CS 545

Homework Assignment 1

Abishek Hariharan - 2743 1905 72

1. a)

$$\dot{x}(t) = A\,x(t) + B\,u(t)$$

input to system

State matrix   input matrix.

$x \Rightarrow$ state - position

$\dot{x} \Rightarrow$ state - velocity.

$$\ddot{x} = b\dot{x} + kx + ru$$

$$x_1 = x$$

$$x_2 = \dot{x} \qquad\qquad x_2 = \dot{x}_1$$

$$\dot{x}_2 = b x_2 + k x_1 + r u$$

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ k & b \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0 \\ r \end{bmatrix} * u$$

$$A = \begin{bmatrix} 0 & 1 \\ k & b \end{bmatrix} \qquad B = \begin{bmatrix} 0 \\ r \end{bmatrix}$$

$u \Rightarrow$ input to the system

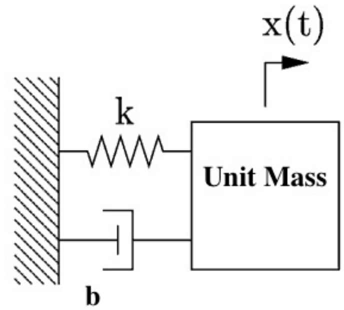
$x \Rightarrow$ position - state

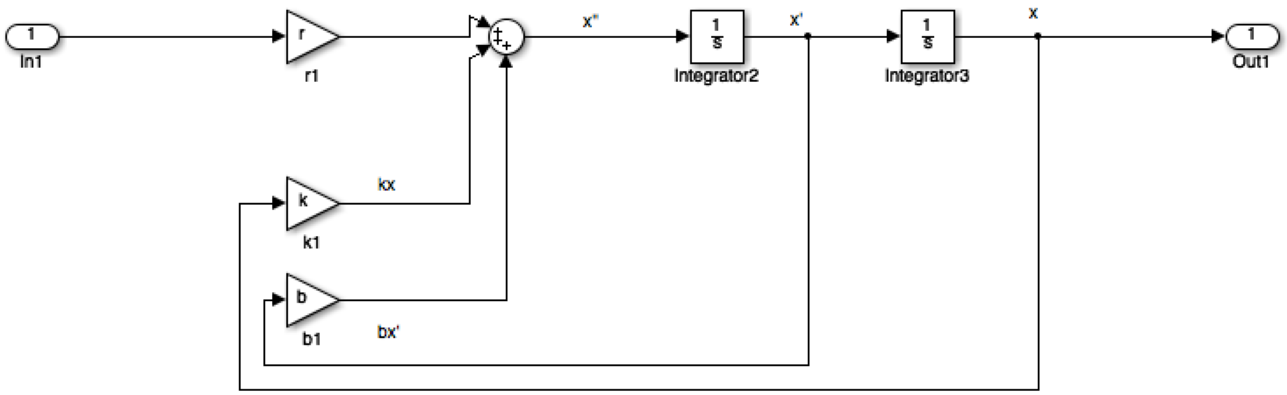
$A \Rightarrow$ State matrix

$B \Rightarrow$ Input matrix

$\dot{x} \Rightarrow$ velocity - state

b) Plant block - graphical representation of physical system





c)Negative feed back control law

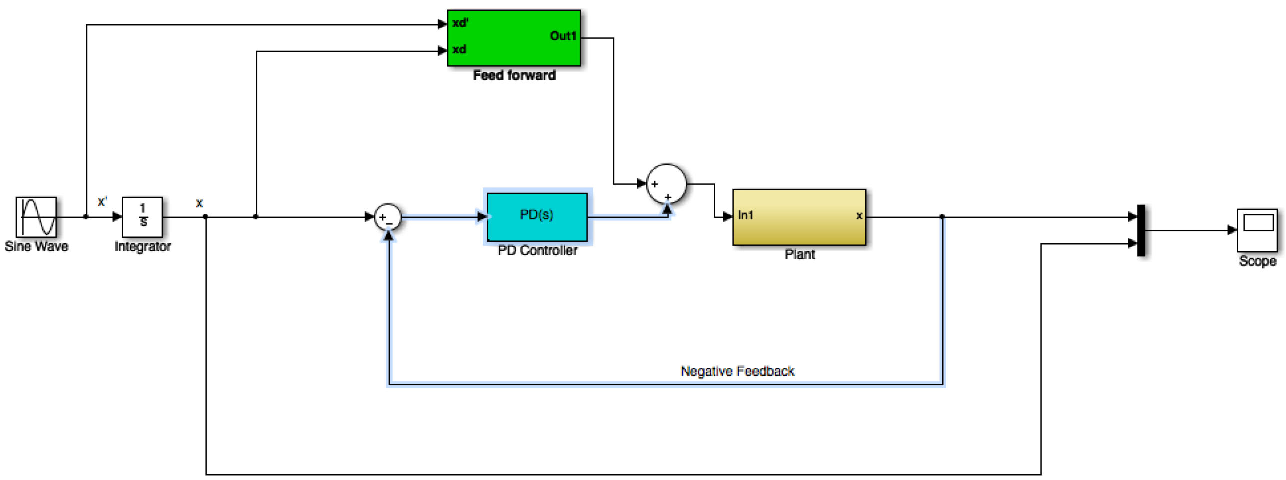
$$u_{fb} = \Pi(x - x_{des}, \alpha, t)$$

$$u_{fb} = K_P(x_d(t) - x(t)) + K_D(\dot{x}_d(t) - \dot{x}(t))$$

$$u_{fb} = K_P(X_{1d}(t) - X_1(t)) + K_D(X_{2d}(t) - X_2(t))$$

$$K_P\left[X_1 d(t) - X_1(t)\right] + K_D\left[X_2 d(t) - X_2(t)\right]$$

d)
PD controller indicated in cyan with the feedback pathway highlighted in light blue.



e) This kind of controller is able to more or less keep up with input signal , there is slight shift in phase , but the controller is able to keep up and not lag . Feedback control is good for error correction but we require a feed forward controller to push the system towards the desired state. The controller has no Integral control so it suffers from steady state errors. By adjusting both the Kp and Kd gains simultaneously we can tune the controller to the system model.

f) Solution for $\mathbf{u_{ff}}$

$$\dot{x}(t) = Ax(t) + Bu(t)$$

$$\dot{x}(t) - Ax(t) = Bu(t)$$

$$\frac{\dot{x}(t) - Ax(t)}{B} = u_{ff}(t)$$

g)

1. If n = m , then matrix B becomes square and is invertible and can be multiplied in the above equation.
2. if n > m , then we have left inverse
3. in n < m , then we have right inverse

Left inverse of a system computes the input to the system given the output , while right inverse on the other hand computes the input required to produce a desired output.In a feed forward system , we care about the right inverse more than the left inverse.However in this case B has more rows than columns ,and it would have to be left inverted.

Calculating the inverse of a matrix

$$X = \begin{bmatrix} a & b \\ c & d \end{bmatrix}$$

taking the determinant $ad - bc$

$$X^{-1} = \frac{1}{\det(X)} \begin{bmatrix} d & -b \\ -c & a \end{bmatrix}$$

Suppose we have

$$X = \begin{bmatrix} 1 & 1 \\ 2 & 3 \end{bmatrix} \qquad x^{-1} = \frac{1}{3-2} \begin{bmatrix} 3 & -2 \\ -1 & 1 \end{bmatrix} = \begin{bmatrix} 3 & -2 \\ -1 & 1 \end{bmatrix}$$

Right inversion of a non square matrix:

Now suppose we have

$$Y = \begin{bmatrix} 1 & 1 & 2 \\ 3 & 5 & 8 \end{bmatrix}$$

the number of columns is bigger than number of rows, hence we take the right inverse using.

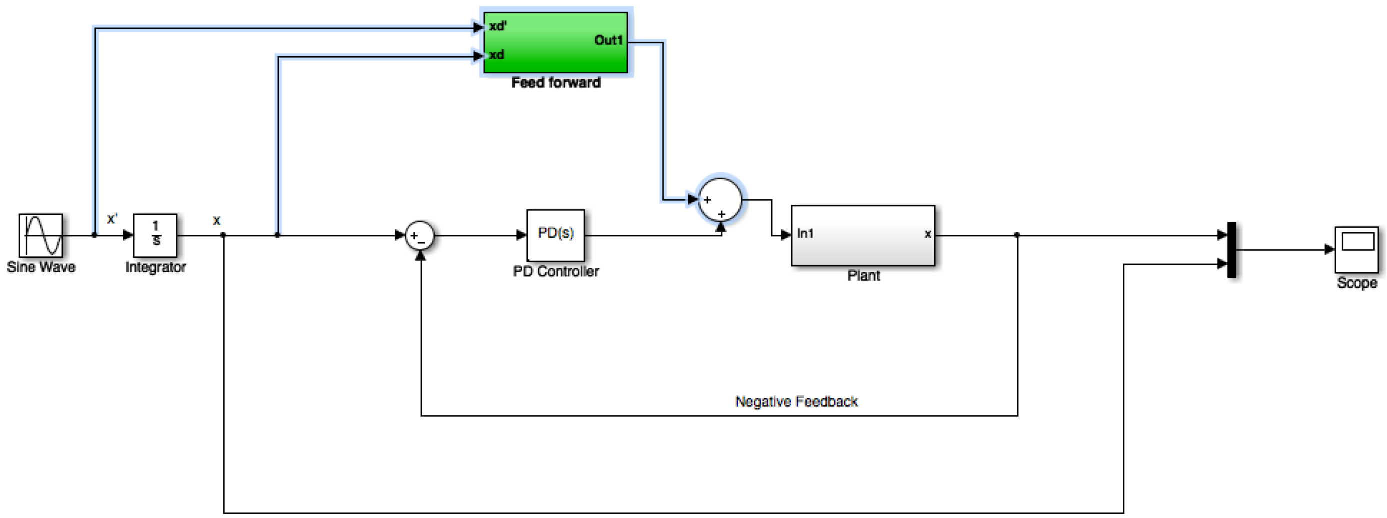
$$YY^T(YY^T)^{-1}$$

$$YY^T = \begin{bmatrix} 1 & 1 & 2 \\ 3 & 5 & 8 \end{bmatrix} \cdot \begin{bmatrix} 1 & 3 \\ 1 & 5 \\ 2 & 8 \end{bmatrix} = \begin{bmatrix} 6 & 24 \\ 24 & 98 \end{bmatrix}$$

$$(YY^T)^{-1} = \begin{bmatrix} 6 & 24 \\ 24 & 98 \end{bmatrix}^{-1} = \frac{1}{12}\begin{bmatrix} 98 & -24 \\ -24 & 6 \end{bmatrix}$$

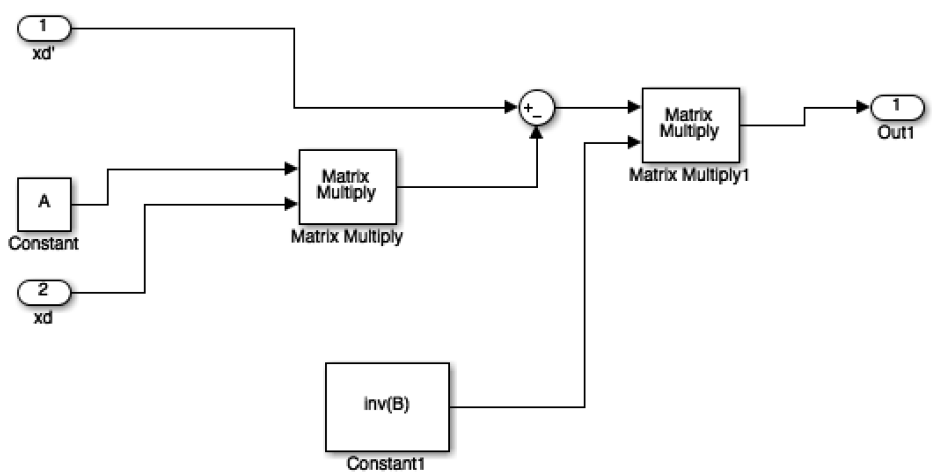
$$\cancel{A^T(AA^T)^{-1} = \frac{1}{12}}$$

$$Y^T(YY^T)^{-1} = \frac{1}{12}\begin{bmatrix} 1 & 3 \\ 1 & 5 \\ 2 & 8 \end{bmatrix} \circ \begin{bmatrix} 98 & -24 \\ -24 & 6 \end{bmatrix} = \begin{bmatrix} 2.1667 & -0.5 \\ -1.8333 & 0.5 \\ 0.3333 & 0 \end{bmatrix}$$



h)Feed forward block in green with feed forward pathway highlighted in blue.

Inside feedforward subsystem:



2)

a)

CS 545

2. $m\dot{x}_1 = bx_1 + ru$

$\dot{x}_2 = k x_2 + x_1$

$m \Rightarrow$ mass
$b \Rightarrow$ viscous friction coeff
$k \Rightarrow$ spring constant.
$r \Rightarrow$ command amplification

a)

Laplace transform

$$m\dot{x}_1(t) = b x_1(t) + r u(t)$$

$$m\, s x_1(s) - x(0) = b x_1(s) + r u(s)$$

$$m\, s x_1(s) - b x_1(s) = r u(s)$$

$$(ms - b)(x(s)) = r u(s)$$

$$\boxed{x_1(s) = \frac{r}{ms - b} u(s)}$$

$$\dot{x}_2(t) = k x_2(t) + x_1(t)$$

$$s x_2(s) - x(0) = k x_2(s) + x_1(s)$$

$$s x_2(s) - k x_2(s) = x_1(s)$$

$$(s-k) x_2(s) = x_1(s)$$

$$x_2(s) = \frac{x_1(s)}{(s-k)}$$

$$\boxed{x_2(s) = \frac{1}{(s-k)} x_1(s)}$$

Substituting,

$$x_2(s) = \frac{1}{(s-k)} \left(\frac{r}{ms - b} u(s) \right)$$

$$x_2(s) = \frac{r}{(s-k)(ms-b)} u(s)$$

$$x_2(s) = \underbrace{\frac{k}{ms^2 - bs - kms + kb}}_{H(s)} \; u(s)$$

b)

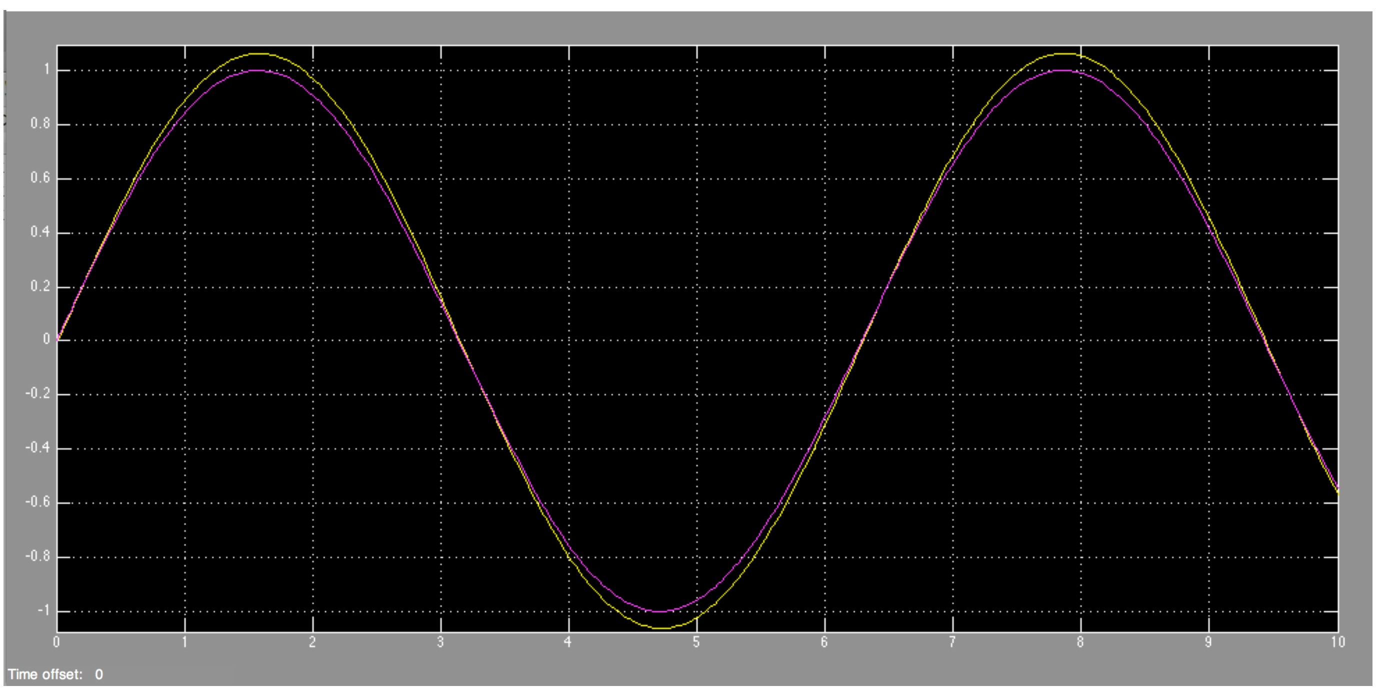
Desired position VS observed position plot after tuning PD controller

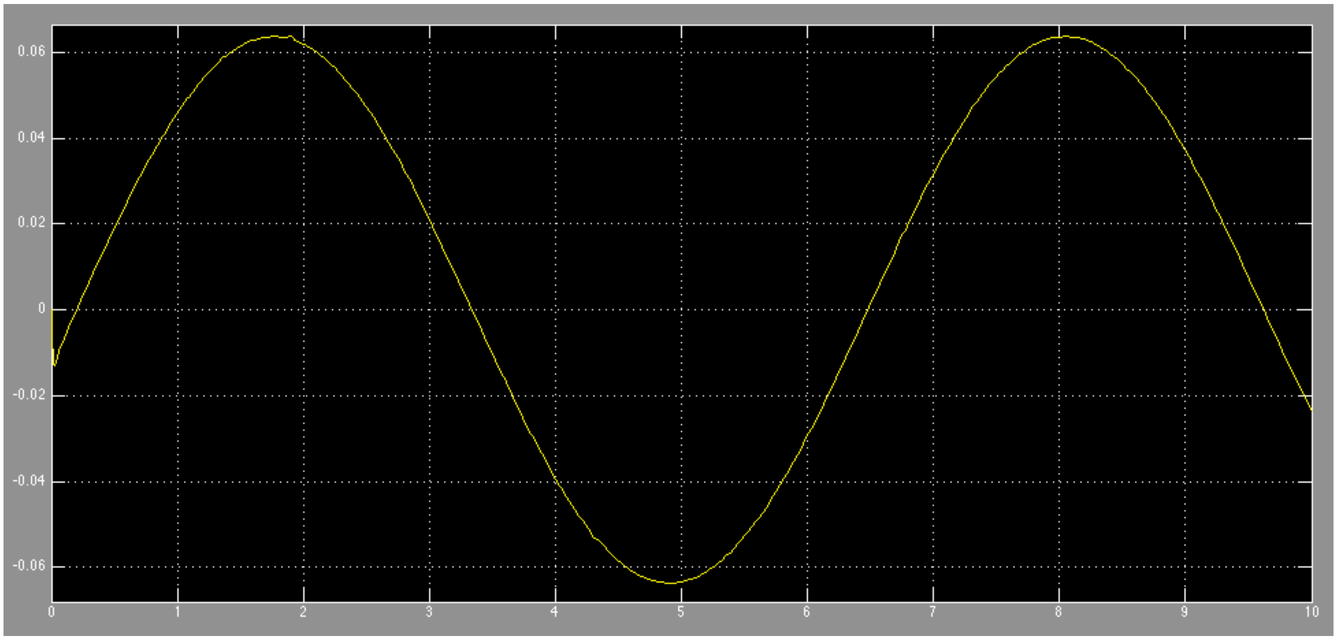
X axis : Time

Y axis = Amplitude of Signal

Magenta = Desired signal

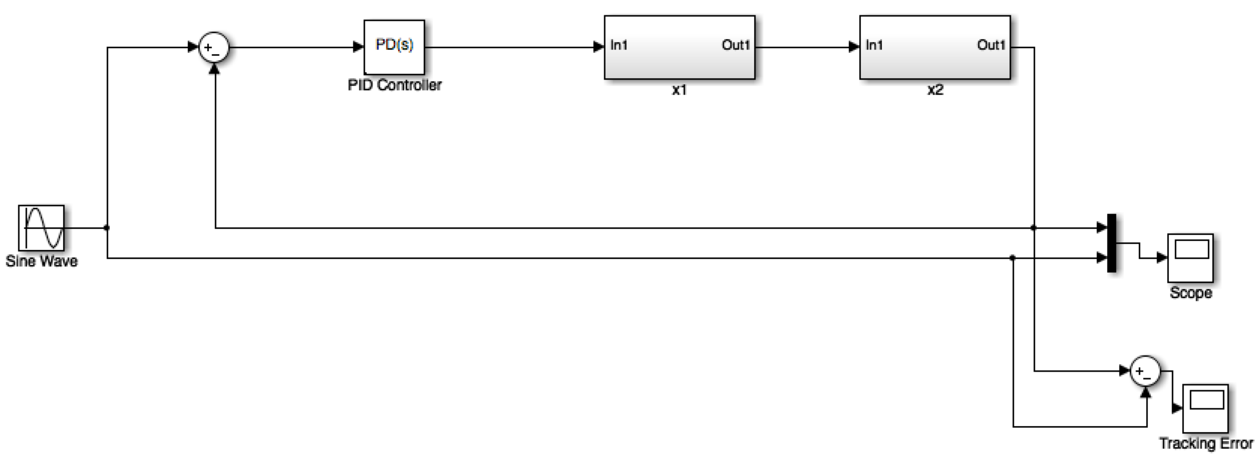
Yellow = Tracking signal



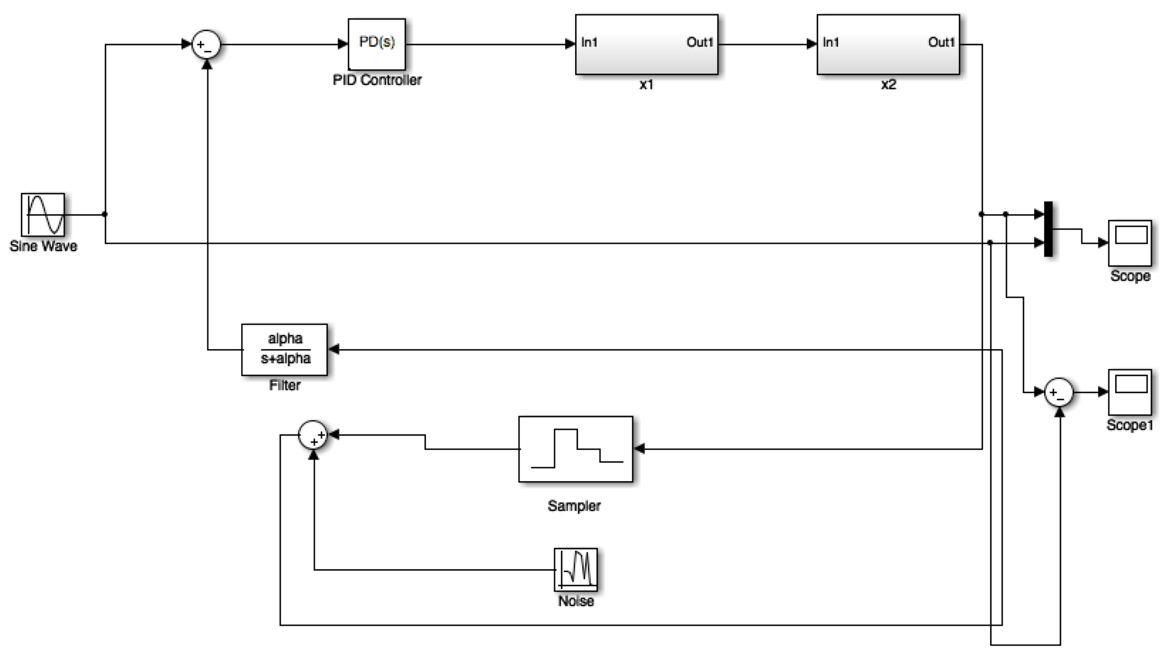


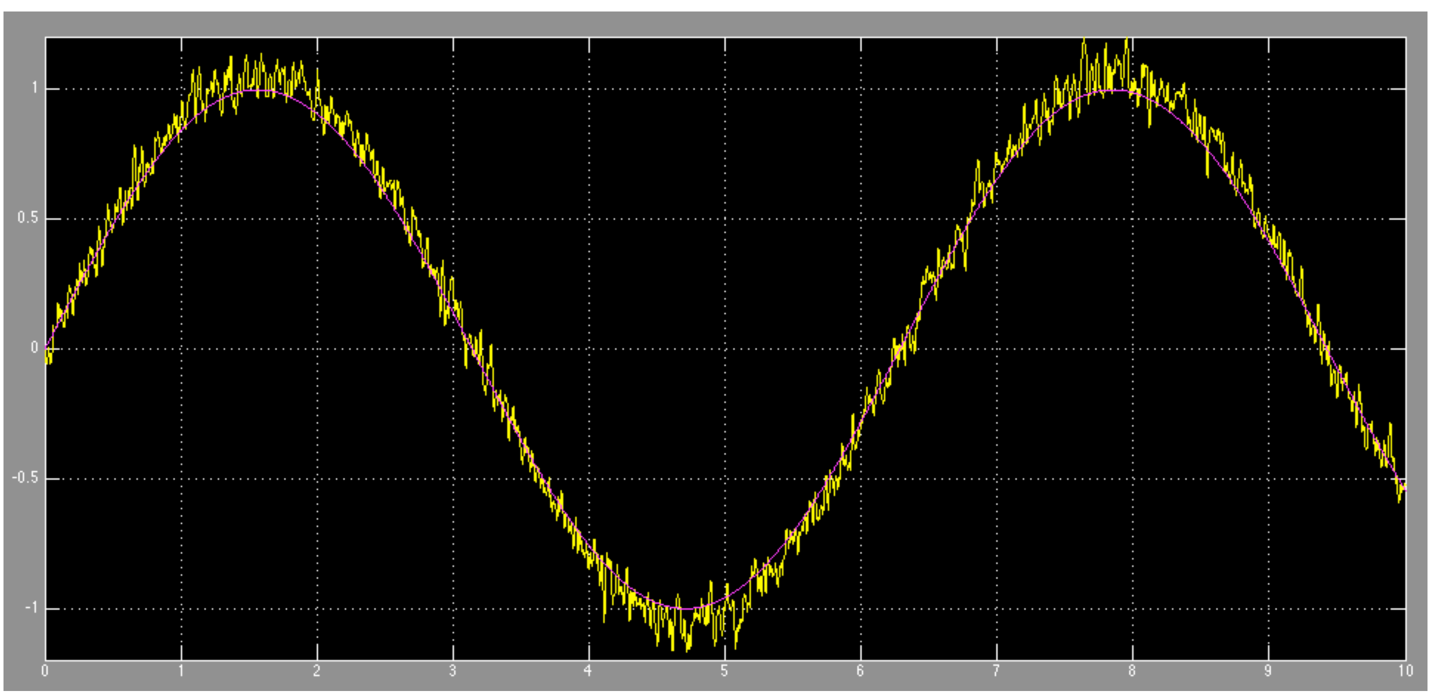
Tracking Error plot

Tracking error grows as the signal grows and decreases as the signal decreases.

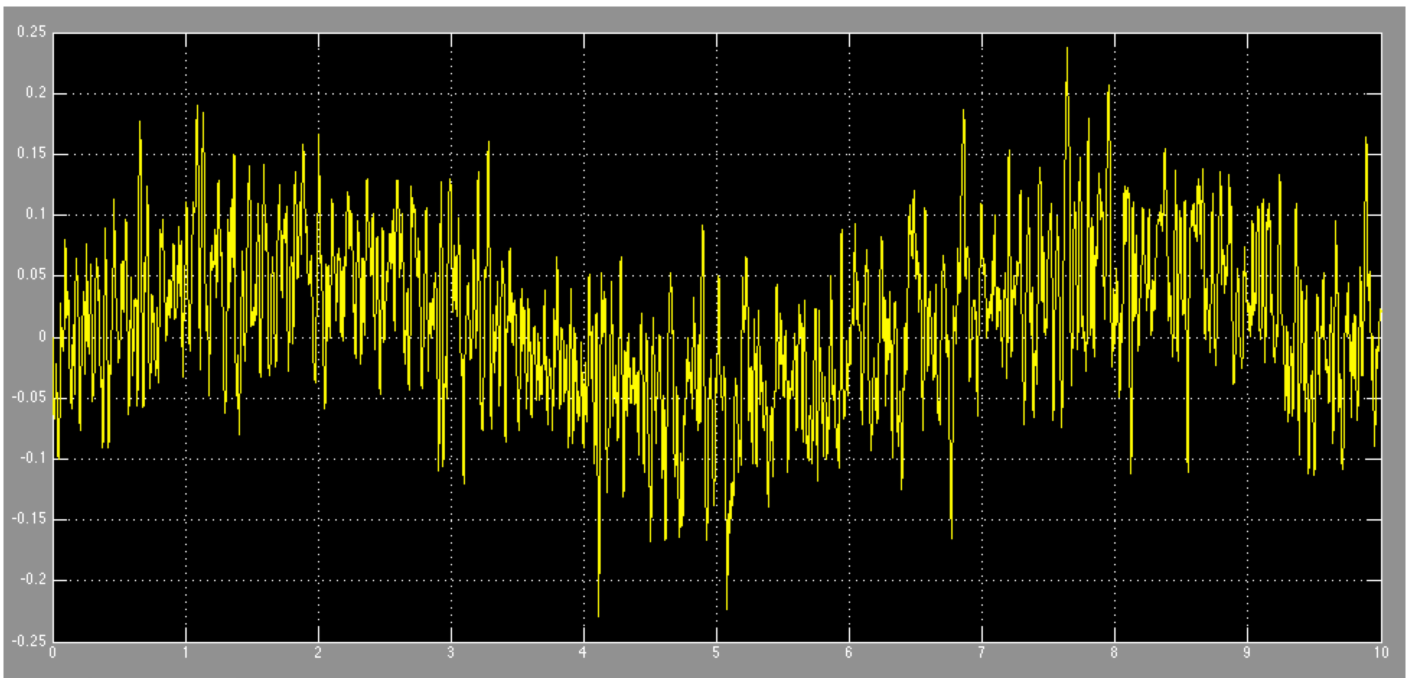


c) Sampler and randomizer plots,Magenta=desired signal,Yellow = response signal,X-axis = time,Y= system state





We can see that the due to the measurement errors from noise and imperfect sampling ,the tracking is not so good as before and there are a lot of fluctuations.The controller has to constantly accommodate the non smooth trajectory and thus results in oscillations such as the ones seen here.Below is a plot of the tracking error.



d)

$$\dot{x} = \alpha(u - x)$$

$$s x(s) = \alpha\left(u(s) - x(s)\right)$$

$$s x(s) = \alpha u(s) - \alpha x(s)$$

$$s x(s) + \alpha x(s) = \alpha u(s)$$

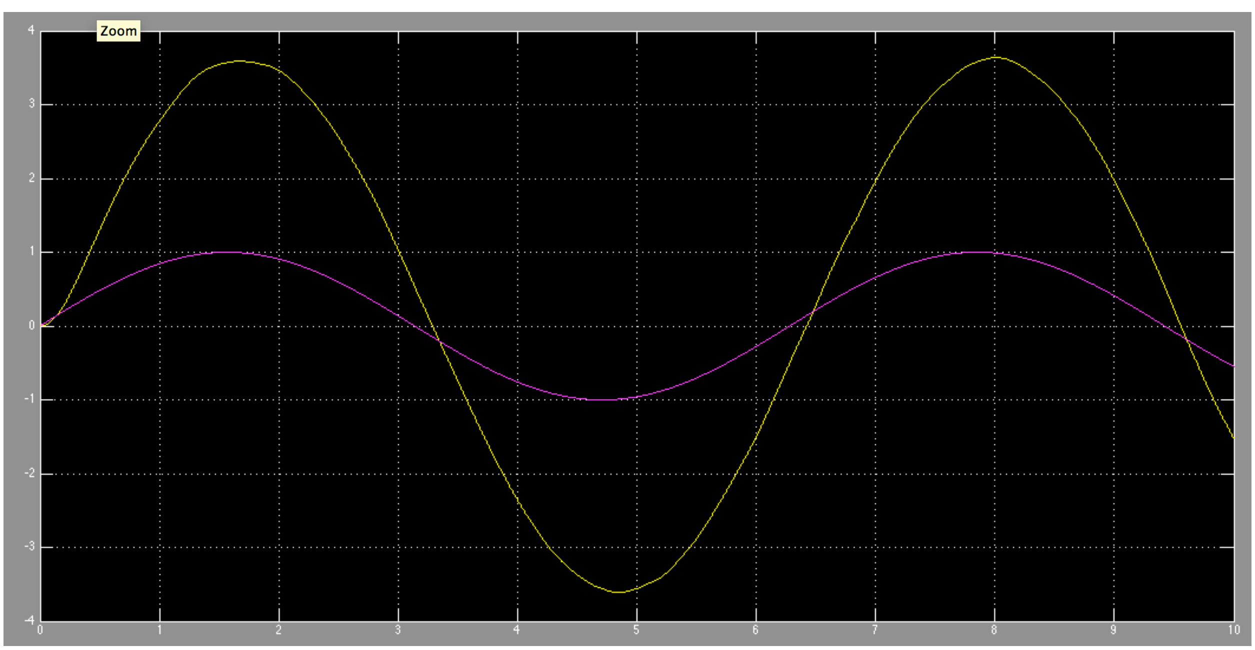
$$(s+\alpha) x(s) = \alpha u(s)$$

$$x(s) = \frac{\alpha u(s)}{(s+\alpha)} \qquad \frac{\alpha}{s+\alpha} u(s)$$

Response plots with filtering. alpha set to 4.77 to maintain phase and remove delay in signal response. P=6.84531551197679 , I=1.69435630048419 , N=1285.39541641463

Magenta=desired signal , Yellow = response signal , X-axis = time,Y= system state

Tracking Error : yellow , X-axis = time,Y= system state



From the above response plot we can see that the noise has been smoothed out by the filtering and that response is smoother but has more error . This is probably due to the approximations made during the filtering.