k)}\}u(k-1), \qquad 0.9^k u(k-2)$$

$$Z\{u(\mathrm{k}\ +\ 2)\}\ =\ z^2\ Z\{u(\mathrm{k})\} = \frac{z^3}{(z-1)}$$

Recall that sin $(\frac{\pi}{10}k-1)$ = sin $(\frac{\pi}{10}k)$ cos(1) – cos $(\frac{\pi}{10}k)$ sin(1). For a single sided transform, (k>=0),

$$Z\{\sin(\frac{\pi}{10}k-1)\} = \cos(1)\frac{z\sin\frac{\pi}{10}}{z^2-2z\cos\frac{\pi}{10}+1} - \sin(1)\frac{z^2-z\cos\frac{\pi}{10}}{z^2-2z\cos\frac{\pi}{10}+1}$$

$$Z\{\{1 - e^{-2k-2+2}\}u(k-1)\} = \frac{1}{(z-1)} - \frac{e^{-2}}{(z-e^{-2})}$$

$$Z\{0.9^{(k-2+2)}u(k-2)\} = \frac{0.81}{z(z-0.9)}$$

Problem 2.
Solve the difference equation $y(k+2)-\frac{1}{4}y(k+1)+\frac{1}{16}y(k) = x(k-1)-x(k)$ with the initial conditions y(0) = 1, y(-1) = 0 and x(k) = u(k).

One approach is to rewrite the ODE so that the correct initial conditions appear for the shifted outputs. The ODE, shifted by one, now becomes,

$$y(\mathrm{k}+1) - \frac{1}{4}\,\mathrm{y}(\mathrm{k}) + \frac{1}{16}\,\mathrm{y}(\mathrm{k}-1) = \mathrm{x}(\mathrm{k}-2) - \mathrm{x}(\mathrm{k}-1).$$

Taking transforms and applying the initial condition property, we get

$$zY(z) - zy(0) - \frac{1}{4}Y(z) + \frac{1}{16}z^{-1}Y(z) + \frac{1}{16}Y(-1)$$
$$= z^{-2}X(z) + z^{-1}X(-1) + X(-2) - z^{-1}X(z) - X(-1)$$

Substituting the IC and X(z),

$$z^{-1}[z^2 - \frac{1}{4}z + \frac{1}{16}]Y(z) = z^{-1}\{\frac{1}{(z-1)} - \frac{z}{(z-1)}\} + z$$

After PFE,

$$Y(z) = 1 + \frac{0.125 + 2.3816j}{z - 0.125 - 0.2165j} + \frac{0.125 - 2.3816j}{z - 0.125 + 0.2165j}$$

In MATLAB, the relevant commands are:

```
>>b= [1 0 -1];
>>a= [1 -0.25 0.0625];
>>[r, p, k] =residue (b, a)
```

Hence,

$$y(k) = \delta(k) + 2Re\{(0.125 + 2.3816j)(0.125 + 0.2165j)^{k-1}u(k-1)\}$$

Problem 3.

Consider the system

$$\begin{aligned} x_{k+1} &= Ax_k + Bu_k \\ y_k &= Cx_k \end{aligned} \quad \text{where} \quad A = \begin{bmatrix} -0.2 & 1 \\ 0 & 0.2 \end{bmatrix} \quad B = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$
$$C = [1 \quad -1]$$

1. Determine whether the system is stable or not
2. Determine whether the system is controllable and/or observable
3. Compute its transfer function
4. Compute the first three samples of its unit-step response.

1. The eigenvalues of A are – 0.2 and 0.2, they are inside the unit circle, hence the system is stable.
2. The controllability matrix [B AB] has rank 2, so the system is completely controllable. The observability matrix [C ; CA] has rank 2, so the system is completely observable.
   In MATLAB, the relevant commands are:
   >>G=ss(A,B,C,D,1);
   >>Qc=ctrb(G); Qo=obsv(G);
3. The transfer function is $C(zI - A)^{-1}B + D = \frac{0.6}{z^2 - 0.04}$. ; >>tf(G)
4. We compute the recursion for the states, starting with x(0) = 0 and u(k) = 1 for k>=0.Then,
   y(0) = 0
   y(1) = 0
   y(2) = 0.6
   y(3) = 0.6
   y(4) = 0.624
   etc. (>>y=step(G))

Problem 4.
Write the differential equation describing the motion of a pendulum with input the torque applied at the pivot point and output the angle of the pendulum. Derive the linearized model around the stable and the unstable equilibria and compute the corresponding transfer functions. Assume that the pendulum is a rigid rod of length 0.5m, with evenly distributed mass 100g, and has a small 200g ball attached to the free end. Its rotation around the pivot point is frictionless.

Newton's law yields, $J\frac{d^2\theta}{dt^2} = -g\left(m\frac{L}{2} + ML\right) sin\theta + u$, where $J = \frac{1}{3}mL^2 + ML^2$.

Hence, substituting the pendulum parameters,

$$\frac{d^2\theta}{dt^2} = -21.02\, sin\theta + 17.14\, u.$$

The linearized system around the stable equilibrium has $\sin\theta \simeq \cos 0\, \theta_L, \theta_0 = \theta_0' = u_0 = 0,$

$$\frac{d^2\theta_L}{dt^2} = -21.02\, \theta_L + 17.14\, u_L, \quad \frac{\theta_L\,(s)}{u_L\,(s} = \frac{17.14}{s^2+21.02}$$

The linearized system around the unstable equilibrium has $\sin\theta \simeq \cos\pi\, \theta_L, \theta_0 = \pi,\ \theta_0' = u_0 = 0,$

$$\frac{d^2\theta_L}{dt^2} = 21.02\, \theta_L + 17.14\, u_L, \quad \frac{\theta_L\,(s)}{u_L\,(s} = \frac{17.14}{s^2-21.02}$$

%%5%%%%%%%%%%%%%

EEE 481 Homework 3

Problem 1.

Consider the continuous time system with transfer function $G(s) = \frac{s+2}{s^2+7s+5}$.

1. Realize G(s) in state-space and use Forward Euler to compute its discretization, using sampling time T = 0.1. Find the transfer function of the corresponding discrete-time system.
2. Use Forward Euler directly on the transfer function G(s) and compute the corresponding discrete-time transfer function. Realize the discrete-time system in state-space.
3. Compute the first five terms of the discrete-time system impulse response using state-space formulae. Compare with the result of MATLAB's *impulse(.)* function.

1. $\dot{x} = Ax + Bu, y = Cx + Du, [A, B, C, D] = \left\{\begin{bmatrix} -7 & -2.5 \\ 2 & 0 \end{bmatrix}, \begin{bmatrix} 2 \\ 0 \end{bmatrix}, [0.5 \quad 0.5], [0]\right\}$, is one possible realization. The FE discretization is found by $\dot{x}(T) \sim \frac{x_{k+1}-x_k}{T} \rightarrow x_{k+1} = (I + TA)x_k + TBu_k, y_k = Cx_k + Du_k$. The transfer function for the discrete time system becomes
$$G_d(z) = C(zI - [I + TA])^{-1}[TB] + D$$
This transfer function can be computed by hand, or by the following MATLAB commands

```
>> G=tf ([1,2],[1,7,5])
>> Gs=ss(G)
>>T=0.1; Gd=ss(eye(size(Gs.a))+Gs.a*T,Gs.b*T,Gs.c,Gs.d,T);tf(Gd)
```

ans = $\frac{0.1z-0.08}{z^2-1.3z+0.35}$

Sample time: 0.1 seconds
Notice that the transfer function does not depend on the choice of realization of the continuous transfer function. The above procedure can therefore be used to find the FE discretization of a continuous time system.

2. Performing the substitution $s = \frac{z-1}{T}$, we find exactly the same discrete transfer function as in Part1. A state space realization is
$x_{k+1} = Ax_k + Bu_k, y_k = Cx_k + Du_k, [A, B, C, D] = \left\{\begin{bmatrix} 0.3 & -0.25 \\ 0.2 & 1 \end{bmatrix}, \begin{bmatrix} 0.2 \\ 0 \end{bmatrix}, [0.5 \quad 0.5], [0]\right\}$, which does not need to be (and is not) the same as the one in Part 1.
3. We can easily compute the recursion $x_{k+1} = Ax_k + Bu_k, y_k = Cx_k + Du_k$, with IC = 0 and $u_k = 1\ for\ k = 0,\ 0\ oth.$:
```
>> x = [0;0];
>> y=c*x, x=a*x+b;
>> y=c*x, x=a*x;
>> y=c*x, x=a*x;
Etc.
```
We find the values for y: 0, 1.0000, 0.5000, 0.3000, 0.2150.
We also find the same values with h = impulse(Gd). The impulse response is independent of the realization.

Problem 2.

The first-principles model of a temperature control system is $\dot{Y} = -0.03(Y - 293) + 10Q$, where Y is the Temperature (Kelvin) and Q is the supplied heat (Watts).

1. Use the Forward Euler approximation of derivative $\dot{Y}(t_k) \cong \frac{Y(t_{k+1}) - Y(t_k)}{T_s}$ to write a corresponding discrete time state-space model for a sampling time of 1sec.
2. What is the discrete-time transfer function of the system?
3. What are the limitations (if any) of this discretization method.

1. $Y_{k+1} - Y_k = -0.03Y_k + 8.79 + 10Q_k$
2. From Q to Y, the discrete transfer function is $\frac{Y(z)}{Q(z)} = \frac{10}{z-0.97}$. (293 can be viewed as an external input, or the output can be interpreted as the incremental output over the equilibrium solution Y = 293 for Q = 0.)
3. The stability constraint for the discrete model is $|1 - 0.03T_s| < 1 \rightarrow T_s < 66.667$. Of course, for a sensible approximation, the sampling time should be much less that this bound. E.g., one-half the value will produce a discretized system with pole at the origin, i.e., the entire dynamic response is modeled by a single delay.

Problem 3.

An analog filter with the transfer function $\dfrac{1}{(s+1)(0.1s+1)}$ is to be replaced by a computer. Determine an appropriate sampling time and the transfer function of the discretized filter. You may use any discretization method you like but you should justify all choices.

A reasonable choice for the sampling time would be related to the system bandwidth (0.988rad/s). One may choose different rules of thumb.

- 6 samples/rise time: tr = 2/BW = 2.02s; T = 2.02/6 = 0.337s or f = 2.96Hz. (Measuring tr from a step response simulation we find 2.215s which is reasonably close).
- Nyquist = 10 x BW = 9.878rad/s = 1.57Hz => f = 2*Nyquist = 3.14Hz, T = 0.318s. (This is similar to the above)
- ZOH adds 6deg phase lag at BW (a feedback-related spec), wT/2 = 0.1 => T = (0.2/BW) = 0.2s.

Since we are trying to replace an analog filter and have a discretization with similar filtering properties, a Tustin discretization is the more reasonable choice. Thus, for T = 0.337s (not a unique choice), the discretized transfer function is

$$H_d(z) = \frac{0.0905z^2 + 0.181z + 0.0905}{z^2 - 0.4565z - 0.1815}$$

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%5%%%%%%%%%%%%

EEE 481 Homework 4

Problem 1.

Consider the following system with transfer function $P(s) = \dfrac{-s+1}{(s+0.1)(s+3)}$.

1. Design a PID so that the closed loop crossover is at 0.7rad/s and the phase margin is 50^o.
2. Select a method and the sampling frequency and discretize the PID.

For a discrete design we should first select the sample time so that the ZOH contributes, say, -3 deg phase at crossover, i.e., w_cT/2=0.105/2 or T= 0.15 sec. The phase of P alone at 0.7 rad/s is -130 deg, so we need a PID to control it. We define:

$$C(s) = \frac{K(s+a)^2}{s(\tau s+1)}$$

To achieve 50 degrees phase margin with the discrete controller, we should compute the PID zero to provide 50+3 deg phase margin. Here, however, the problem asks for 50 degrees PM: $2tan^{-1}\frac{0.7}{a} - 90^{\circ} - tan^{-1}0.7\tau - 130 = -180 + 50 \Rightarrow 2tan^{-1}\frac{0.7}{a} = 96, for\ \tau = T.$ Then, we compute $a = \frac{0.7}{tan48} = 0.63$. Substituting back to the gain equation $|(j0.7)C(j0.7)|=1\Rightarrow$ K=1.416. Computing the margins for PC we verify the design.

The sampling frequency is now 1/T=6.667 Hz and the preferred method of discretization of the PID is Tustin, for which we expect a phase margin of 47 deg., since we did not pre-compensate for the ZOH. The controller has the transfer function

$$C_d(z) = \frac{6.902z^2 - 12.56z + 5.713}{(z-1)(z-0.333)}$$

If we evaluate its margins, it provides a 46.98 degree PM, very close to the expected value. The step and frequency responses are also very close to the continuous time versions.

Problem 2.

1. Ziegler-Nichols Tuning: Apply the two Z-N methods from the notes to tune a PID for the plants:
$$P_1(s) = \frac{-0.5s+4}{s^2+6s+3} \quad P_2(s) = \frac{27(-0.5s+1)}{s^2+3s+27}$$

2. Compare the results with a PID designed for a gain crossover frequency of open-loop bandwidth and 45deg. phase margin.

*Hint: Define P as a transfer function object and use step(P) to get an estimate of R,L for the first Z-N tuning. Then iterate k on step(feedback(k*P,1)) until the system is marginally stable (slowly increasing or slowly decreasing response). Then estimate Ku,Pu for the second Z-N tuning. Define the compensators and compare step responses and bode plots for the transfer functions command-to-output and input disturbance-to-output*

We compute the approximate slopes from the step responses as R1 = 0.41, L1 = 0.206, R2 = 7.25, L2 = 0.436. The corresponding controllers are

$$C_1(s) = \frac{1.22s^2 + 14.21s + 34.49}{s}$$

$$C_2(s) = \frac{0.06897s^2 + 0.3796s + 0.4354}{s}$$

For the second method, we try closing the loop with different gains, until oscillatory response is observed. For the first system we find Ku1 = 12, Pu1 = 0.88 and for the second system Ku2 = 0.21, Pu1=1.1. The corresponding controllers are

$$C_1(s) = \frac{0.792s^2 + 7.2s + 16.36}{s}$$

$$C_2(s) = \frac{0.01733s^2 + 0.126s + 0.2291}{s}$$

Note that while these gain values happened to produce an exact oscillatory response (due to the round numbers in the system transfer functions), this does not need to be the case in general; for practical applications, sufficient approximation can be obtained by gains that produce decaying oscillations with low damping.

Finally, we design a controller for crossovers at the open-loop bandwidth: The first system has BW = 0.5463rad/s and the second has BW = 20.2584rad/s. Performing the design, we find

$$C_1(s) = \frac{0.0798s - 0.3266}{s}; \qquad C_2(s) = \frac{0.06217s^2 - 0.3259s + 0.427}{s}$$

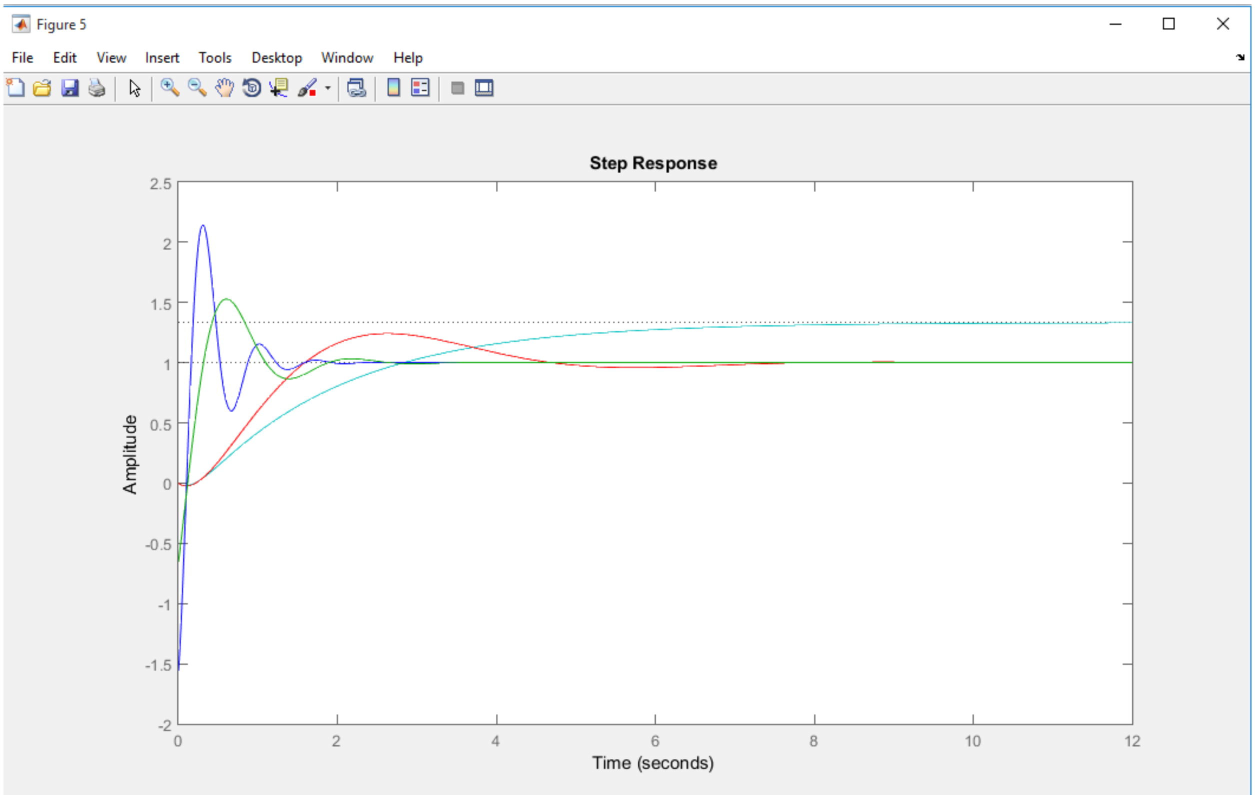
Redesigning the first controller with a crossover at 3x BW:

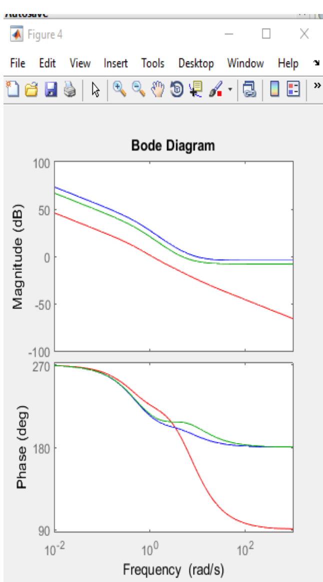
$$C_1(s) = \frac{1.022s + 1.462}{s}$$

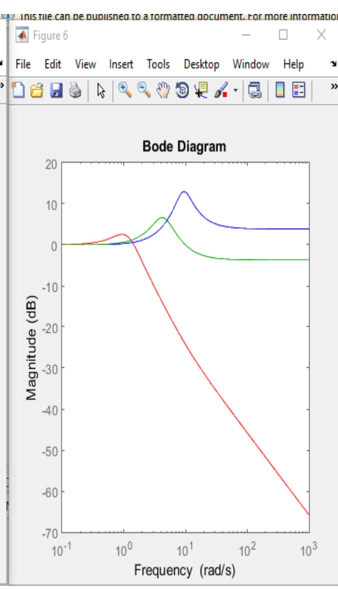
The step responses, command frequency responses, and input disturbance frequency responses are shown below.

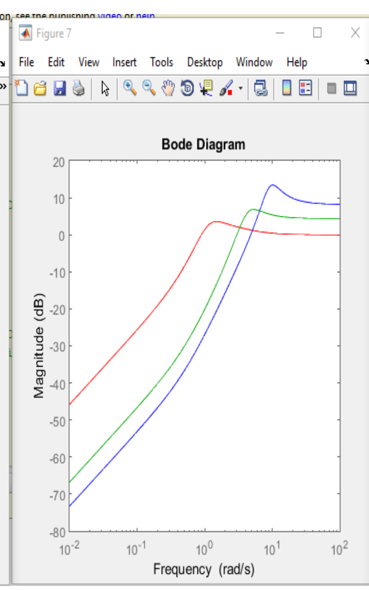
Legends: red: **PM-tuned**; blue: **OL-ZN**; green: **CL-ZN,** cyan: **slow one**

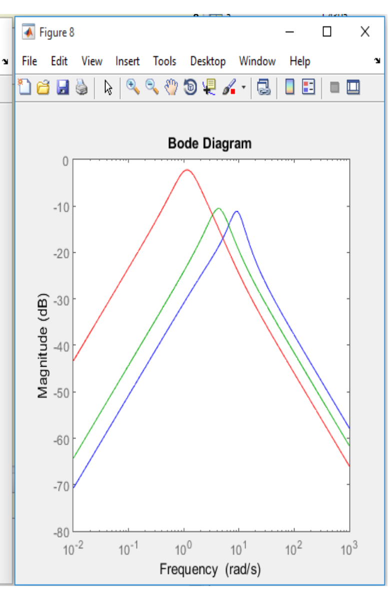
Plant 1:











Plant 2:

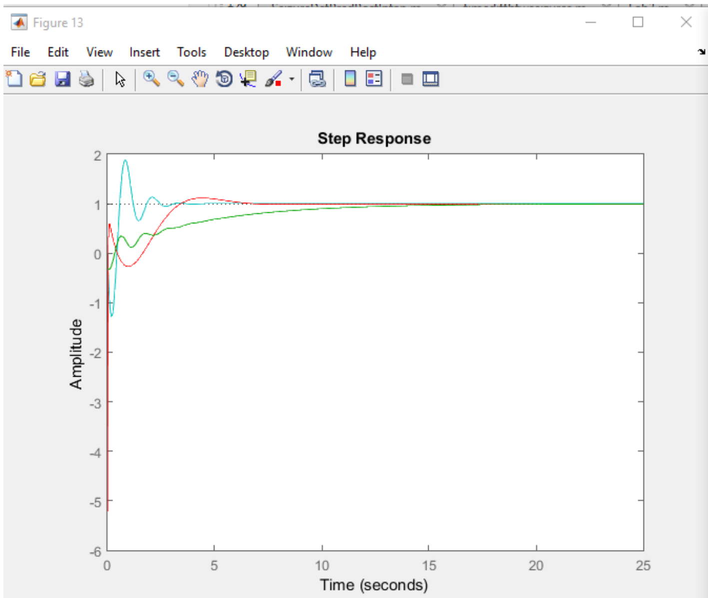


Figure 13
Step Response
Amplitude
Time (seconds)

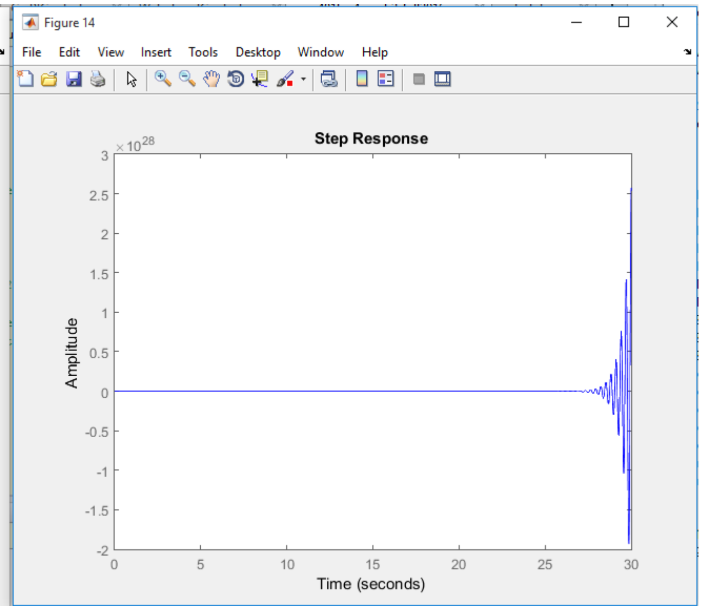


Figure 14
Step Response
Amplitude
Time (seconds)

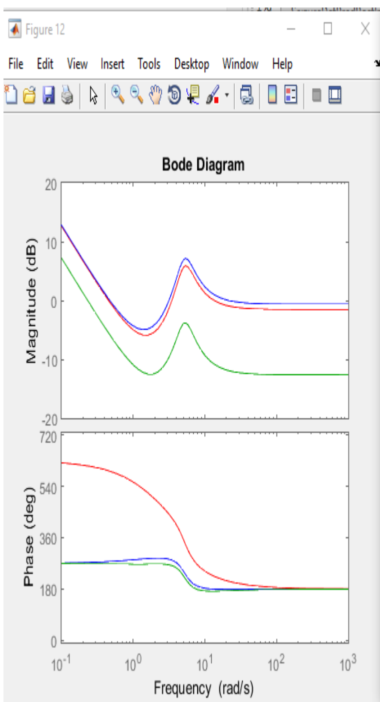


Figure 12
Bode Diagram
Magnitude (dB)
Phase (deg)
Frequency (rad/s)

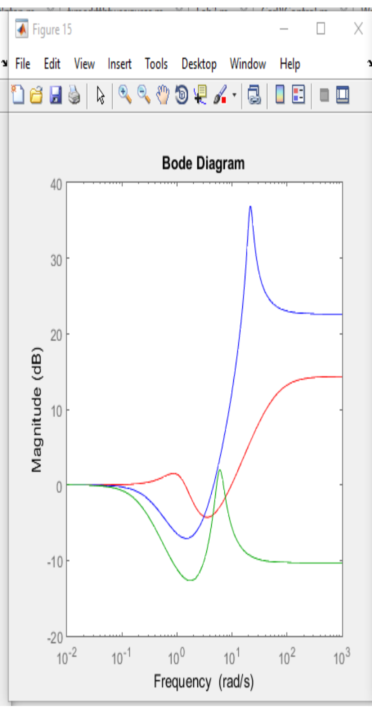


Figure 15
Bode Diagram
Magnitude (dB)
Frequency (rad/s)

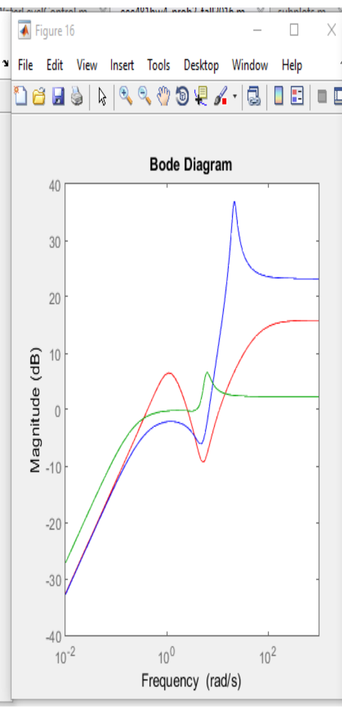


Figure 16
Bode Diagram
Magnitude (dB)
Frequency (rad/s)

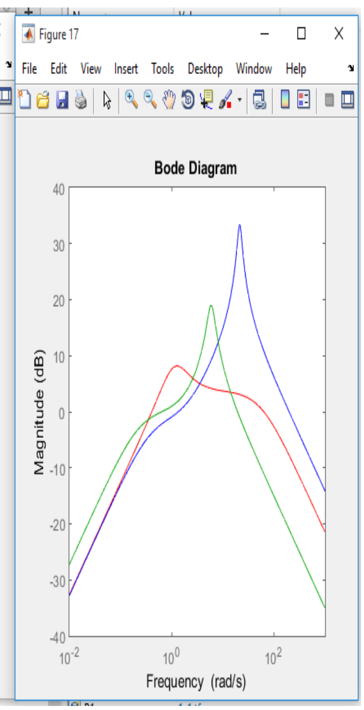


Figure 17
Bode Diagram
Magnitude (dB)
Frequency (rad/s)

Problem 3.

1. Design a PID controller to achieve a bandwidth of 1Hz, 50deg phase margin, and to be discretized with a sampling frequency of 10Hz for the system with transfer function
$$P_1(s) = \frac{-0.5s + 4}{s^2 + 3s + 3}$$
2. Compare the results with a design in discrete time directly, where the plant is discretized and the parameters of a discrete-time PID are calculated to achieve the same specifications.

The angle condition for this problem is
$$angle\{P_1\} + angle\{PID\} + angle\{ZOH\} = -180 + 50$$
evaluated at 1 Hz=6.28 rad/sec. First, we estimate the crossover value as 6.28/1.5=4.2 rad/sec. The ZOH discretized at 10 Hz (T=0.1 sec) has angle –wT/2=0.21 rad= -12 deg. The plant P_1 has angle [m,p]=bode(P_1,4.2); p = -166 deg. Hence, $angle\{PID\}$= 48 deg. This (positive phase) can only be achieved by a PID. Furthermore, since we are trying to design a controller for a plant with roll-off rate only -20db/dec and the sampling rate is not very high relative to the desired bandwidth, it is advisable to include a pseudo-differentiator pole (sT/2+1) in the PID. If not, then the controller discretization will not approximate well the continuous design. Moreover, it will have no high frequency roll-off and will be susceptible to high frequency noise. For consistency in our comparison, we select this pole to be at 2/T so that its Tustin discretization will be simply "z". We will make the same choice later in the discrete design. Adding the pseudo differentiator pole will add lag in the PID. We now have
$$angle\{PID\} = N tan^{-1}\frac{4.2}{a} - 90 - tan^{-1}\frac{T4.2}{2} = N tan^{-1}\frac{4.2}{a} - 102 = 48$$
The only choice for N is 2, for which we find
$$a = \frac{4.2}{\tan(75)} = 1.125$$
Next, we compute the controller gain from the magnitude equation

```
>> C=tf(conv([1 1.125],[1 1.125]),[0.05 1 0]);
>> [m,p]=bode(P1*C,4.2); m = 1.0304
>> C=tf(conv([1 1.125],[1 1.125]),[0.05 1 0])/m
```

Transfer function:
$$C = \frac{s^2+2.25s+1.266}{0.05152s^2+1.03s}$$

Cd=c2d(C,0.1,'tustin')

Transfer function:
$$Cd = \frac{10.83z^2-19.35z+8.644}{z^2-z}$$

Next, we will perform the design entirely in discrete time. Here we consider the PID of the form
$$C(z) = K\frac{(z-a)^2}{z(z-1)}$$
while the plant is Pd=c2d(P_1,0.1,'zoh')
Transfer function:
$$Pd = \frac{-0.02488z + 0.05936}{z^2 - 1.715z + 0.7408}$$
Sampling time:0.1 sec

We compute the angle of the plant and the PID poles at crossover:
[m,p]=bode(Pd*dp,4.2) where dp=1/(z^2-z)

p – 360 = -305.53

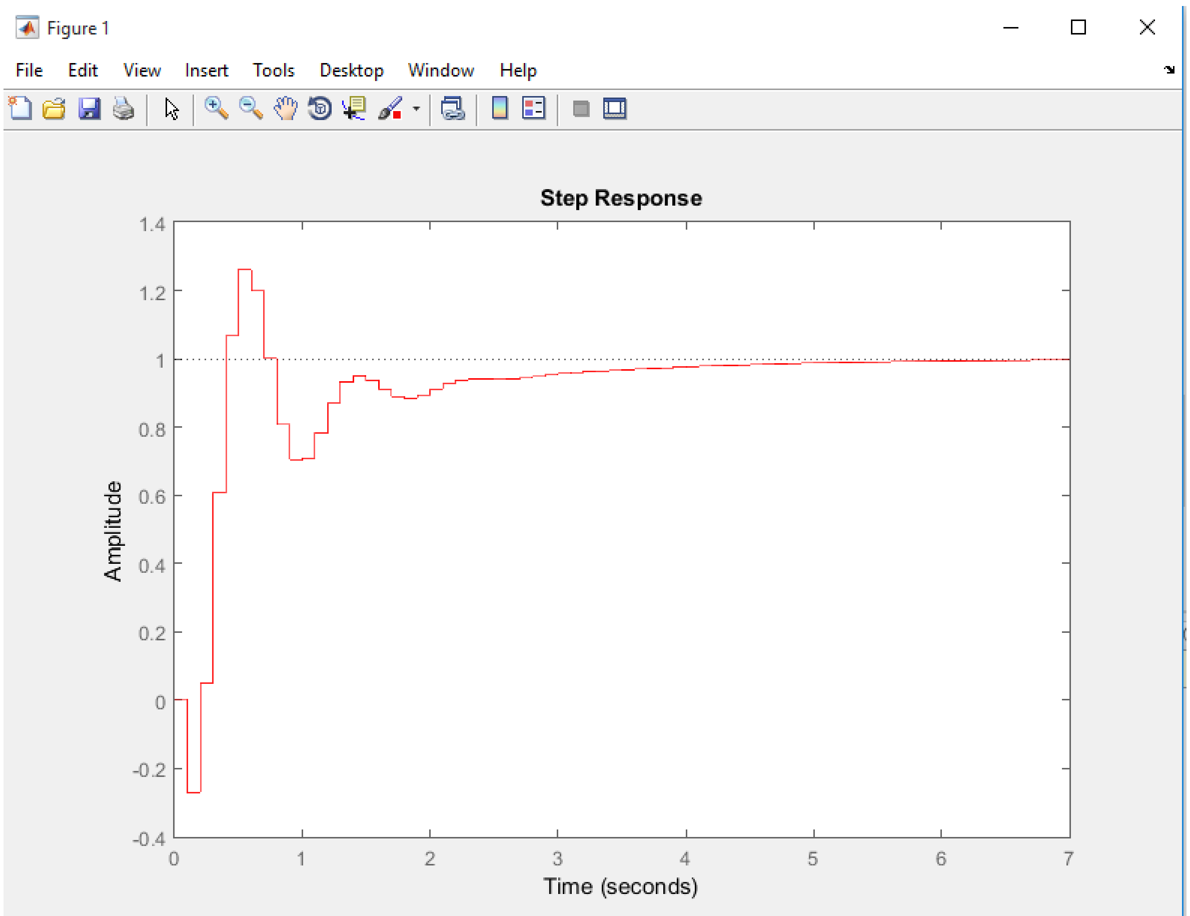
For 50 deg phase margin, this requires an angle contribution from each zero of {-130-(-305.53)}/2 = 87.5 deg.

$$a: atan\frac{\sin(0.42)}{\cos(0.42)-a} = 87.5 \Rightarrow a = 0.897$$

We then compute the gain K so that the crossover is at 4.2 rad/s (0.42 rad/sample), K=10.95. The final controller is

$$\frac{10.95z^2 - 19.64z + 8.8104}{z^2 - z}$$

Obviously, both methods yield very similar controllers and responses. (Any differences are expected to appear much closer to the Nyquist frequency.)



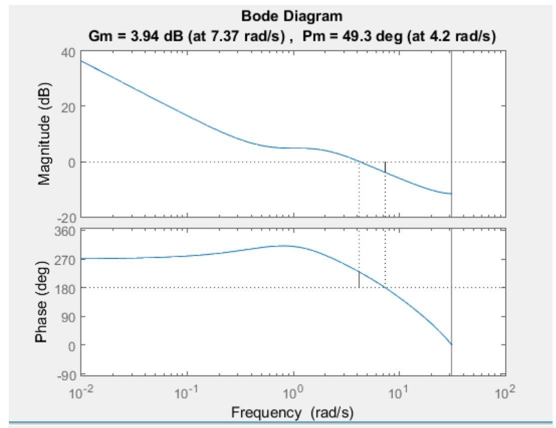
%%5%%%%%%%%%%%%

EEE 481 Homework 5

Problem 1.

1. Design a PID controller to achieve a bandwidth of 1Hz, 50deg phase margin, and to be discretized with a sampling frequency of 10Hz for the system with transfer function
$$P_1(s) = \frac{-0.5s + 4}{s^2 + 3s + 3}$$
2. An additive disturbance enters the plant output with transfer function $P_2(s) = \frac{1}{0.1s+1}$. Design a feedforward component for the PID controller, also discretized at 10Hz, to reduce the effect of the disturbance on the output.

1. From the BW specification, we make a first guess for the crossover frequency 1/1.5 Hz = 0.66Hz = 4.2 rad/s. With the sampling time at T = 0.1sec (10Hz), the ZOH phase is that of half-sample delay at crossover, or –(4.2)(0.1)/2 = 12deg. The continuous PID is

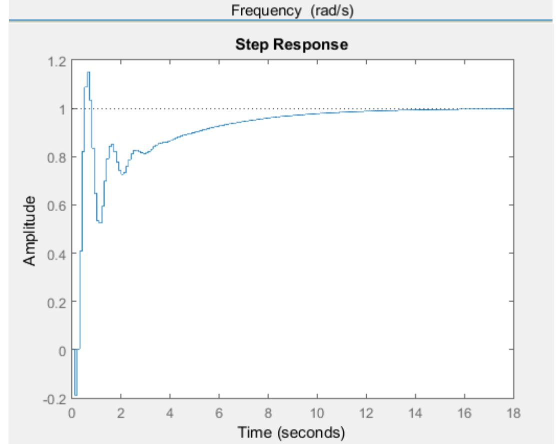


cpid =

0.91 s^2 + 2.875 s + 2.27

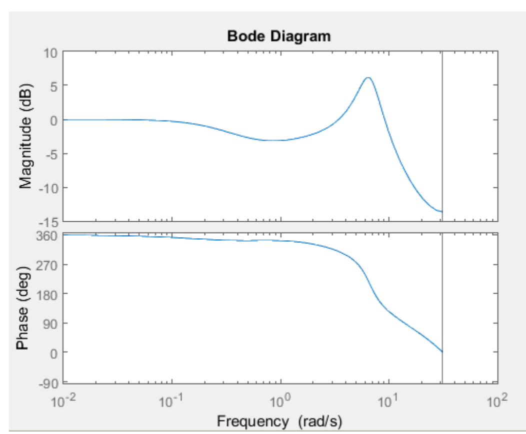
0.05 s^2 + s

The discretized PID (Tustin) is



dpid =
10.81 z^2 - 19.38 z + 8.68
----
z^2 - z
Sample time: 0.1 seconds

2. For the prefilter, we can work either in continuous time (but that requires the redesign of the PID without the ZOH to obtain the equivalent continuous time closed loop system) or in discrete time (but here the factorization function operate in continuous time so a Tustin transform is necessary). Other than that, we follow the stable projection algorithm described in the notes.



As alternatives, we note the approximation of the Plant-inverse by the inverse of its outer part (i.e., the invertible part, where the RHP zeros are replaced by their mirror images (for C-T systems, or their inverses for D-T systems). This approximation has the correct magnitude but its phase is approximately correct only for low frequencies. Similarly, one can also use the DC of the plant as an approximation so the FF is simply$(P(0)^{-1}) * Q$.

A code implementing the solution is given below.

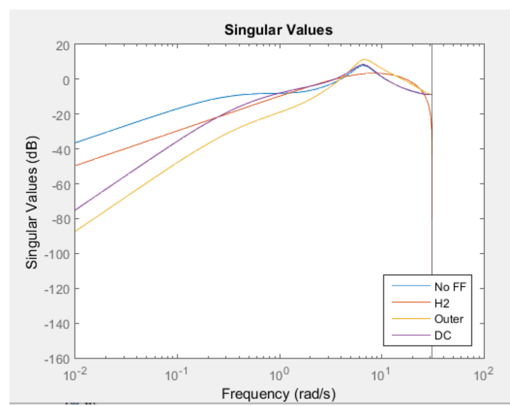
++

```
%EEE 481, HW 5, Problem 1

P=tf([-.5 4],[1 3 3]);
Q=tf(1,[.1 1]);
bw=1*2*pi;
PM=50;
Ts=1/10;
p_correction = bw/1.5*Ts/2*180/pi
[pid1,cpid]=pidpmtune(bw,P,Ts,PM+p_correction);
dpid=c2d(cpid,Ts,'tustin');

Pd=c2d(P,Ts);
Qd=c2d(Q,Ts);

 step(fbk(Pd*dpid,1));pause
 bode(fbk(Pd*dpid,1));pause
 margin(Pd*dpid);pause

Sd=feedback(1,Pd*dpid); SPd=feedback(Pd,dpid); SQd=Sd*Qd;
W=1,r=1e-6,W=c2d(tf([.1 1],[1 1e-4]),.1,'tustin');
Gd=[W*SPd;r]; WTd=([W*SQd;0]);
[SPi,SPip,SPo]=iofr(ss(d2c(Gd,'tustin'))); Stil=inv([SPi,SPip]);
R=minreal(Stil*d2c(WTd,'tustin'));
X2=stabproj(R-R.d)+R.d;  H2o=minreal(inv(SPo)*[1 0]*X2);
eig(H2o)
cut=[];
while isempty(cut), cut=input('cut '),end           % choose 1 here!
[H2s,H2f]=slowfast(H2o-H2o.d,cut);H2f=H2f+H2o.d;
Hd=c2d(H2f,.1,'Tustin');    % H2 optimal design

[Hi,Hip,Ho]=iofr(ss(d2c(Pd,'tustin')));Hd_alt=c2d(inv(Ho)*Q,.1,'Tustin');
                    % Outer Approximation
Hd_dc=inv(dcgain(Pd))*Qd;   % DC-gain Approximation

step(Sd*Qd,SPd*Hd,SPd*Hd_alt,SPd*Hd_dc);
legend('No FF','H2','Outer','DC','location','SouthEast')
pause
sigma(Sd*Qd,Sd*Qd-SPd*Hd,Sd*Qd-SPd*Hd_alt,Sd*Qd-SPd*Hd_dc)
legend('No FF','H2','Outer','DC','location','SouthEast')

disp('Error system Norms:')
disp('    No-FF,  H2-optimal, Outer app,  DC-gain')
disp([norm(W*(Sd*Qd)),norm(W*(Sd*Qd-SPd*Hd)), ...
norm(W*(Sd*Qd-SPd*Hd_alt)),norm(W*(Sd*Qd-SPd*Hd_dc))])

+++++++++++++++++++++++++++++++++++++++++++++++

function [pid,cpid]=pidpmtune(bw,g,tau,pm,n)
% [pid,cpid]=pidpmtune(bw,g,tau,pm)
% bw = bandwidth
% g = system
% tau = derivative TC
% pm = phase margin
% n = PI/PID order (forced)

if nargin<3,tau=.01; end
```

```
if isempty(tau),tau=.01;end
if nargin<4;pm=50;end
if isempty(pm);pm=50;end
if nargin<5;n=[];end

gc=bw/1.5;

cpid=tf(1,[1,0]);
[m,p]=bode(g*cpid,gc);
p=mod(p,360);
if p>0;p=p-360;end
th=(-180-p+pm)
if isempty(n)
    if abs(th)>75; n=2;else;n=1;end
end

if n==1;cpid=tf(1,[1 0]);else;cpid=tf(1,[tau,1,0]);end
[m,p]=bode(g*cpid,gc);
p=mod(p,360);
if p>0;p=p-360;end
th=(-180-p+pm);
a=gc/tan(abs(th)/n*pi/180)
if n==2;cpid=tf([1 2*a a*a],[tau,1,0]);else;cpid=tf([1 a],[1,0]);end
[m,p]=bode(g*cpid,gc);
cpid=cpid*(1/m);

[nu,de]=tfdata(cpid,'v');

if length(nu)<3;nu=[0,nu];end
pid=[nu(2)-nu(3)*tau,nu(3),nu(1)-tau*(nu(2)-nu(3)*tau)];
```

++

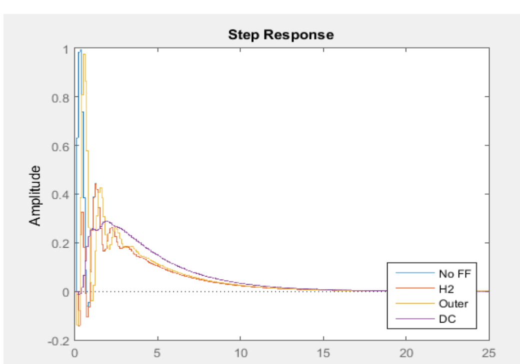
As a metric of the success of the design, we compute the error system norms, which describe the amplification of the variance of the disturbance signal. We observe that the Outer approximation produces the lowest error at low frequencies but the highest variance amplification. Also notice that the proximity of the RHP zero to the bandwidth results in a small performance improvement over the no-feedforward case.




```
Error system Norms:
No-FF,      H2-optimal,    Outer app,  DC-gain
0.2396        0.1790         0.2355     0.2054
```

(The Ad hoc solution of the outer approximation is actually worse than no feedforward, in terms of variance to white noise disturbance!)

The feedforward has a high pass transfer function (looks simple but is actually 5th order.)



Problem 2.

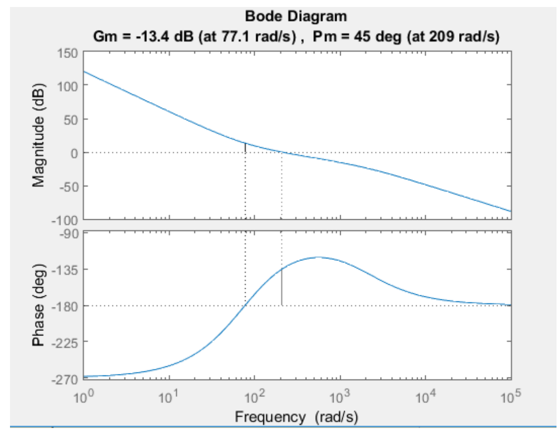
The read arm on a computer disk drive has transfer function

$$H(s) = \frac{1000}{s^2}$$

1. Design an analog PID controller to achieve a bandwidth of approx. 50Hz with 45deg. phase margin.
2. Design a discrete PID for the same bandwidth and phase margin, with a sampling frequency 1kHz and simulate the closed loop step response.
3. What is the minimum sampling rate that can be used, while achieving 50Hz bandwidth and 45deg phase margin with a PID?
4. Design a prefilter to achieve overshoot to step reference changes under 5%.

Hint: You need a complete PID for this problem (2-zeros). Use a filter for the pseudo-differentiator with T = 0.001, consistent with the 1ms sampling time.

The continuous-time design is following the standard procedure, with 45 deg. phase margin and 314rad/s as intended closed loop bandwidth. The resulting closed loop has the correct phase margin and almost exact bandwidth.

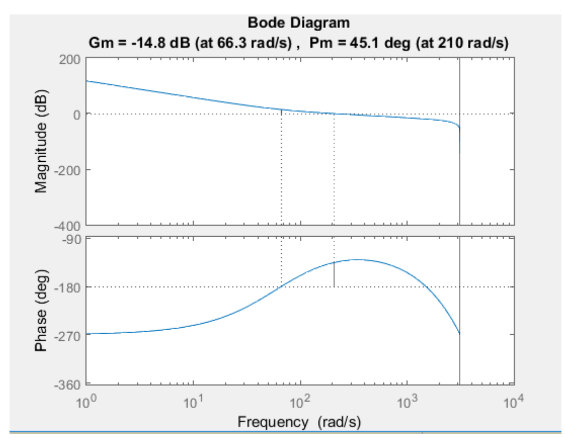



```
>> [pid1,cpid1]=pidpmtune(314,P,Ts/2,50)

cpid1 =

0.187 s^2 + 27.8 s + 1030
 --------------------------------
        0.0005 s^2 + s
```

Next, the discrete-time design will be computed by discretizing the continuous controller but with an adjusted phase margin to compensate for the ZOH. This angle is $\frac{wT}{2} = \frac{314}{1.5} \times \frac{0.001}{2} (rad) = 6^o$

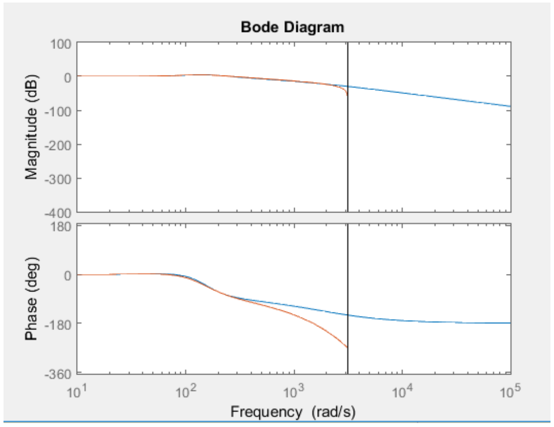
```
>> [pid2,cpid2]=pidpmtune(314,P Ts/2,50+6)

cpid2 =

 0.194 s^2 + 24.01 s + 746
 ---------------------------------
          0.0005 s^2 + s
```

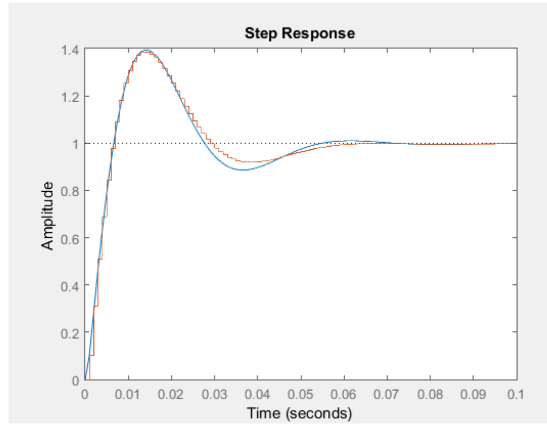
And the discrete pid is found as its Tustin-equivalent:




```
>> dpid=c2d(cpid2,.001,'tustin')

dpid =

 206 z^2 - 387 z + 182
 -----------------------------------
     z^2 - z

Sample time: 0.001 seconds
```

The discrete time loop has phase margin 45 deg. and a slightly larger 362rad/s bandwidth. The frequency and step responses of the CT and DT closed-loop systems are fairly close to each other.





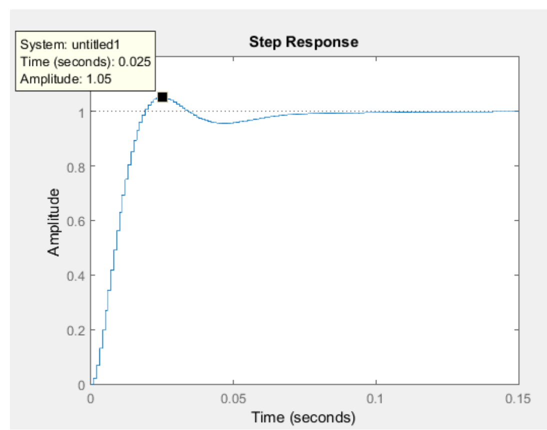
The phase margin equation for this system is

$$-180 - 90 - \operatorname{atan}\frac{w}{2000} - \frac{w}{2000} \times \frac{180}{\pi} + 2\operatorname{atan}\frac{w}{a} = -180 + 45$$

Where a is the PID zero; this yields the constraint

$$2\operatorname{atan}\frac{w}{a} = 135 + \operatorname{atan}\frac{w}{2000} + \frac{w}{2000} \times \frac{180}{\pi} \Rightarrow \operatorname{atan}\frac{w}{2000} + \frac{w}{2000} \times \frac{180}{\pi} \leq 45$$

We plot the left-hand side as a function of w to find that the maximum possible crossover frequency is 712 rad/s corresponding to BW ~ 1068 rad/s or 170 Hz. (Notice, here, we selected the pseudo-derivative time constant as 0.0005s). Of course, for a reasonable integral action, the PID zero should not contribute more than 75-80 degrees.



4. For a prefilter, we can use the general procedure of an additive signal at the plant input, as in the previous problem. Alternatively, a simpler design is to use a low-pass filter or a 2-DOF implementation of the PID with the slow zeros in the feedback path. We will only discuss the last two options here.

A simple first order low-pass set-point filter can be designed approximately based on the frequency response of the closed-loop transfer function. We can then iterate on the filter pole to meet the specification:

```
>> p=85; step(c2d(tf([1/500 1],[1/p 1]),.001,'Tustin')*fbk(c2d(P,.001)*dpid,1))
```

Problem 3.

Design a PID controller for the flexible inverted pendulum with transfer function

$$\frac{\{1.478\}}{\{s^2 + 0.0635s - 19.54\}} + \frac{\{0.000332\, s^2 + 0.3785\, s + 177.5\}}{\{s^2 + 15.52\, s + 64750\}}$$

For this problem, the PID should be augmented by a low-pass filter to increase roll-off beyond bandwidth and avoid the excessive excitation of the flexible modes. The sampling frequency is 1000Hz and the choice of closed-loop bandwidth is left as a design parameter. Use a 3rd order low-pass filter, with bandwidth roughly at 2x or 3x of the crossover frequency. In your design, include a prefilter to maintain overshoot to step reference changes under 5%. Verify the stability of your controller with simulations.

The filter is needed to attenuate the resonance peak of the flexible mode so that it does not cause the loop magnitude with the PID to exceed unity. At that frequency, the PID will be in its high frequency gain that is expected to be large, since considerable phase lead is required to stabilize the plant. On the other hand, the crossover frequency should be higher than the instability (4.4 rad/s). So we need to determine a sensible filter F to allow us to iterate on crossover/phase margin.

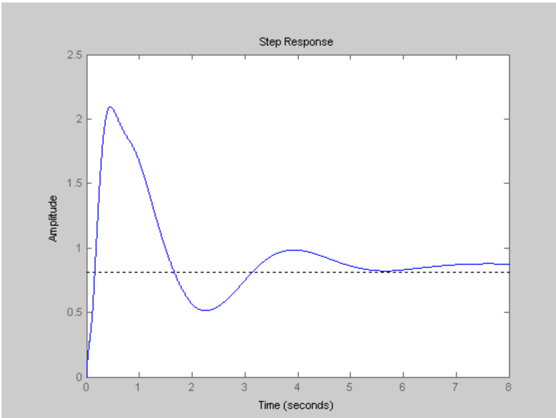
Roughly, the design equation is $\angle P + \angle F + \angle C = -180 + PM$. We expect that the PID zeros do not contribute more than 150 deg and since we are looking at a crossover around 10 rad/s, the ZOH will have a minimal effect. So we can iterate very quickly $\angle P + \angle F \geq -200^o$ $(or +160^o)$ and adjust the pole of F so that this inequality holds for some frequencies above 5 rad/s. We arrive at a value of 40 for the filter pole. We then set-up an iteration to compute a reasonable PID tuning:

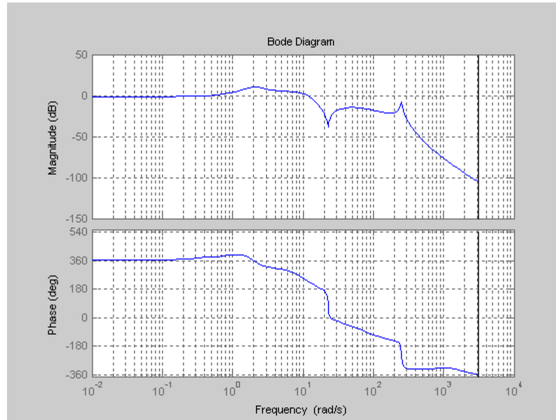
```
%EEE 481, HW 5, Problem 3

P=tf(1.478,[1  0.0635  -19.54])+tf([0.000332  0.3785  177.5],[1 15.52
64750]);
T=0.001;
wc=[];
while isempty(wc), wc=input('crossover '),end

F=tf(1,[1/40,1]);F=F*F*F;
zoh = T/2*wc*180/pi;

[pid1,cpid]=pidpmtune(wc*1.5,P*F,.001,35+zoh)
dpid=c2d(cpid,.001,'tustin')
Pd=c2d(P*F,.001);

 step(fbk(Pd*dpid,1));pause
 bode(fbk(Pd*dpid,1));pause
 margin(Pd*dpid);pause
```

After comparing the responses of a few controllers with different wc and PM, we select PM = 35 and wc = 7 as the best one. The final controller (C-T, D-T) is:

```
cpid =

  7.572 s^2 + 23.44 s + 18.14
  ---------------------------
        0.001 s^2 + s

dpid =

  5056 z^2 - 1.01e04 z + 5040
  ---------------------------
    z^2 - 1.333 z + 0.3333

Sample time: 0.001 seconds
```

It yields the following step and frequency responses:





The large overshoot is due to the proximity of the RHP pole of the plant to the closed-loop bandwidth. It does necessitate the use of a prefilter.We use a simplified prefilter $\frac{0.05s+1}{as+1}$ and try different values of a to get the overshoot below 5%, arriving at the value $a = 1.7$. The response is not great (significant undershoot) but for the reduced complexity prefilter, it is adequate.

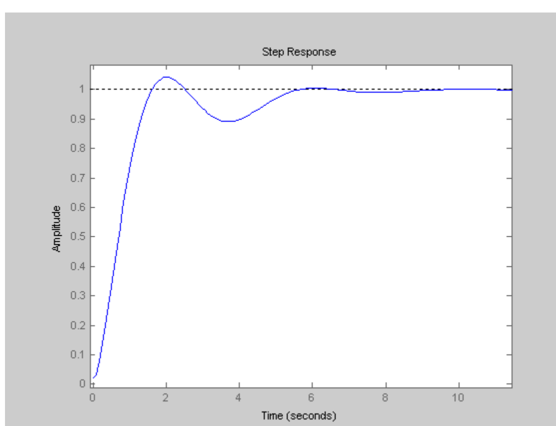
```
>> step((tf([.05 1],[1.7 1]))*fbk(P*cpid,1))
```

The discretized prefilter is

```
0.0297 z - 0.02911
------------------
    z - 0.9994
```

Sample time: 0.001 seconds

(Note: The discrete simulation
>> step(c2d(tf([.05 1],[1.7 1]),.001,'tustin')*fbk(Pd*dpid,1))
diverges due to numerical sensitivity issues. To obtain a correct result P, F, and cpid should all be converted to state-space from the beginning)



EEE 481 Homework 6

Problem 1.
Consider the pendulum model with input the torque applied at the pivot point and output the angle of the pendulum. (Assume that the pendulum is a rigid rod of length 0.5m, mass 50g evenly distributed, with a 300g point mass at its end, and its rotation around the pivot point is frictionless.)

1. Design a state observer to estimate the angle and angular velocity from noisy angle measurements.
2. Collect 20s of simulation data at 100Hz with random 10Hz excitation around the stable equilibrium such that the amplitude of oscillation is approximately 6degrees. Implement a 12-bit quantization on the angle measurement for the 360degree range and a 10-bit quantization on the torque for the range [-1, 1]. Formulate the parameter estimation problem and use the batch least-squares algorithm to estimate the parameters of the corresponding transfer function.
3. Repeat Part 3 for an Arduino-based controller with 10-bit ADC quantization and 8-bit DAC quantization.

Illustrate your findings with a few well-chosen simulations.

We start with the pendulum model

$$J\ddot{\theta} = T - g(\frac{m_1 l}{2} + m_2 l)\sin\theta - \epsilon\dot{\theta}|\dot{\theta}|$$

Where m is the mass, l is the length, $J = \frac{m_1 L^2}{3} + m_2 L^2$ is the inertia, and ϵ is the friction coefficient for the pendulum, and $[T, \theta]$ is the I/O pair. The torque T is proportional to the current driving the pendulum motor, but since we have no further data, we will assume a proportionality constant of 1. Linearizing the model around the stable equilibrium$[0, 0]$, we obtain the transfer function

$$P(s) = \frac{12.63}{s^2 + 20.12}$$

And the state-space realization in terms of angle and angular velocity

$$\dot{x} = \begin{bmatrix} 0 & 1 \\ -20.12 & 0 \end{bmatrix} x + \begin{bmatrix} 0 \\ 12.63 \end{bmatrix} u,$$
$$y = [1 \quad 0]x$$

For the discrete-time model, to be used for state estimation, we find the ZOH equivalent:

$$x_{k+1} = \left(\begin{bmatrix} 0.999 & 0.009997 \\ -0.2011 & 0.999 \end{bmatrix}\right) x_k + \begin{bmatrix} 0.0006315 \\ 0.1263 \end{bmatrix} u_k,$$
$$y_k = [1 \quad 0]x_k$$

For this we define the state observer

$$\hat{x}_{k+1} = A\hat{x}_k + Bu_k + L(y_k - \hat{y}_k),$$
$$\hat{y}_k = C\hat{x}_k$$

where L is the observer gain which can computed using a variety of approaches. One, particularly attractive method is by using the Kalman Filter equations in their steady-state solution, given by the discrete Riccati equation $L = A\Sigma C^T[C\Sigma C^T + R]^{-1}, \Sigma = A\Sigma A^T + GQG^T - A\Sigma C^T[C\Sigma C^T + R]^{-1}C\Sigma A^T$. While this equation, taken as a recursion, will converge to the steady-state solution, MATLAB also implements efficient numerical methods to solve it:

```
>> L = dlqe(A,G,C,Q,R)
```

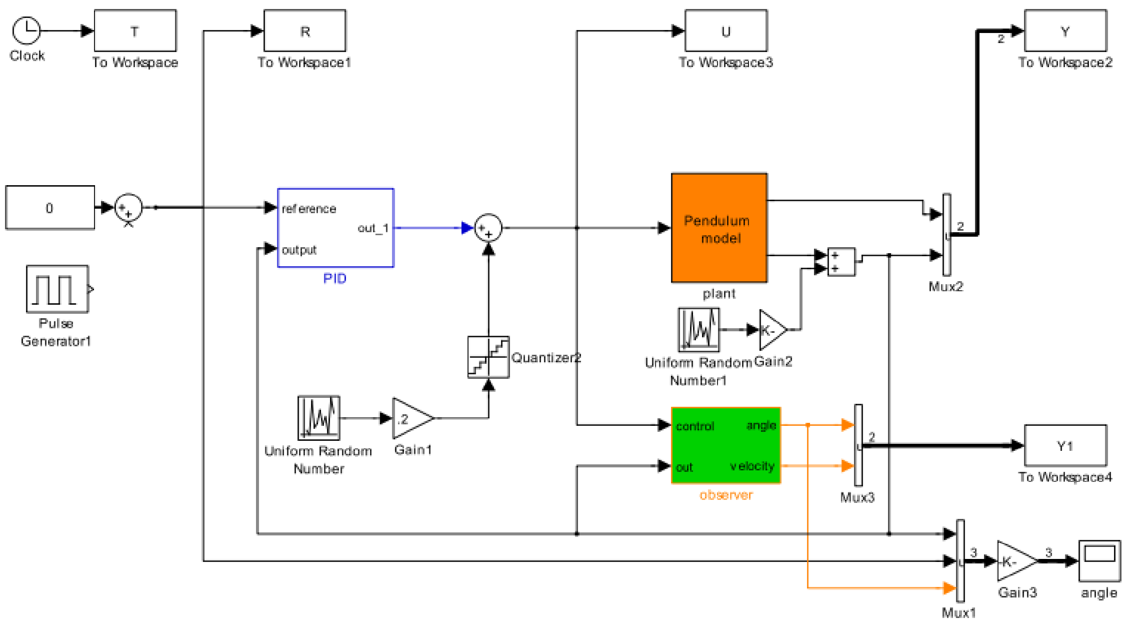
Here, G,Q are the input and intensity (covariance) matrices for the state noise and R is the intensity of the output noise. Since we do not have any additional information to model the noise, or optimize specific aspects of the Kalman Filter response, we will simply choose, G = I, Q = BB', and R = a small scalar, to be iterated until a "reasonable" speed of convergence is obtained. For example,

```
>> Hd = c2d(H,0.01,'zoh')
>> L=dlqe(Hd.a,eye(2,2),Hd.c,Hd.b*Hd.b',0.01),abs(eig(Hd.a-L*Hd.c))
```

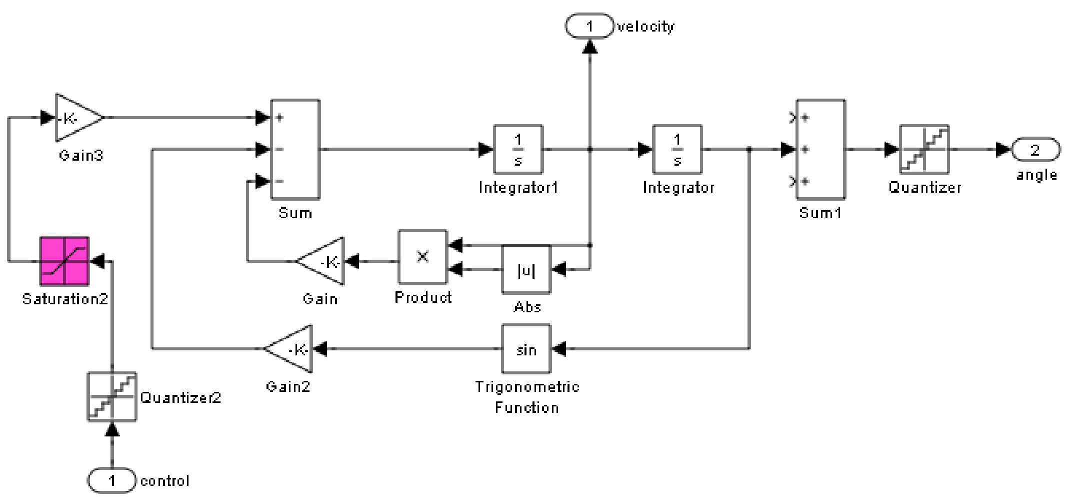
This yieds
L = [0.1365; 1.0013]
and magnitude of the observer error system eigenvalues 0.9347.

Finally, for implementation purposes, it is often a good idea to use a controller to stabilize the system so that its response stays bounded for any possible test condition. (Especially, for system identification applications.) Omitting the details, here we design a PID to provide 50deg phase margin at 13rad/s:
[Kp, Ki, Kd] = [6.2839 12.1727 0.7799]

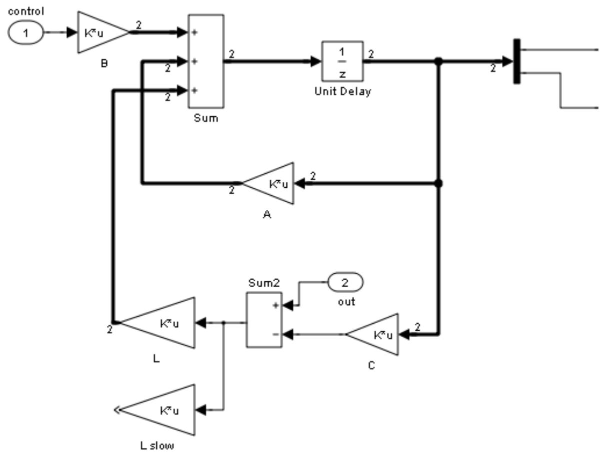
Next, we construct a simulation model to solve the nonlinear pendulum equation, and connect the observer to the system I/O.

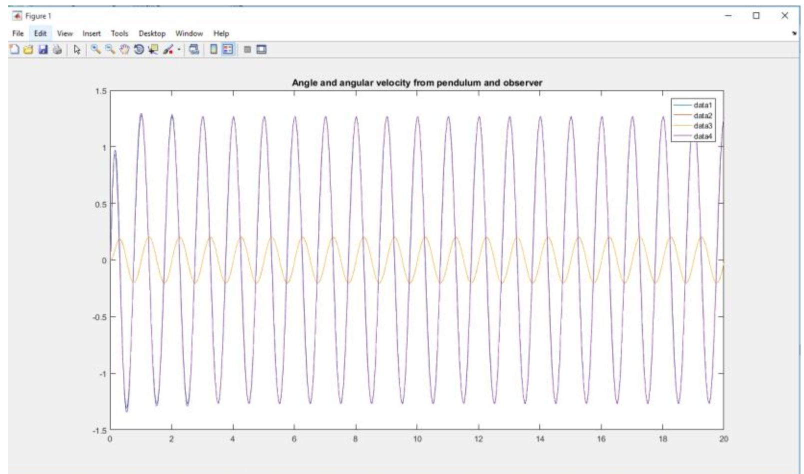


Pendulum Subsystem:



Observer Subsystem:

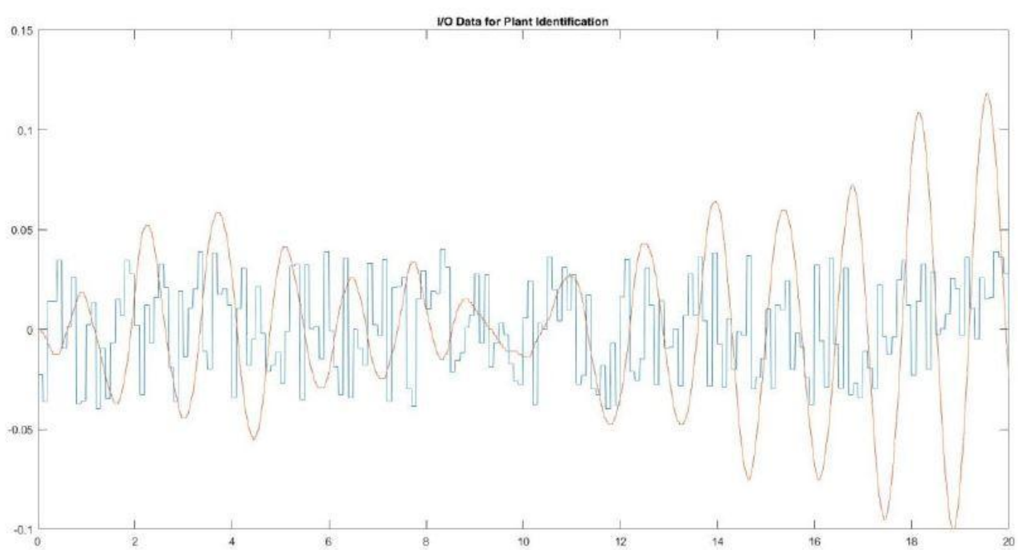




This simulation model allows the study of observer and identification problems under a variety of conditions. We list some below:

- Convergence for different initial conditions (defined in the Pendulum mask)
- Convergence with and without the PID controller, with and without random excitation, with and without output noise
- Use of different observer gains, obtained with different output noise weights (R) in the Riccati equation
- Stable and unstable equilibrium (requires adjustment of the observer model).

For the identification experiment, we connect the excitation at the pendulum input. With zero I.C., and after some trial-and-error we find a gain for the excitation (0.02) which causes the angle deviations to be below 6 degrees. (This is necessary to keep the system near the linearization point where $\sin\theta \simeq \theta$.)



We collect the data (U,Y) and form a regressor for a second order system. For a generalization, we define the filter F (e.g., a delay) and then write the regressor as:

$$w = [Fy, FFy, Fu, FFu]$$

(for a general case of regressor construction, see a system identification text). Then, the LS approximation problem has a solution

$$q = w\backslash y = (w^T w)^{-1} w^T y$$

From which the identified system can be expressed as

$$H = \frac{q(3)F + q(4)FF}{1 - q(1)F - q(2)FF}$$

The MATLAB implementation of this algorithm is shown below

```
>> F=c2d(tf(1,[.1 1]),.01)
>> w=[lsim(F,Y(:,2)),lsim(F*F,Y(:,2)),lsim(F,U),lsim(F*F,U)];q=w\Y(:,2)
>> Hd=minreal((q(3)*F+q(4)*F*F)/(1-q(1)*F-q(2)*F*F)), H=d2c(Hd)
```

Then

$$H_d(z) = \frac{0.001287z - 6.598e-05}{z^2 - 1.997\, z + 0.9993}, \qquad H_c(s) = \frac{0.06767\, s + 12.21}{s^2 + 0.07018s + 19.69}$$

Notice that, even though the identified model is unstable, its coefficients are fairly “close” to the true linearization (P), implying that a controller designed for the identified system will also work for the actual system. (The theory behind this statement is “coprime factor perturbations”, “gap metric” is discussed in graduate courses.) However, the identification of the resonance is usually a difficult task and some “smearing” of the peak occurs. A similar result is obtained with the controller in feedback, but now the excitation must be increased by an order of magnitude to achieve the same range of output variation. Otherwise, the output noise causes the signal to noise ratio (SNR) to decrease and the accuracy of the identification deteriorates. Finally, identification with the pure ARX regressor (delay, F = tf(1,[1 0],.01)) is unsuccessful for this case, because it puts too much emphasis on the high frequencies.