Kalman Filter

$$data := \begin{bmatrix} 18.55 \\ 18.59 \\ 18.19 \\ 18.45 \\ 18.62 \\ 18.88 \\ 19.23 \\ 19.16 \\ 18.9 \\ 18.92 \\ 18.73 \\ 18.75 \\ 18.38 \\ 18.1 \\ 18.13 \\ 17.84 \\ 17.63 \\ 17.76 \\ 17.99 \\ 18.33 \\ 18.0 \\ 19.13 \\ 19.57 \\ 19.98 \\ 19.92 \\ 19.61 \\ 19.74 \\ 19.57 \\ 19.61 \\ 19.31 \\ 19 \\ 19.51 \\ 18.86 \\ 18.72 \\ 19.23 \\ 19.22 \\ 19.44 \\ 19.37 \\ 19.09 \\ 18.78 \\ 18.24 \\ 18.52 \\ 18.87 \\ 18.88 \\ 19.63 \\ 20.1 \\ 19.8 \\ 19.9 \end{bmatrix}$$

```
%Original Matlab Code by
% Tim Hawkins, 1999
% Kalman Filter For 1st Order System Sensor Input
% Edited by Mitchell Kotler and John Penning
clear
%load NoisySen.dat  % Actual Noisy Data Taken from the
Feedback from
%a Temperature Sensor
load dec_z.log
x = dec_z(:,1);
y = dec_z(:,2);
z = dec_z(:,3);
NoisySen = z;
len = length(NoisySen)
Q = 1;
R = 200; % Adjust R for Degree of Damping
for n = 1: 1: len;
   zz(n) = NoisySen(n);
end;
len
Q
R
for n = 1: 1: len;
   index(n) = n;
end;
Pmin1 = 0;
Pmin1
K = Pmin1/(Pmin1+R)
X_hat(1) = NoisySen(1) + K*(zz(1) - NoisySen(1))
P = (1-K)*Pmin1
Pmin = P + Q
X_hat_min(1) = X_hat(1)
for n = 2: 1: len;
   K = Pmin/(Pmin+R);
          X_hat(n) = X_hat_min(n-1) + K*(X_hat(n-1) -
X_hat_min(n-1));
   P = (1-K)*Pmin;
   Pmin = P + Q;
   X_hat_min(n) = X_hat(n);
end;
figure
plot
(index,zz,'g-',index,zz,'gx',index,X_hat,'b--',index,X_h
at,'b*')
legend('Z','Z','X^','X^')
title(['1st Order Kalman Filter Results; R = ',num2str
(R)])
axis([0 length(zz) 0 max(zz)]);
```

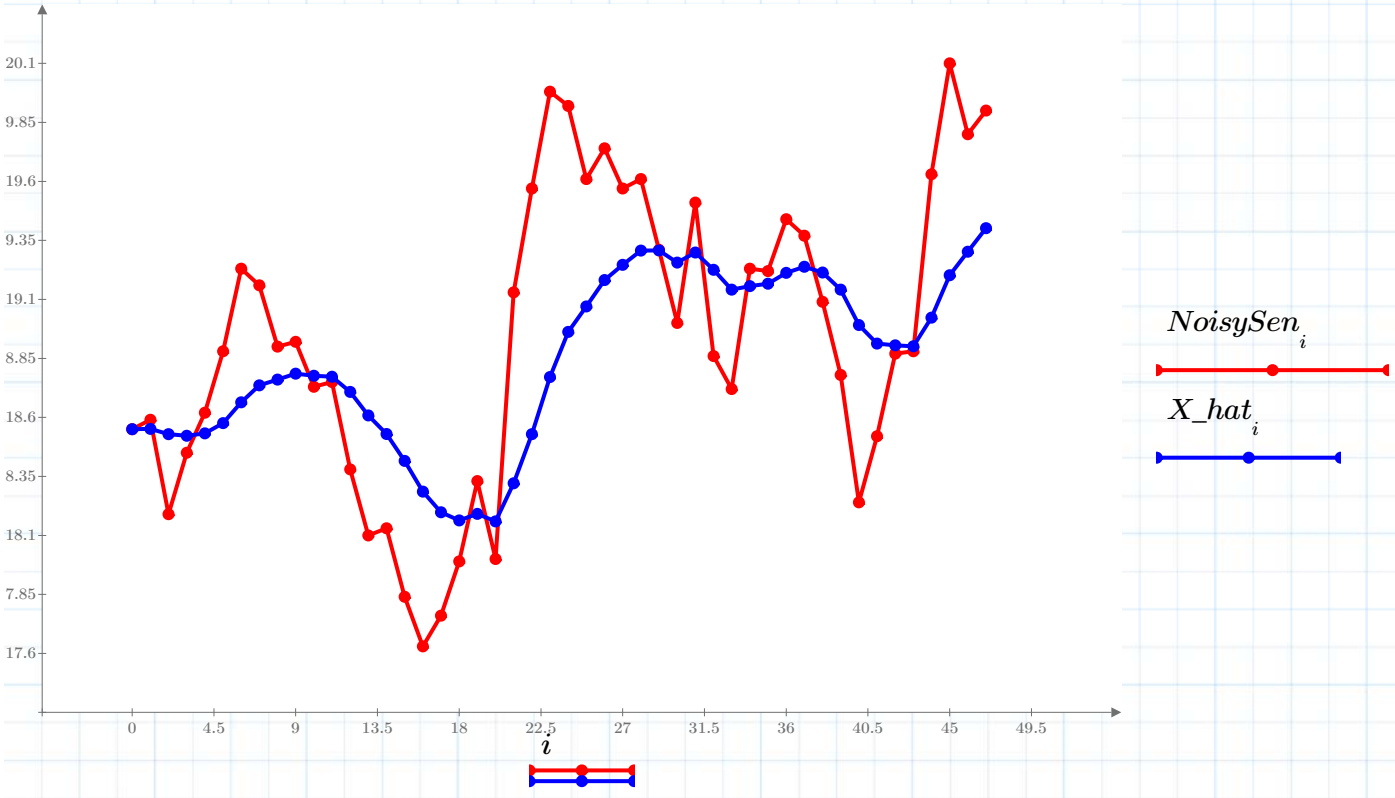
kwws=22z z z 1fefw|1gh2gdv0ndqp dq0ibhu0hlqidfk0hunodhuw0whk04
kwws=22z z z 1fefw|1gh2gdv0ndqp dq0ibhu0hlqidfk0hunodhuw0whk05
Ndqp dq0Ibhu#p dc#jdq}#hlqidfk1sgi

$NoisySen := data$

$len := \text{last}(NoisySen) = 47 \qquad Q := 1 \qquad R := 30$

$$
\begin{aligned}
X\_hat := \;& Pmin1 \leftarrow 0 \\
& K \leftarrow \frac{Pmin1}{(Pmin1 + R)} \\
& X\_hat_0 \leftarrow NoisySen_0 \\
& poise \leftarrow (1 - K) \cdot Pmin1 \\
& Pmin \leftarrow poise + Q \\
& X\_hat\_min_0 \leftarrow X\_hat_0 \\
& \text{for } n \in 1 .. len \\
& \quad K \leftarrow \frac{Pmin}{(Pmin + R)} \\
& \quad X\_hat_n \leftarrow X\_hat\_min_{n-1} + K \cdot \left(NoisySen_n - X\_hat\_min_{n-1}\right) \\
& \quad poise \leftarrow (1 - K) \cdot Pmin \\
& \quad Pmin \leftarrow poise + Q \\
& \quad X\_hat\_min_n \leftarrow X\_hat_n \\
& X\_hat
\end{aligned}
$$

$i := 0 .. \text{last}(NoisySen)$



Xp vhw}xqj#lq#FRGHV\V

```
FUNCTION_BLOCK KalmanFilter
VAR_INPUT
        rNoisySensor: REAL;
        rQ: REAL;
        rR: REAL;
END_VAR
VAR_OUTPUT
        rSensor: REAL;
END_VAR
VAR
        X_hat: REAL;
        X_hat_min: REAL;
        poise: REAL;
        Pmin: REAL;
        K: REAL;
END_VAR

K := Pmin/(Pmin + rR);
X_hat := X_hat_min + K*(rNoisySensor - X_hat_min);
poise := (1 - K)*Pmin;
Pmin := poise + rQ;
X_hat_min := X_hat;
rSensor := X_hat;
```

WrGr=#Xp vhw}xqj#lq#F111