CARNEGIE MELLON UNIVERSITY
APPLIED STOCHASTIC PROCESSES
(COURSE 18-751)
HOMEWORK 7

Daniel Marew

October 16, 2017

I collaborated with :

Nebyou Yismaw
Daniel Nkemelu
Agatha Niwomugizi

Q.1

Q.2

(b)

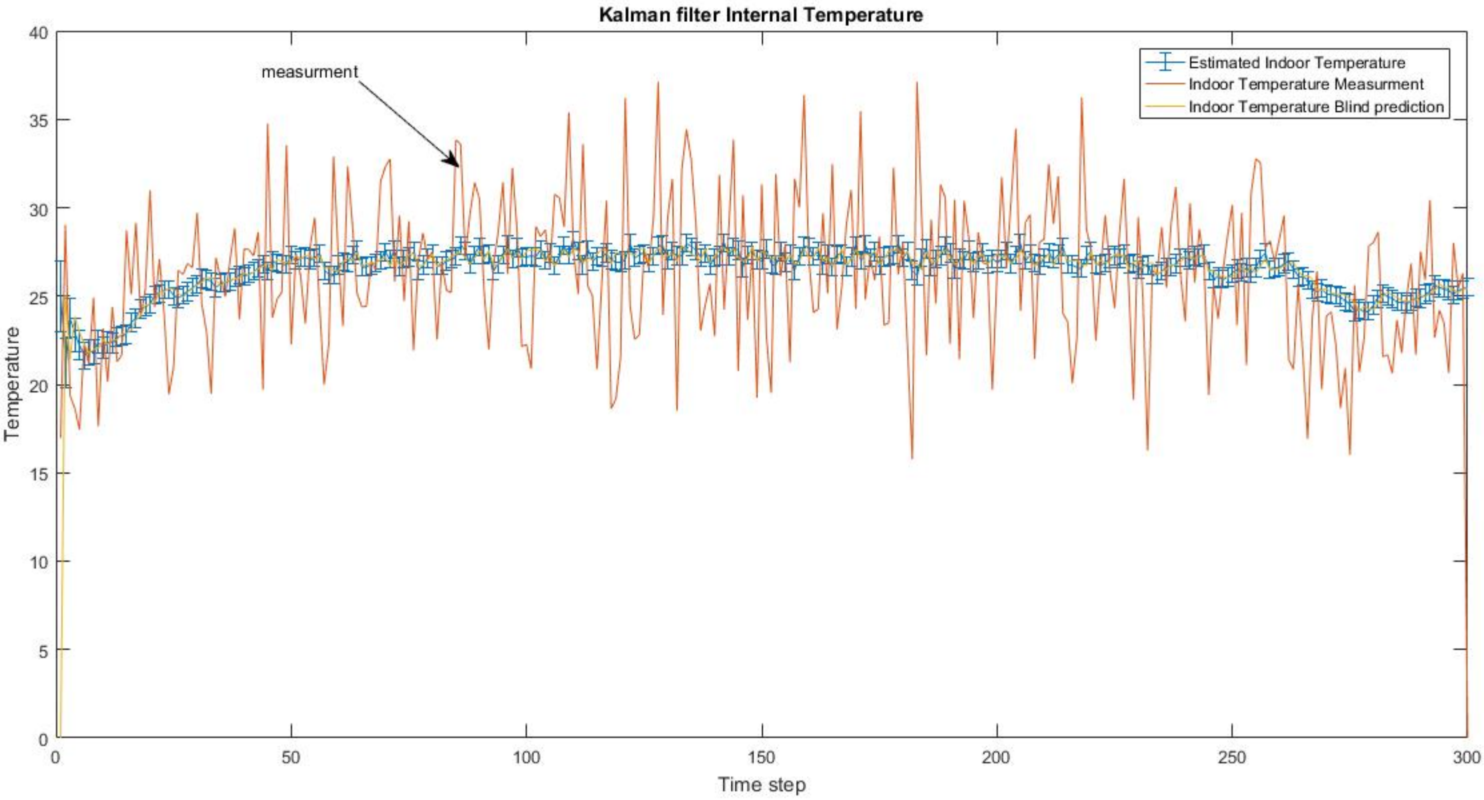

Figure 1: Kalman filter Internal Temperature

(c)

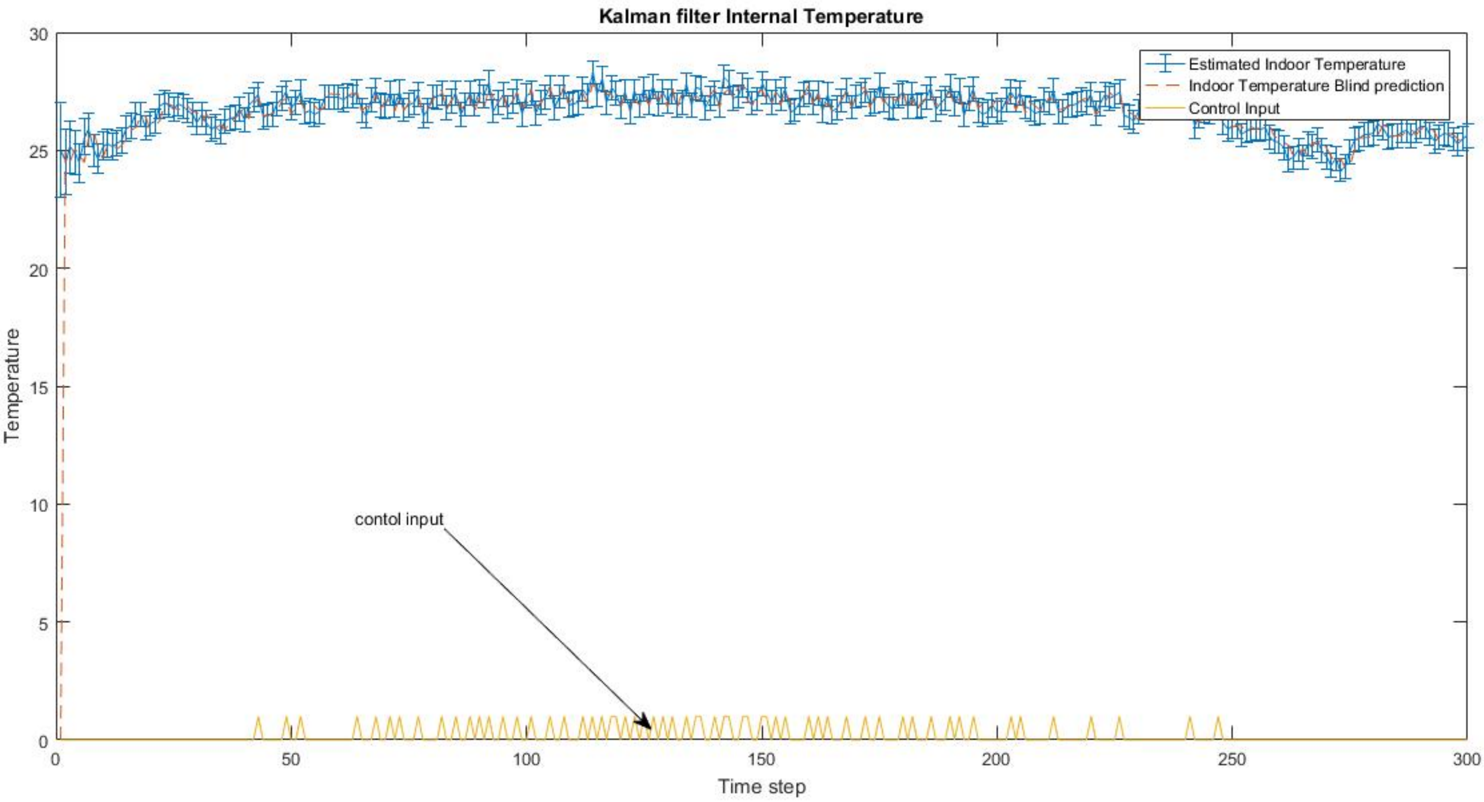

Figure 2: Kalman filter Internal Temperature with control input

Q.3

(a) $\alpha = 0.2$ $\beta = 1$

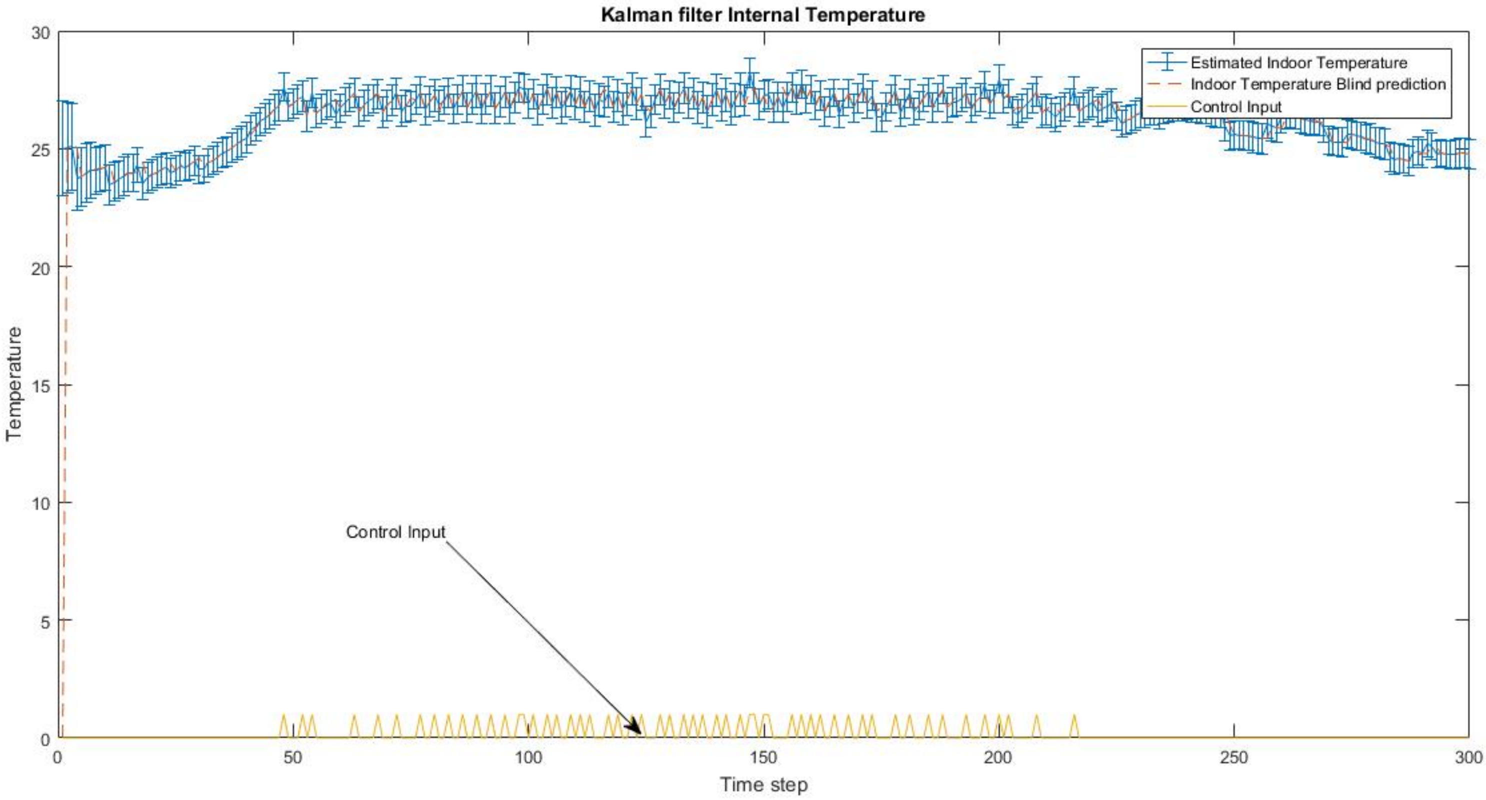

Figure 3: Kalman filter Internal Temperature with control input measurment simulated by MC with $\alpha = 0.2$ $\beta = 1$

(b) $\alpha = 0.2$ $\beta = 0.2$

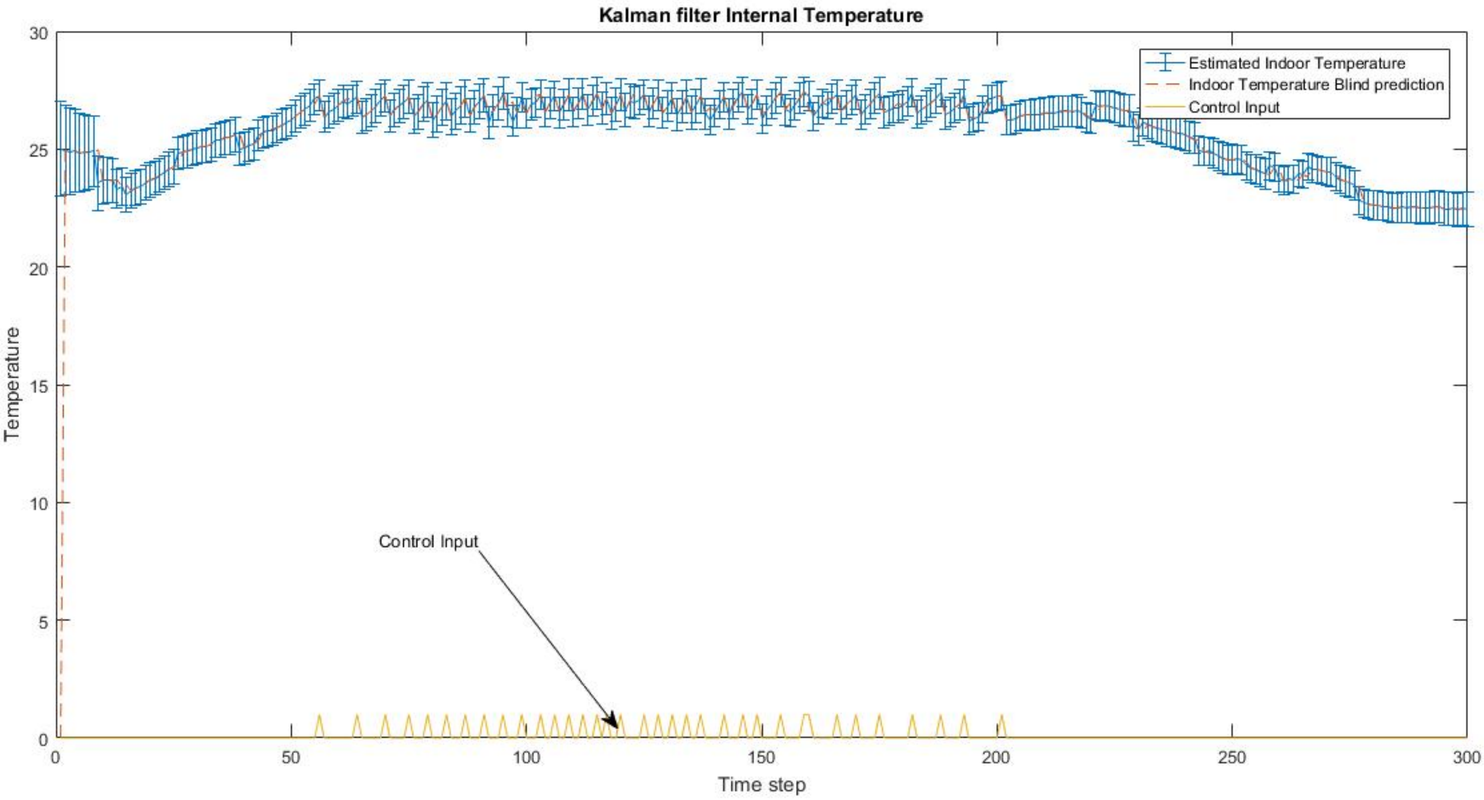

Figure 4: Kalman filter Internal Temperature with control input measurment simulated by MC with $\alpha = 0.2$ $\beta = 0.2$

Code Appendix

```
clear;
clc;
A=[0.95 0.05;0 1];%state transtion matric
H=[1 0;0 1];%measurment transition
X= zeros(2,300);% blind prediction
Var = zeros(2,300);
Xh=X;%estimate
P = zeros(2,2,300);%estimate covariance
K=P;%gain
Xh(:,1)= [25,25];%initial state estimate
numberOfTimeSteps = 299;
Q = [0.04,0 ; 0,0.01];%model covariance
R = [4 0;0 1];%measurement covariance
Z = zeros(2,300); %measurment
turnOn = 0;%comand to turn on an off air conditioner
u_k = 0;%control input
P(:,:,1)= inv(H)*R*inv(H');%initialize estimate covariance to some big value
U_k = zeros(300,1);%hold command history
B = [-1,0]';%B
meanEstimateInternal = zeros(300,1);
varianceEstimate = zeros(300,1);
alpha =0.2;%contol the MC
beta = 0.2;%control the MC
mcP = [1-alpha,alpha;beta,1-beta];%markov chain transition matrix
state = 1;%start state of MC
for n=1:numberOfTimeSteps
    u_k = turnOn;
    U_k(n) = u_k;
    v = [normrnd(0,0.04); normrnd(0,0.01)];%model noise
    %X(:,n+1)= A*X(:,n) + 0.1*sin((2*pi/300)*n) + B*u_k + v ;%B*u_k + v;
    X(:,n+1)= A*Xh(:,n) + 0.1*sin((2*pi/300)*n) + B*u_k + v ;%B*u_k + v; %prediction

    Pp = A*P(:,:,n)*A' + Q;%predicted covariance

    if state==1%if in prediction state
        Xh(:,n+1)= X(:,n+1);%take the current prediction as final estimate
        P(:,:,n+1)=Pp;
        state = discrete(mcP(state,:));
    end
    while state==2%while measurment is available update
        state = discrete(mcP(state,:));
        Z(:,n) = H*X(:,n+1) + [normrnd(0,4); normrnd(0,1)];
        %Pp = A*P(:,:,n)*A' + Q;
        K = Pp * H' * ((H*Pp*H' +R))^(-1);
        P(:,:,n+1) = (eye(2)-K*H)*Pp;
        %Xh(:,n+1)= (A-K*H*A)*Xh(:,n)+ K*Z(:,n);
```

```
        Xh(:,n+1)= X(:,n+1)+K*(Z(:,n)-H*X(:,n+1));
    end
    meanEstimateInternal(n) = Xh(1,n+1);
    varianceEstimate(n) = P(1,1,n+1);
    %if prob. temp>28 is >10%=0.1
    if qfunc((28-meanEstimateInternal(n))/sqrt(varianceEstimate(
    n)))>0.1
        turnOn = 1;
    else
        turnOn = 0;
    end
end


internalTempError = sqrt(P(1,1,:));%error matric
externalTempError = sqrt(P(2,2,:));
figure;
errorbar(1:300,Xh(1,:),internalTempError(:));
hold on;
%plot(Z(1,:));
hold on;
plot(X(1,:),'--');
plot(U_k(:,1));
title('Kalman filter Internal Temperature');
xlabel('Time step')
ylabel('Temperature')
legend('Estimated Indoor Temperature','Indoor Temperature Blind
    prediction',...
    'Control Input');
```