BY S.VISHAL
CH18B020

Assignment-5 CH5115

Q1)

a) Formulating a model

We can formulate a combined model as:

$$y[k] = a_1 y[k-1] + a_2 y[k-2] + b_1 u[k-1] + b_2 u[k-2] + b_3 u[k-3]$$

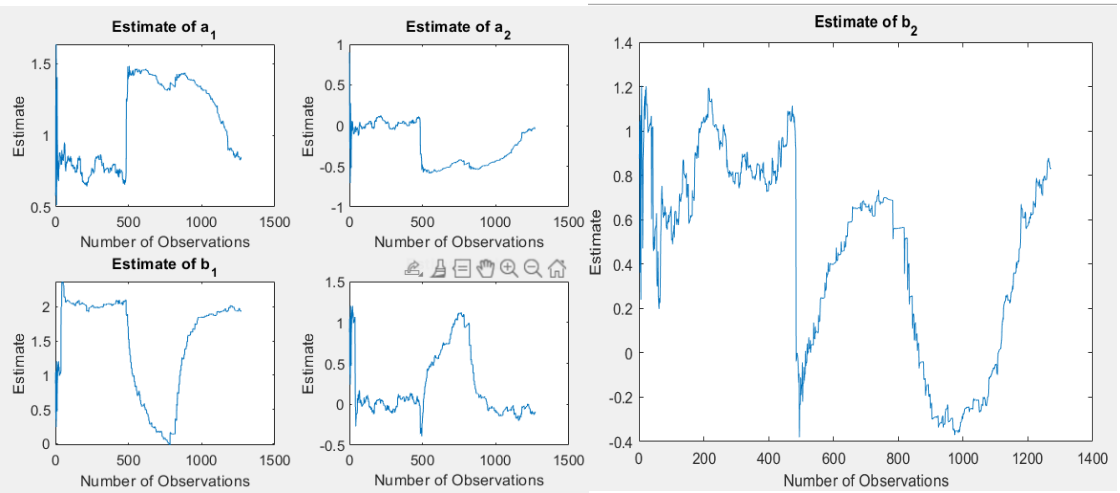
Recursive Least squares (with a forgetting factor) can be performed on this, and switching time can be identified if we observe the following:

i. There is a sudden change in the value of a_1
ii. a_2, b_3 show a sudden jump from about zero to non-zero (insignificant to significant)
iii. b_1 shows a sudden jump from zero to non-zero (insignificant to significant)
iv. Or vice-versa of two and three

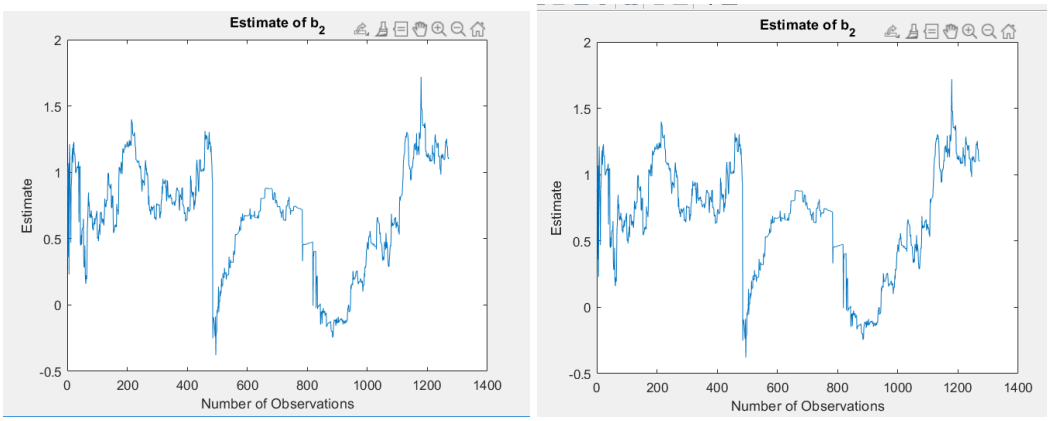
b) Optimising the forgetting factor

High forgetting factor (close to 1 => more memory)

λ = 0.99

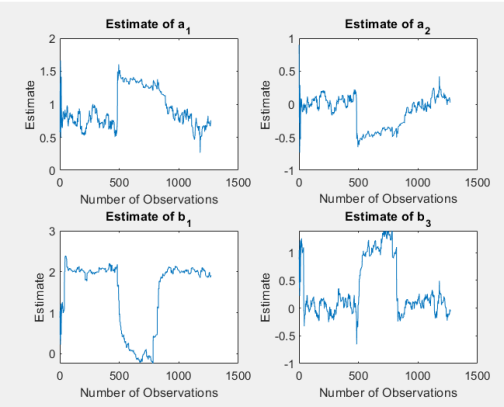


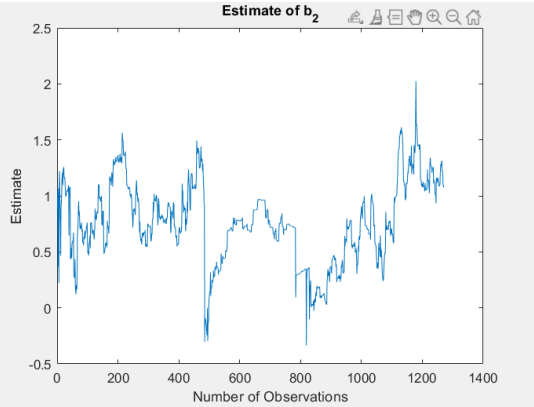
λ = 0.98



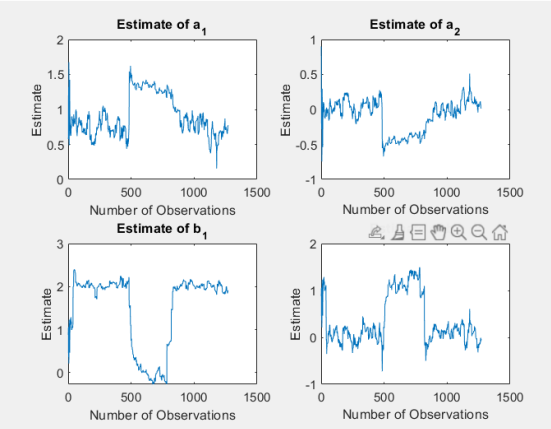
BY S.VISHAL
CH18B020

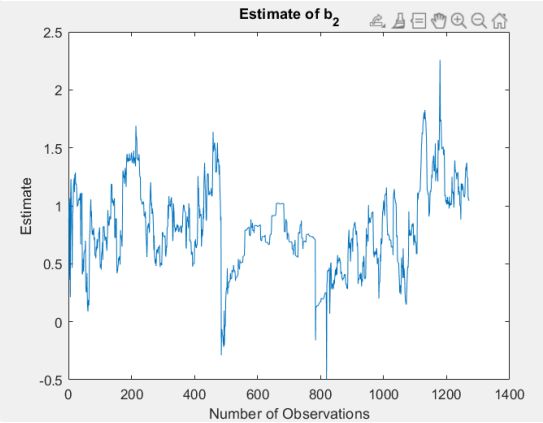
λ = 0.97



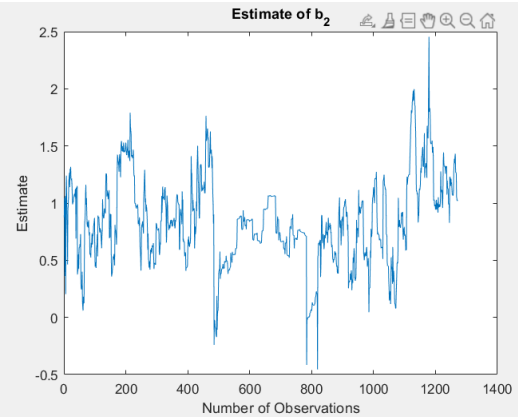


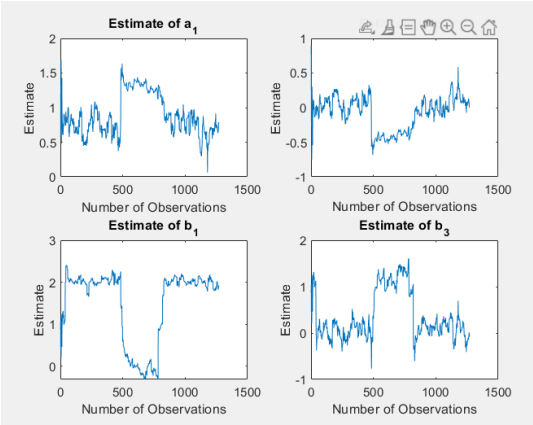
λ = 0.96





λ = 0.95





We want a constant b_2 because it is common to the two models. It varies because of two reasons: i) There is too less forgetting (high λ) which causes the recursive least squares to miss the regime change or ii) There is too much forgetting which causes the algorithm to oscillate parameter values randomly due to randomness in data.

Accounting for these two reasons, λ = 0.97 seems to be a reasonable factor.

When b1 is around 0, model 2 regime is active. When a2 and b3 are around zero, model 1 regime is active.

From the graphs approximately, samples 500-800 is the region where model 2 is active.

Switching times: 500 & 800

Final estimate here is the mean of the estimates in the regime where the particular coefficient is active.

a1_1 = 0.759

a1_2 = 1.338

a2 = -0.442

b1 = 1.979

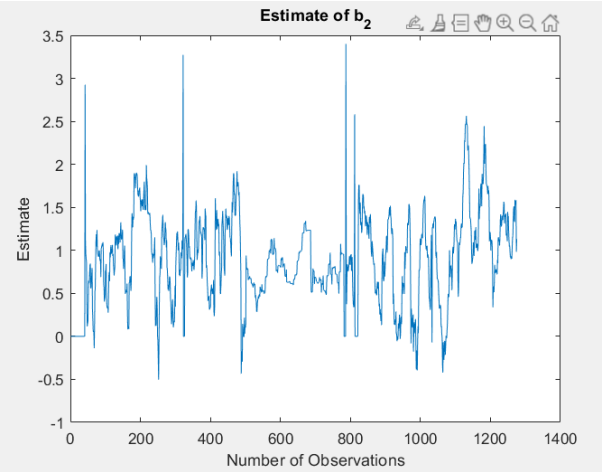
b2 = 0.761

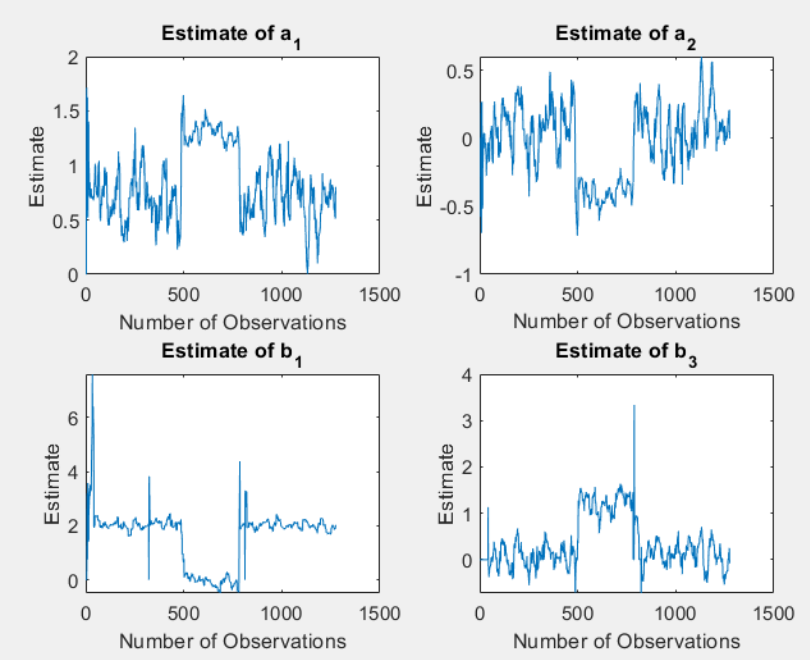
b3 = 1.076

c) sliding window model

$1/(1-\lambda_{opt}) = 33.33$. This must be the value around which the optimal window length might lie.

Among possible window lengths tried out (5, 15, 25, 27, 30, 33, 50, 75), using similar reasons as compared to the previous question, window length of 27 seems optimal.





From the graphs approximately, samples 503-797 is the region where model 2 is active.

Switching times: 500 & 800

a1_1 = 0.7082

a1_2 = 1.257

a2 = -0.3733

b1 = 2.089

b2 = 0.885

b3 = 1.212

d) sliding window vs forgetting factor

Comparing only the mse,

Forgetting factor model mse = 300 < sliding window model mse= 296.

They are roughly the same with **sliding window** showing superior results. This might be because by choosing the appropriate window size, we completely discard the irrelevant past, ensuring better estimates for the active regime.

However given that the regimes were selected by just viewing the graphs the and since the mse-s are close by, it could be that either the methods are equally good or forgetting factor might be the better one.

BY S.VISHAL
CH18B020

Q2)

a) Minimal realization

The controllability and the observability matrices are full rank, so the system is a minimal realization.

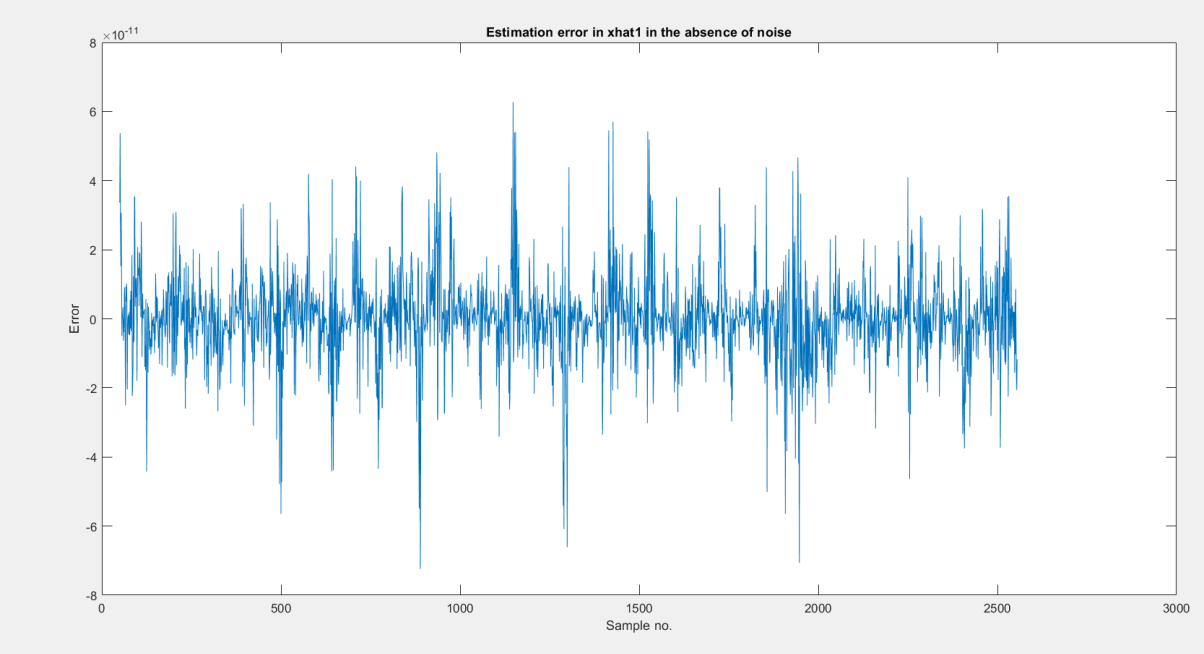
b) Observer Design

The required observer is designed using the '*place*' function.

$L = [0.7, -8098.815, -9583.126]^T$

c) Implement Observer

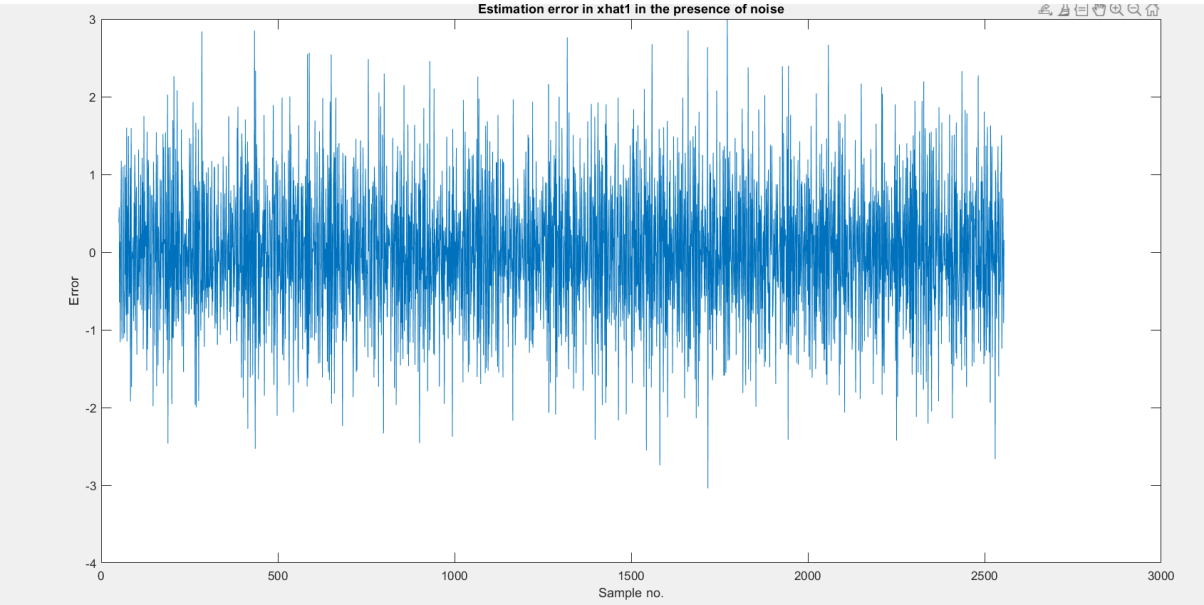
Initial state of the system is given as [0,0,0] to both the model simulation function (lsim). The observer is given a noisy version of the same (a realization of a random normal variable with mean 0 and variance 1).



The plot of error is made from the 50th observation, to ensure the error is stabilized (the high initial error would be because of errors in the initial estimate X0). As you can see after the 50th observation the error stabilizes to zero. (the above graph is in the order of 10^{-11})

d) Behaviour under noisy conditions

Initial state of the system is given as [0,0,0] to both the model simulation function (lsim). The observer is given the same X0 as in earlier case so that the errors are comparable.



Although on an average errors are 0 (and looks bounded), we don't see a convergence to 0.

Varying the Observer poles:

Qualitatively we can say that lower eigen values will give more weightage to the measurements, and higher eigen value gives more weight to the initial estimate. Our initial estimate can be wrong, there can be modelling errors and because of noise, the measurement available is a corrupted form of the actual output. This means we have error contributions from both places, so we need to strike a balance.

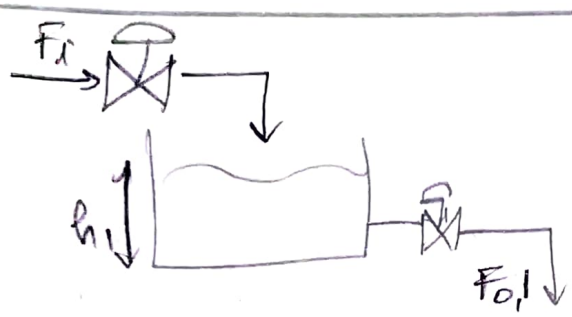
Eigen values used:

i) [0.1,0.2,0.3]
ii) [0.912,0.99,0.999]
iii) [0.05,0.055,0.059]

```
Mean Absolute Error in the absence of noise =
    0.0003    3.1128    3.6830

Mean Absolute Errors in the presence of noise =
   1.0e+04 *

    0.0001    1.0074    1.1919

Mean Absolute Errors(high eigen value case) =
   1.0e+08 *

    0.0000    1.4806    1.7520

Mean Absolute Errors(low eigen value case) =
   1.0e+04 *

    0.0002    4.4988    5.3230
```

Low eigen value works better. High eigen value give less weightage to the measurements. Since we have a noisy condition (modelling errors as well as measurement errors), high eigen value results in a higher mean absolute error. To account for the noise, we would want to use the measurements to correct the states. That is why low eigen value works better. (And also, the error in measurements is low, so the measurements are useful to a certain extent)

③

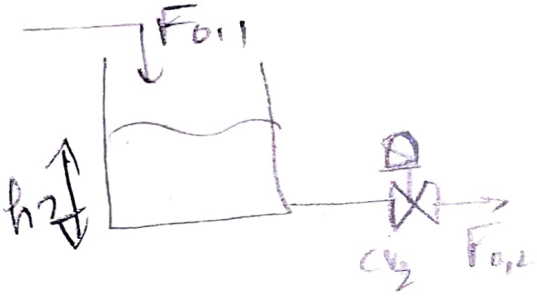


Mass balance over tank 1:

$$\frac{A_1 dh_1}{dt} = F_i - F_{o,1}$$

$$= F_i - C_{v_1} h_1$$

$$\Rightarrow \frac{A_1 dh_1}{dt} = \frac{F_i}{A_1} - \frac{C_{v_1} h_1}{A_1} \quad \text{—①}$$



Mass balance over tank 2:

$$\frac{A_2 dh_2}{dt} = F_{o,1} - F_{o,2}$$

$$= C_{v_1} h_1 - C_{v_2} h_2$$

$$\Rightarrow \frac{dh_2}{dt} = \frac{C_{v_1}}{A_2} h_1 - \frac{C_{v_2}}{A_2} h_2$$

$$\Rightarrow \begin{bmatrix} \frac{dh_1}{dt} \\ \frac{dh_2}{dt} \end{bmatrix} = \begin{bmatrix} \frac{-C_{v_1}}{A_1} & 0 \\ \frac{C_{v_1}}{A_2} & \frac{-C_{v_2}}{A_2} \end{bmatrix} \begin{bmatrix} h_1 \\ h_2 \end{bmatrix} + \begin{bmatrix} 1 \\ 0 \end{bmatrix} F_i$$

$$\Rightarrow \dot{\underline{x}} = A_c \underline{x} + B_c u$$

$$\underline{x} = \begin{bmatrix} h_1 \\ h_2 \end{bmatrix} ; \; u = F_i$$

$$A_c = \begin{bmatrix} \frac{-C_{v_1}}{A_1} & 0 \\ \frac{C_{v_1}}{A_2} & \frac{-C_{v_2}}{A_2} \end{bmatrix} \quad \text{and} \quad B_c = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$$

If only $F_{o,2}$ is measured

$$F_{o,2} = C_{v_2} h_2 \Rightarrow y = \begin{bmatrix} 0 & C_{v_2} \end{bmatrix} \begin{bmatrix} h_1 \\ h_2 \end{bmatrix}$$

$$\Rightarrow C_C = \begin{bmatrix} 0 & C_{V_2} \end{bmatrix}$$

BY S.VISHAL
CH18B020

Q3)

b) Discretising the state space model

The model is discretized using c2d method in MATLAB. Sampling time t_s = 0.1 s. Stored in object named *discrete.*

```
A =                    B =

                                C =
    0.9355         0       0.0806
    0.0624    0.9355       0.0027
                                        0    0.8000
```

c) Observability

In estimation terms, the question is basically asking us to find whether it is possible to estimate the states. (since, in this model, the states are the heights of the tank.) We answer this using the observability condition.

```
On =

         0    0.8000
    0.0499    0.7484
```

Rank(O_n) = 2; So the system is observable.

We can estimate the heights by measuring the outlet flow rate alone.

d)

The system is assumed to be in steady state initially.

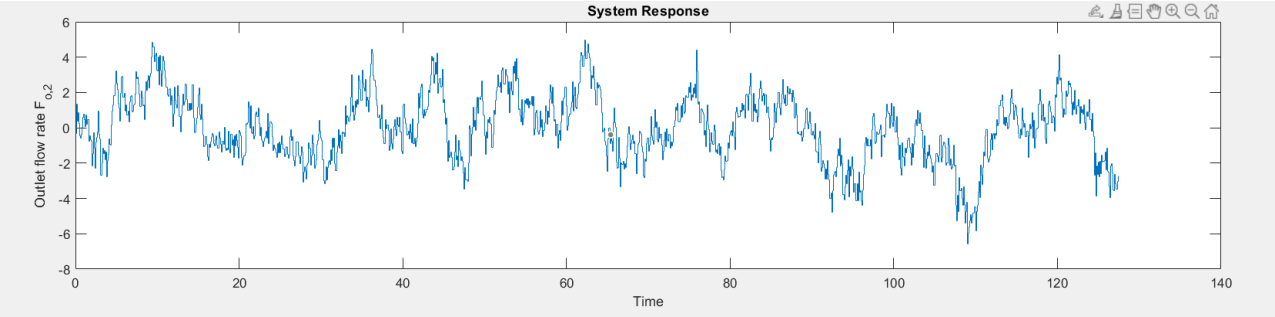


Figure 3.1: System response (Output Flow Rate)

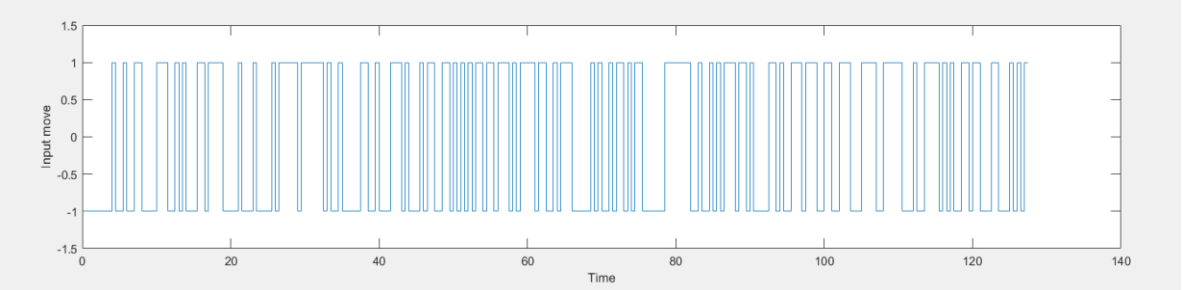


Figure 3.2: Input Generated using idinput (F_i)

We note that there are 'unphysical' outputs < 0. That is because we have given some of the input moves to be -1. (because flow rates don't go in reverse direction in such systems generally)

e)

When I tried giving two different noise inputs, the kalman function takes only the last column as the one corresponding to noise. Because of the nature of Q variable asked in Kalman Filter ($Q = E(ww^T)$), and w that is accepted (single column), I input 'w' as a standard normal and adjusted G appropriately so that the variance is as required. (G is as defined in the *kalman* function documentation)

G can then be appropriately adjusted to adjust the 'Q'. (the Q implied in the question)

To minimise variance of estimates of both the states and measurements, Q was found to be diag([0.2 0.1]). **Thus, the Q value which minimises the variance is same as the Q value used in simulation.**

I did not extensively search over the whole 2-D plane but rather just tried tweaking the individual terms in Q and run it manually for different values. These were my observations:

i) Increase in Q (more than the Q used in simulation) increases the error in estimating states while simultaneously decreases the error in estimation of measurement. This makes sense because, higher Q means modelling error is high so we value the measurements more.

ii) Decrease in Q increases both the variances.

Best Variance of state and output filtered estimates: [3.24 0.62]

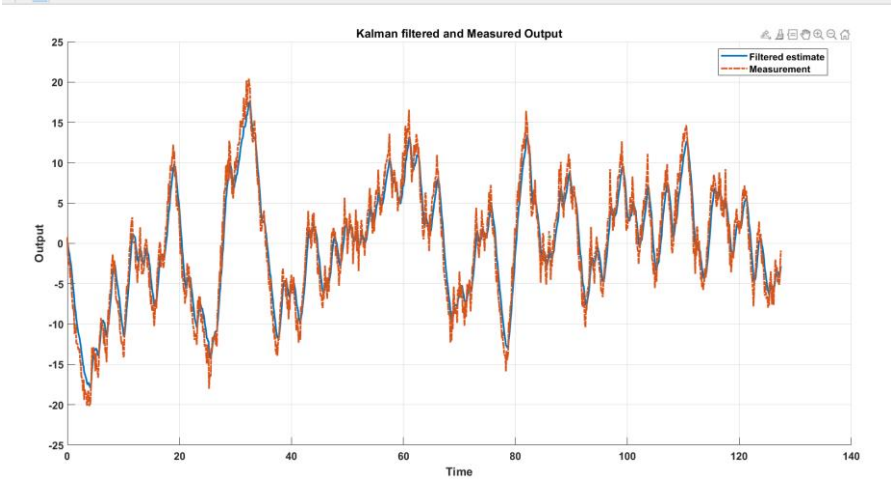


Figure 3.3 Measured and filtered output (flow rate)

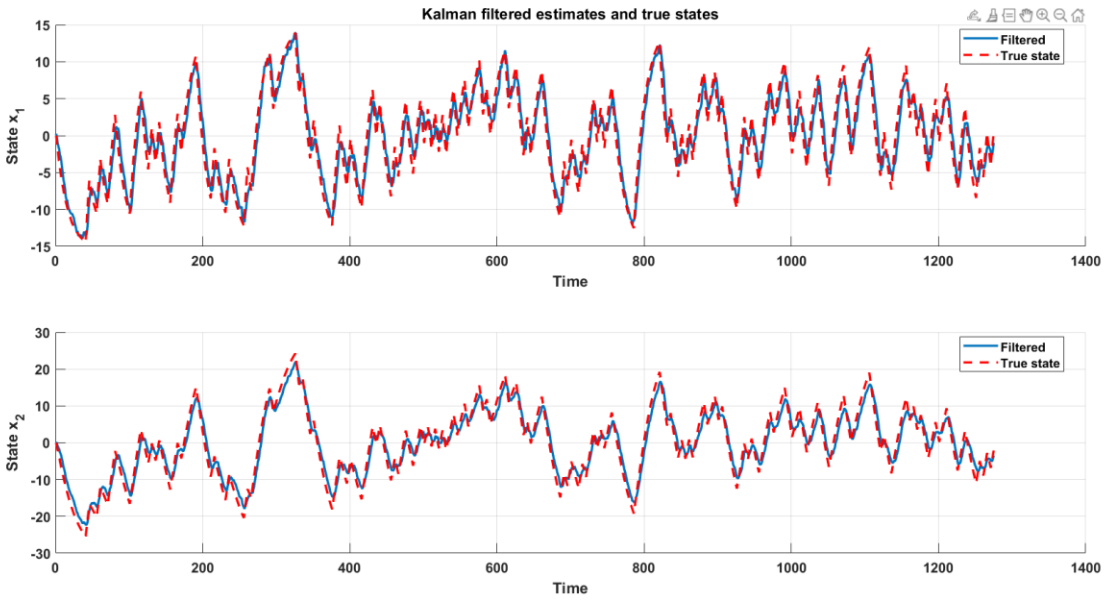


Figure 3.4 Filtered and True States (heights)

④ The equations will be similar to that of previous question except that flow rate will be proportional to $\sqrt{h}$.

$$\therefore \frac{dh_1}{dt} = \frac{F_i}{A_1} - \frac{C_{v1}\sqrt{h_1}}{A_1} + w_1 \quad \text{——①}$$

$$\frac{dh_2}{dt} = \frac{C_{v1}\sqrt{h_1}}{A_1} - \frac{C_{v2}\sqrt{h_2}}{A_2} + w_2 \quad \text{——②}$$

There will also be a noise term w_1 and w_2

& Eqns ① & ② will be supplied to the DEE block in Simulink.

also $y = C_{v2}\sqrt{h_2}$ (flow rate)

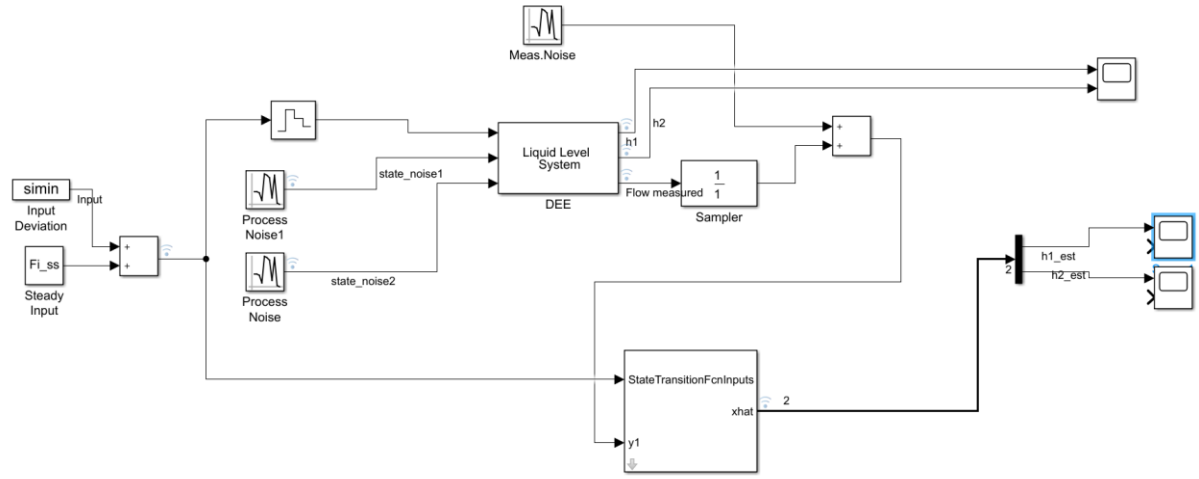
Also, steady state values: $h_{1,ss} = \left(\frac{F_1}{C_{v1}}\right)^2$

$$h_{2,ss} = \left(\frac{C_{v1}}{C_{v2}}\frac{A_2}{A_1}\right)^2 h_{1,ss}$$

BY S.VISHAL
CH18B020

Q4)

At t = 0, the system is assumed to be in steady state.



One noise term is added to each of the states.

The Q and R set in the External Kalman Filter are the true Q and R.

The input given is the steady input plus the PRBS input.

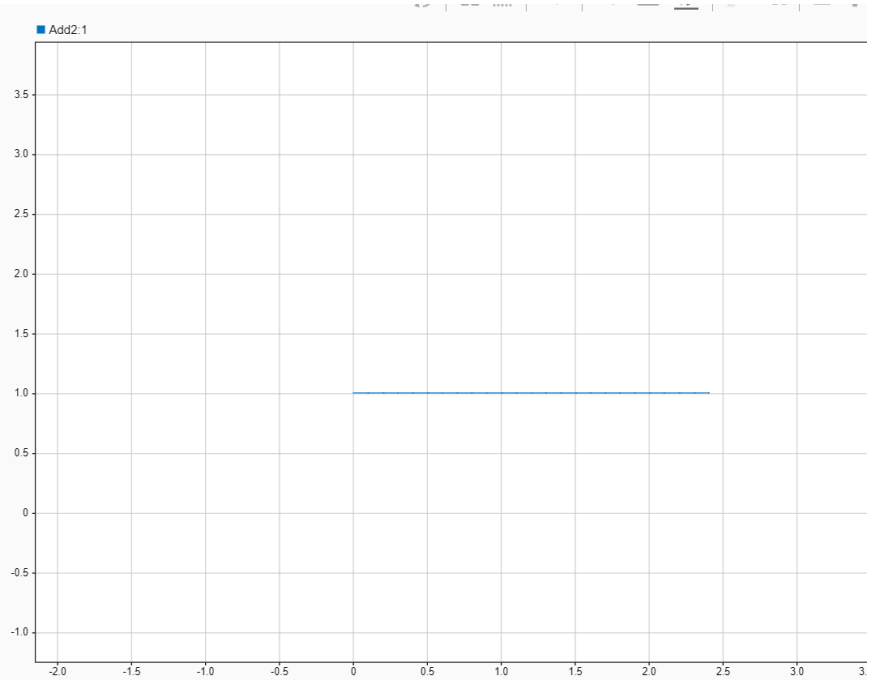


Figure 4.1: Input moves, plot using data inspector

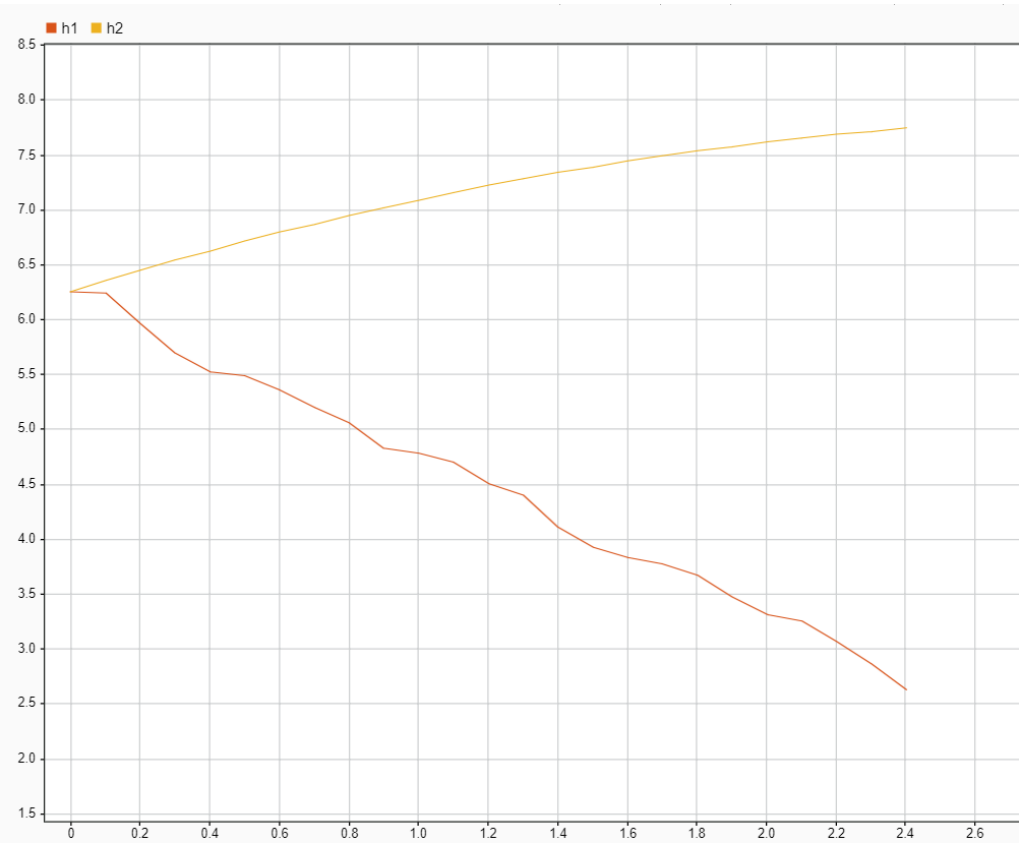


Figure 4.2: Height of simulated tank with respect to time.

However I am facing an issue. For some reason, one of the estimated xhat (namely, h2_hat) goes negative after certain time. And since we can't evaluate square root of negative numbers we can't continue the simulation.

I tried remedying by changing Q given to filter, changing process noise added in the DEE block and increasing input flow rate. But none of it worked.

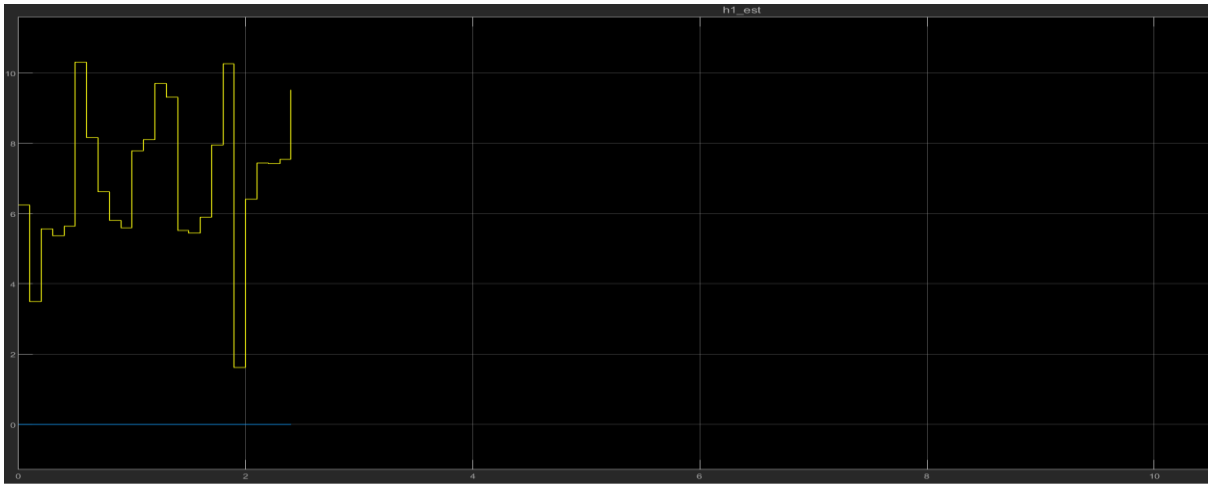


Figure 4.3: h1_est plot using 'scope' block

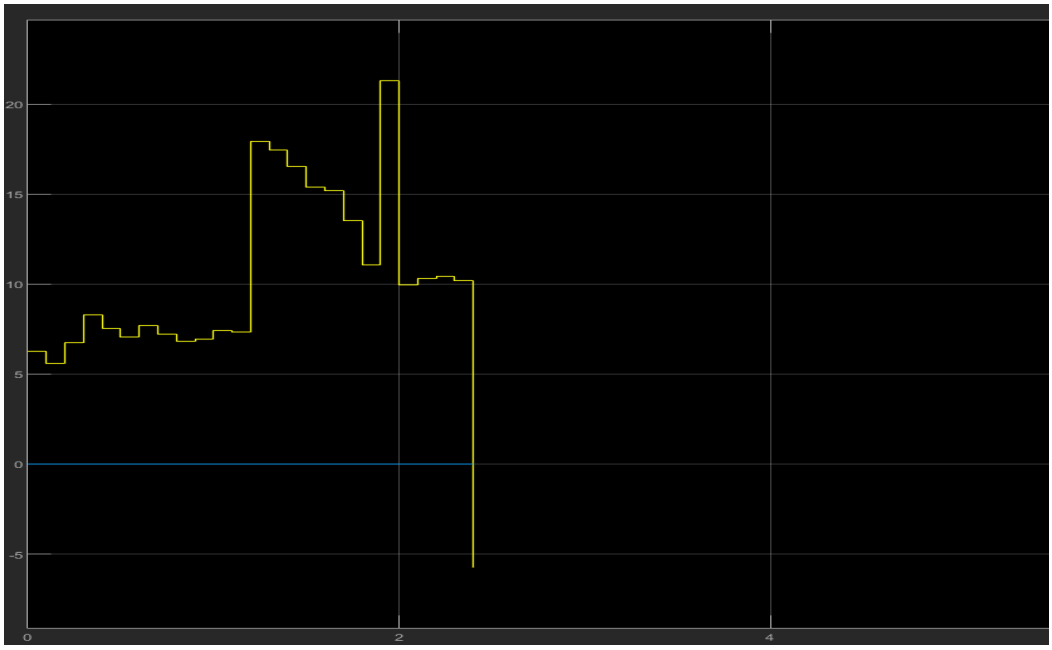


Figure 4.4: h2_est plot using 'scope' block

As we can see, the simulation terminates after estimated height of tank 2 (h2) becomes negative.

(5) We know that Kalman filter gives us the MMSE estimate.

We can consider the given scenario as,

$$x(k+1) = \varepsilon(k+1) \quad \text{---(1)}$$

$(\varepsilon \sim WN(\mu, \sigma_x^2)$ assuming the observations are i.i.d.)

and $y(k) = x(k) + v(k)$ ---(2)

So we have,

$A = 0$, $B = 1$, $C = 1$, $q = \sigma_x^2$, $r = \sigma_v^2$

Substituting these values in the Kalman filter expression for a first order system,

$$K_{f,k} = \frac{(a^2 P_{k-1} + q)C}{(c^2(a^2 P_{k-1} + q) + r)} = \frac{\sigma_x^2}{\sigma_x^2 + \sigma_y^2} = \text{constant} = K_f$$

Now $\hat{x}(k|k-1) = \hat{x}(k-1|k-1)$ ---(3)

and $\hat{x}(k|k) = \hat{x}(k-1|k-1) + K_{f,k}\left(y(k) - \hat{x}(k-1)\right)$

$\Rightarrow \hat{x}(k|k) = (1 - K_f)\hat{x}(k-1|k-1) + K_{f,k}\, y(k)$

$= (1 - K_f)\hat{x}(k-1|k-1) + K_{fk}\, y(k)$ ---(4)

and $\hat{x}(0|-1) = \bar{x}$ ---(5)

$\Rightarrow \hat{x}(0|0) = \bar{x} + K_f\left(y(k) - \bar{x}\right)$ ---(6)

By recursive eliminating terms until $\hat{x}(0|0)$

$$\hat{x}(N|N) = \sum_{k=0}^{N} K_f (1 - K_f)^k \left(y(N-k)\right) + (1 - K_f)^{N+1} \bar{x}$$

$$\Rightarrow \hat{x}(N|N) = \sum K_f (1-K_f)^k (y(N-k)) + (1-K_f)^{N+1} \bar{x}$$

So $\beta(N) = (1-K_f)^{N+1}$

ie we don't have a clear $\alpha(N).\Sigma y(k)$ is actually a weighted ~~sum~~ sum, with less weight given to older data

In the limit $N \to \infty$,

The ~~sum~~ $\beta(N)$ vanishes, since,

$$1-K_f = \frac{\sigma^2 v}{\sigma^2 x - \sigma^2 v} < 1.$$

For $\alpha(N)$, consider $y(k) = \mu + \underbrace{\varepsilon(k)}_{\sim WN(0,\sigma_x^2)} + \underbrace{v(k)}_{\sim WN(0,\sigma_y^2)}$

So $\lim_{N\to\infty} E\left(\sum (K_f (1-K_f)^k y(N-k))\right)$

$$\Rightarrow \lim_{N\to\infty} E(\alpha(N)) = E\sum_{k=0}^{\infty} K_f (1-K_f)^k \mu.$$

$$= \mu \quad \left(\because E(\varepsilon) = E(v) = 0\right)$$

So as $\lim_{N\to\infty}$, we completely rely on the data, and the average value of the estimate is the mean of the R.V. x.

METHOD - (2)

Since the ε used in (1) has non-zero mean, I tried using a different formulation having 0 mean ε

$$\begin{bmatrix} x(k+1) \\ \mu \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x(k) \\ \mu \end{bmatrix} + \begin{bmatrix} 1 \\ 0 \end{bmatrix} \varepsilon(k)$$

and $y(k) = \begin{bmatrix} 1 & 0 \end{bmatrix} \begin{bmatrix} x(k) \\ \mu \end{bmatrix} + v(k)$

However $O_n = \begin{pmatrix} C \\ CA \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$ $\therefore$ system is observable

rank $(O_n) = 2$

We note that both $x(k)$ and μ are unknowns.

(If μ is known, $\hat{x}_{MMSE} = E(X) = \mu$, irrespective of N)

We can jointly estimate both in an MMSE fashion using Kalman filtering.

($\varepsilon \sim WN(0, \sigma_x^2)$ and $v \sim WN(0, \sigma_v^2)$ and in addition both are Gaussian).

$$A = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} \qquad C = \begin{bmatrix} 1 & 0 \end{bmatrix}$$

$$Q = \begin{bmatrix} \sigma_x^2 & 0 \\ 0 & 0 \end{bmatrix}$$ ($\because$ μ is a fixed value without an error)

$R = \sigma_v^2$

~~Riccatti equn~~: $\bar{P}_k = A P_{k-1} A^T + Q$ — (1)

$$K_{f,k} = (\bar{P}_k)(C^T)(R + C\bar{P}_k C^T)^{-1} \quad \text{— (2)}$$

$$P_k = (I - K_{f,k} C)(\bar{P}_k) \quad \text{— (3)}$$

Let P_{k-1} be a 2×2 matrix $\begin{pmatrix} a & b \\ c & d \end{pmatrix}$

$$\bar{P}_k = \begin{pmatrix} 0 & 1 \\ 0 & 1 \end{pmatrix}\begin{pmatrix} a & b \\ c & d \end{pmatrix}\begin{pmatrix} 0 & 0 \\ 1 & 1 \end{pmatrix} + \begin{pmatrix} \sigma_x^2 & 0 \\ 0 & 0 \end{pmatrix}$$

$$= \begin{pmatrix} c & d \\ c & d \end{pmatrix}\begin{pmatrix} 0 & 0 \\ 1 & 1 \end{pmatrix} + \begin{pmatrix} \sigma_x^2 & 0 \\ 0 & 0 \end{pmatrix}$$

$$= \begin{pmatrix} \sigma_x^2 + d & d \\ d & d \end{pmatrix}$$

So, $K_{f,k} = \begin{pmatrix} \sigma_x^2 + d & d \\ d & d \end{pmatrix}\begin{pmatrix} 1 \\ 0 \end{pmatrix}(R + C\bar{P}_k C^T)^{-1}$ — (4)

Consider $R + C\bar{P}_k C^T$

$$= \sigma_v^2 + \begin{pmatrix} 1 & 0 \end{pmatrix}\begin{pmatrix} \sigma_x^2 + d & d \\ d & d \end{pmatrix}\begin{pmatrix} 1 \\ 0 \end{pmatrix}$$

$$= \sigma_v^2 + \begin{pmatrix} \sigma_x^2 + d & d \end{pmatrix}\begin{pmatrix} 1 \\ 0 \end{pmatrix}$$

$$\Rightarrow R + C\bar{P}_k C^T = \sigma_v^2 + \sigma_x^2 + d \quad \text{— (5)}$$

Substituting ~~from~~ (5) in (4)

$$\Rightarrow K_{f|k} = \begin{pmatrix} \dfrac{\sigma^2_x + d}{\sigma^2_v + \sigma^2_x + d} \\ \dfrac{d}{\sigma^2_v + \sigma^2_x + d} \end{pmatrix} \quad \text{---(6)}$$

$$I - K_{f|k} C =$$

$$\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} - \frac{1}{\sigma^2_v + \sigma^2_x + d} \begin{pmatrix} \sigma^2_x + d \\ d \end{pmatrix} \begin{pmatrix} 1 & 0 \end{pmatrix}$$

$$\therefore (3) \Rightarrow P_k = \begin{bmatrix} \dfrac{\sigma^2_v}{\sigma^2_v + \sigma^2_x + d} & 0 \\ - \dfrac{\sigma^2_x d}{\sigma^2_v + \sigma^2_x + d} & 1 \end{bmatrix} \begin{bmatrix} \sigma^2_x + d & d \\ d & d \end{bmatrix}$$

Since the 2nd diagonal term $(2,2)$, is the only ~~important~~ relevant term,

$$d_k = \left(\frac{-d_{k-1}}{\sigma^2_v + \sigma^2_x + d_{k-1}} \right) d_{k-1} + d_{k-1}$$

$$\Rightarrow d_k = \frac{(\sigma^2_v + \sigma^2_x)\, d_{k-1}}{\sigma^2_v + \sigma^2_x + d_{k-1}} \quad \text{---(7)}$$

We know: $\hat{s}(k|k) = \hat{s}(k+1|k-1) + K_{f|k} \left(y(k) - C \hat{s}(k|k-1) \right)$

where $\hat{s}(k) = \begin{bmatrix} \hat{x}(k) \\ \hat{y}(k) \end{bmatrix}$

$$\begin{bmatrix} \hat{x}(k|k-1) \\ \hat{y}(k|k-1) \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \hat{x}(k-1|k-1) \\ \hat{y}(k-1|k-1) \end{bmatrix}$$

$$\Rightarrow \begin{bmatrix} \hat{x}(k|k-1) \\ \hat{y}(k|k-1) \end{bmatrix} = \begin{bmatrix} \hat{y}(k-1|k-1) \\ \hat{y}(k-1|k-1) \end{bmatrix}$$

$$\therefore \begin{pmatrix} \hat{x}(k|k) \\ \hat{y}(k|k) \end{pmatrix} = \begin{pmatrix} \hat{y}(k-1|k-1) \\ \hat{y}(k-1|k-1) \end{pmatrix} +$$

$$\begin{pmatrix} \frac{\sigma^2_x + d}{\sigma^2_v + \sigma^2_x + d} \\ \frac{d}{\sigma^2_v + \sigma^2_x + d} \end{pmatrix} \left(y(k) - (1 \;\; 1) \begin{pmatrix} \hat{x} \\ \hat{y} \end{pmatrix} \right)$$

$$= \begin{pmatrix} \hat{y}(k-1|k-1) + \frac{\sigma^2_x + d}{\sigma^2_v + \sigma^2_x + d} \left(y(k) - \hat{y}(k-1|k-1) \right) \\ \hat{y}(k-1|k-1) + \frac{d}{\sigma^2_v + \sigma^2_x + d} \left(y(k) - \hat{y}(k-1|k-1) \right) \end{pmatrix}$$

$$= \begin{pmatrix} \hat{y}(k-1|k-1) \left(\frac{\sigma^2_v}{\sigma^2_x + \sigma^2_v + d} \right) - \left(\frac{\sigma^2_x + d}{\sigma^2_v + \sigma^2_x + d} \right) y(k) \\ \hat{y}(k-1|k-1) \left(\frac{\sigma^2_v}{\sigma^2_x + \sigma^2_v + d} \right) \end{pmatrix}$$

$$\Rightarrow \hat{\mu}(k|k) = \hat{\mu}(k-1|k-1)\left(\frac{\sigma^2_v}{\sigma^2_v+\sigma^2_x+d}\right) + \left(\frac{\sigma^2_x+d}{\sigma^2_v+\sigma^2_x+d}\right) y(k)$$

$$\hat{\mu}(k|k) = \hat{\mu}(k-1|k-1)\left(\frac{\sigma^2_v+\sigma^2_x}{\sigma^2_v+\sigma^2_x+d}\right)_{k-1} + \frac{d_{k-1}}{\sigma^2_v+\sigma^2_x+d_{k-1}}\, y(k) \quad \text{---(8)}$$

$$\text{lt}_{k\to\infty} \hat{\mu}(k|k) = \hat{\mu}(k-1|k-1)$$

($\because$ (7) $\Rightarrow d_k \to 0$ because $\frac{\sigma^2_v+\sigma^2_x}{\sigma^2_v+\sigma^2_x+d_{k-1}} < 1$)

→ To conclude, since d_k is itself in a non-linear recursion, I am not able to write it in the expected form

→ Nonetheless by performing the recursion in (7) and (8), we can obtain the MMSE estimate of μ.

→ And as $N \to \infty$, our estimate from the data converges.