Aastav Sen *20812878*

ME780 - Assignment 2

Q1.

a) Linear parametric model with uncertainty:

$$\begin{cases} k(y_1 - y_2) = (1 + \Delta_u)u \\ k(y_1 - y_2) = m\ddot{y}_2 + \beta\dot{y}_2 \end{cases} \rightarrow u + \Delta_u u = m\ddot{y}_2 + \beta\dot{y}_2 \qquad [z = u]$$

The plant(s) are formed as follows:

$$\text{(1)}\quad u + \eta = k\underbrace{(y_1 - y_2)}_{\text{output}} \rightarrow \frac{(y_1 - y_2)}{u + \eta} = \frac{1}{k}$$

$$\text{(2)}\quad u + \eta = ms^2 \underbrace{y_2}_{\text{output}} + \beta s y_2 \rightarrow \frac{y_2}{u + \eta} = \frac{1}{ms^2 + \beta s}$$

Thus the linear parametric model with uncertainty is:

$$\begin{bmatrix} ky_1 - ky_2 = u + \Delta_u u \\ m\ddot{y}_2 + \beta\dot{y}_2 = u + \Delta_u u \end{bmatrix}$$

where... $\Lambda(s) = (s+1)^2$

measured... y_1, y_2, u

$$u = ky_1 - ky_2 - \overbrace{\Delta_u u}^{d} \rightarrow \overbrace{u}^{z_1} = k(y_1 - y_2) - d = [k][y_1 - y_2] - \eta$$

$$u = m\ddot{y}_2 + \beta\dot{y}_2 - \underbrace{\Delta_u u}_{d} \rightarrow \underbrace{\frac{u}{\Lambda(s)}}_{z_2} = m\frac{s^2 y_2}{\Lambda(s)} + \beta\frac{s y_2}{\Lambda(s)} - \frac{d}{\Lambda(s)} = [m \ \ \beta]\begin{bmatrix} \frac{s^2 y_2}{\Lambda(s)} \\ \frac{s y_2}{\Lambda(s)} \end{bmatrix} - \eta$$

$\rightarrow z = \left[\frac{k}{m} \right.$

$\rightarrow z - \hat{z}$

b) On-line gradient based parameter estimator with parameter projection to estimate the constants k, m and β

$$\begin{bmatrix} 0.3 \le \beta \le 1.2 \\ \& \ldots \ k \ge 0.2,\ m \ge 5 \end{bmatrix}$$

$$\left\{ \begin{matrix}\text{Gradient} \\ \text{Based}\end{matrix} \right\} + \left\{ \text{Projection} \right\}$$

SPM: $z = \theta^{*T}\phi + \eta$

Gradient Algorithm with Projection:

$$\dot{\theta} = \begin{cases} \Gamma\epsilon\phi & \text{... if parameter bounds are obeyed} \\ 0 & \text{... if parameter bounds are broken} \end{cases}$$

thus...

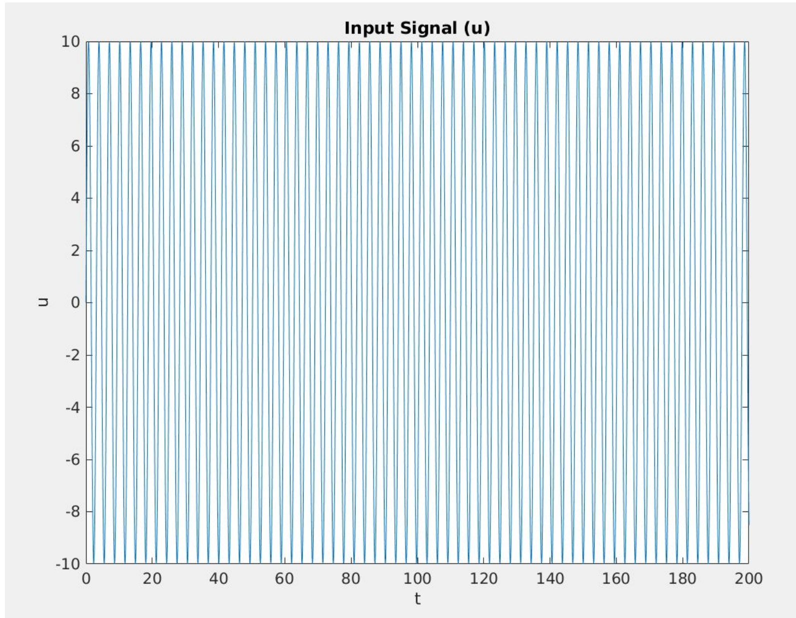
$$\dot{k} = \begin{cases} \Gamma_1 \epsilon_1 \phi_1 = \overset{=1}{\gamma_1}\,(z_1 - \hat{z}_1)\,\overbrace{[y_1 - y_2]}^{\phi_1} & \text{...}\ k \geq 0.2 \\ 0 & \text{...}\ k < 0.2 \end{cases}$$

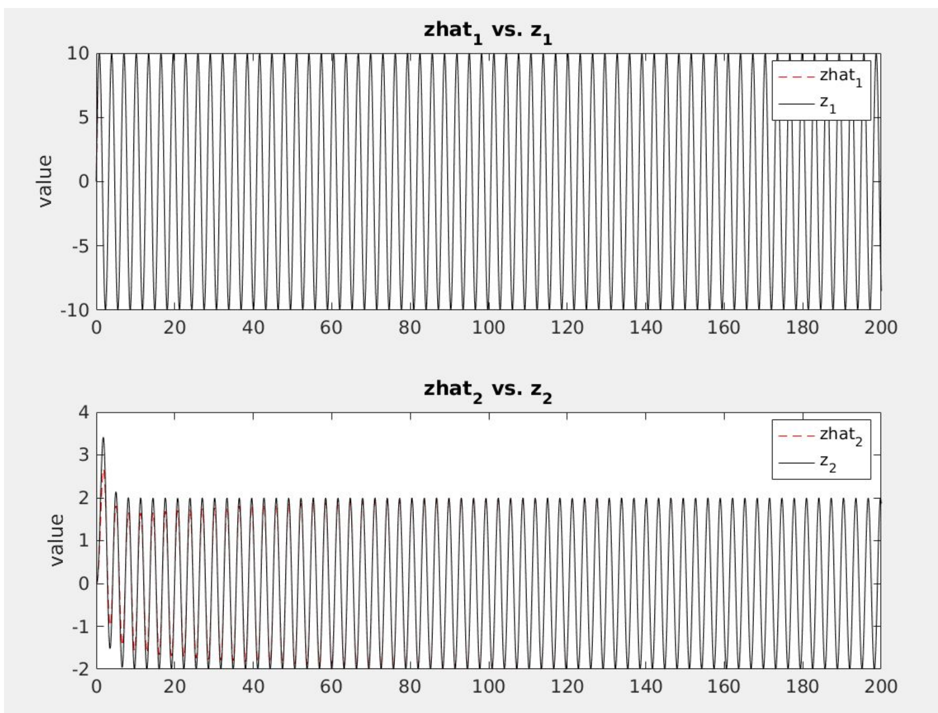
$$\dot{m} = \begin{cases} \Gamma_{21} \epsilon_2 \phi_{21} = \overset{=1}{\gamma_2}\,(z_2 - \hat{z}_2)\,[s^2 y_2/\Lambda(s)] & \text{...}\ m \geq 5 \\ 0 & \text{...}\ m < 5 \end{cases}$$

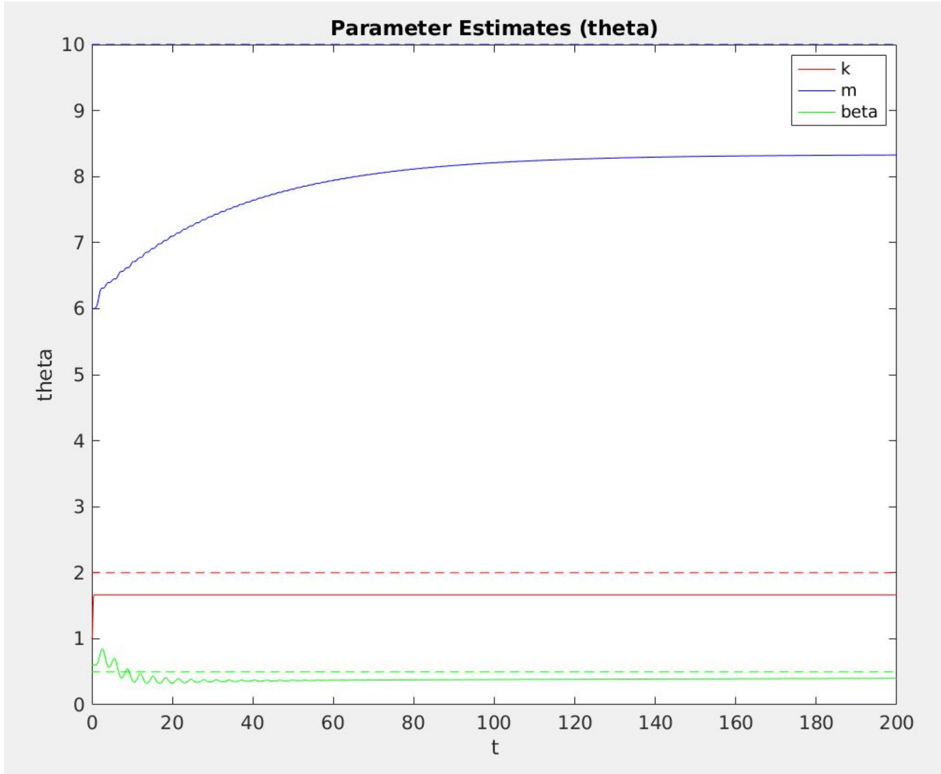
$$\dot{\beta} = \begin{cases} \Gamma_{22} \epsilon_2 \phi_2 = \overset{=1}{\gamma_3}\,(z_2 - \hat{z}_2)\,[s y_2/\Lambda(s)] & \text{...}\ 0.3 \leq \beta \leq 1.2 \\ 0 & \text{...}\ \beta < 0.3 \text{ or } \beta > 1.2 \end{cases}$$

c) Simulate the design for 2 different input signals:

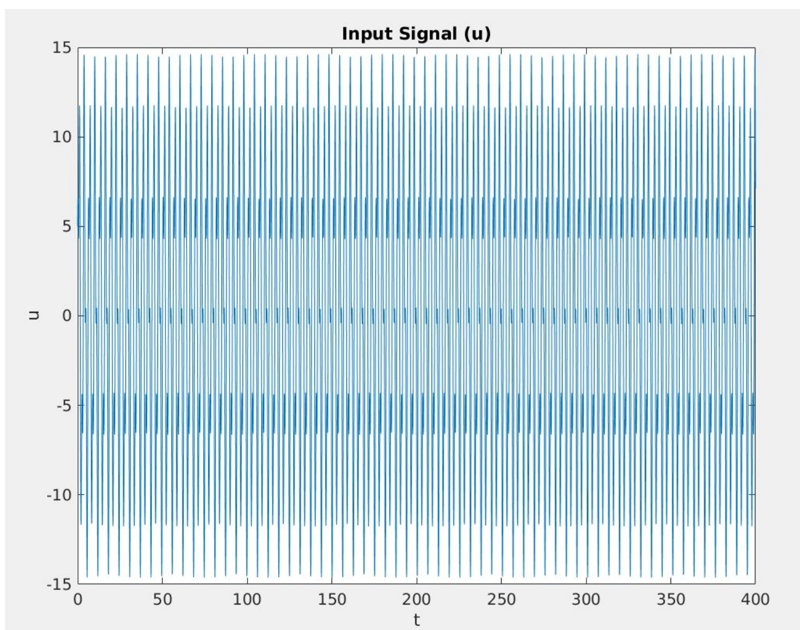
i) u = 10sin(2t) ...and… u_noise = 0.2

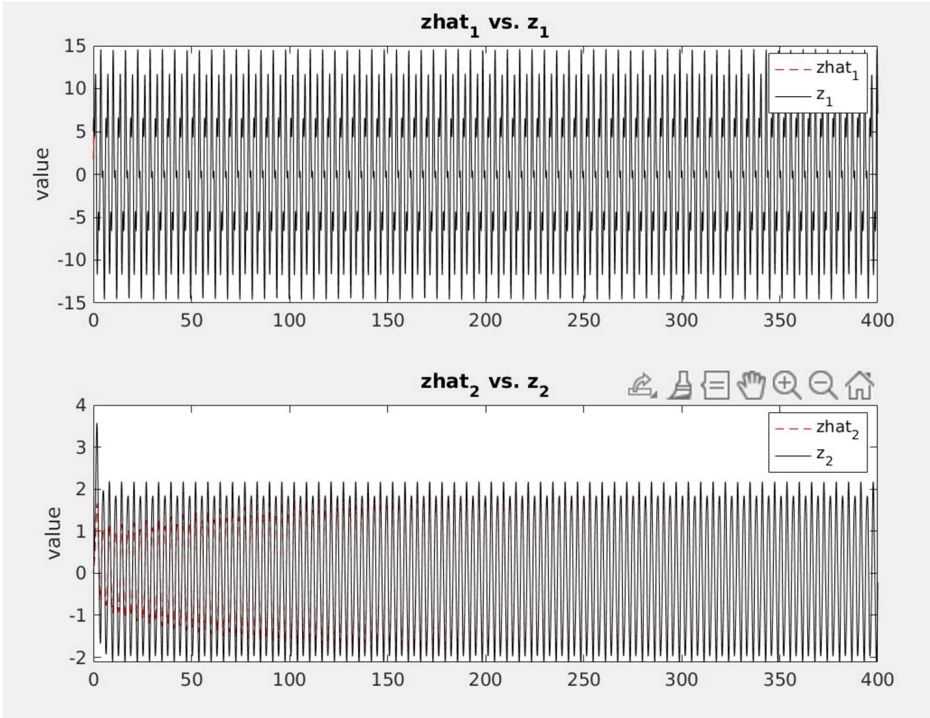

Input Signal (u)
u
t
10
8
6
4
2
0
-2
-4
-6
-8
-10
0 20 40 60 80 100 120 140 160 180 200

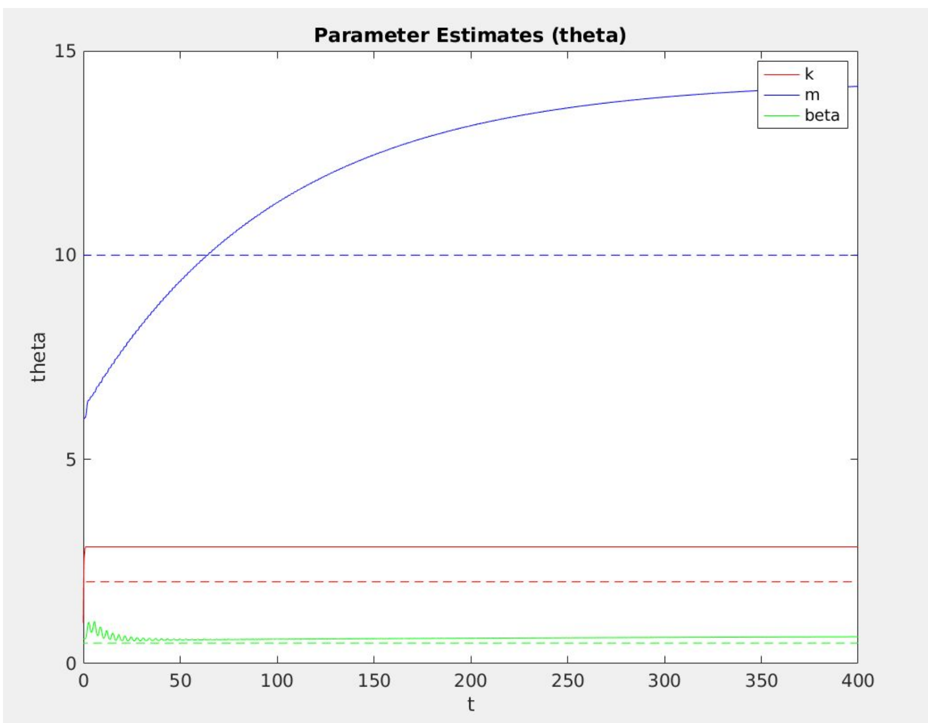

zhat1 vs. z1
zhat1
z1
value
10
5
0
-5
-10
0 20 40 60 80 100 120 140 160 180 200
zhat2 vs. z2
zhat2
z2
value
4
3
2
1
0
-1
-2
0 20 40 60 80 100 120 140 160 180 200



ii) u = 10sin(2t) + 5cos(5t) ...and… u_noise = -0.3







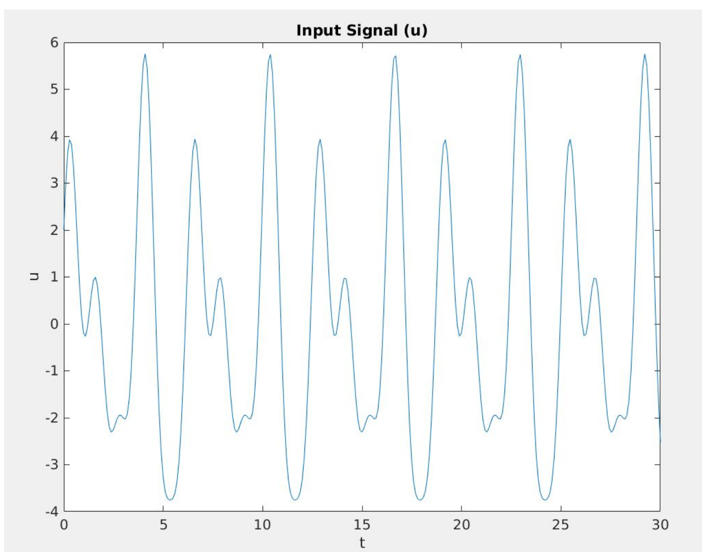
Note that low visibility of the zhat (red dotted line) on the zhat vs. z plots is due to close tracking of z.

- Without the noise our parameter estimates do converge to the true parameter values.
- Positive noise (input case (i)) results in the estimates being too low and negative noise (input case (ii)) results in the estimates being too high, due to integral action of adaptive law that integrates the noise as well.
- Thus, in neither of the cases do the parameter estimates converge to their true values
- For the parameter estimates to converge to the actual parameter values we would require dominantly rich excitation, meaning that the input signal (u) should be sufficiently rich in the low-frequency range such that the amplitude should be greater than the amplitude of the noise. Since this is not true for both our input/ noise signal cases the parameter estimates do not converge to the true parameter values. If we increase the amplitude of the low frequency component in our input signal to be much greater than the amplitude of the low frequency component of the noise signal than our parameter estimates will converge to the true parameter values.

Q2.

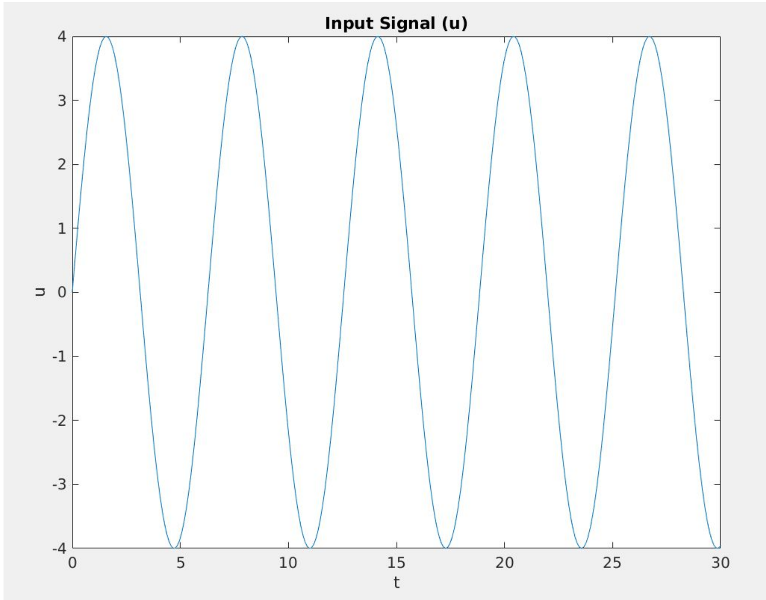
For both problems it is assumed that we do not have access to the derivatives of the output ‘y’ and the input ‘u’, thus a second order stable filter was used to approximate these.

The chosen input ‘u’ in both cases is an oscillating signal that contains 5 frequencies of different amplitudes. The input signal is chosen to be u(t) = 4sin(t)+3sin(2t)+2sin(3t)+sin(5t)+sin(10t)



The input was chosen to contain 5 different frequencies to ensure that it is sufficiently rich of order 2n (when n in this case = length of the unknown parameter vector) then the adaptive observer guarantees that the state observation error as well as the parameter estimate error converges to zero.

The other input signal chosen was a sine wave with a single frequency ie) u(t) = 4sin(t).



$\boxed{\dot{\hat{x}} = A\hat{x} + Bu + K_o(y - C^T\hat{x})}$

$\dot{\tilde{x}} = (A - K_oC^T)\tilde{x}$

Use MATLAB!

$s^3 + (a_2+k_1)s^2 + (a_1+k_2)s + (a_0+k_3) = 0$

want... $s^3 + 12s^2 + 48s + 64$

$\rightarrow s^3 + (2+k_1)s^2 + (2+k_2)s + (2+k_3) = 0$

$$\begin{cases} k_1 = 12-2 = 10 \\ k_2 = 48-2 = 46 \\ k_3 = 64-2 = 62 \end{cases}$$

$A = \begin{bmatrix} -a_p & I_2 \\ & [0] \end{bmatrix}$, $a_p = \begin{bmatrix} a_2 \\ a_1 \\ a_0 \end{bmatrix}$, $\underbrace{B = b_p}$, $\begin{bmatrix} b_2 \\ b_1 \\ 1 \end{bmatrix}$

know $b_1 > 0$
$b_2 > 0$

$$\begin{cases} \dot{x} = \begin{bmatrix} -2 & 1 & 0 \\ -2 & 0 & 1 \\ -2 & 0 & 0 \end{bmatrix} x + \begin{bmatrix} 5 \\ 5 \\ 1 \end{bmatrix} u \\ y = [1\ 0\ 0]\, x \end{cases}$$

In Observer Canonical Form $x_o = \begin{bmatrix} 3 \\ -3 \\ 2 \end{bmatrix}$

$\underline{K_o = a^* - a_p}$ (characteristic eq. coeffs), $K_o = \begin{bmatrix} a_2^* \\ a_1^* \\ a_0^* \end{bmatrix} - \begin{bmatrix} a_2 \\ a_1 \\ a_0 \end{bmatrix} = \begin{bmatrix} 12-2 \\ 48-2 \\ 64-2 \end{bmatrix} = \begin{bmatrix} 10 \\ 46 \\ 62 \end{bmatrix}$

$z = \theta^{*T}\phi \mid \theta^* = \begin{bmatrix} b_p \\ a_p \end{bmatrix}$ $\rightarrow$ since we know $b_0 = 1$... re-order to find $z_{reduced}$ $\Rightarrow$ $y = \dfrac{(b_2s^2 + b_1s + 1)u}{s^3 + s^2a_2 + sa_1 + a_0}$

$ys^3 + ys^2a_2 + ysa_1 + ya_0 = us^2b_2 + usb_1 + u \rightarrow \underbrace{\dfrac{ys^3 - u}{\Lambda(s)}}_{z_r} = \underbrace{[b_2\ b_1\ a_2\ a_1\ a_0]}_{\theta_r^T} \underbrace{\begin{bmatrix} us^2 \\ us \\ -ys^2 \\ -ys \\ -y \end{bmatrix}}_{\phi_r}$ ...δ.. $\Lambda(s) = 1/(s+1)^3$

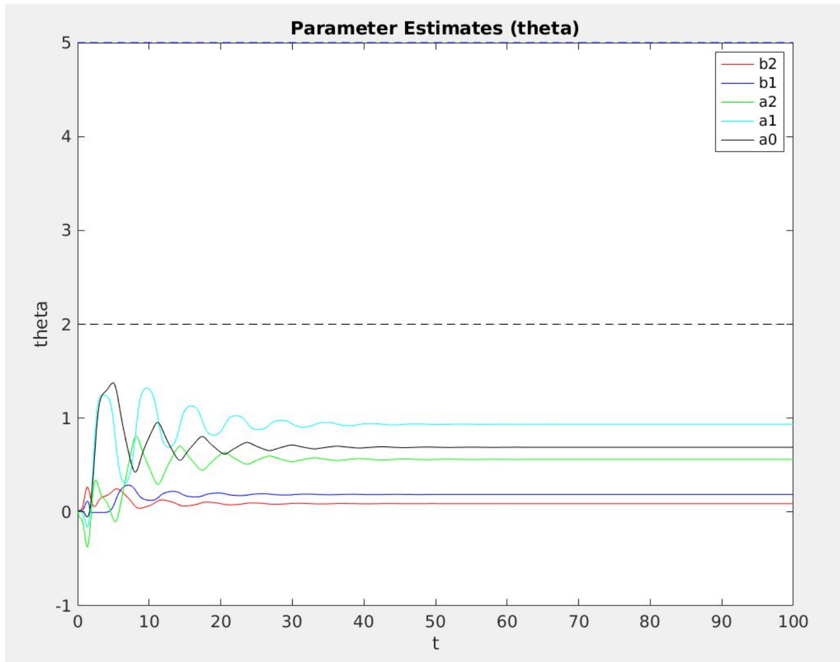
$(s+1)^3 = (s+1)(s^2+2s+1)$
$= s^3 + 2s^2 + s + s^2 + 2s + 1$
$= s^3 + 3s^2 + 3s + 1$

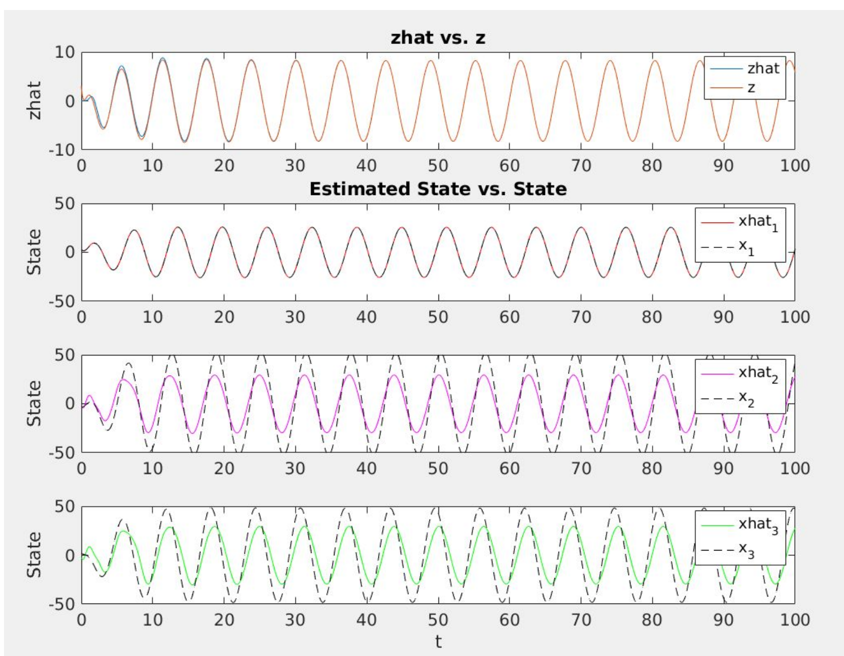
Note: on the following parameter estimate plots, the dashed lines correspond to the true parameter values where the colors are the same for each parameter.

(i) Gradient Estimator (with normalization) (alpha=0.1) (theta initial = [0.01 0.01 0.01 0.01 0.01]) (xhat initial = [3;-3;2])

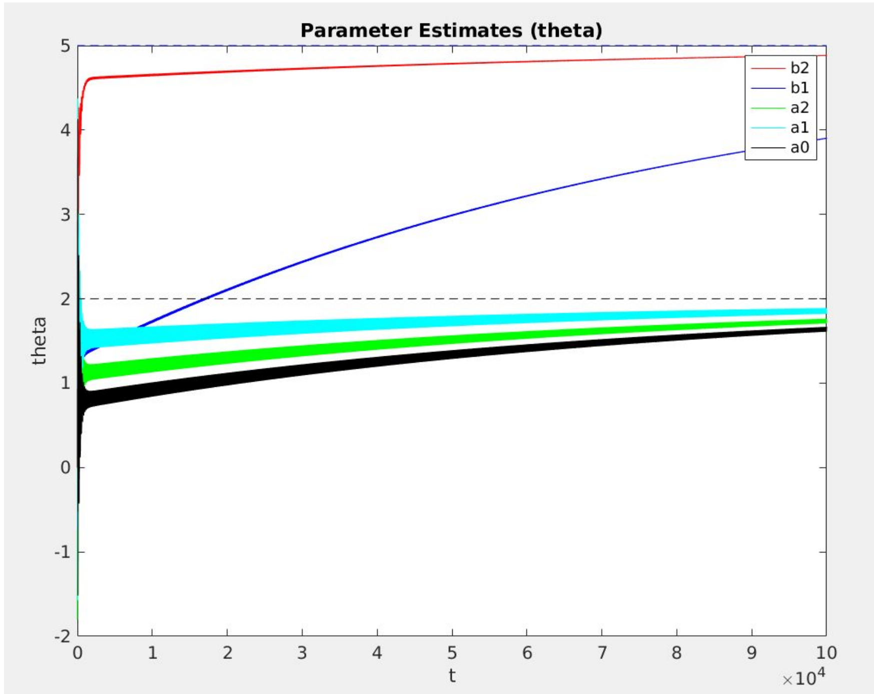
$$\dot{\theta} = \Gamma \varepsilon \phi \qquad \theta(0) = \theta_0$$

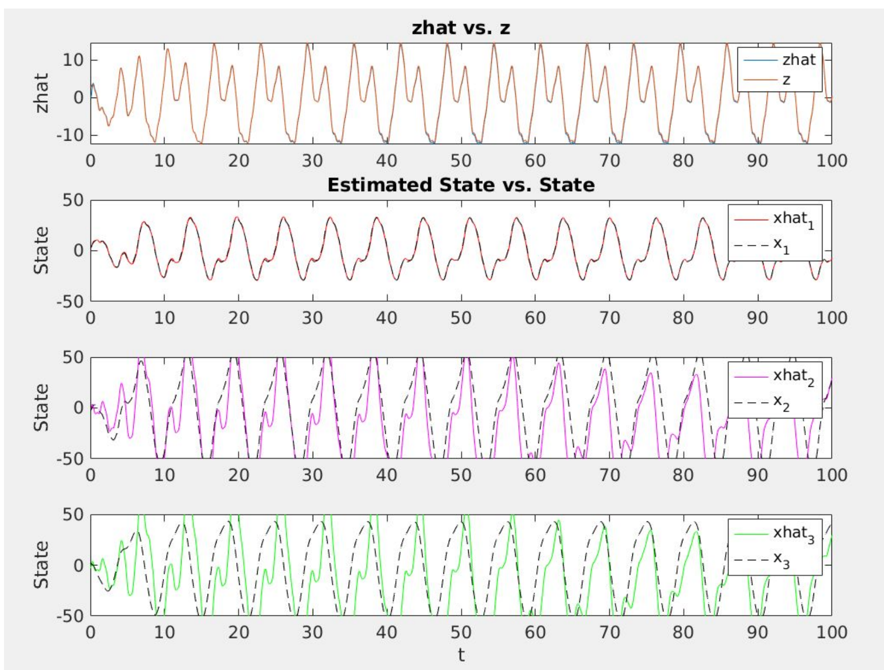
For an input signal of u(t) = 4sin(t), where **(gamma = 10*eye(5))**





For an input signal of u(t) = 4sin(t)+3sin(2t)+2sin(3t)+sin(5t)+sin(10t), where **(gamma = eye(5))**





As can be seen from the above figures, the gradient estimator is able to allow for the parameter estimates to converge to the true parameter values for the second sufficiently rich input case and not for the first (as expected). However, notice the long period of time (100,000 + seconds) required for convergence. Also note that suitable tracking is not achieved in the first input case (where although the frequencies appear correct the amplitudes are incorrect).

(ii) Least-Squares Estimator with Forgetting Factor (with normalization) (alpha=0.1) (beta=0.9) (theta initial = [0.01 0.01 0.01 0.01 0.01]) (P initial = eye(5)) (xhat initial = [3;-3;2])

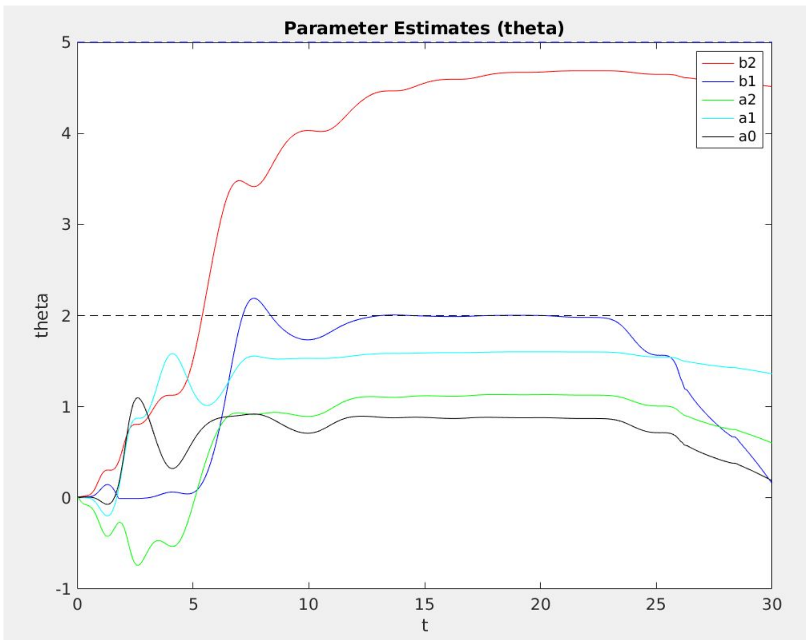
$$\dot{\theta}(t) = P(t)\varepsilon(t)\phi(t), \qquad \theta(0) = \theta_0$$

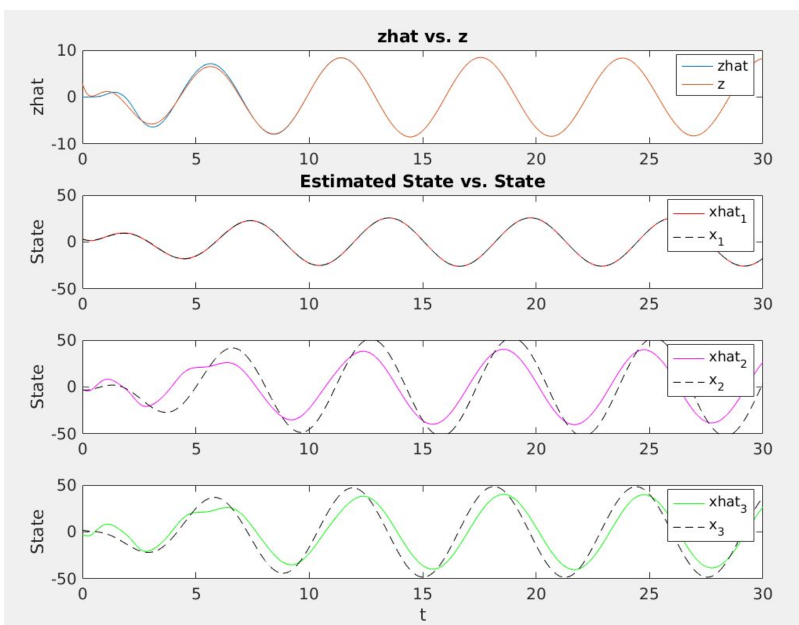
$$\dot{P}(t) = \beta P(t) - P(t)\phi(t)\phi^T(t)P(t), \qquad P(0) = P_0 = Q_0^{-1}$$

β: forgetting factor

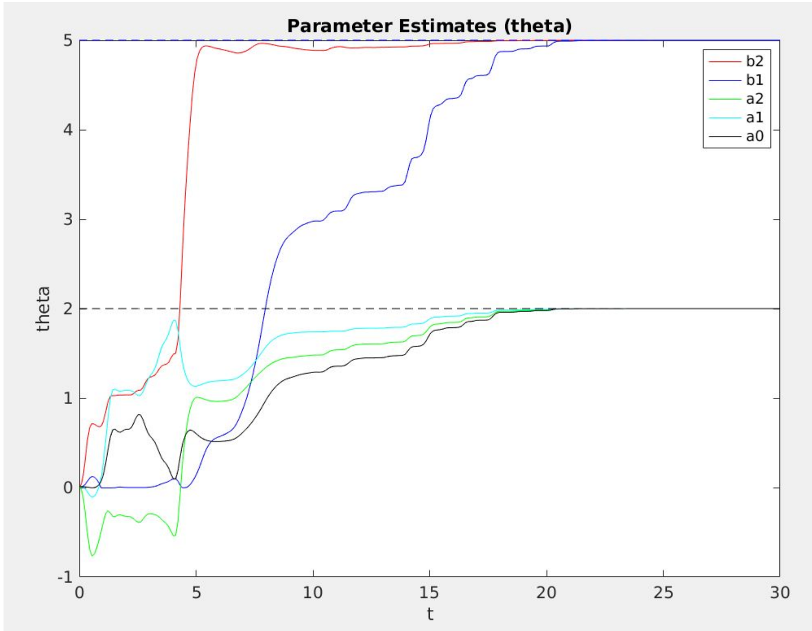
Version with normalization: $\dot{P}(t) = \beta P(t) - P(t)\dfrac{\phi(t)\phi^T(t)}{m_s{}^2(t)}P(t),$

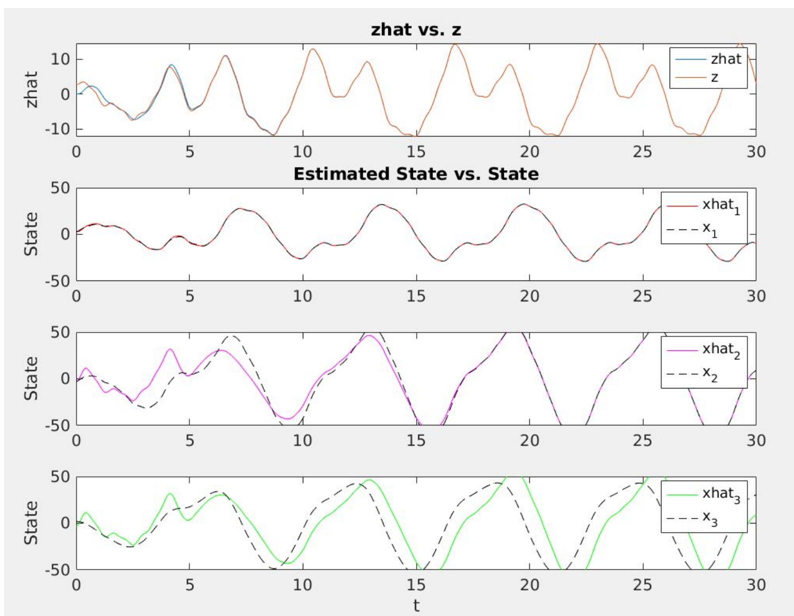
For an input signal of u(t) = 4sin(t)





For an input signal of u(t) = 4sin(t)+3sin(2t)+2sin(3t)+sin(5t)+sin(10t)





As can be seen from the above figures, the least-squares estimator is able to allow for the estimated parameter values to converge to the true parameter values in the second input case but not for the first (as expected). Also note that suitable tracking is not achieved in the first input case (where although the frequencies appear correct the amplitudes are incorrect), in fact, beyond 30s the state estimates diverge.

The least squares estimator for the sufficiently rich input case allows for the parameter estimates to converge to the true parameter values as in the gradient estimator, however, the converge is much faster (~20s as opposed to 100000+s). This is due to the fact that the least squares estimator is derived with a time integral cost function while the gradient estimator is derived from an instantaneous cost function. As zhat closely follows and oscillates about z, the gradient estimator makes very minute oscillating adjustments to the parameter estimates while the least squares estimator is able to identify the ‘overall’ direction of the z error and make adjustments to the parameter estimates in the correct direction, leading to a much faster convergence rate.

Q3.

a) LQR design:

a) state (x) → measurable

$$J = \int_0^{\infty} \left(x^2(t) + \overbrace{0.25}^{\lambda}\, u^2(t) \right) dt$$

↳ $y(t) = x(t)$

$$\dot{x} = \begin{bmatrix} -2 & 1 & 0 \\ -2 & 0 & 1 \\ -2 & 0 & 0 \end{bmatrix} x + \begin{bmatrix} 5 \\ 5 \\ 1 \end{bmatrix} u$$

$$y = [1 \ 0 \ 0]\, x = x$$

$u = -K_c x$

↳ $K_c = \lambda^{-1} B^T P$

Solution with Riccati Eq: $A^T P + PA - PB \overbrace{\lambda^{-1}}^{R} B^T P + \overbrace{CC^T}^{Q} = 0$

↳ solve for P

Solve using $[P, \overset{\text{Feedback gain}}{K}, L] = \text{icare}(A, B, \overset{\to CC^T}{Q}, \underset{\to \lambda}{R}, \overset{\to [\,]=0}{S}, \overset{\to[\,]=0}{E}, \underset{\to [\,] = I}{G})$

↳ closed-loop eigenvalues

↳ get... $K_c = \begin{bmatrix} 3.1308 \\ 0.3260 \\ -2.5351 \end{bmatrix}$ using MATLAB ‘icare’

$$u = -K_c x = -\begin{bmatrix} 3.1308 \\ 0.3260 \\ -2.5351 \end{bmatrix} x$$

b) ALQC block diagram and parameters:

b) $u = -\hat{K}_c \hat{x}$

$\hat{K}_c = \lambda^{-1}\hat{B}^T\hat{P}$ — $\lambda = 0.25$; $\hat{P}$ solved @ each timestep (Riccati Eq.)

$\dot{\hat{x}} = \hat{A}\hat{x} + \hat{B}u - \hat{K}_o(C^T\hat{x} - y)$

Place poles @ -4: $\hat{K}_o = \begin{bmatrix} a_2^* \\ a_1^* \\ a_0^* \end{bmatrix} - \begin{bmatrix} \hat{a}_2 \\ \hat{a}_1 \\ \hat{a}_0 \end{bmatrix} = \begin{bmatrix} 12-\hat{a}_2 \\ 48-\hat{a}_1 \\ 64-\hat{a}_0 \end{bmatrix}$

-> Adaptive Leunberger chosen -> Least Squares with forgetting factor & normalization

-> The same Parametric model as in Q2 was used

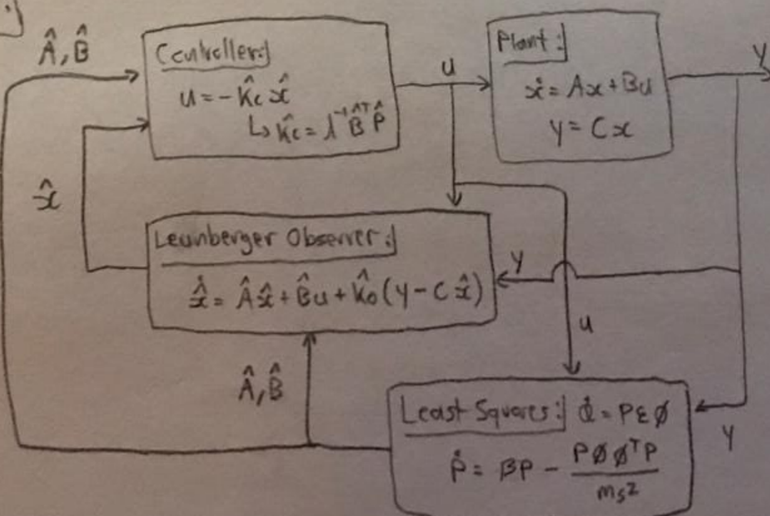
$\left\{ P_0 = eye(5),\ \theta_0 = [1, 1, -1, -1, -1],\ \beta = 0.9,\ \alpha = 0.1 \right\}$

$\hat{x}_0 = \begin{bmatrix} 1 \\ -1 \\ 1 \end{bmatrix}$

$\hat{A} = \begin{bmatrix} -\hat{a}_2 & 1 & 0 \\ -\hat{a}_1 & 0 & 1 \\ -\hat{a}_0 & 0 & 0 \end{bmatrix}$ $\hat{B} = \begin{bmatrix} \hat{b}_2 \\ \hat{b}_1 \\ 1 \end{bmatrix}$

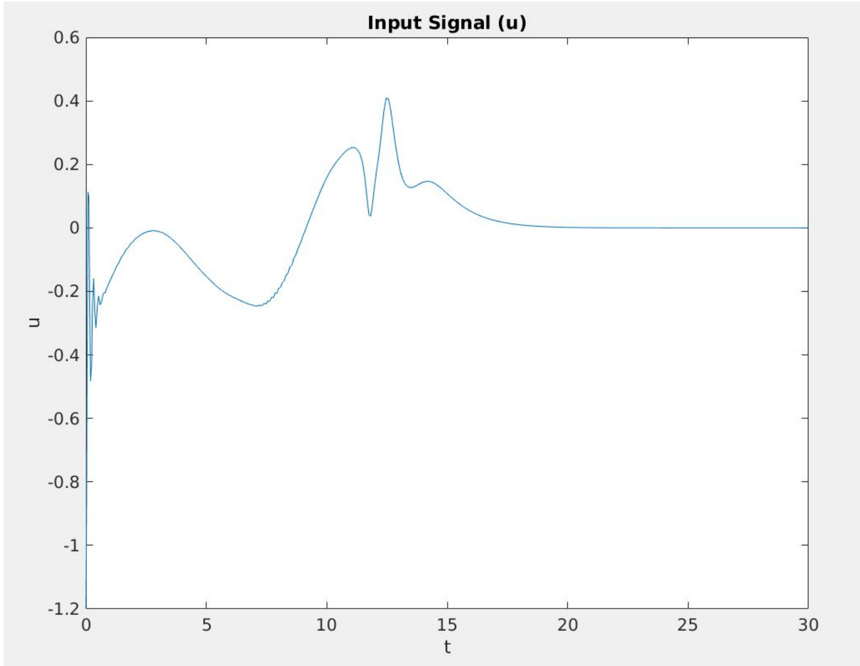
$C = [1\ 0\ 0]$

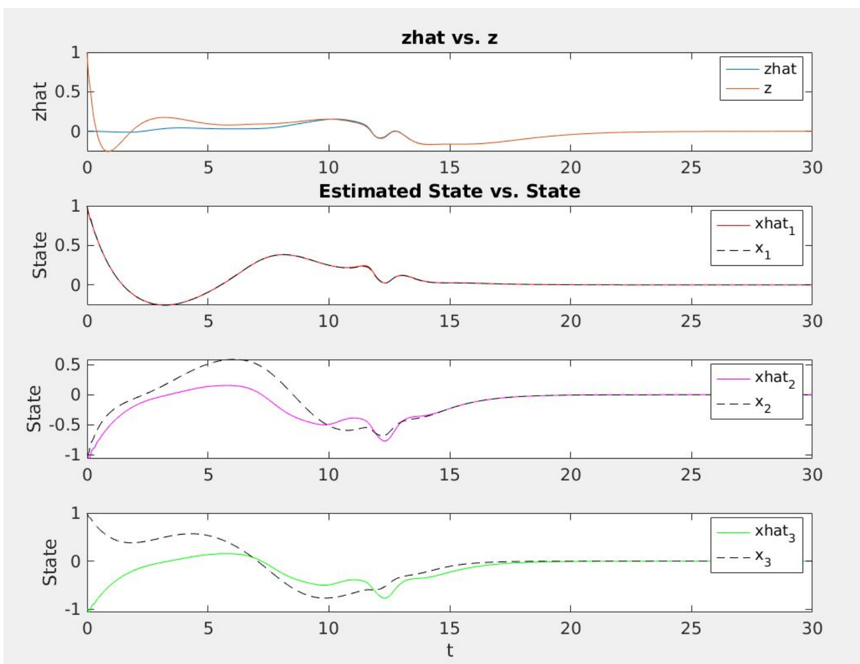
Block Diagram:

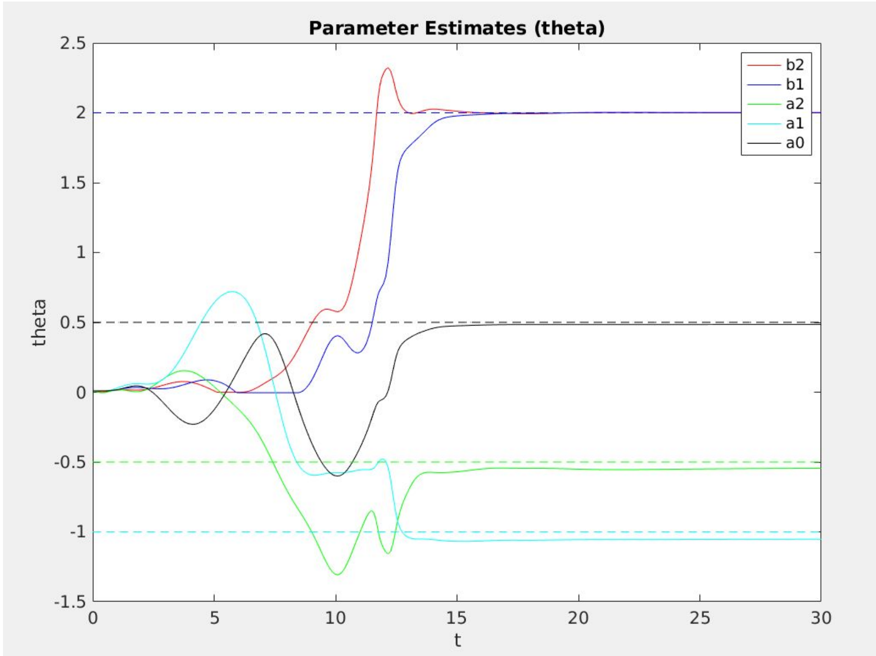


Also note that all initial parameter estimate values = 0.01

c)


Input Signal (u)
u
t
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
-1
-1.2
0
5
10
15
20
25
30


zhat vs. z
zhat
z
Estimated State vs. State
State
xhat$_1$
x$_1$
xhat$_2$
x$_2$
xhat$_3$
x$_3$
t



The plant is controllable and observable, however, it is unstable which implies that the open-loop system poles are in the open right hand plane.

There is no guarantee that one can find a lambda such that the closed loop poles are equal to the roots of a desired polynomial A*, rather LQ control design is used only to ensure that the closed loop system has good robustness properties meaning that the closed loop eigenvalues are in the open left hand side of the plane.

The location of the closed loop system poles (A-BKc) depends on the choice of lambda. A low value of lambda corresponds to low cost control and implies that the input u may become unbounded. A higher value of lambda (high cost control) is required to ensure that the input 'u' is bounded. However, since the plant is unstable (the open loop system is unstable), the value of lambda can not be too high. For example, changing lambda to a value of 1 results in a bounded input 'u' and convergence of y(t)->0. On the other hand making lambda small (=0.01) results in 'u' becoming unbounded. The value of lambda = 0.25 (as in our simulation) results in converging state and parameter estimates suggesting that the ALQC successfully places the closed loop poles in the left hand plane. The output of the LQR controller converges to zero as the closed loop poles (of (A-BKc)) are made stable.

Q4.

Q4.

$$\dot{x} = \begin{bmatrix} -a_2 & 1 & 0 \\ -a_1 & 0 & 1 \\ -a_0 & 0 & 0 \end{bmatrix} x + \begin{bmatrix} b_2 \\ b_1 \\ 1 \end{bmatrix} u + \mathcal{N}\begin{bmatrix} 0.1 & 0 & 0 \\ 0 & 0.2 & 0 \\ 0 & 0 & 0.3 \end{bmatrix} \text{(zero mean)} \quad \text{where } x_0 = \begin{bmatrix} 3 \\ -3 \\ 2 \end{bmatrix}$$

$$y = [1\ 0\ 0]\, x + \mathcal{N}(0, 0.1)$$

a) Discrete time approx. $\Delta t = 0.1$ s:

$$x[k+1] = x[k] + \overset{0.1}{(\Delta t)} f(x[k], u[k]) = \overset{I}{x[k]} + (0.1)\begin{bmatrix} -a_2 & 1 & 0 \\ -a_1 & 0 & 1 \\ -a_0 & 0 & 0 \end{bmatrix} x + (0.1)\begin{bmatrix} b_2 \\ b_1 \\ 1 \end{bmatrix} u + (0.1)\mathcal{N}\overbrace{\begin{bmatrix} 0.1 & 0 & 0 \\ 0 & 0.2 & 0 \\ 0 & 0 & 0.3 \end{bmatrix}}^{\text{zero mean}}$$

$$x[k] = \begin{bmatrix} 1-(0.1)(2) & 0.1 & 0 \\ -(0.1)(2) & 1 & 0.1 \\ -(0.1)(2) & 0 & 1 \end{bmatrix} x[k-1] + \begin{bmatrix} 0.5 \\ 0.5 \\ 0.1 \end{bmatrix} u[k-1] + \mathcal{N}\underbrace{\begin{bmatrix} 0.01 & 0 & 0 \\ 0 & 0.02 & 0 \\ 0 & 0 & 0.03 \end{bmatrix}}_{\text{(zero mean)}}$$

$$y[k] = [1\ 0\ 0]\, x[k] + \mathcal{N}(0, 0.1)$$

b) Design Kalman Filter:

Predict:

$$\hat{x}^-[k] = A\hat{x}[k-1] + Bu[k-1]$$

$$P^-[k] = AP[k-1]A^T + Q$$

Correct:

$$\hat{x}[k] = \hat{x}^-[k] + K[k]\,(y[k] - C^T\hat{x}^-[k])$$

$$K[k] = P^-[k]C\,(C^TP^-[k]C + R]^{-1}$$

$$P[k] = (I - K[k]C^T)\,P^-[k]$$

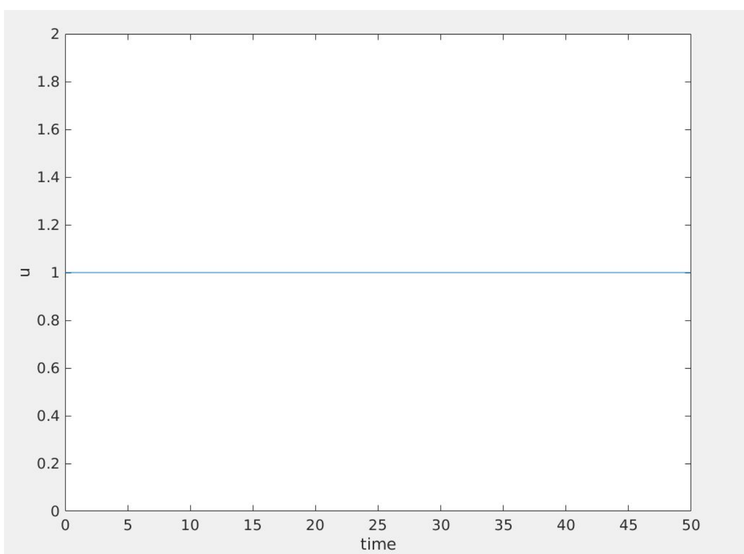
where $A = \begin{bmatrix} -2 & 1 & 0 \\ -2 & 0 & 1 \\ -2 & 0 & 0 \end{bmatrix}$ & $B = \begin{bmatrix} 5 \\ 5 \\ 1 \end{bmatrix}$ $\quad Q = \begin{bmatrix} 0.1 & 0 & 0 \\ 0 & 0.2 & 0 \\ 0 & 0 & 0.3 \end{bmatrix}$ & $R = 0.1$

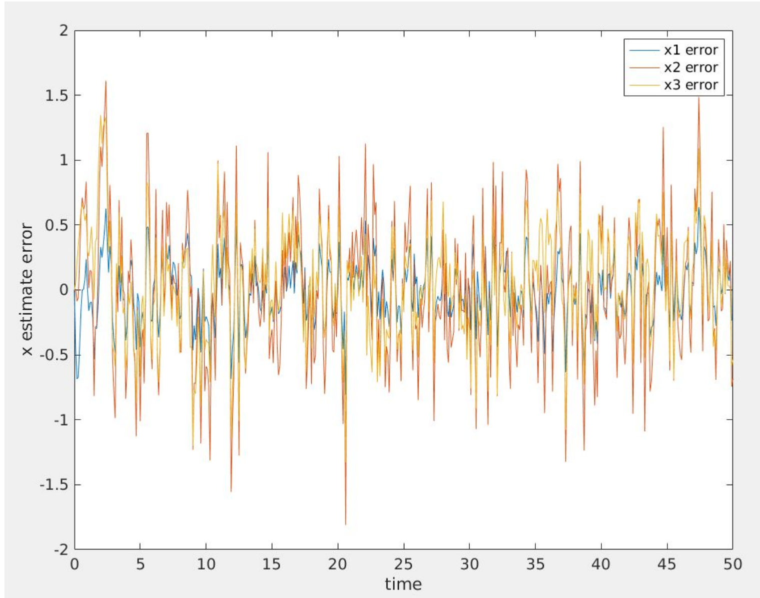
$C = [1\ 0\ 0]$ $\quad P_0 = \text{eye}(3)$

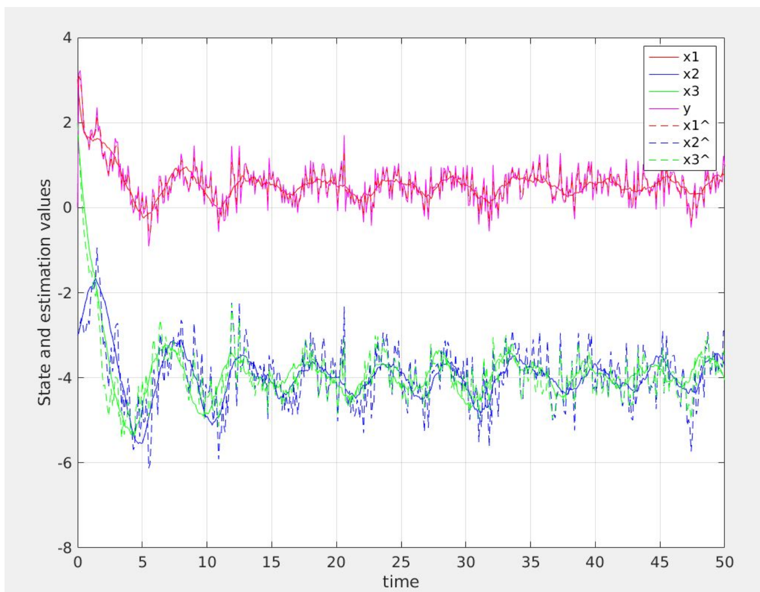
$x_0 = \begin{bmatrix} 3 \\ -3 \\ 2 \end{bmatrix}$

c)

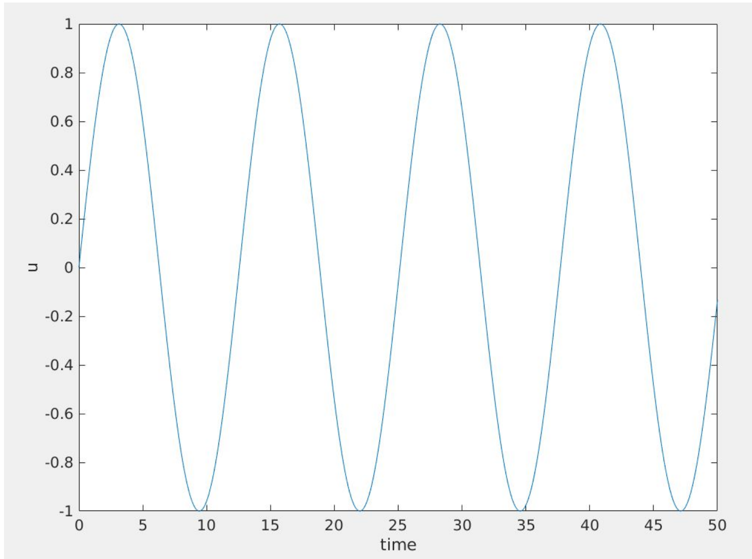
i) u(t) = 1: constant input case

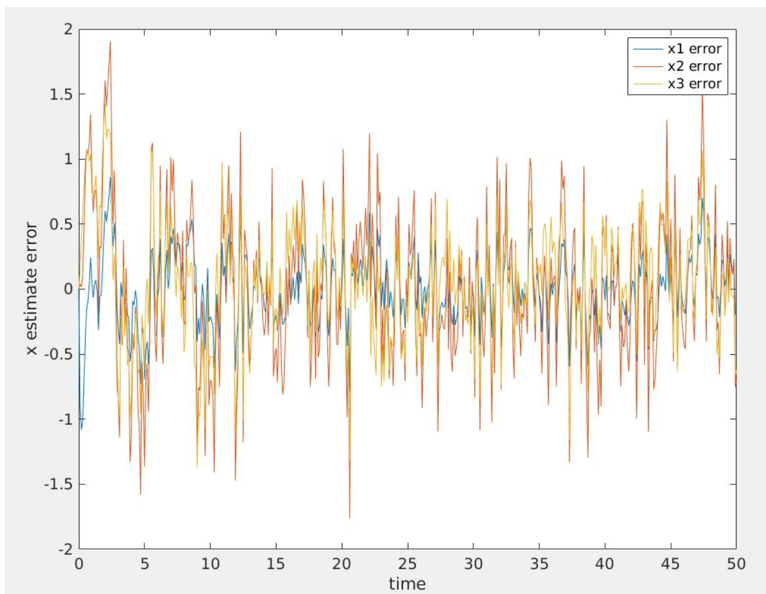


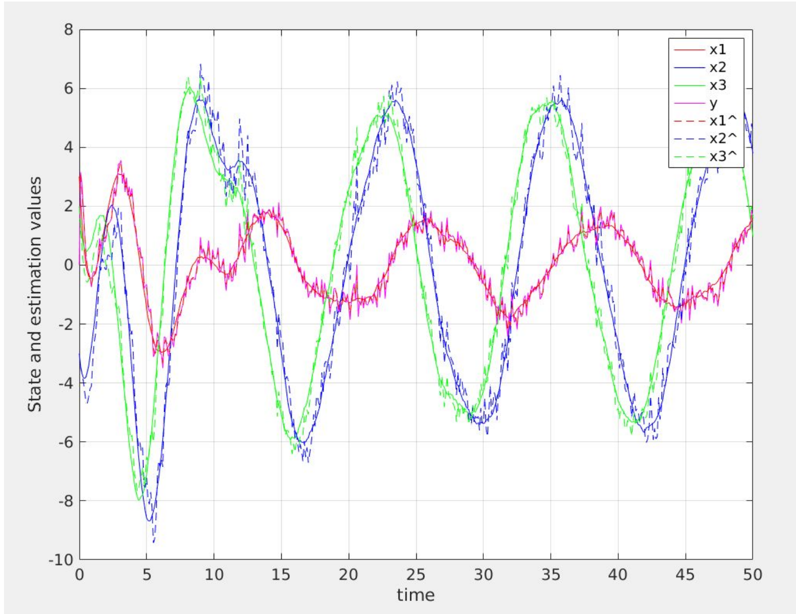

x1 error
x2 error
x3 error
x estimate error
time


x1
x2
x3
y
x1^
x2^
x3^
State and estimation values
time

ii) u(t) = sin(0.5t): sinusoidal input case







System is completely observable (as can be seen from the observability matrix). In both input cases it appears that the state estimates track the actual state reasonably well.

It should also be mentioned that this is the ideal case in which the sensor noise (R) was assumed to have the true value (0.1), however, in application the true sensor noise variance is often not known where assuming a higher value of R would result in a fast response and bad tracking, while assuming a lower value of R would result in a slow response with reasonable tracking. Here in both cases we see an ideal response with good tracking of the state.

Unlike the Leunberge observer (as seen in Q2) which is deterministic, the Kalman Filter explicitly incorporates a noise model for both state and output processes and is more appropriate and, in general, performs better for stochastic systems. This is our case in Q4 as we have noise in both our plant and measurement model. On the contrary (as seen in Q2) the Leunberger observer needs sufficient excitation to ‘overcome’ the noise and produce good state estimates.