Notes on the Kalman filter

W.S. Harwin[*], University of Reading

March 25, 2009

1 Introduction

Credited to R.E. Kalman but similar algorithms found by T.N. Thiele and P. Swerling. Developed by R.S. Bucy.

The Kalman filter can be seen in action on most GPS satnav systems. As a car enters a tunnel the GPS signal is lost but the satnav will continue to try to maintain the position based on the Kalman filter model.

Similar to Wiener filter.

Note: The notation here follows that of Welch and Bishop but see end.

2 Continuous time

State equations are

$$\dot{x} = Ax + Bu + w$$

and the Measurement equations are

$$z = Hx + v$$

where v and w are zero mean noise with $E[ww^T] = Q$, and $E[vv^T] = R$. Although not necessarily so, we make the assumption that the state and the measurement noise is uncorrelated, ie $E[wv^T] = 0$,

Note: $E[x]$ is known as the expected value of variable x. In the following we will compute an estimate of P which is the expected value of the correlation matrix $(x_k - \hat{x}_k)(x_k - \hat{x}_k)^T$ i.e. $P_k = E[(x_k - \hat{x}_k)(x_k - \hat{x}_k)^T]$

2.1 Time update (prediction)

$$\begin{aligned} \dot{\hat{x}}_t^- &= A\hat{x}_t + Bu_t \\ \dot{\hat{P}}_t^- &= A\hat{P}_t + \hat{P}_t A^T + Q_t \text{ (Ricatti equation)} \end{aligned}$$

In a non-realtime system a solution to the full Ricatti equation would be used.

Then integrate e.g. numerical Euler integration

$$\hat{x}_t^- = \hat{x}_t + \Delta\dot{\hat{x}}_t^-$$

$$\hat{P}_t^- = \hat{P}_t + \Delta\dot{\hat{P}}_t^-$$

2.2 Measurement update

$$\begin{aligned} K_t &= \hat{P}_t^- H_k^T (H_t \hat{P}_t^- H_t^T + R_t)^{-1} \\ \hat{x}_t &= \hat{x}_t^- + K_t(z_t - H\hat{x}_t^-) \\ \hat{P}_t &= (I - K_t H_t)\hat{P}_t^- \end{aligned}$$

In the above we use the predict the measurement ($\hat{z}_t^- = H\hat{x}_t^-$) to reconcile the system state.

3 In sampled time i.e. the discrete Kalman filter

State equations are

$$x_k = Ax_{k-1} + Bu_{k-1} + Gw_{k-1}$$

Measurement equations

$$z_k = Hx_k + v_k$$

where again $E[ww^T] = Q$, $E[vv^T] = R$, and $E[wv^T] = 0$,

[*]Digital Notebook © February 2007

3.1 Time update (prediction)

$$\begin{aligned}\hat{x}_k^- &= A\hat{x}_{k-1} + Bu_{k-1} \\ \hat{P}_k^- &= A\hat{P}_{k-1}A^T + Q_{k-1}\end{aligned}$$

3.2 Measurement update

$$\begin{aligned}K_k &= \hat{P}_k^- H_k^T (H_k \hat{P}_k^- H_k^T + R_k)^{-1} \\ \hat{x}_k &= \hat{x}_k^- + K_k(z_k - H\hat{x}_k^-) \\ \hat{P}_k &= (I - K_k H_k)\hat{P}_k^-\end{aligned}$$

3.3 The discrete extended Kalman filter

State equations are

$$x_k = f(x_{k-1}, u_k, w_k)$$

Measurement equations

$$z_k = h(x_k, v_k)$$

where again $E[ww^T] = Q$, $E[vv^T] = R$, and $E[wv^T] = 0$,

3.4 Time update (prediction)

$$\begin{aligned}\hat{x}_k^- &= f(\hat{x}_{k-1}, u_{k-1}, 0) \\ \hat{P}_k^- &= A_k\hat{P}_{k-1}A_k^T + W_k Q_{k-1} W_k^T\end{aligned}$$

where $A = \frac{\partial f_i}{\partial x_j}$ and $W = \frac{\partial f_i}{\partial w_j}$. Where there is a poor noise model there is little help in calculating W so most people treat it as an identity matrix.

3.5 Measurement update

$$\begin{aligned}K_k &= \hat{P}_k^- H_k^T (H_k \hat{P}_k^- H_k^T + V_k R_k V_k^T)^{-1} \\ \hat{x}_k &= \hat{x}_k^- + K_k(z_k - h(\hat{x}_k^-, 0)) \\ \hat{P}_k &= (I - K_k H_k)\hat{P}_k^-\end{aligned}$$

where $H = \frac{\partial h_i}{\partial x_j}$ and $V = \frac{\partial h_i}{\partial v_j}$. Similar argument to W, treat as an identity matrix.

3.6 Explanation

The Kalman filter has two ways to estimate state, the first is 'open loop' relying on perfect knowledge of A and B and any inputs to the system u. In this case the Kalman filter gain K_k is zero and the filter applies the normal state space equations. This condition would require R to be large compared with HPH^T approaching infinitely noisy measurements. The second case relies on perfect knowledge of the measurements of the output. If an inverse to $h(x_k)$ or H existed then the state could be recovered as $x = H^{-1}z$. Since this is not normally true the Kalman filter defaults to a modivied pseuo-inverse of H and sets the filter gain to $K_k = PH(HPH^T)^{-1}$. Under these circumstances the measurement update equation (in zero noise) becomes $\hat{x}_k = K_k z_k$

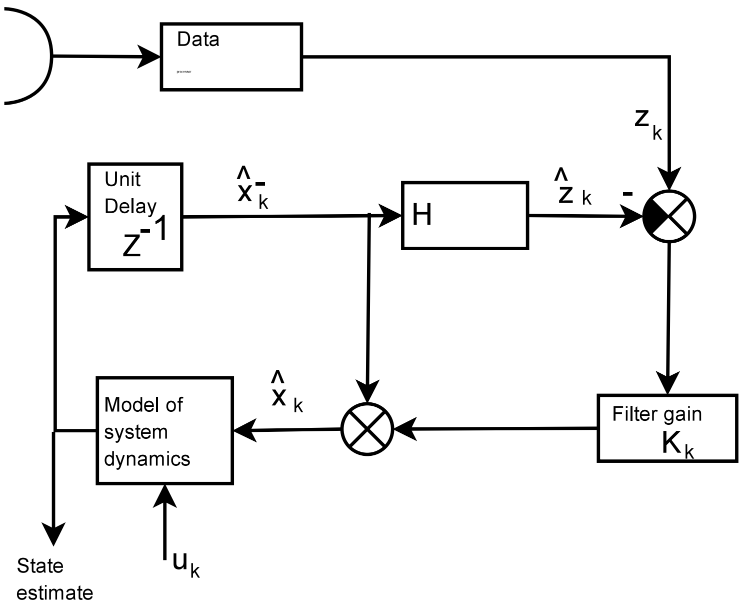

Figure 1. Kalman filter based on Singer and Behnke

3.7 Random walk

We can use the Kalman filter to predict the movement of a person or robot. The following should run in matlab or octave.

```
% Where's Wally
% A Kalman filter to predict the 2D location of a 1st order system
% with integrator
% Should be able to play with the time constant, the sample time,
% Q and R, the measurement noise, the model incorrectness
T=3; % time constant for Wally
Atrue=[0 1;0 -1/T];

% This is the actual model of wally walking, i.e. a first order
% system followed by an integral. Put them into a great big state matrix
nothing=zeros(2,2);
m1=ss([Atrue nothing;nothing Atrue],[0 0;0 1;0 0;1 0],...
      [1 0 0 0;0 0 1 0],[0 0 ;0 0]);

% Simulate a walk of 20 seconds where the input is a [0,1] random variable
tt=0:.1:20;
u=randn(length(tt),2);

[x_w,x]=lsim(m1,u,tt); % simulate the walk

% if matlab use the transpose
x_w=x_w';


% Will predict from an incorrect discrete time version of the model
% The correct model would have the numerator of A(2,2) as -1
Akf=[0 1;0 -1.10/T]; % currently A(2,2) is 10% too large

sampletime=.1;
```

```
m2d=c2d(ss([Akf nothing;nothing Akf],[0 0;0 1;0 0;1 0],...
      [1 0 0 0;0 0 1 0],[0 0 ;0 0]),sampletime);

% Could use the (incorrect) discrete time model to compare to the actual walk
% Xx=[0 0 0 0]'; for ii=2:length(u);Xx(:,ii)=m2d.a*Xx(:,ii-1)+m2d.b*u(ii,:)'; end

% Now set up the Kalman filter
A=m2d.a;  % A is a (probably incorrect) model of the system
H=[1 0 0 0;0 0 1 0]; % we will be able to make a direct measure of
                     % only the x/y positions
Xmeas=[0 0 0 0]'; % our first prediction, where it starts
Pmeas=eye(4); % first guess at the covariance matrix
Q=.1*[ones(2,2) zeros(2,2);zeros(2,2) ones(2,2)]; % our estimate of
                                                  % model noise variance
R=.1;  % our estimate of the sensor noise variance
Xallpred=[]; % place to put states after predictions
Xallmeas=[]; % place to put states after measurement
Xnpred=[0;0];

tk=tt; % tk=0:sampletime:20;

for jj=1:length(tk)
  % Prediction
  Xnewpred = A*Xmeas;
  Pnewpred = A*Pmeas*A'+Q;

  Xpred=Xnewpred; % save to old values
  Ppred=Pnewpred;
  Xallpred=[Xallpred Xnewpred]; % history of prediction state
  % Measurement
  znew=x_w(:,jj) +.01*randn;
  K=Pnewpred*H'/(H*Pnewpred*H'+R);
  Xnewmeas=Xnewpred+K*(znew -H*Xnewpred);
  Pnewmeas=(eye(4)-K*H)*Pnewpred;

  Xmeas=Xnewmeas; % save to old values
  Pmeas=Pnewmeas;
  Xallmeas=[Xallmeas Xnewmeas]; % history of measurement state
end

% Make a prediction at each time point on where Wally is going
% These depend on the estimate state, and the incorrect model
Predict1=(A-eye(4))*Xallmeas; % 1*sampletime prediction
Predict10=(A^10-eye(4))*Xallmeas; % 10*sampletime prediction

deltax=[x_w(1,:); x_w(1,:)+Predict10(1,:)];
deltay=[x_w(2,:); x_w(2,:)+Predict10(3,:)];

figure(1)
plot(x_w(1,:),x_w(2,:),Xallmeas(1,:),Xallmeas(3,:));shg
title('The true path and the Kalman estimate')
figure(2)
plot(tt,x_w,tk,Xallmeas',tk,Xallpred)
title('The model states over time')
figure(3)
```

```
plot(x_w(1,:),x_w(2,:),deltax,deltay)
title('Predicting the future direction of Wally')
```

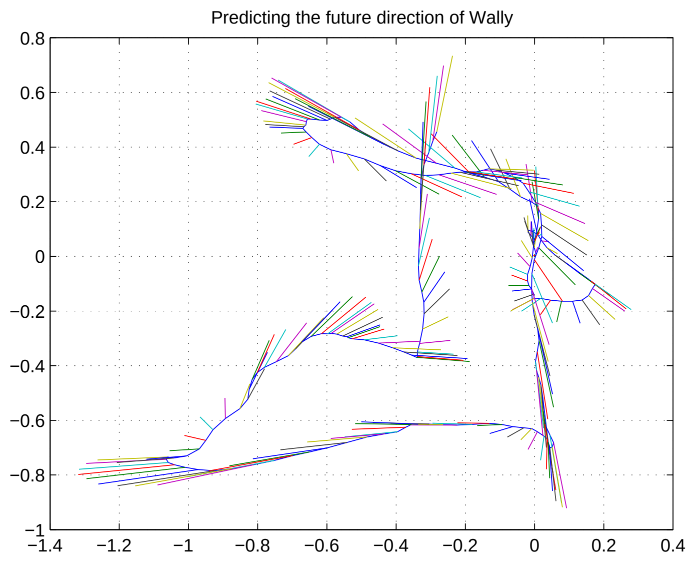

Figure 2. Predictions of a random walk

3.8 Notation

Walsh and Bishop use a slightly different notation for the development of x and P

here		W & B		Others
x_k^-	$=$	$x_{k\|k-1}$		
x_{k-1}	$=$	$x_{k-1\|k-1}$		
x_k	$=$	$x_{k\|k}$		
P_k^-	$=$	$P_{k\|k-1}$		
P_k	$=$	$P_{k\|k}$		
A	$=$	A	$=$	Φ
H	$=$		$=$	C

References

G. Welch and G. Bishop An Introduction to the Kalman Filter , Department of Computer Science at the University of North Carolina at Chapel Hill Tech. report TR 95-041 (2006)
-Online- **http://www.cs.unc.edu/˜welch/kalman/kalmanIntro.html**

R.A. Singer and K.W. Behnke Real-Time Tracking Filter Evaluation and Selection for Tactical Applications "IEEE Transactions on Aerospace and Electronic Systems " **AES-7** (1) , (doi 10.1109/TAES.1971.310257) pp. 100 - 110 (1971)