①

Name: Soumya Ganguly

PID: A53277333

MAE 288B: Optimal Estimation

email: s1gangul@eng.ucsd.edu

Final Assignment

Task 1

The given nonlinear system with scalar state looks like:

$$x_{t+1} = \sin(x_t) + \omega_t$$
$$y_t = x_t^3 + v_t.$$

where the noise processes $\{\omega_t\}$ & $\{v_t\}$ are independent, white & uniformly distributed as $\mathcal{U}[-d, d]$ for ω_t and $\mathcal{U}[-h, h]$ for v_t.

The process described by the system here can be assumed to be a Markov process, by the equations given & the fact that noises are indep. of the states.

now, Given is: $pdf(x_0) = \mathcal{U}[-\pi, \pi]$.

$$pdf(x_0 | Y^{-1}) = pdf(x_0)$$

where,

$$Y^t = \{y_t, y_{t-1}, \dots y_0, pdf(x_0)\}.$$

Now, by def^n of Y^t:

$$pdf(x_t | Y^t) = pdf(x_t | y_t, Y^{t-1})$$

$$= \frac{pdf(x_t, Y^{t-1} | y_t)}{pdf(Y^{t-1} | y_t)}$$ (by form result no. 1)

(2)

$$= \frac{pdf(y_t | x_t, Y^{t-1}) \cdot pdf(x_t, Y^{t-1})}{pdf(y_t) \cdot pdf(Y^{t-1} | y_t)}$$

(by Bayes' rule)

$$= \frac{pdf(y_t | x_t)\, pdf(x_t | Y^{t-1}) . pdf(y^{t-1})}{pdf(y_t) . pdf(y^{t-1} | y_t)}$$ (by def^n of conditional prob)

$$= \frac{pdf(y_t | x_t) \cdot pdf(x_t | y^{t-1}) . pdf(y^{t-1})}{pdf(y_t, Y^{t-1})}$$

(by def^n of conditional probability)

$$= \frac{pdf(y_t | x_t) \cdot pdf(x_t | y^{t-1}) . pdf(y^{t-1})}{\int pdf(y_t, y^{t-1} | x_t)\, pdf(x_t)\, dx_t}$$

(by total probability theorem)

$$= \frac{pdf(y_t | x_t)\, pdf(x_t | y^{t-1}) . pdf(y^{t-1})}{\int pdf(y_t | Y^{t-1}, x_t) \cdot pdf(Y^{t-1} | x_t) . pdf(x_t)\, dx_t}$$

(by fun-result #1)

$$= \frac{pdf(y_t | x_t)\, pdf(x_t | Y^{t-1}) . pdf(Y^{t-1})}{\int pdf(y_t | x_t)\, pdf(x_t, y^{t-1})\, dx_t}$$

(by def^n of conditional prob & the fact that the process is markov so y_t is indep.

③

of Y^{t-1}.

$$= \frac{pdf(y_t|x_t)\, pdf(x_t|y^{t-1})}{\int pdf(y_t|x_t) \frac{pdf(x_t, y^{t-1})}{pdf(y^{t-1})} dx_t}.$$

(as in this process y^{t-1} does not depend on x_t, so we can insert it in the integral in the denominator)

$$= \frac{pdf(y_t|x_t)\, pdf(x_t|Y^{t-1})}{\int pdf(y_t|x_t) \cdot pdf(x_t|Y^{t-1})\, dx_t}.$$

(Proved).

Next

$$pdf(x_{t+1}|Y^t) = \frac{pdf(x_{t+1}, Y^t)}{pdf(Y^t)}.$$

$$= \int \frac{dx_t\, p(x_{t+1}, x_t, Y^t)}{pdf(Y^t)} dx_t$$

(by total prob. theorem)

$$= \int \frac{dx_t\, p(x_{t+1}|x_t, Y^t)\, pdf(x_t, Y^t)}{pdf(Y^t)}.$$

(by def^n of conditional expect.)

$$= \int dx_t \frac{\text{pdf}(x_{t+1} | x_t, Y^t)\, \text{pdf}(x_t | Y^t)\, \cancel{\text{pdf}(Y^t)}}{\cancel{\text{pdf}(Y^t)}}$$

(by defn of conditional prob)

$$= \int dx_t \; \text{pdf}(x_{t+1} | x_t, Y^t)\, \text{pdf}(x_t | Y^t)$$

$$= \int dx_t \; \text{pdf}(x_{t+1} | x_t) \cdot \text{pdf}(x_t | Y^t)$$

(since it is a markov process, x_{t+1} does not depend on Y^t)

because, w_t & v_t does not depend on x_t.

(proved)

Name: Soumya Ganguly

PID : A53274333

Email : s1gangul@eng.ucsd.edu

Task 2:

Here we have written a MATLAB code to implement an approximate Bayesian Filter and Extended Kalman Filtering (code attached in the end). The (d,h) values will be taken from the user at the beginning, each time the programme is run.

For specific values of d and h, our programme here, is cognizant of the following facts:

(i) The range of state values: The state, x_t has a range $[-\pi, \pi]$ at t=0. The states, after t=0, have a range of [-1-d, 1+d] – obvious from the state update equation of our system.

(ii) The range of output values: The outputs y_t have a range of $[-\pi^3 - h, \pi^3 + h]$ at t=0. The outputs, after t=0, have a range of $[(-1-d)^\wedge 3 - h, (1+d)^\wedge 3 + h]$ because of the output equation of our system and the last point in this task.

(iii) Number of sample points: Our Bayesian filter is aware of number of sample points, which is denoted by ‘n_sample’ in the code and which is kept as 1000 in this code.

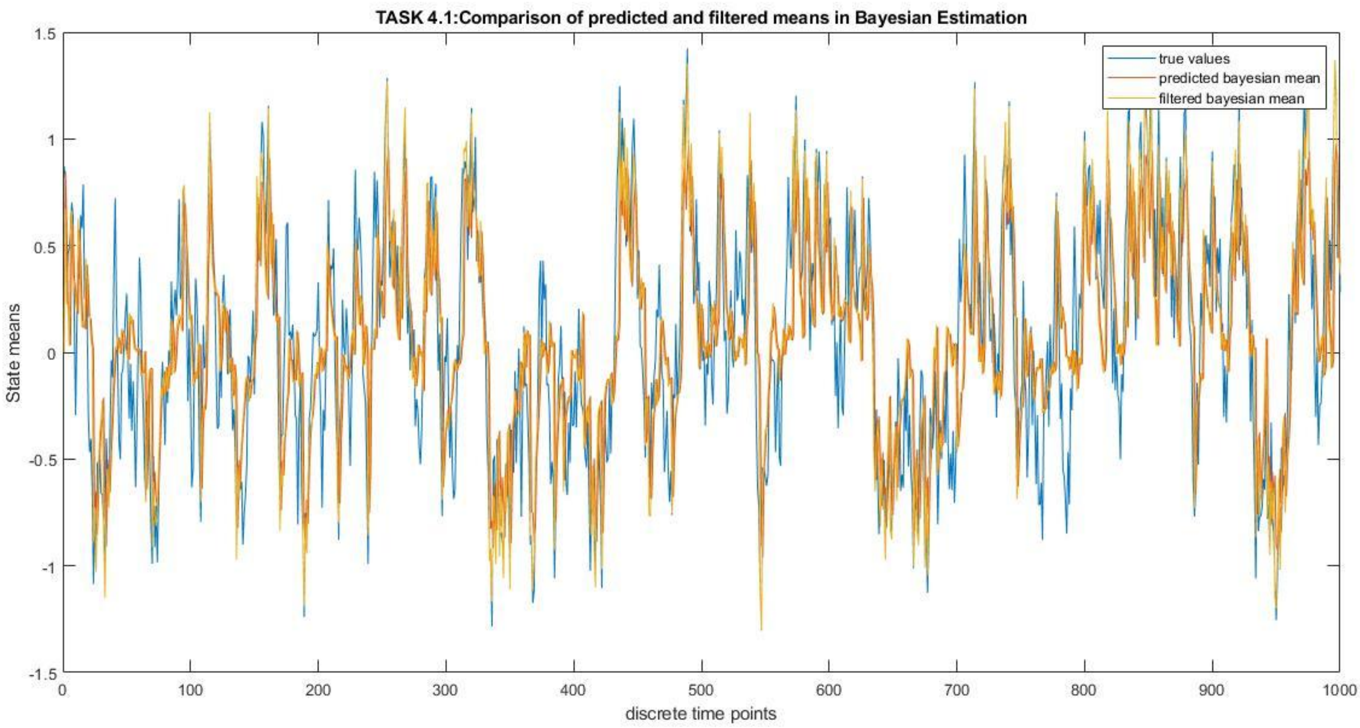
Task 3:

In our programme, we have generated sample states and output values with the help of the system equations given and MATLAB’s *rand* function. These outputs have been referred throughout as the ‘measurement values’. Estimators have been tested later, with respect to these ‘true state values’ and ‘measurement values’ only.

Task 4:

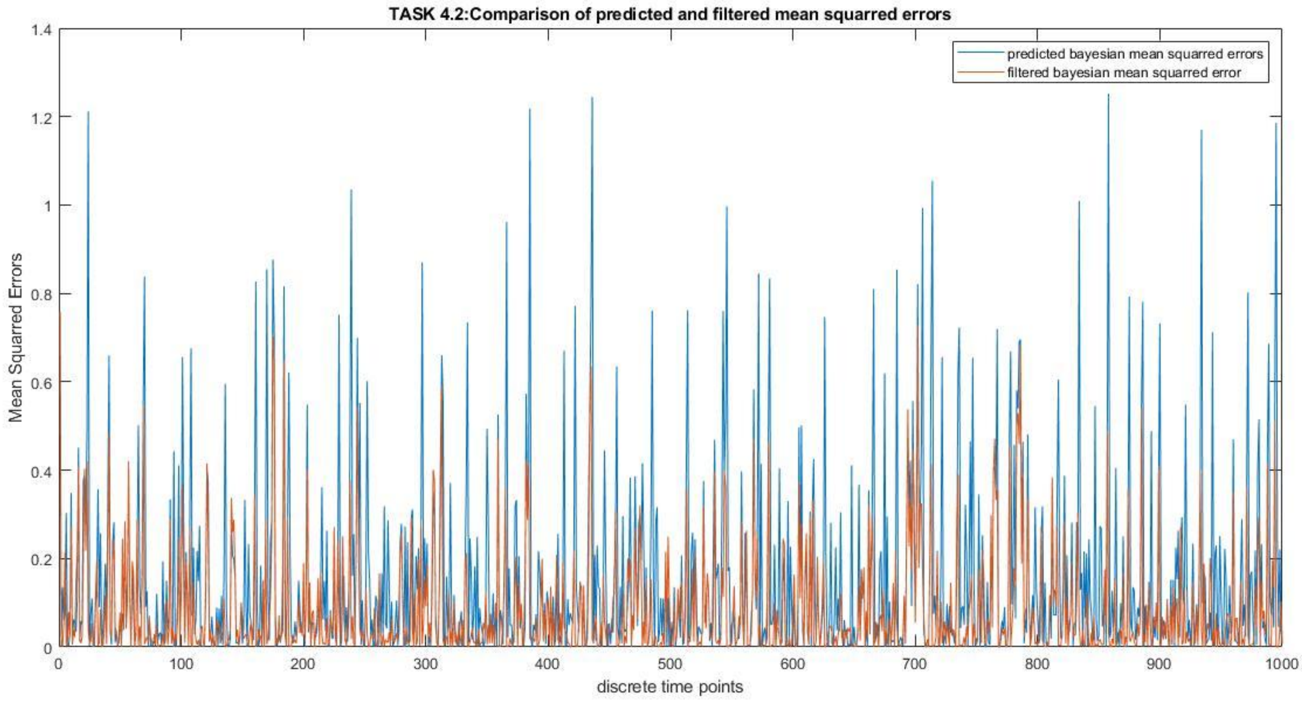
For specific values of d,h and time steps, Bayesian Filter returns conditional predicted and filtered pdfs (probability density functions).Since one sensible way to compare the conditional pdfs is comparing the conditional means, hence we have compared the conditional means of the states obtained from the predicted and filtered pdfs for d=0.5 and h=0.5 (with number of time steps=1000). The true values of the states are also given for a clearer comparison.

Fig 1



Also, for the same parameter values, we have compared the mean squared error (with respect to the true state values) in case of predicted and filtered pdfs, as shown below. This shows that, generally filtered values have smaller error than predicted ones which means filtered ones are closer to the true values. This is intuitively clear as theoretically, the filtered values normally push the predicted values towards the true values.

Fig 2.



Similar exercise has been done for d=1 and h=1 (keeping number of time steps and samples same) The first plot is of the mean and the next one is of the mean squared error comparison.

Fig. 3

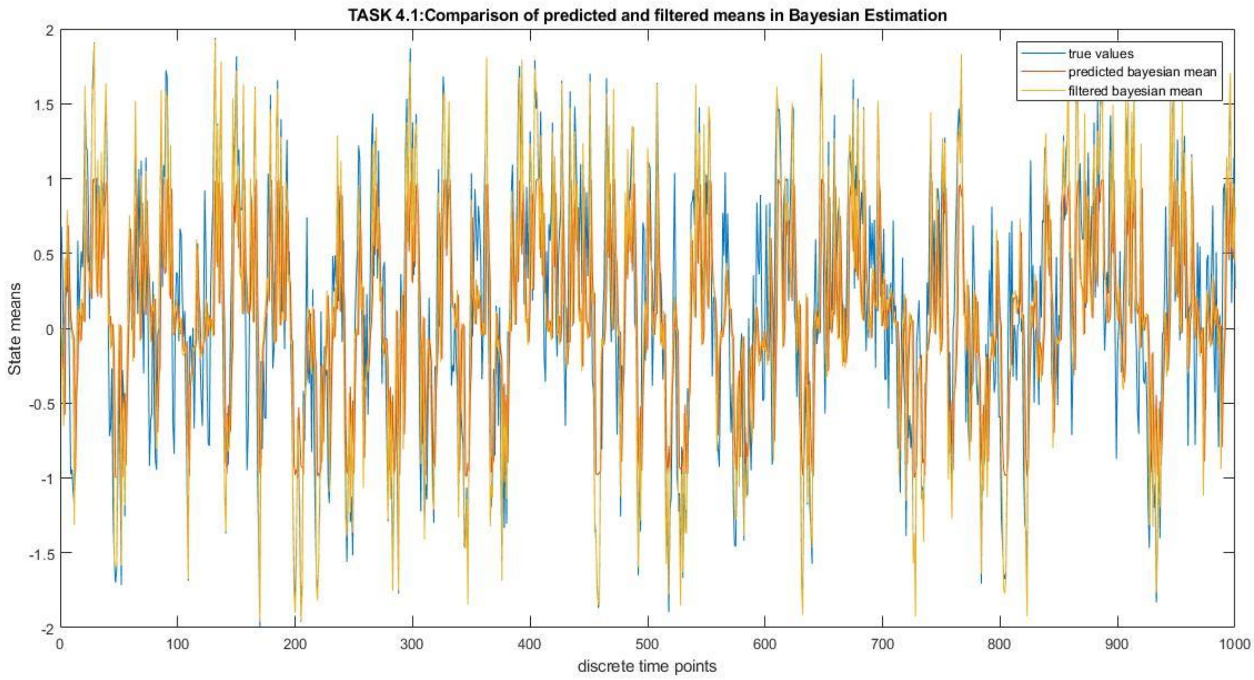
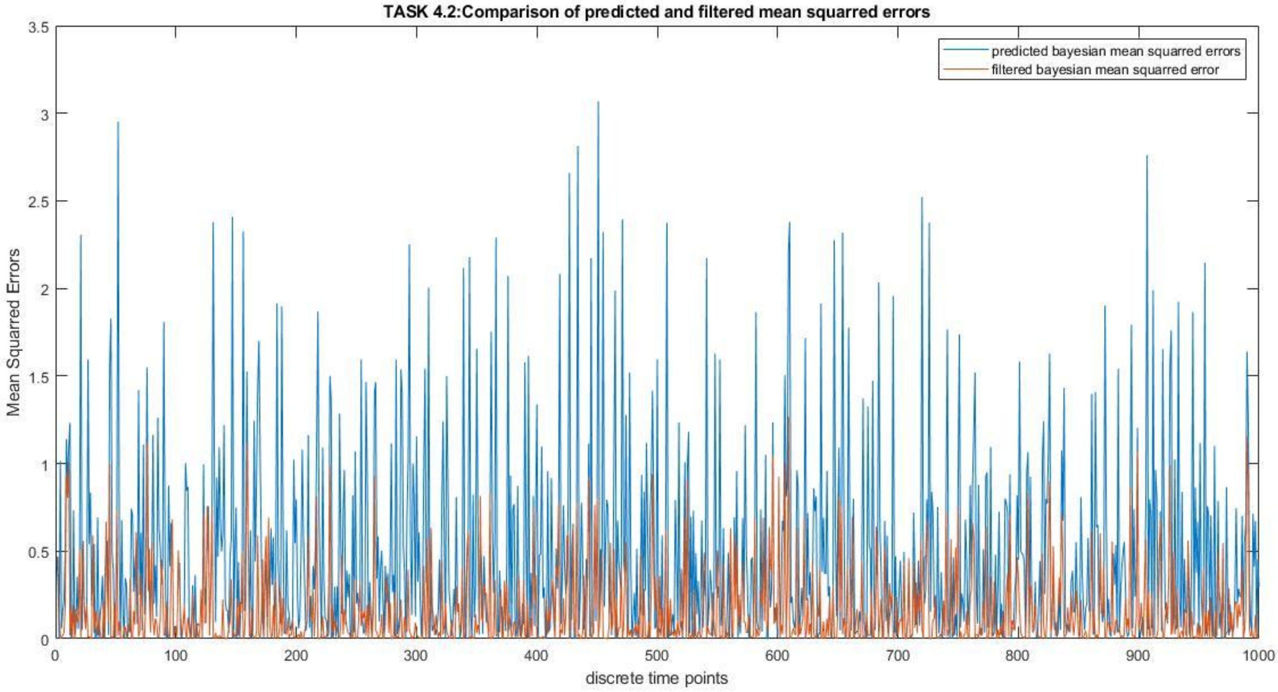


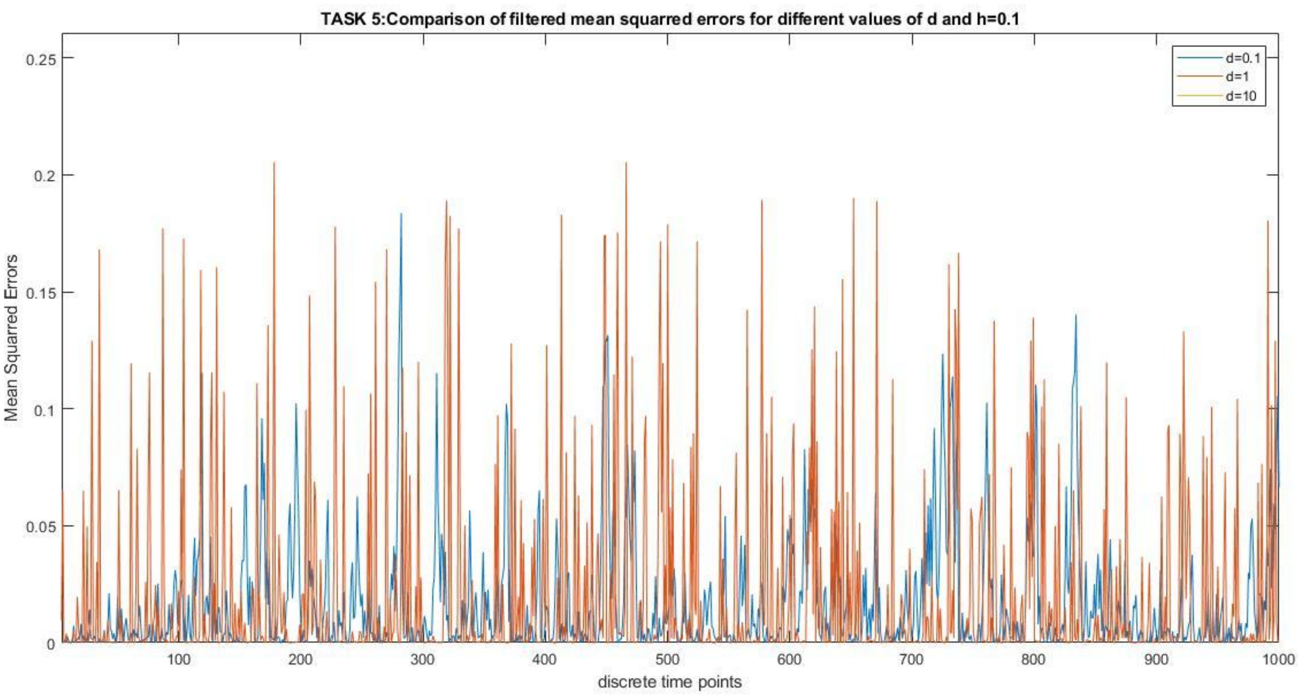
Fig 4.



Task 5:

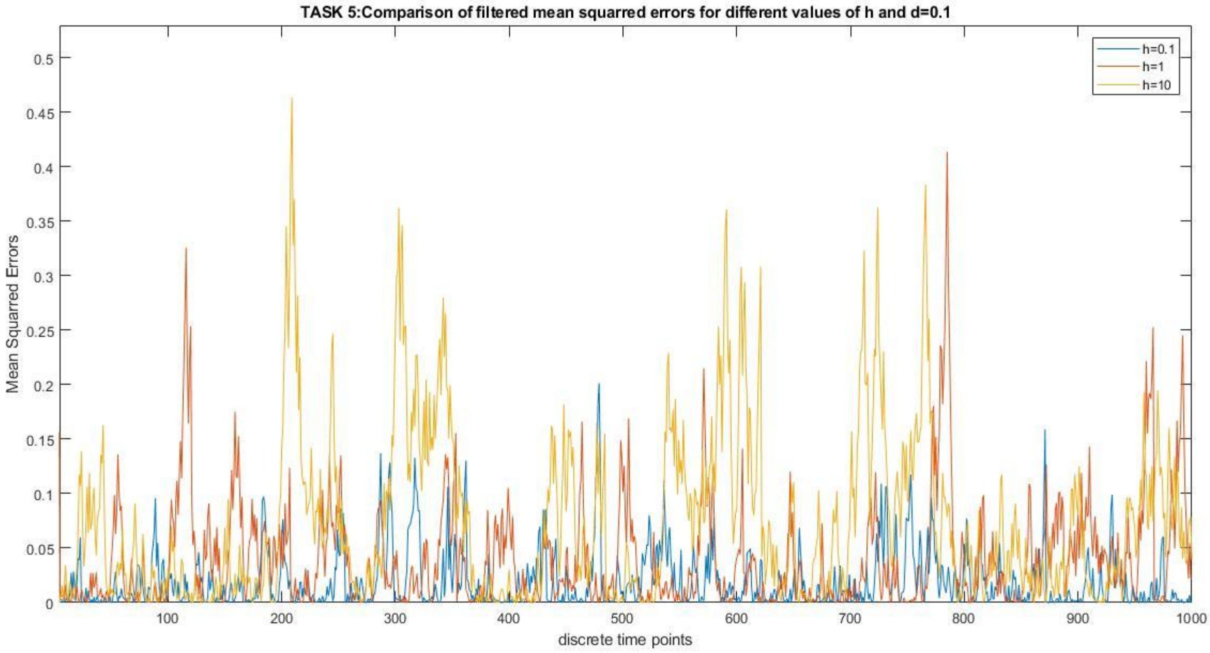
In the following, results of our programme for some d and h value-combinations are given. We have compared only the mean squared errors of the filtered values with respect to the true values for these different combinations because that gives us a clear idea of which combination can give us a filtered result that is closer to the true value.

Fig 5: Only changing the d values first



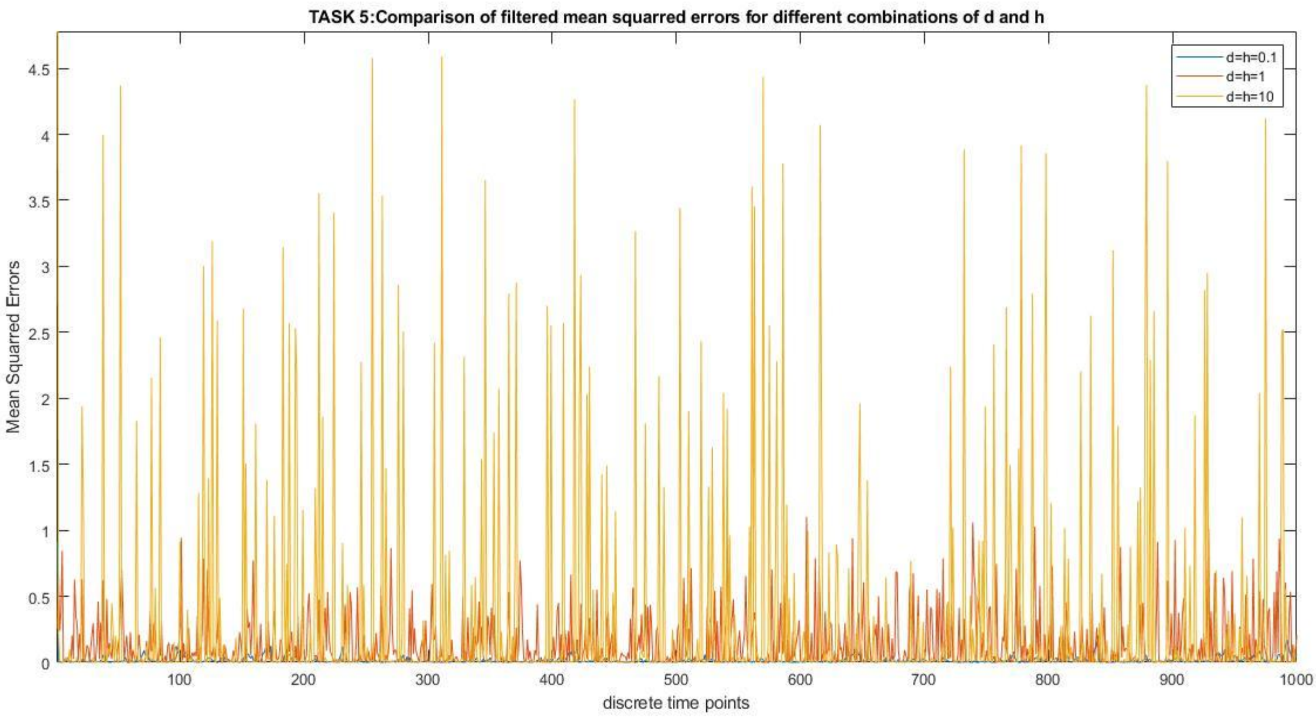
We can observe one thing that for d, the effect of increment does not have a very profound impact on the plots. From 0.1 to 1, there has been a significant change in the mean squared error, but for the increment from 1 to 10, the effect is not that proportional. This proves our system is lesser sensitive to changes in d than in h (shown after the following paragraph). However, as d increases, system has more disturbance in it, which increases the mean squared error.

Fig 6:Only changing h values



However, when we are changing h values gradually, we can see considerable changes in the mean squared errors. This tells that our system is more sensitive to changes in h than in d. Obviously, as h increases, the noise level increases which means the error in the estimation grows, as evident from the plot above.

Fig 7: Changing both h and d values



When we are changing both d and h values, the system is showing expected result as when the (d,h) both increase, the system has more noise and disturbance in it, which means the mean squared error must increase.

Task 6:

In the following some comparative plots between the true state values, filtered state means by Bayesian Filter and filtered state means by Extended Kalman Filter are given. The first plot is for d=0.3 and h =0.3 (for number of time steps= =number of sample points=1000).

Fig 8:Comparison of means and true values of states

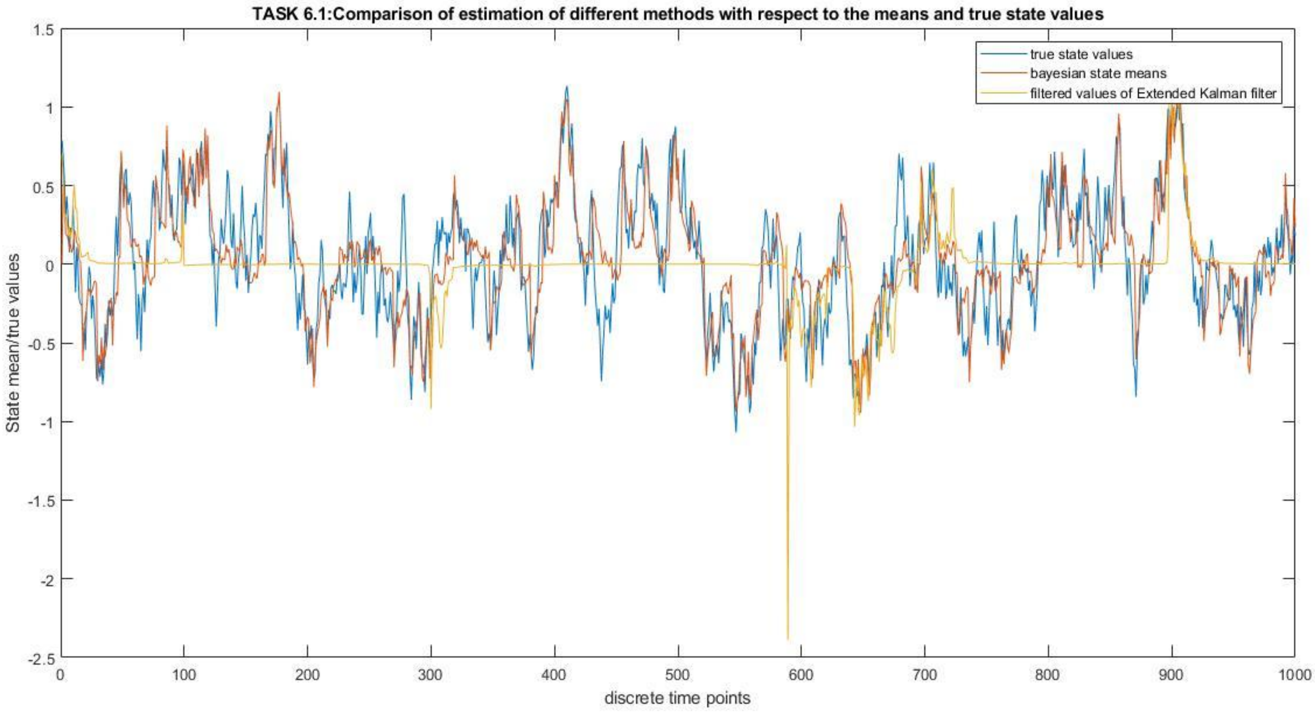
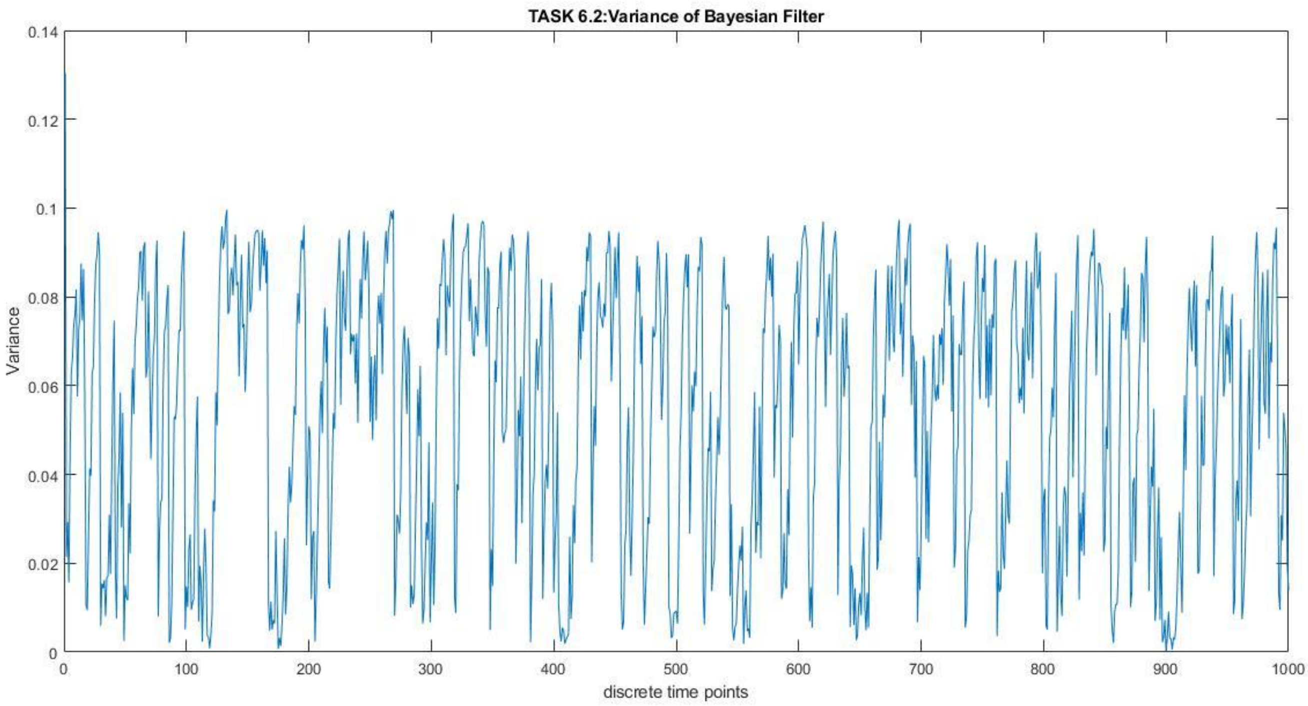
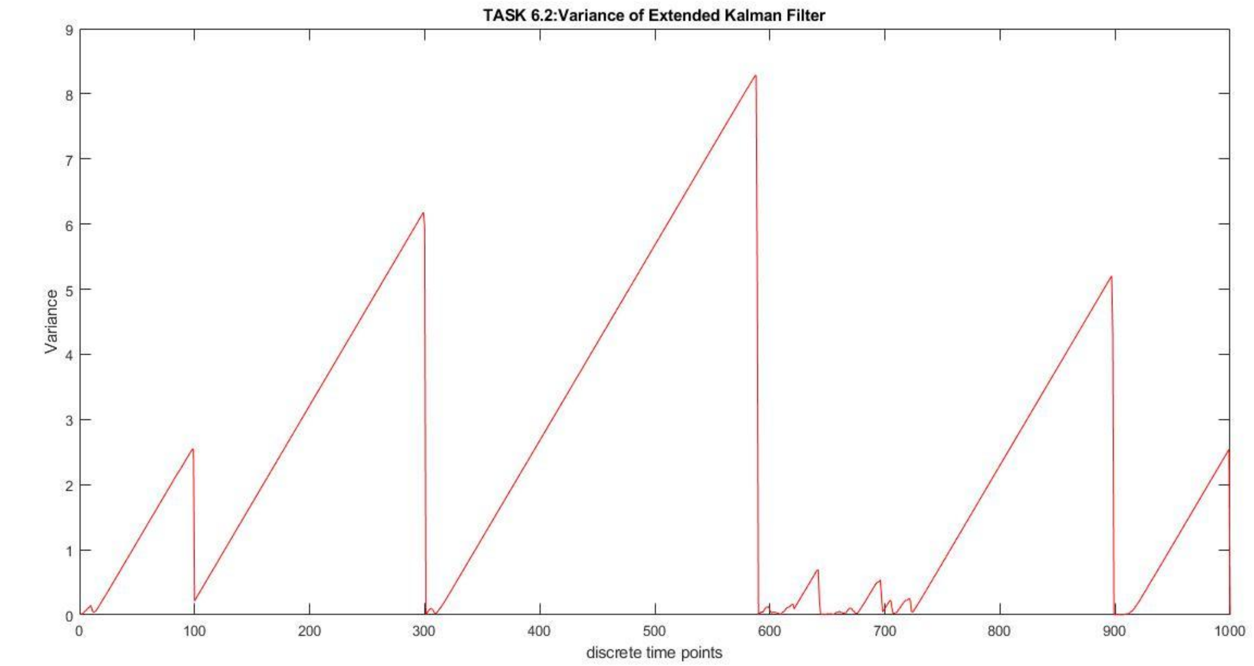


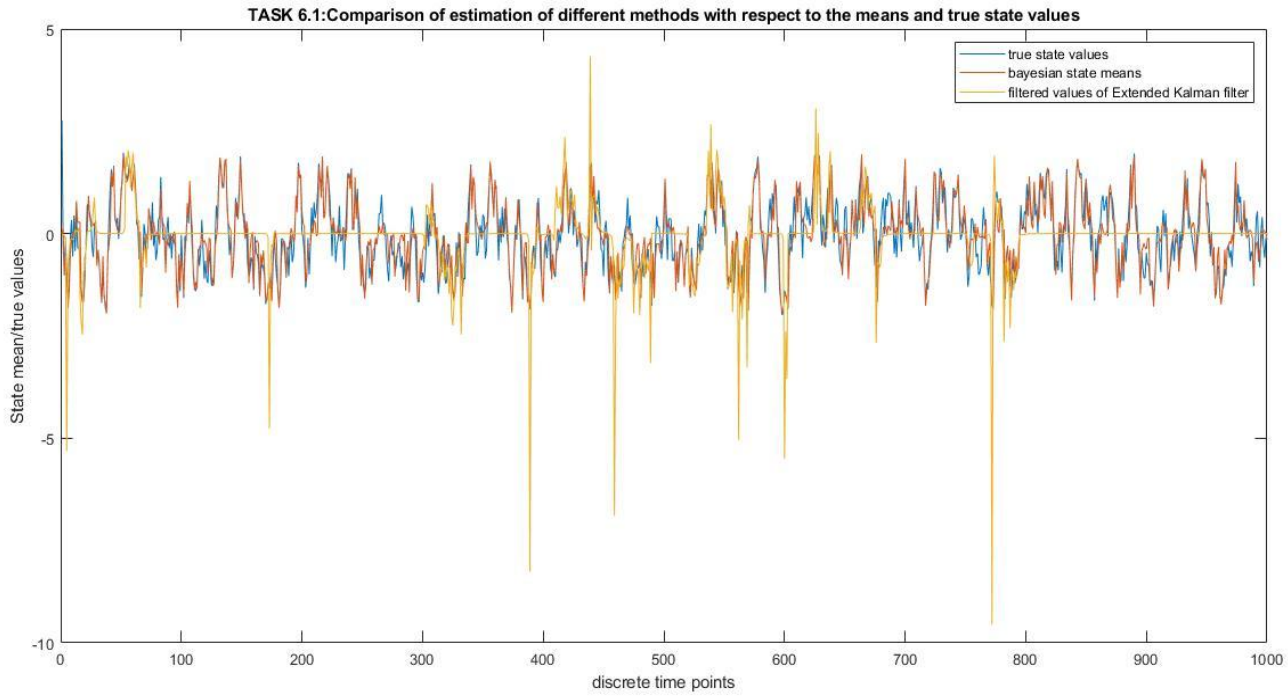
Fig 9. Comparison of variances of different methods in consecutive plots (for different scales)





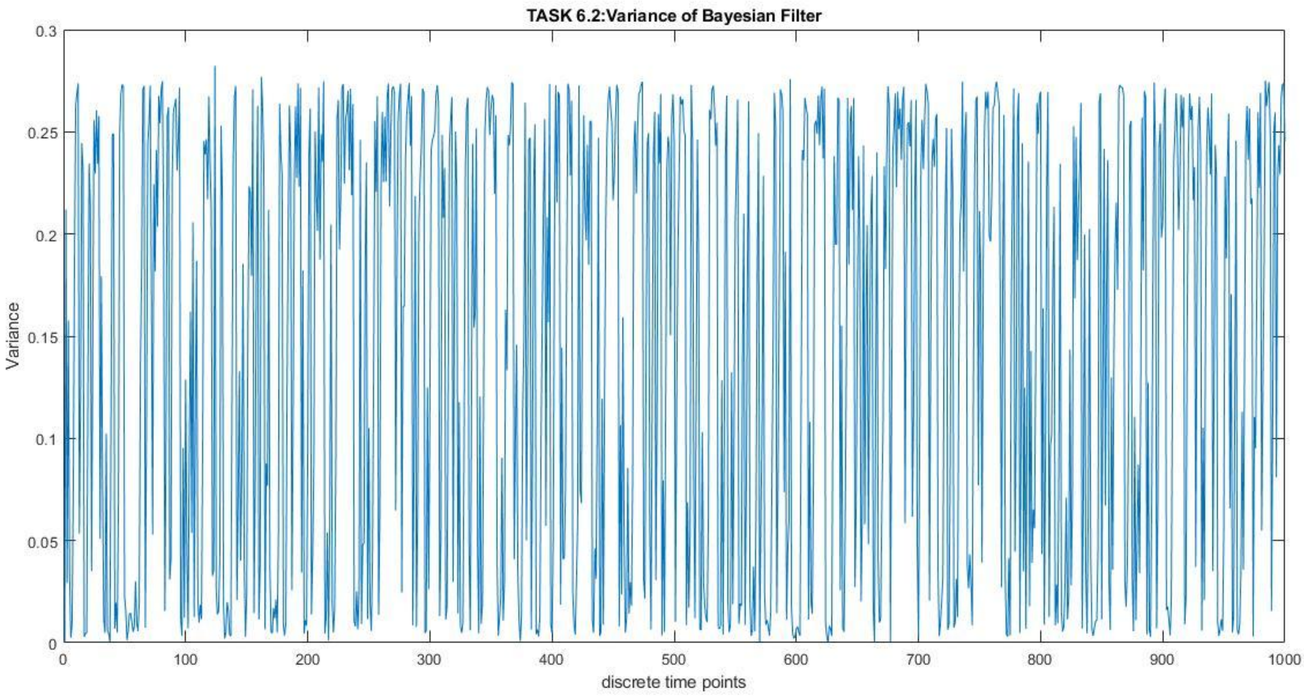
Another similar exercise is done for d=1 and h =1 (for number of time steps= =number of sample points=1000). Firstly, we get:

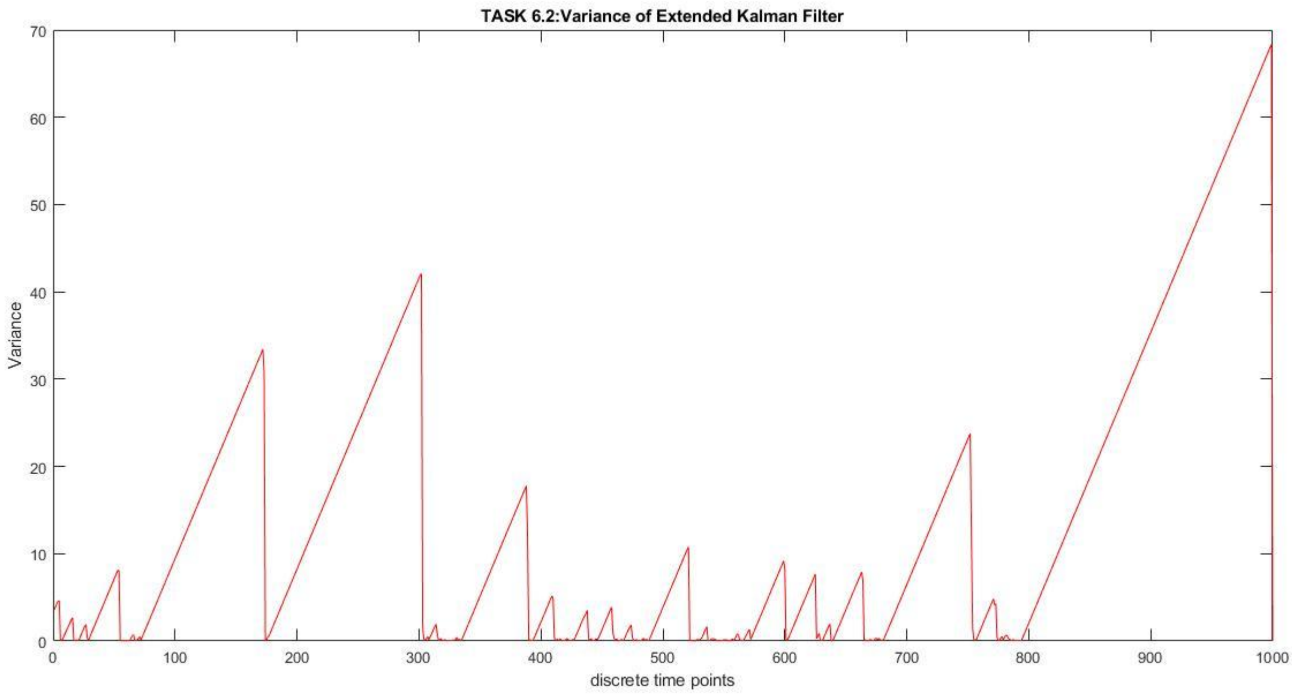
Fig 10: Mean comparison of different methods with true values of states



And similar comparison between variances is also given below, like in previous case:

Fig 11: Variance comparison of different methods in consecutive plots





We can see from the two cases considered above, that for nonlinear systems, Bayesian estimates(mean values) follow the true values more than estimates by

Extended Kalman Filtering. From the variances also, we can see such an inference only. This is because, Bayesian Filter implements bayes law directly and which gives an exact estimation algorithm. Extended Kalman Filter algorithm however, implements a linearization scheme which is just an approximation of the original estimation.

We can thus infer that Extended Kalman Filter is worse than Bayesian Filter in these two particular scenarios, for this nonlinear system.

Task 7:

I am giving myself 2280 dollars (my gross one month stipend in this Ph.D. programme, something that I never get totally in hand because of taxes! Hence atleast once, I want to get it fully.) for this performance in the estimation task assigned.

The MATLAB code that has been used for almost all the tasks is copied below:

```
% Soumya Ganguly
% PID : A53274333
% Final project
% MAE 288 B: Optimal Estimation
clc;
clear all;

%%
% Generation of data

% Number of timesteps
n_t = 1000;

% Noise parameters

prompt = 'Please input the range of the uniform distribution of
disturbance';
d = input(prompt)

prompt = 'Please input the range of the uniform distribution of noise';
h=input(prompt)

Y_true = zeros(1, n_t);
X_true = zeros(1, n_t);
X_true(1,1) = -pi + 2*pi*rand;
```

```
Y_true(1,1)=X_true(1, 1)^3+(-h + 2*h*rand);
for i = 2 : n_t
    X_true(1, i) = sin(X_true(1, i-1))+(-d + 2*d*rand); % generation of
true state values
    Y_true(1, i) = X_true(1, i)^3+(-h + 2*h*rand); %generation of
measurement

end

%%
% Bayesian filter

% Number of sample points for pdf
n_samp = 1000;

% Initializing filter
pdfX_f = zeros(n_t, n_samp);

X0 = linspace(-pi, pi, n_samp);
X = linspace(-1-d, 1+d, n_samp);

% Initializing predictor
pdfX_p = zeros(n_t, n_samp);
pdfX_p(1,:)= ones(1, n_samp)/(2*pi);

pdfX_f(1,:)= (1/(2*h))*(abs(Y_true(1, 1)-X0.^3) < h).*pdfX_p(1,:);
pdfX_f(1,:)= (n_samp/(2*pi))*pdfX_f(1,:)/sum(pdfX_f(1,:));
pdfX_p(2, :) = (2*pi/n_samp)*((1/(2*d))*(abs(X'-sin(X0)) <
d)*pdfX_f(1,:)')';

X_mean_filtered(1,1)=((2*pi/n_samp)*pdfX_f(1,:)*X0');
var_f_X(1,1)=((2*pi/n_samp)*pdfX_f(1,:)*(X0.^2)') -
(((2*pi/n_samp)*pdfX_f(1,:)*X0')^2);
X_mean_predicted(1,1)=((2*pi/n_samp)*pdfX_p(1,:)*X0');

for i = 2 : n_t
    % Filter update
    pdfX_f(i, :) = (1/(2*h))*(abs(Y_true(1, i)-X.^3) < h).*pdfX_p(i, :);
    pdfX_f(i, :) = (n_samp/(2*(1+d)))*pdfX_f(i, :)/(sum(pdfX_f(i, :)));
    % Predictor update
    pdfX_p(i+1, :) = (2*(1+d)/n_samp)*((1/(2*d))*(abs(X'-sin(X)) <
d)*pdfX_f(i, :)')';
    X_mean_filtered_bayes(1,i) = ((2*(1+d)/n_samp)*pdfX_f(i,:)*X')';
    var_f_X(1,i)=((2*(1+d)/n_samp)*pdfX_f(i,:)*(X.^2)') -
(((2*(1+d)/n_samp)*pdfX_f(i,:)*X')^2);
    X_mean_predicted_bayes(1,i)= ((2*(1+d)/n_samp)*pdfX_p(i,:)*X')';
end


figure(1)
plot(X_true)
hold on
plot(X_mean_predicted_bayes)
hold on
plot(X_mean_filtered_bayes)
legend('true values','predicted bayesian mean','filtered bayesian mean')
xlabel('discrete time points')
ylabel('State means')
```

```
title('TASK 4.1:Comparison of predicted and filtered means in Bayesian
Estimation')


figure(2)
plot((X_mean_predicted_bayes - X_true).^2)
hold on
plot((X_mean_filtered_bayes-X_true).^2)
legend('predicted bayesian mean squarred errors','filtered bayesian mean
squarred error')
xlabel('discrete time points')
ylabel('Mean Squarred Errors')
title('TASK 4.2:Comparison of predicted and filtered mean squarred errors')
%%
Extended Kalman filter

% Initializing predictor
Pp = zeros(1, n_t+2);          %predictor covariance
xp = zeros(1, n_t+2);          % predictor state
yp = zeros(1, n_t+2);          % predictor output
Pp(1, 1) = pi^2/3;             % initial predictor covariance for uniform
distribution
xp(1, 1) = 1e-1;               % initial predictor mean value, cannot take
as zero,
% because then every subsequent result will be zero

% Initializing filter
Pf = zeros(1, n_t+1); %filter covariance
xf = zeros(1, n_t+1); %filtered state
yf = zeros(1, n_t+1); % filtered output
% Variance of noise
Q = d^2/3; % true for uniform distribution
R = h^2/3;

% Run the Extended Kalman filter
for i = 1 : n_t
    % Filter update
    H = 3*xp(1, i)^2;
    K = Pp(1, i)*H/(H^2*Pp(1, i)+R); % kalman gain
    xf(1, i) = xp(1, i) + K*(Y_true(1, i)-xp(1, i)^3);
    Pf(1, i) = Pp(1, i)-Pp(1, i)^2*H^2/(H^2*Pp(1, i)+R);

    % Predictor update
    xp(1, i+1) = sin(xf(1, i));
    F = cos(xf(1, i));
    Pp(1, i+1) = F^2*Pf(1, i)+Q;
end

%
Plots of Task 6
figure(3)
plot(X_true)
hold on
% plotting expected value of state obtained from bayesian filter
plot(X_mean_filtered_bayes)
hold on
%plotting EKF state data
plot(xf(1, 2 : n_t+1))
legend('true state values','bayesian state means','filtered values of
Extended Kalman filter')
```

```
xlabel('discrete time points')
ylabel('State mean/true values')
title('TASK 6.1:Comparison of estimation of different methods with respect
to the means and true state values')

figure(4)
plot(var_f_X)
%legend('variance of bayesian filter')
xlabel('discrete time points')
ylabel('Variance')
title('TASK 6.2:Variance of Bayesian Filter')

figure(5)
plot(Pf(1,2:n_t+1),'r')
xlabel('discrete time points')
ylabel('Variance')
%legend('variance of Extended Kalman Filter')
title('TASK 6.2:Variance of Extended Kalman Filter')
```