Q1.

a> No, Robot action's don't always increase uncertainity. It is true that when robot's take actions in an environment, the uncertainity regarding them increases, but at the same time the actions can also tell/gives us more information about the environment. This information can help us make informed decisions to achieve the desired goal.

Example: ~~Only when robot decides to take action~~ ~~in m~~ Only when a robot moves forward and reaches it's goal state, we can know that it has completed it's mission. i.e in this case robot action did not cause uncertainity.

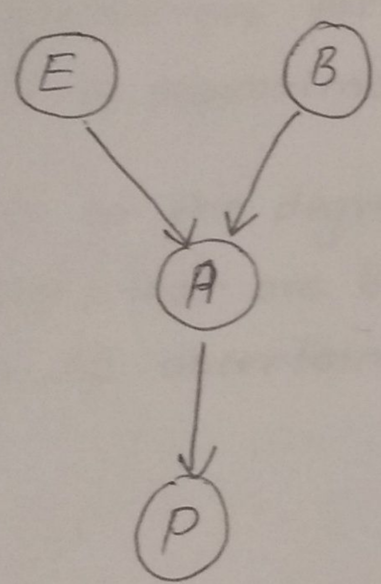
b>

→ If the probability of a state assignment becomes 1, then bayes filter might get over-confident in the belief of it's state & may ~~mot~~ not incorporate future beliefs that may contain the actual position of a state.

- One way that probability of state estimation may become 1 is if ~~two~~ too many observations from one state are added to the filter too quickly
- One way to prevent this is to be aware of biases in measurement that may arise due to uncertainities in the environment.

Q1.

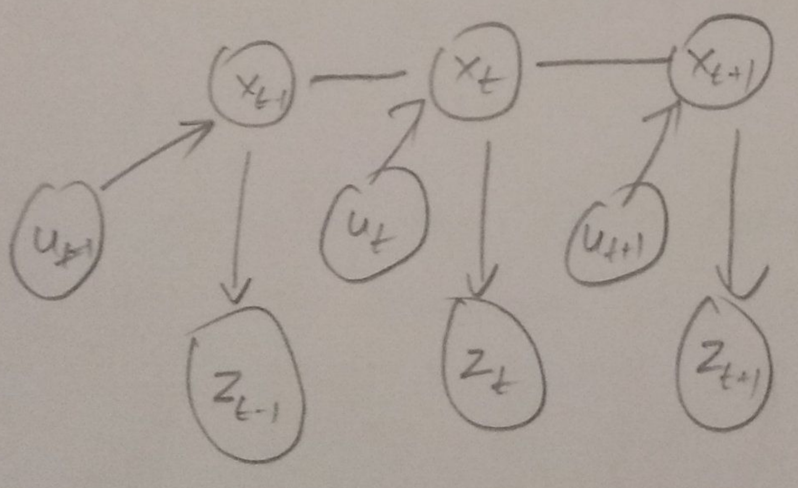
c) If Earthquake occurs or there is a burglary, the alarm is likely to go off.
If alarm goes off, a police may arrive.
Design a Basian network illustrating the causal relationship.

→ 

E: Earthquake occurs
B: Burglary
A: Alarm goes off
P: police arrives

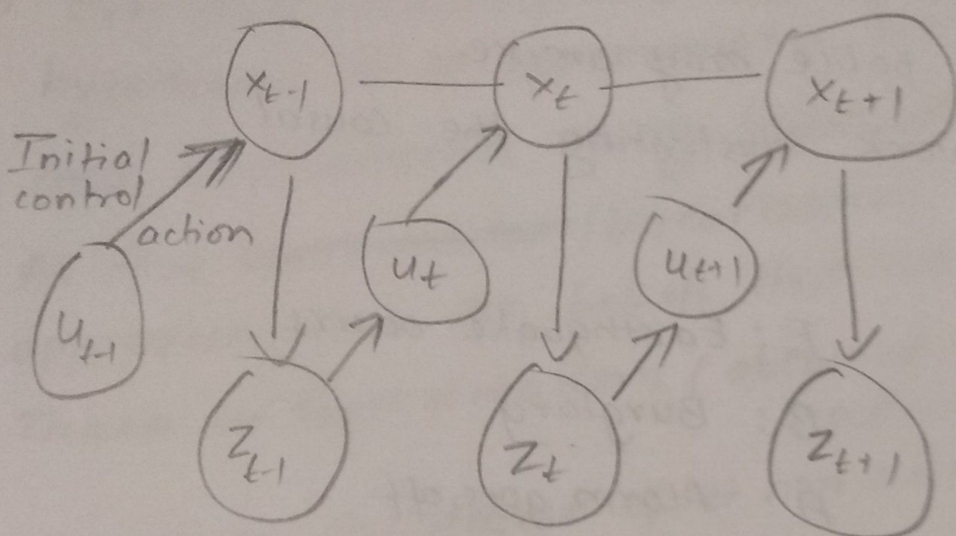
Q1.

d) Normal Baysean Network:



Q1.

d) If controls were dependent on observations



Q.1.

e)

→ Extended Kalman filters fail in handling multiple hypotheses as it approximates state transitions and measurements using linear taylor expansion

As this ~~approximation~~/linearization is a poor approximation for beliefs that are ~~non-l~~ multi-modal

These ~~of~~ approximations (obtained using linear taylor expansion) depend on the degree of ~~t~~ non-linearity of the functions that are being approximated & the degree of uncertainity.

* Problem 3 – initialization.

$$x(t+1) = \alpha x(t) + w(t) \qquad w(t) \sim N(0,R)$$

$$z(t) = \sqrt{x(t)^2+1} + v(t) \qquad v(t) \sim N(0,Q)$$

→

a) Equations for the kalman filter to estimate the unknown parameter α.

→ Augmenting the state with the unknown parameter α.

$$\dot{x}(t) = \begin{bmatrix} x(t) \\ \alpha \end{bmatrix}$$

i.e According to our problem statement,

$$\dot{x}(t) = \cancel{\begin{bmatrix} x(t-1) \\ \alpha_{t-1} \end{bmatrix}} \begin{bmatrix} (\alpha_{t-1}) \cdot x(t-1) + w(t) \\ \alpha_{t-1} \end{bmatrix}$$

Although α is a constant, it is a changing on every time step. Hence, α_{t-1}

$$\dot{x}(t) = \begin{bmatrix} \alpha_{t-1} \cdot x(t-1) + w(t) \\ \alpha_{t-1} \end{bmatrix} = \begin{bmatrix} g_1(x,\alpha) \\ g_2(x,\alpha) \end{bmatrix}$$

i.e $g_1(x,\alpha) = \alpha_{t-1} \cdot x(t-1) + w(t)$

$g_2(x,\alpha) = \alpha_{t-1}$

∴ Linearizing the above functions g_1 & g_2 to form the Jacobian G

$$G_t = \begin{bmatrix} \frac{\partial g_1}{\partial x} & \frac{\partial g_1}{\partial \alpha} \\ \frac{\partial g_2}{\partial x} & \frac{\partial g_2}{\partial \alpha} \end{bmatrix} = \begin{bmatrix} \alpha_{t-1} & x(t-1) \\ 0 & 1 \end{bmatrix}$$

$w(t)$ has no effect in Jacobian as we are differentiating w.r.t x & α.

(1) As we are linearizing, this problem is related to Extended kalman filter. (EKF)

$$\bar{\mu}_t = g(\mu(t-1))$$

$$\mu(t-1) = E[x(t-1)]$$

$$\therefore \bar{\mu}_t = \begin{bmatrix} \alpha_{t-1} \cdot \mu(t-1) \\ \alpha_{t-1} \end{bmatrix}_{2\times 1}$$

(2) $$G_t = \begin{bmatrix} \alpha_{t-1} & x(t-1) \\ 0 & 1 \end{bmatrix}_{2\times 2}$$

$$\therefore \bar{\Sigma}_t = G_t \Sigma_{t-1} G_t^T + R$$

(3) $$h(t) = \sqrt{x(t)^2 + 1}$$

$$H_t = \begin{bmatrix} \frac{\partial h(t)}{\partial x} & \frac{\partial h(t)}{\partial \alpha} \end{bmatrix}$$

$$\frac{\partial}{\partial x}\sqrt{x^2+1} = \frac{1}{2\sqrt{x^2+1}} \cdot 2x.$$

$$\therefore H_t = \begin{bmatrix} \frac{x(t)}{\sqrt{x(t)^2+1}} & 0 \end{bmatrix}_{1\times 2}$$

$$= \frac{x}{\sqrt{x^2+1}}$$

(4) $$K_t = \bar{\Sigma}_t \cdot H_t^T \left[H_t \cdot \bar{\Sigma}_t \cdot H_t^T + Q \right]^{-1}$$

Dimensional analysis for K_t:

$$K_t = [2\times 2][2\times 1] \cdot \left[1\times 2 \times 2\times 2 \times 2\times 1 + 1\times 1 \right]^{-1}$$

$$= [2\times 1][1\times 1]^{-1}$$

$$= [2\times 1]$$

(5) $\mu_t = \bar{\mu}_t + K_t \cdot [z_t - h(\bar{\mu}_t)]$

As augmenting our state should not affect our sensor values.

$\cancel{h(\bar{\mu}_t) = \sqrt{\mu(t)^2 + 1}\ ;\ \mu(t) = E[x(t)]}$

$$h(\bar{\mu}_t) = \sqrt{\overline{\mu(t)}^2 + 1} \qquad \bar{\mu}_t = \begin{bmatrix} \alpha_{t-1} \cdot \mu(t-1) \\ \alpha_{t-1} \end{bmatrix}$$

$$= \sqrt{(\alpha_{t-1} \cdot \mu(t-1))^2 + 1} \qquad \mu(t) = \alpha_{t-1} \cdot \mu(t-1)$$

$$\therefore \boxed{h(\bar{\mu}_t) = \sqrt{[\alpha_{t-1} \cdot \mu(t-1)]^2 + 1}}$$

$$z(t) = \sqrt{x(t)^2 + 1} + v(t)$$

i.e $\boxed{z(t) = \sqrt{[\alpha_{t-1} \cdot x(t-1)]^2 + 1} + v(t)}$

$\therefore \cancel{\bar{\mu}_t = \mu_t}$

$$\mu_t = \bar{\mu}_t + K_t \cdot [z_t - h(\bar{\mu}_t)]$$

(6) $\boxed{\Sigma_t = (I - K_t H_T) \cdot \bar{\Sigma}_t}$

Q.3.

(b) $Q = 1$ $\quad$ $x(0) = 2$ $\quad$ Ground-truth

$R = 0.5$ $\quad$ $\alpha = 0.1$

$\mu_0 = 1$ $\quad$ $E[(x(0) - \mu_0)(x(0) - \mu_0)] = 2$

$\hat{a} = 2$ $\quad$ ~~$E[(x(0) - \mu_0)(x(0)) = a$~~

$E[(x(0) - \hat{a})(x(0) - \hat{a})] = 2$

$\therefore \bar{\mu}_0 = \begin{bmatrix} 1 \\ 2 \end{bmatrix}$ $\quad$ $\bar{\Sigma}_0 = \begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix}$

Assuming α & x are uncorrelated.

$R = \begin{bmatrix} 1/2 & 0 \\ 0 & 1/2 \end{bmatrix}$

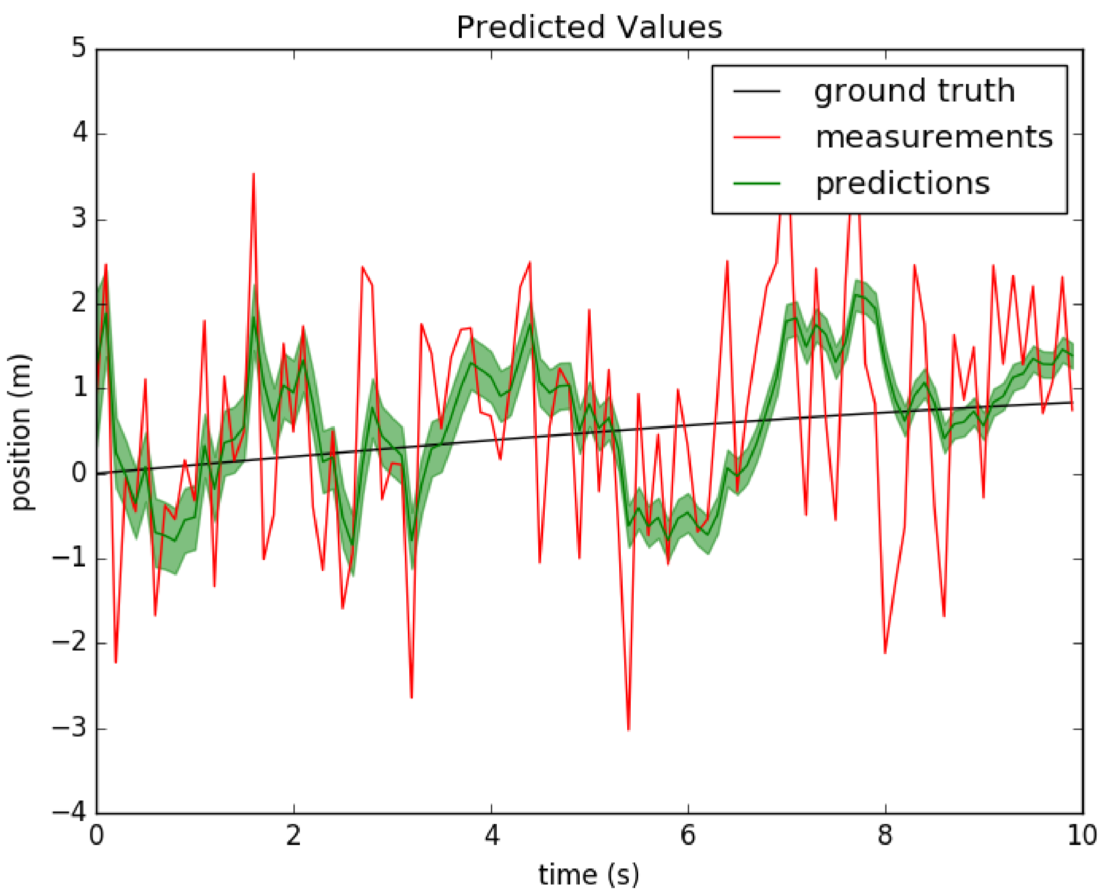
Problem 2 :

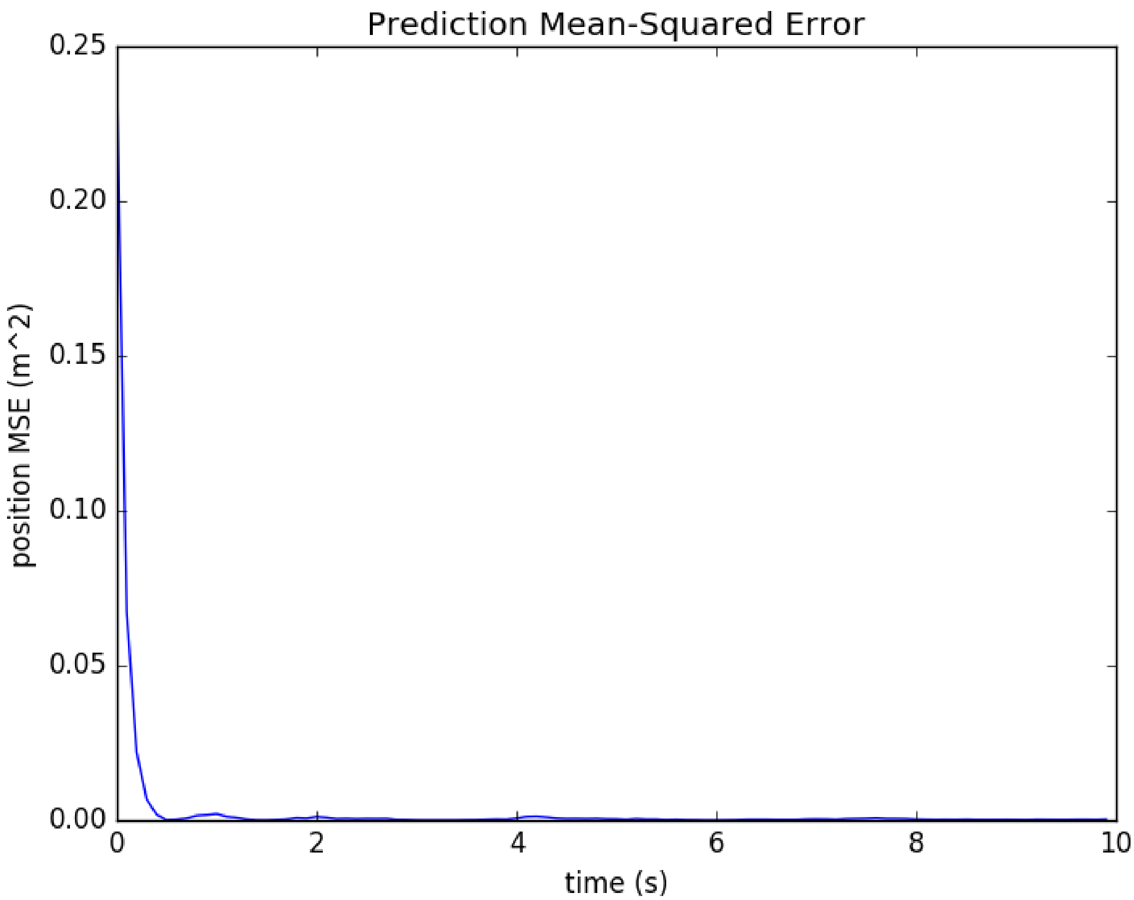
In this problem, I have implemented the Kalman filter for two cases.

1. In this case, the time step = 0.1. As a result, sine takes values from 0 to 10.
For this case, the solutions are as follows:

Part a:

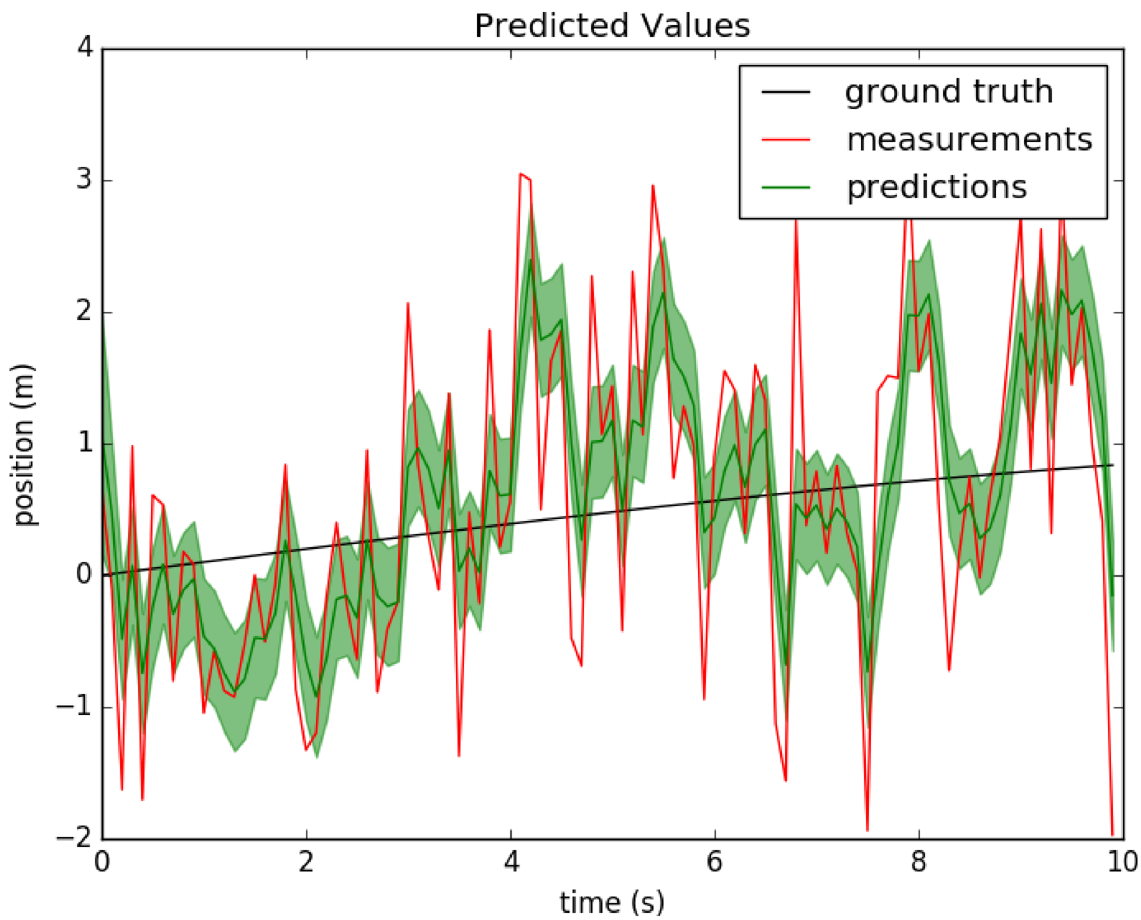
MSE averaged over 10000 trials = 0.0002855

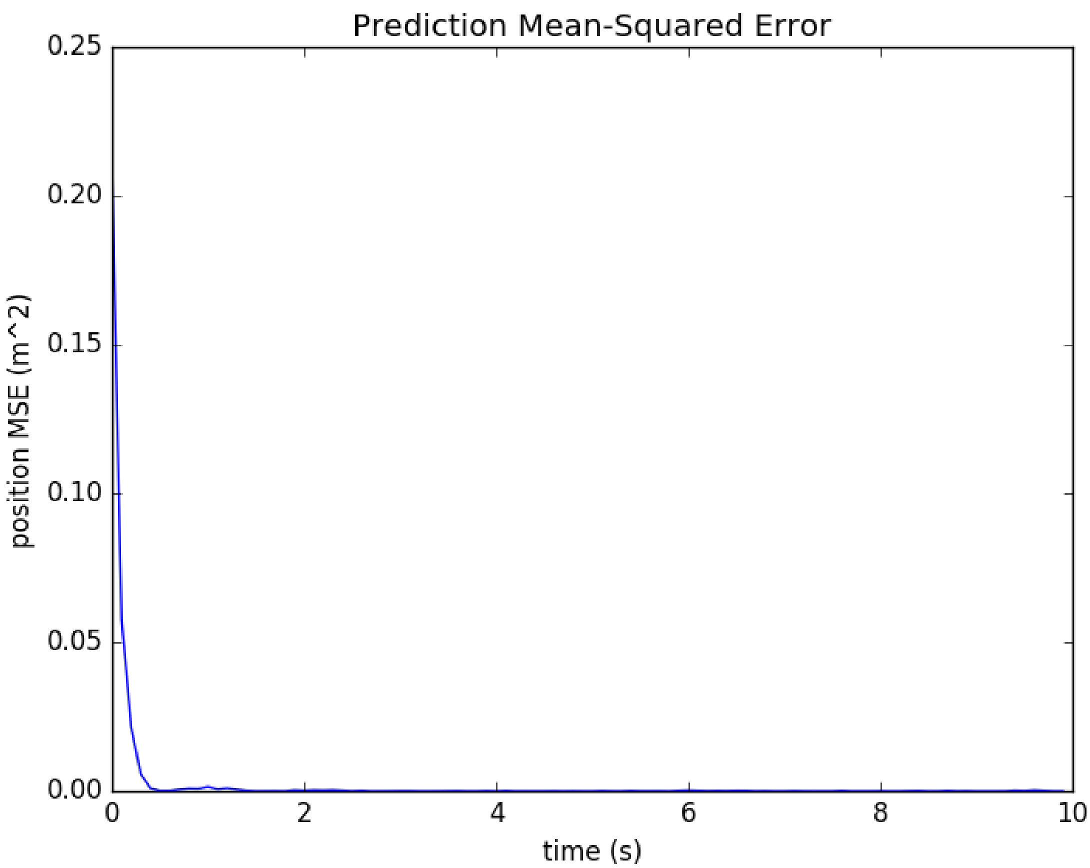



Prediction Mean-Squared Error
position MSE (m^2)
time (s)
0.25
0.20
0.15
0.10
0.05
0.00
0
2
4
6
8
10

Part b:

MSE averaged over 10000 trials = 7.7036578863e-06



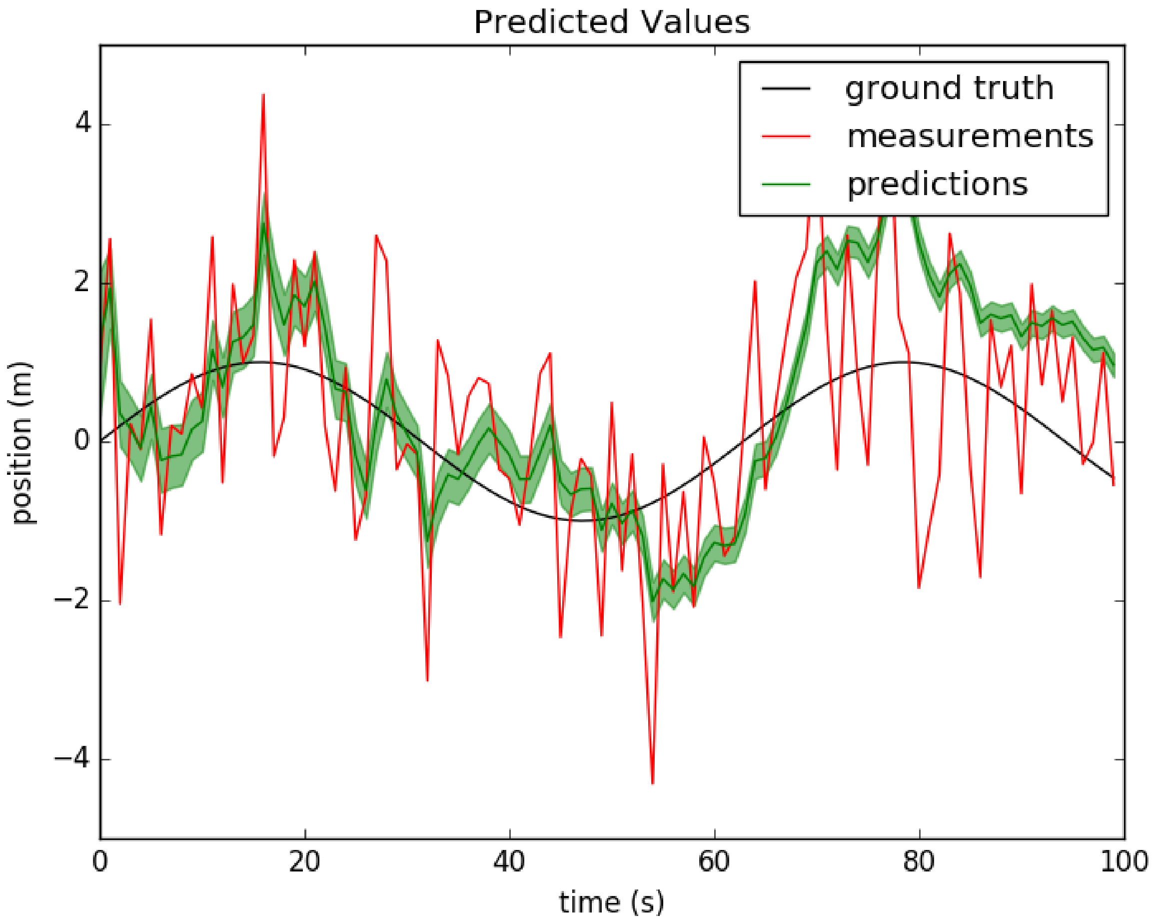


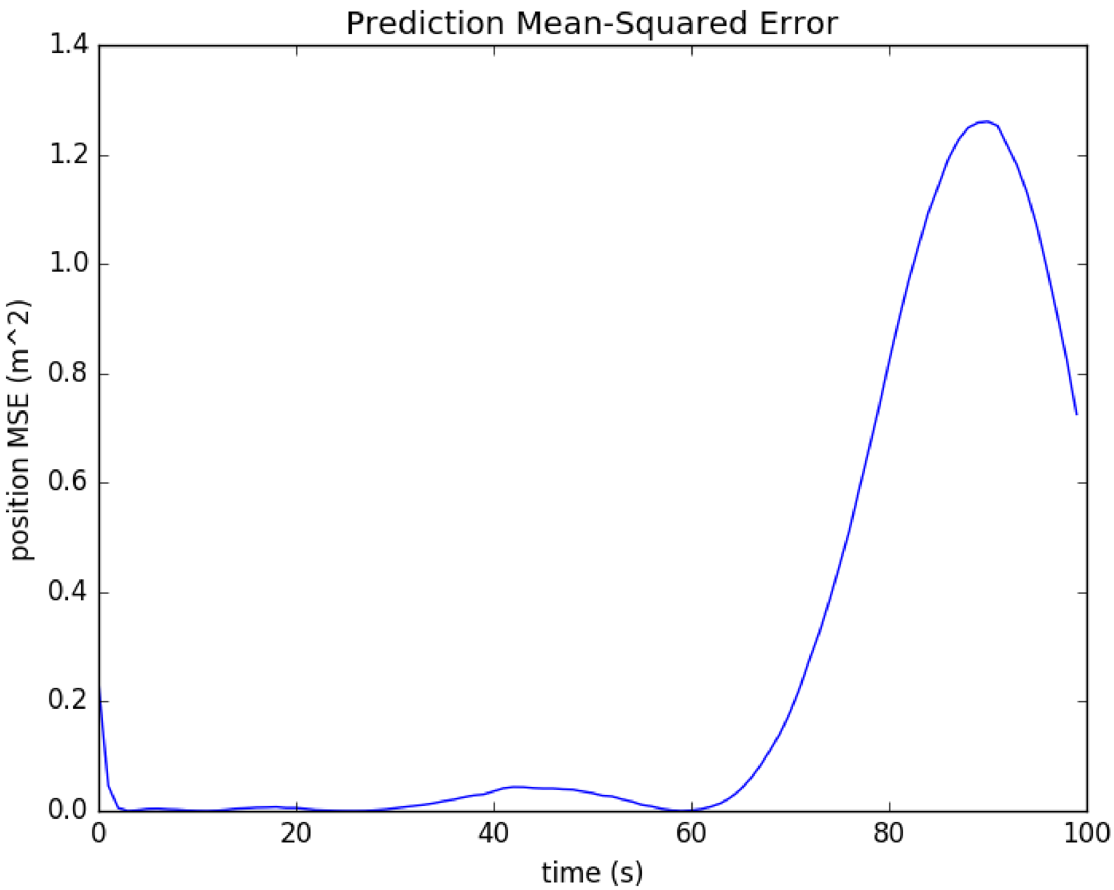
As we can see that in part b the MSE is very low as compared to part a.

2. In this case, I have considered time step = 1, As a result the sine takes values from 0 to 100. For this case, the solution are as follow:

Part a:

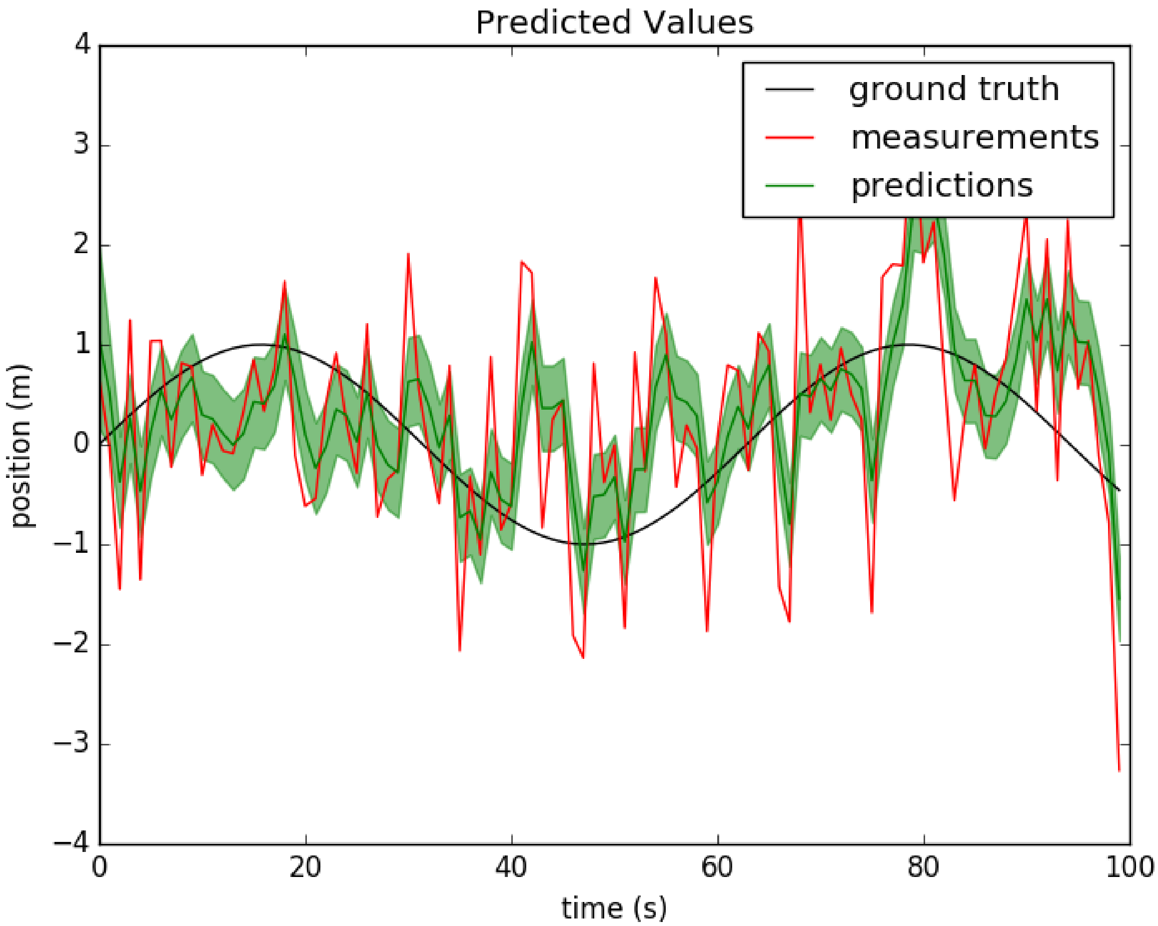
MSE averaged over 10000 trials = 0.72518



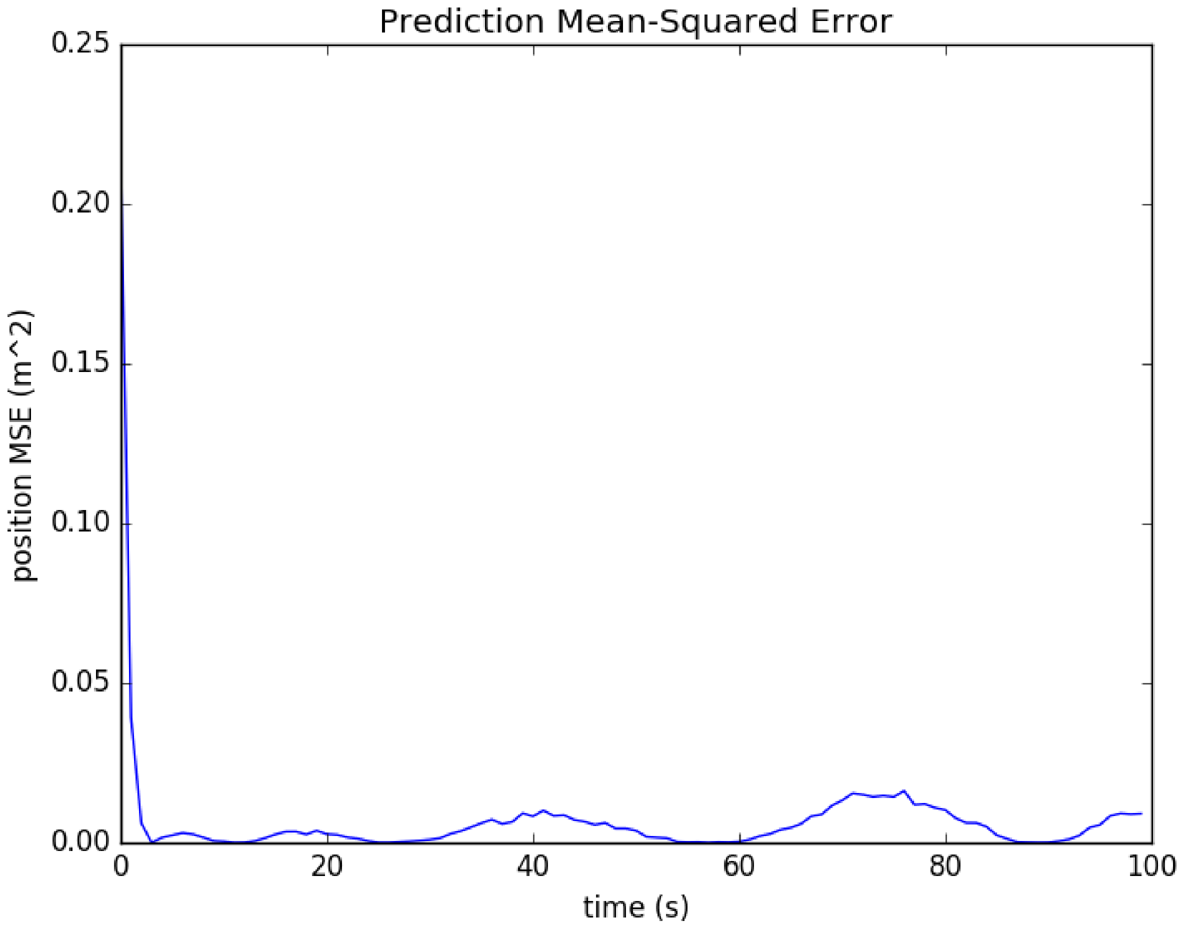


Part b:

MSE averaged over 10000 trials = 0.00912



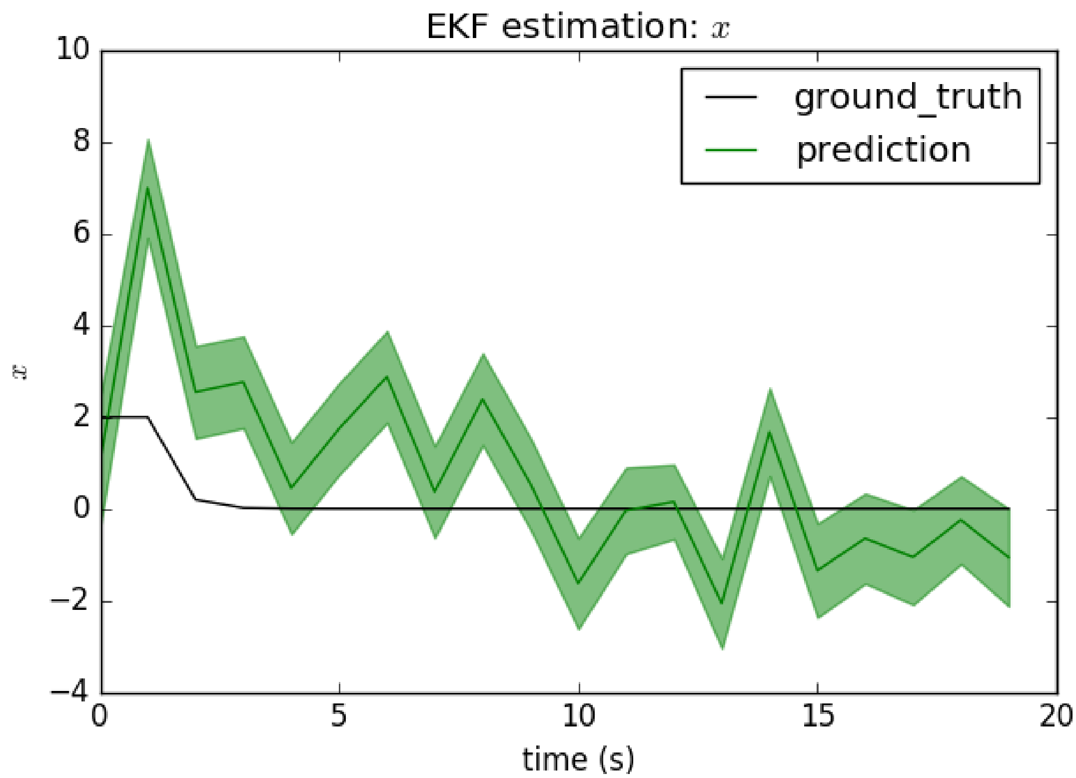
Predicted Values
ground truth
measurements
predictions
position (m)
time (s)
4
3
2
1
0
−1
−2
−3
−4
0
20
40
60
80
100

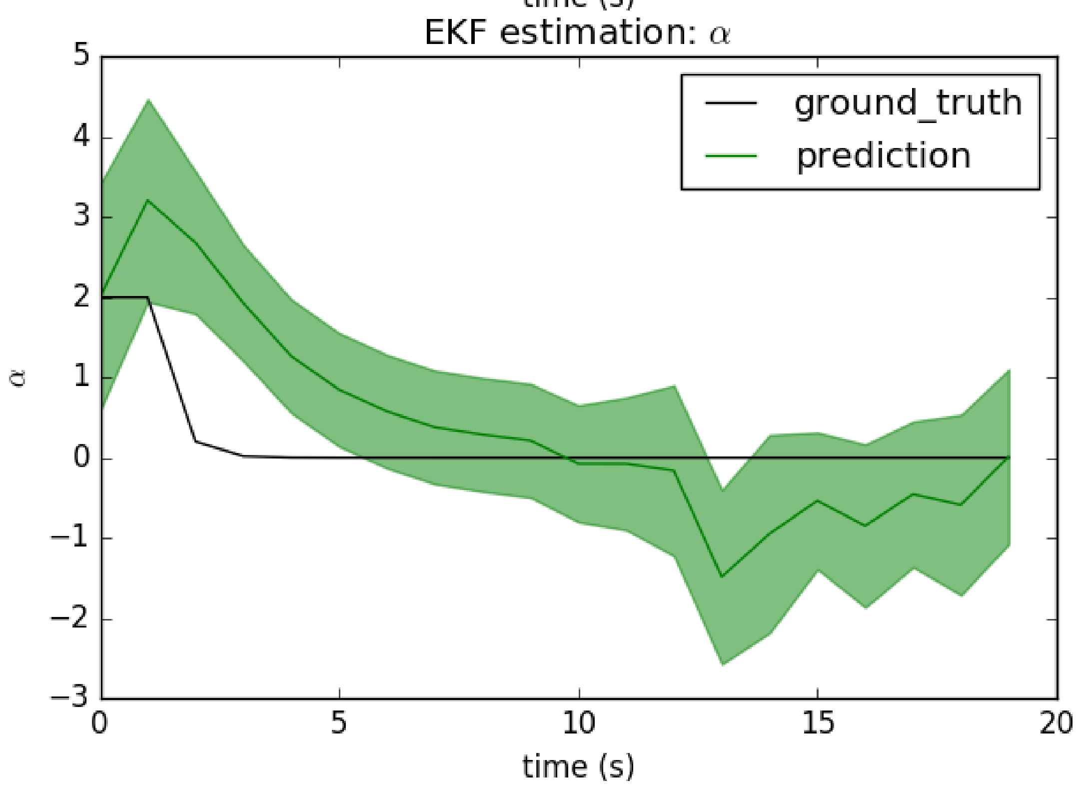


As we can see that even in this case, the MSE without process noise, is more as compared to with process noise.

Problem 3:

The results of EKF estimation is as follows:





The estimation works really well for both x(t) and alpha. As we can see, that in 20 time steps itself, the predicted c and alpha came very close to the actual values.