Optimal Estimation Methods

(Lecture 13 – Kalman Filtering: Part II)

Dr. John L. Crassidis

University at Buffalo – State University of New York
Department of Mechanical & Aerospace Engineering
Amherst, NY 14260-4400
johnc@buffalo.edu
http://www.buffalo.edu/~johnc

University at Buffalo The State University of New York

Kalman Filter Stability (i)

- Stability proven through a Lyapunov function

$$V(\tilde{\mathbf{x}}) = \tilde{\mathbf{x}}_k^T P_k^{-1} \tilde{\mathbf{x}}_k, \qquad \tilde{\mathbf{x}}_k \equiv \hat{\mathbf{x}}_k - \mathbf{x}_k$$ (error)

  - The increment is given by - show $\Delta V < 0$ for stability

$$\Delta V(\tilde{\mathbf{x}}) = \tilde{\mathbf{x}}_{k+1}^T P_{k+1}^{-1} \tilde{\mathbf{x}}_{k+1} - \tilde{\mathbf{x}}_k^T P_k^{-1} \tilde{\mathbf{x}}_k$$

  - Substitute the following into $\tilde{\mathbf{x}}_{k+1} = \hat{\mathbf{x}}_{k+1} - \mathbf{x}_{k+1}$

$$\hat{\mathbf{x}}_{k+1} = \Phi_k \hat{\mathbf{x}}_k + \Gamma_k \mathbf{u}_k + \Phi_k K_k [\tilde{\mathbf{y}}_k - H_k \hat{\mathbf{x}}_k]$$

$$\mathbf{x}_{k+1} = \Phi_k \mathbf{x}_k + \Gamma_k \mathbf{u}_k + \Upsilon_k \mathbf{w}_k$$

$$\tilde{\mathbf{y}}_k = H_k \mathbf{x}_k + \mathbf{v}_k$$

to give

measurement noise

process noise

$$\tilde{\mathbf{x}}_{k+1} = \Phi_k [I - K_k H_k] \tilde{\mathbf{x}}_k + \Phi_k K_k \mathbf{v}_k - \Upsilon_k \mathbf{w}_k$$

  - Ignore the inputs since the matrix $\Phi_k [I - K_k H_k]$ defines the stability of the filter $|\lambda| < 1$
    - Note, in reality a stochastic stability analysis should be done since random inputs exist, but we'll ignore this here (does not change final result) (v_k and w_k are unbounded by Gaussian Distribution)

- Then the increment leads to the following condition

$$\tilde{\mathbf{x}}_k^T \left\{ [I - K_k H_k]^T \Phi_k^T P_{k+1}^{-1} \Phi_k [I - K_k H_k] - P_k^{-1} \right\} \tilde{\mathbf{x}}_k < 0$$

- Therefore, stability is achieved if the matrix within the brackets can be shown to be negative definite (eigenvalues negative)

$$[I - K_k H_k]^T \Phi_k^T P_{k+1}^{-1} \Phi_k [I - K_k H_k] - P_k^{-1} < 0$$

- This can be rewritten as

$$I - P_{k+1} \Phi_k^{-T} [I - K_k H_k]^{-T} P_k^{-1} [I - K_k H_k]^{-1} \Phi_k^{-1} < 0 \qquad (1)$$

- Substitute the following

$$P_k^+ = [I - K_k H_k] P_k^- [I - K_k H_k]^T + K_k R_k K_k^T$$ – Update Joseph stabilized form

into

$$P_{k+1}^- = \Phi_k P_k^+ \Phi_k^T + \Upsilon_k Q_k \Upsilon_k^T$$ – propagate

to give

$$P_{k+1} = \Phi_k [I - K_k H_k] P_k [I - K_k H_k]^T \Phi_k^T + \Phi_k K_k R K_k^T \Phi_k^T + \Upsilon_k Q_k \Upsilon_k^T$$

Same as P_{k+1}^-

Kalman Filter Stability (iii)

- Substitute this into Eq. (1) to yield

$$
\begin{aligned}
&-[\Phi_k K_k R_k K_k^T \Phi_k^T + \Upsilon_k Q_k \Upsilon_k^T] \\
&\times \Phi_k^{-T}[I - K_k H_k]^{-T} P_k^{-1} [I - K_k H_k]^{-1} \Phi_k^{-1} < 0
\end{aligned}
$$

- The second term is always positive definite
- Then the condition reduces down to

$$
-[\Phi_k K_k R_k K_k^T \Phi_k^T + \Upsilon_k Q_k \Upsilon_k^T] < 0
$$

Inside bracket (positive definite)

- Clearly, if R_k is positive definite and Q_k is at least positive semi-definite, then the Lyapunov condition is satisfied and the discrete-time Kalman filter is stable
  - This means that the filter will "track" the measurements even if the measurements are unbounded!
  - These conditions are usually always met
  - Sometimes Q_k is zero, for example when estimating for a constant parameter

Steady-State Kalman Filter

- Say we have an autonomous system
  - Filter state and output matrices as well as measurement and process noise covariance matrices are all time-invariant
  - Then the error-covariance reaches steady-state quickly
  - Covariance found by solving the *discrete algebraic Riccati equation* (DARE)

Model	$\mathbf{x}_{k+1} = \Phi\,\mathbf{x}_k + \Gamma\,\mathbf{u}_k + \Upsilon\,\mathbf{w}_k, \quad \mathbf{w}_k \sim N(\mathbf{0}, Q)$ $\tilde{\mathbf{y}}_k = H\,\mathbf{x}_k + \mathbf{v}_k, \quad \mathbf{v}_k \sim N(\mathbf{0}, R)$
Initialize	$\hat{\mathbf{x}}(t_0) = \hat{\mathbf{x}}_0$
Gain	$K = P\,H^T[H\,P\,H^T + R]^{-1}$
Covariance	$P = \Phi\,P\,\Phi^T - \Phi\,P\,H^T[H\,P\,H^T + R]^{-1}H\,P\,\Phi^T + \Upsilon\,Q\,\Upsilon^T$
Estimate	$\hat{\mathbf{x}}_{k+1} = \Phi\,\hat{\mathbf{x}}_k + \Gamma\,\mathbf{u}_k + \Phi\,K\,[\tilde{\mathbf{y}}_k - H\,\hat{\mathbf{x}}_k]$

(DARE)

Farrenkopf's Analysis (i)

- Finds the steady-state covariance for the single-axis attitude estimation problem
  - Recall the model is given by

$$\dot{\theta} = \tilde{\omega} - \beta - \eta_v$$
$$\dot{\beta} = \eta_u$$

  - Define the following propagated and updated covariances

$$P^- \equiv \begin{bmatrix} p_{\theta\theta}^- & p_{\theta\beta}^- \\ p_{\theta\beta}^- & p_{\beta\beta}^- \end{bmatrix}, \quad P^+ \equiv \begin{bmatrix} p_{\theta\theta}^+ & p_{\theta\beta}^+ \\ p_{\theta\beta}^+ & p_{\beta\beta}^+ \end{bmatrix}$$

  - Also define the following variables

$$\xi \equiv p_{\theta\beta}^- \Delta t \Big/ \sigma_n^2$$
$$S_u \equiv \sigma_u \Delta t^{3/2} \Big/ \sigma_n, \quad S_v \equiv \sigma_v \Delta t^{1/2} \Big/ \sigma_n$$

Farrenkopf's Analysis (ii)

- Farrenkopf showed (after a lot of algebra)

$$p_{\theta\theta}^{-} = \sigma_n^2 \left[\left(\frac{\xi}{S_u} \right)^2 - 1 \right]$$

$$p_{\beta\beta}^{-} = \left(\frac{\sigma_n}{\Delta t} \right)^2 \left[S_u^2 \left(\frac{1}{\xi} + \frac{1}{2} \right) - \xi \right]$$

$$p_{\theta\theta}^{+} = \sigma_n^2 \left[1 - \left(\frac{S_u}{\xi} \right)^2 \right]$$

$$p_{\beta\beta}^{+} = \left(\frac{\sigma_n}{\Delta t} \right)^2 \left[S_u^2 \left(\frac{1}{\xi} - \frac{1}{2} \right) - \xi \right]$$

where

$$\xi = -\frac{1}{2} \left[\left(\frac{S_u^2}{2} + \vartheta \right) + \sqrt{\left(\frac{S_u^2}{2} + \vartheta \right)^2 - 4 S_u^2} \right]$$

$$\vartheta = \left[S_u^2 (4 + S_v^2) + S_u^4 / 12 \right]^{1/2}$$

Farrenkopf's Analysis (iii)

- In the limiting case of very frequent updates we have

$$\sqrt{p^-_{\theta\theta}} = \sqrt{p^+_{\theta\theta}} \equiv \sigma_c = \Delta t^{1/4}\sigma_n^{1/2}\left(\sigma_v^2 + 2\sigma_u\sigma_v\Delta t^{1/2}\right)^{1/4}$$

- The even simpler limiting form when the contribution of σ_u to the attitude error is negligible is given by

$$\sigma_c = \Delta t^{1/4}\sigma_n^{1/2}\sigma_v^{1/2}$$

attitude noise

gyro noise

- Indicates a one-half power dependence on both σ_n and σ_v, and a one-fourth power dependence on the update time Δt
- This shows why it is extremely difficult to improve the attitude performance by simply increasing the update frequency
- Farrenkopf's equations are useful for an initial estimate on attitude performance
  - For a star tracker with many stars available and gyro system, Farrenkopf's equations agree very well with a detailed three-axis attitude Kalman filter

- Goal is to solve for the discrete-time Riccati equation

DARE —
$$P = \Phi\, P\, \Phi^T - \Phi\, P\, H^T [H\, P\, H^T + R]^{-1} H\, P\, \Phi^T + \Upsilon\, Q\, \Upsilon^T$$

  - First need to show that the propagation can be factored into

$$P_k = S_k Z_k^{-1} \qquad (1)$$

  for some $n \times n$ matrices S_k and Z_k

  - Rewrite the covariance propagation using the matrix inversion lemma

$$P_{k+1} = \Phi\,[\bar{H} + P_k^{-1}]^{-1}\Phi^T + \bar{Q}$$

  where $\bar{H} \equiv H^T R^{-1} H$ and $\bar{Q} \equiv \Upsilon\, Q \Upsilon^T$

  - Factor P_k and multiply the right by an identity $\Phi^{-T}\Phi^T$

$$P_{k+1} = \Phi\, P_k [\bar{H}\, P_k + I]^{-1}\Phi^T + \bar{Q}\,\Phi^{-T}\Phi^T$$

  - Factoring $[\bar{H}\, P_k + I]$ gives

$$P_{k+1} = \left\{\Phi\, P_k + \bar{Q}\,\Phi^{-T}[\bar{H}\, P_k + I]\right\}[\bar{H}\, P_k + I]^{-1}\Phi^T$$

Riccati Equation (ii)

- Collecting P_k terms gives

$$P_{k+1} = \{[\Phi + \bar{Q}\,\Phi^{-T}\bar{H}]P_k + \bar{Q}\,\Phi^{-T}\}\,[\bar{H}\,P_k + I]^{-1}\Phi^T$$

- Substituting Eq. (1) and factoring Z_k gives

$$P_{k+1} = \{[\Phi + \bar{Q}\,\Phi^{-T}\bar{H}]S_k + \bar{Q}\,\Phi^{-T}Z_k\}\,Z_k^{-1}[\bar{H}\,S_k Z_k^{-1} + I]^{-1}\Phi^T$$

- Factoring Z_k^{-1} and Φ^T into the last inverse gives

$$\begin{aligned} P_{k+1} &= \{[\Phi + \bar{Q}\,\Phi^{-T}\bar{H}]S_k + \bar{Q}\,\Phi^{-T}Z_k\}\,[\Phi^{-T}Z_k + \Phi^{-T}\bar{H}\,S_k]^{-1} \\ &\equiv S_{k+1}Z_{k+1}^{-1} \end{aligned}$$

- Therefore we now have

$$\begin{bmatrix} Z_{k+1} \\ S_{k+1} \end{bmatrix} = \mathcal{H}\begin{bmatrix} Z_k \\ S_k \end{bmatrix}$$

where the *Hamiltonian matrix* is defined as ← B is in optimal control

$$\mathcal{H} \equiv \begin{bmatrix} \Phi^{-T} & \Phi^{-T}H^T R^{-1} H \\ \Upsilon\, Q \Upsilon^T \Phi^{-T} & \Phi + \Upsilon\, Q \Upsilon^T \Phi^{-T} H^T R^{-1} H \end{bmatrix}$$

Riccati Equation (iii)

- It can be shown that if λ is an eigenvalue of $\mathcal{H}$, then λ^{-1} is also an eigenvalue of $\mathcal{H}$
  - $\mathcal{H}$ is a *symplectic* matrix
- Arrange the eigenvalues into a diagonal matrix

$$\mathcal{H}_\Lambda = \begin{bmatrix} \Lambda & 0 \\ 0 & \Lambda^{-1} \end{bmatrix}$$

where Λ is a diagonal matrix of the n eigenvalues outside of the unit circle

- It can shown that if the system is observable then all the eigenvalues are distinct. System $\neq$ observable then Kalman filter $\neq$ observable
- Then the eigenvalue/eigenvector decomposition of $\mathcal{H}$ leads to

$$\mathcal{H}_\Lambda = W^{-1}\mathcal{H}W \qquad (2)$$

where W is the matrix of eigenvectors
  - Note that the matrix $\mathcal{H}$ is not symmetric in general so the eigenvector matrix is not orthogonal in this case

Riccati Equation (iv)

- Partition W into $n \times n$ block matrices

$$W = \begin{bmatrix} W_{11} & W_{12} \\ W_{21} & W_{22} \end{bmatrix}$$

- At steady-state the unstable eigenvalues will dominate the response of P_k
- Using only the unstable eigenvalues, Eq. (2) can be partitioned as

$$\begin{bmatrix} W_{11} \\ W_{21} \end{bmatrix} \Lambda = \mathcal{H} \begin{bmatrix} W_{11} \\ W_{21} \end{bmatrix}$$

- If we make the analogy that $Z \rightarrow W_{11}$ and $S \rightarrow W_{21}$, then the steady-state solution for P with $k \rightarrow k+1$ is given by

$$\boxed{P = [W_{21}\Lambda][W_{11}\Lambda]^{-1} = W_{21}W_{11}^{-1}}$$

- Therefore, the gain K can be computed off-line and remains constant
  - This can significantly reduce the onboard computational load

Riccati Equation (v)

```
function p=dare_crass(phi,h,r,q)
% function p=dare_crass(phi,h,r,q)
% This function solves the discrete-time algebraic Riccati equation
%  The inputs are:
%    phi = state matrix (n x n)
%       h = output matrix (m x n)
%       r = measurement covariance matrix (m x m)
%       q = process noise covariance matrix (n x n)

% Form Hamiltonian Matrix
phi_i=inv(phi);n=length(phi);
ham=[phi_i' phi_i'*h'*inv(r)*h;q*phi_i' phi+q*phi_i'*h'*inv(r)*h];

% Eigenvalue/Eigenvector Decomposition
[w,lam]=eig(ham);

% Sort Eigenvalues and Get Solution
i=find(abs(diag(lam))>1);
w_sort=w(:,i);
p=real(w_sort(n+1:2*n,:)*inv(w_sort(1:n,:)));
```

Riccati Equation (vi)

```
>> phi=randn(4);
>> h=randn(2,4);
>> r=randn(2);r=r*r';
>> q=randn(4);q=q*q';
>> p=dare_crass(phi,h,r,q);
>> phi*p*phi'-phi*p*h'*inv(h*p*h'+r)*h*p*phi'+q-p

ans =

   1.0e-12 *

    0.0728    0.0977   -0.0071   -0.0457
   -0.0320   -0.1510   -0.0320    0.0875
   -0.0024   -0.0353    0.0053    0.0226
    0.0259    0.1159    0.0073   -0.0657
```

Information Filter (i)

- Gain and Covariance Equations

$$K_k = P_k^- H_k^T [H_k P_k^- H_k^T + R_k]^{-1} = P_k^+ H_k^T R_k^{-1} \quad (1)$$

$$P_k^+ = [I - K_k H_k] P_k^- = [(P_k^-)^{-1} + H_k^T R_k^{-1} H_k]^{-1} \quad (2)$$

$$P_{k+1}^- = \Phi_k P_k^+ \Phi_k^T + \Upsilon_k Q_k \Upsilon_k^T \quad (3)$$

  - The gain K_k requires an inverse of order R_k, which may cause computational and numerical difficulties for large measurement sets
    - In order to circumvent these difficulties the *information* form of the Kalman filter can be used
  - The information matrix, denoted by $\mathcal{P}$, is simply the inverse of the covariance matrix, with $\mathcal{P} \equiv P^{-1}$
  - From Eq. (2) the update equation is given by

Assumes Q^{-1} & Φ^{-1} exists

$$\mathcal{P}_k^+ = \mathcal{P}_k^- + H_k^T R_k^{-1} H_k$$

  - Equation (3) can be rewritten using the matrix inversion lemma as

$$\mathcal{P}_{k+1}^- = \left[I - \Psi_k \Upsilon_k \left(\Upsilon_k^T \Psi_k \Upsilon_k + Q_k^{-1}\right)^{-1} \Upsilon_k^T\right] \Psi_k, \quad \Psi_k \equiv \Phi_k^{-T} \mathcal{P}_k^+ \Phi_k^{-1}$$

Information Filter (ii)

- The gain can be computed directly from Eq. (1)

$$K_k = (\mathcal{P}_k^+)^{-1} H_k^T R_k^{-1}$$

- The information form clearly requires inverses of Φ_k and Q_k
  - The inverse of Φ_k exists in most cases, unless a deadbeat response (i.e., a discrete pole at zero) is given in the model . Didn't sample fast enough
  - However, Q_k may be zero in some cases, and the information filter cannot be used in this case
  - Furthermore, the inverse of $\mathcal{P}_k^+$ is required in the gain calculation
- Also, if the initial state is known precisely then $P(t_0) = 0$, and the information filter cannot be initialized
- The advantage of the information filter is that the largest dimension matrix inverse required is equivalent to the size of the state
- Even though more inverses are needed, the information filter may be more computationally efficient than the traditional Kalman filter when the size of the measurement vector is much larger than the size of the state vector

Sequential Processing (i)

- Approach processes one measurement at a time, repeated in sequence at each sampling instant
  - The gain and covariance are updated until all measurements at each sampling instant have been processed
  - The result produces estimates that are equivalent to processing all measurements together at one time instant
    - The underlying principle of this approach is rooted in the linearity of the Kalman filter update equation, where the rules of superposition apply
  - Perform a linear transformation of the measurement

$$\begin{aligned}\tilde{\mathbf{z}}_k \equiv T_k^T \tilde{\mathbf{y}}_k &= T_k^T H_k \mathbf{x}_k + T_k^T \mathbf{v}_k \\ &\equiv \mathcal{H}_k \mathbf{x}_k + \boldsymbol{\upsilon}_k\end{aligned}$$

where

$$\mathcal{H}_k \equiv T_k^T H_k$$

$$\boldsymbol{\upsilon}_k \equiv T_k^T \mathbf{v}_k$$

Sequential Processing (ii)

- The covariance of the new measurement noise is given by

$$\mathcal{R}_k \equiv E\left\{\boldsymbol{\upsilon}_k \boldsymbol{\upsilon}_k^T\right\} = T_k^T R_k T_k$$

- Clearly, since R_k is real and symmetric then if T_k is chosen to be the matrix whose columns are the eigenvectors of R_k, then $\mathcal{R}_k$ is a diagonal matrix with elements given by the eigenvalues of R_k
- Note that this decomposition has to be applied at each time instant
  - However, for many systems the measurement error process is *stationary* so that R_k is constant for all times, denoted simply by R
  - In this case, the decomposition needs to be performed only once, which can significantly reduce the computational load
- The Kalman gain and covariance update can now be performed using a sequential procedure, given by

$$K_{i_k} = \frac{P^+_{i-1_k}\mathcal{H}^T_{i_k}}{\mathcal{H}_{i_k}P^+_{i-1_k}\mathcal{H}^T_{i_k} + \mathcal{R}_{i_k}}, \quad P^+_{0_k} = P^-_k$$

$$P^+_{i_k} = [I - K_{i_k}\mathcal{H}_{i_k}]P^+_{i-1_k}, \quad P^+_{0_k} = P^-_k$$

Sequential Processing (iii)

where i represents the i^{th} measurement, $\mathcal{R}_i$ is the i^{th} diagonal element of $\mathcal{R}$, and $\mathcal{H}_i$ is the i^{th} row of $\mathcal{H}$

- The process continues until all m measurements are processed (i.e., $i = 1, 2, \ldots, m$), with

$$P_k^+ = P_{m_k}^+$$

- The state update can now be computed using

$$\hat{\mathbf{x}}_k^+ = \hat{\mathbf{x}}_k^- + P_k^+ \mathcal{H}_k^T \mathcal{R}_k^{-1} [\tilde{\mathbf{z}}_k - \mathcal{H}_k \hat{\mathbf{x}}_k^-]$$

- Note that the transformed measurement is now used in the state update equation

Example (i)

- Consider the following state-space model

$$F = \begin{bmatrix} -4 & -3 & -4 & -1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}, \quad H = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

  - Choosing a sampling interval of 0.1 seconds gives

$$\Phi = \begin{bmatrix} 0.6583 & -0.2637 & -0.3324 & -0.0820 \\ 0.0820 & 0.9862 & -0.0177 & -0.0044 \\ 0.0044 & 0.0995 & 0.9994 & -0.0002 \\ 0.0002 & 0.0050 & 0.1000 & 1.0000 \end{bmatrix}$$

  - Initial conditions for the true and estimated states

$$\mathbf{x}_0 = \begin{bmatrix} 1 & 0 & 2 & 0 \end{bmatrix}^T, \quad \hat{\mathbf{x}}_0 = \begin{bmatrix} 0 & 0 & 0 & 0 \end{bmatrix}^T$$

  - Initial covariance

$$P_0 = \text{diag}\begin{bmatrix} (2/3)^2 & 0.001 & (4/3)^2 & 0.001 \end{bmatrix}$$

    - First state has a 3σ bound of 2, third state has a 3σ bound of 4, and second and fourth states are known pretty accurately

Example (ii)

- Measurement error covariance

$$R = \begin{bmatrix} 0.01 & 0.005 \\ 0.005 & 0.02 \end{bmatrix}$$

- Simulating discrete noise (as shown before) with $\mathbf{v} \sim N(\mathbf{0}, R)$
- One way is to take eigenvalue/eigenvector decomposition

$$R = T\,\mathcal{R}\,T^T$$

- Define new variable $\mathbf{z} \sim N(\mathbf{0}, \mathcal{R})$
  - Can easily sample $\mathbf{z}$ now since its covariance is a diagonal matrix
- Then $\mathbf{v} = T\mathbf{z}$
- Note that the eigenvalue and eigenvector matrices are used in the sequential processor
- Process noise covariance set to zero
- Ran both the standard Kalman filter and the sequential processor
  - Note that there are two outputs
  - Gave nearly identical results to almost within machine precision

Example (iii)

```
% Output
h=[1 0 0 0;0 0 1 0];

% Sampling Interval and Time
dt=0.1;tf=60;t=[0:dt:tf]';m=length(t);

% State Model
f=[-4 -3 -4 -1;eye(3) zeros(3,1)];
phi=expm(f*dt);

% Measurement Covariance
r=[0.01 0.005;0.005 0.02];

% Generate Correlated Noise
[tt,lam]=eig(r);
v_uncorr=randn(m,2)*lam.^(0.5);
v=(tt*v_uncorr')';
```

Example (iv)

```
% Get Measurements
x=zeros(m,4);x(1,:)=[1 0 2 0];
for i=1:m-1
 x(i+1,:)=(phi*x(i,:)')';
end
y=(h*x')';
ym=y+[v(:,1) v(:,2)];

% Kalman Filter Parameters
xe=zeros(m,4);p_cov=zeros(m,4);
p=diag([(2/3)^2 0.001 (4/3)^2 0.001]);
p_cov(1,:)=diag(p)';
```

Example (v)

```
% Kalman Filter Loop
for i=1:m-1

% Update
gain=p*h'*inv(h*p*h'+r);
p=(eye(4)-gain*h)*p;
xe(i,:)=xe(i,:)+(gain*(ym(i,:)'-h*xe(i,:)'))';

% Propagation
p=phi*p*phi';
xe(i+1,:)=(phi*xe(i,:)')';
p_cov(i+1,:)=diag(p)';

end

% Sequential Filter Parameters
xes=zeros(m,4);p_covs=zeros(m,4);
ps=diag([(2/3)^2 0.001 (4/3)^2 0.001]);
p_cosv(1,:)=diag(ps)';
```

Example (vi)

```
% Transformed Variables
ht=tt'*h;zm=(tt'*ym')';

% Sequential Filter Loop
for i=1:m-1

% Update
for j=1:2
 hj=ht(j,:);
 gains=ps*hj'/(hj*ps*hj'+lam(j,j));
 ps=(eye(4)-gains*hj)*ps;
end
xes(i,:)=xes(i,:)+(ps*ht'*inv(lam)*(zm(i,:)'-ht*xes(i,:)'))';

% Propagation
ps=phi*ps*phi';
xes(i+1,:)=(phi*xes(i,:)')';
p_covs(i+1,:)=diag(ps)';

end
```

Example (vii)

```
% Plot Results
sig3=p_covs.^(0.5)*3;
clf
plot(t,sig3(:,1),'r',t,xes(:,1)-x(:,1),'b',t,-sig3(:,1),'r')
set(gca,'fontsize',12)
axis([0 60 -0.03 0.03])
ylabel('{x_1} Errors and Bounds')
xlabel('Time (sec)')

pause

subplot(221)
plot(t,xe(:,1)-xes(:,1))
set(gca,'fontsize',12)
axis([0 60 -2e-15 2e-15])
ylabel('{x_1}')
xlabel('Time (sec)')
```

Example (viii)

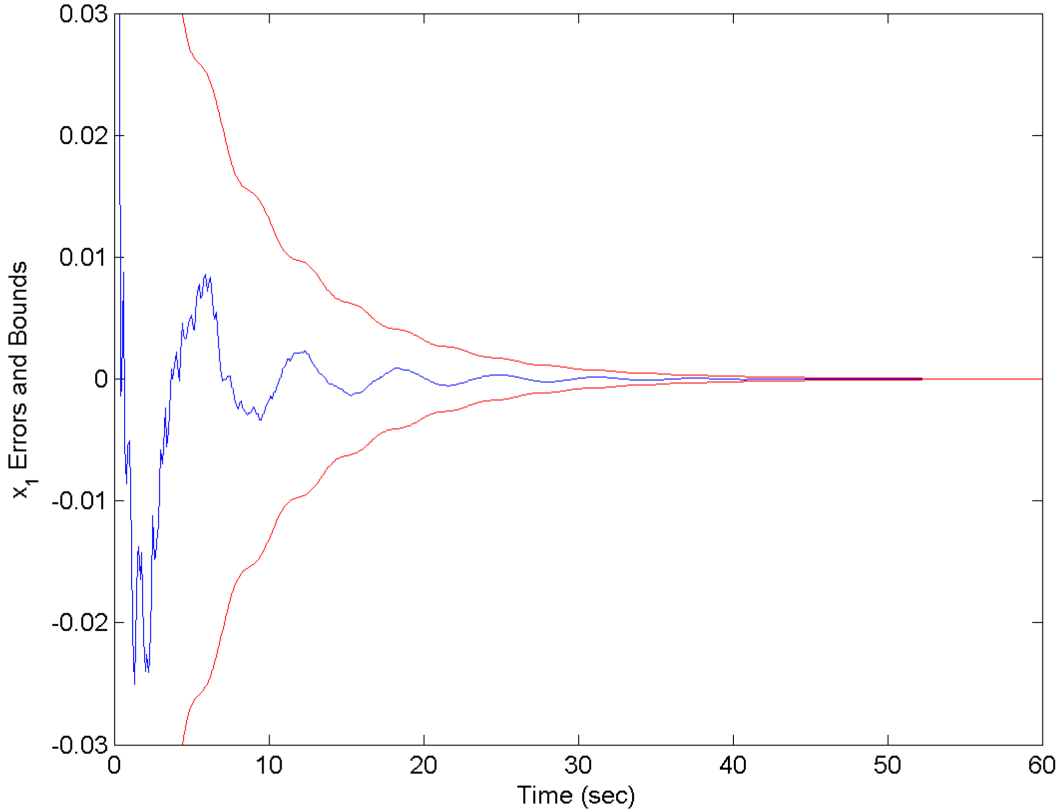
```
subplot(222)
plot(t,xe(:,2)-xes(:,2))
set(gca,'fontsize',12)
axis([0 60 -2e-15 2e-15])
ylabel('{x_2}')
xlabel('Time (sec)')

subplot(223)
plot(t,xe(:,3)-xes(:,3))
set(gca,'fontsize',12)
axis([0 60 -2e-15 2e-15])
ylabel('{x_3}')
xlabel('Time (sec)')

subplot(224)
plot(t,xe(:,4)-xes(:,4))
set(gca,'fontsize',12)
axis([0 60 -2e-15 2e-15])
ylabel('{x_4}')
xlabel('Time (sec)')
```

Example (ix)

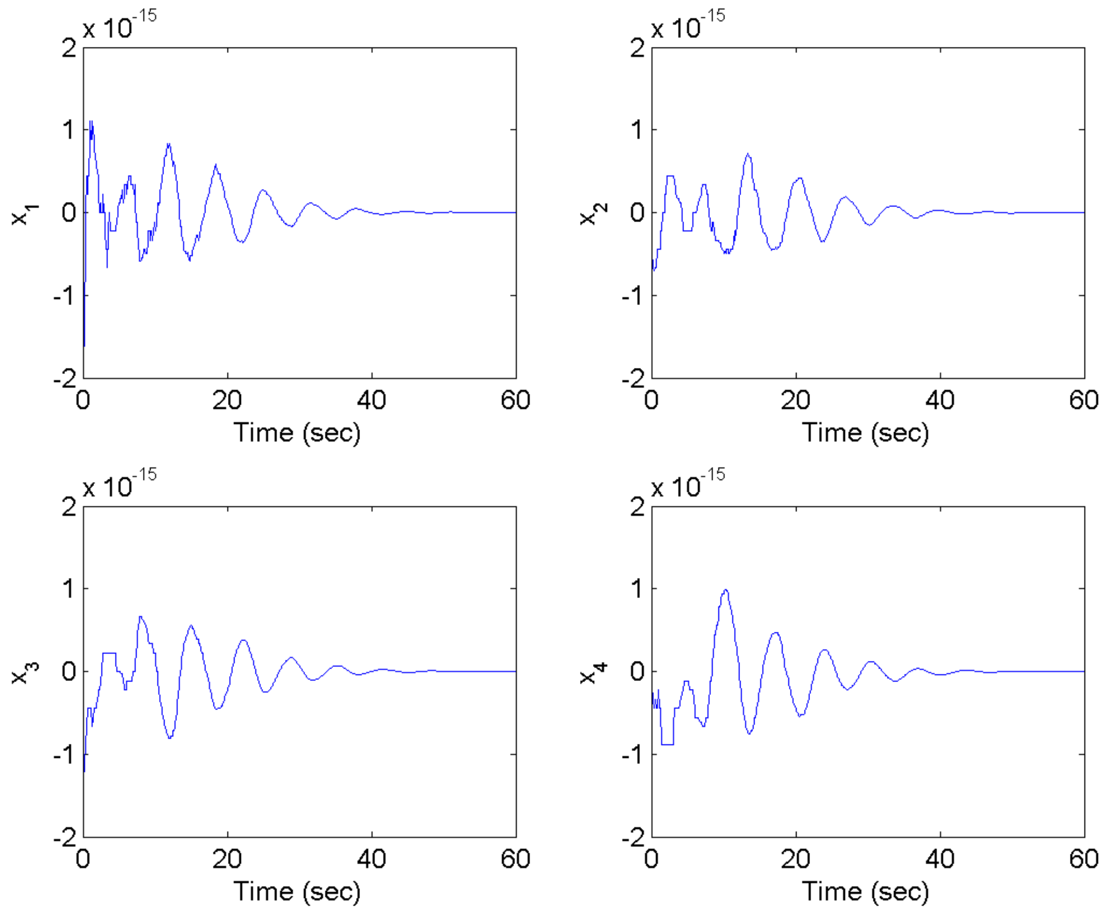
- First state estimate errors and 3σ bounds



- Estimates are well bounded
- Similar results for other states

Example (x)

- State differences between standard Kalman filter and sequential processor estimates



Cramér-Rao Inequality (i)

- We consider the following density $p(\tilde{\mathbf{Y}}|\mathbf{X})$, where $\tilde{\mathbf{Y}}_k$ denotes the sequence $\{\tilde{\mathbf{y}}_0, \tilde{\mathbf{y}}_1, \ldots, \tilde{\mathbf{y}}_k\}$ and $\mathbf{X}_k$ denotes the sequence $\{\mathbf{x}_0, \mathbf{x}_1, \ldots, \mathbf{x}_k\}$. We also denote $\hat{\mathbf{X}}_k^+$ by the sequence $\{\hat{\mathbf{x}}_0^+, \hat{\mathbf{x}}_1^+, \ldots, \hat{\mathbf{x}}_k^+\}$
  - Assuming unbiased estimates, the covariance of $\hat{\mathbf{X}}_k^+$ has a Cramér-Rao lower bound denoted by

$$\boxed{E\left\{\left(\hat{\mathbf{X}}_k^+ - \mathbf{X}_k\right)\left(\hat{\mathbf{X}}_k^+ - \mathbf{X}_k\right)^T\right\} \geq \mathcal{F}_k^{-1}}$$

where the *trajectory information* matrix is given by

$$\mathcal{F}_k = -E\left\{\frac{\partial^2}{\partial \mathbf{X}_k \partial \mathbf{X}_k^T} \ln p(\tilde{\mathbf{Y}}_k, \mathbf{X}_k)\right\}$$

  - Here the joint probability density is used because the state is stochastic in nature, due to process noise
    - If zero process noise exists then $p(\tilde{\mathbf{Y}}_k, \mathbf{X}_k)$ can be replaced with $p(\tilde{\mathbf{Y}}_k|\mathbf{X}_k)$

Cramér-Rao Inequality (ii)

- The matrix $\mathcal{F}_k$ is of dimension $(kn) \times (kn)$, which grows in time
- We are more interested as to how this matrix is related to P_k^+
- Let $\mathbf{X}_{k+1} = [\mathbf{X}_{k-1}^T \;\; \mathbf{x}_k^T \;\; \mathbf{x}_{k+1}^T]^T$ and

$$\mathcal{F}_{k+1} = \begin{bmatrix} A_{k+1} & B_{k+1} & L_{k+1} \\ B_{k+1}^T & C_{k+1} & E_{k+1} \\ L_{k+1}^T & E_{k+1}^T & G_{k+1} \end{bmatrix}$$

- We are interested in the inverse of the middle $n \times n$ matrix of $\mathcal{F}_{k+1}$, because it is related to $\mathbf{x}_k$. Denote this inverse by J_{k+1}
- Before we derive expression for these sub-matrices we first establish a recursion for the joint density

$$\begin{aligned} p(\tilde{\mathbf{Y}}_{k+1}, \mathbf{X}_{k+1}) &= p(\tilde{\mathbf{y}}_{k+1}, \tilde{\mathbf{Y}}_k, \mathbf{x}_{k+1}, \mathbf{X}_k) \\ &= p(\tilde{\mathbf{y}}_{k+1} | \mathbf{x}_{k+1}, \tilde{\mathbf{Y}}_k, \mathbf{X}_k)\, p(\mathbf{x}_{k+1} | \tilde{\mathbf{Y}}_k, \mathbf{X}_k)\, p(\tilde{\mathbf{Y}}_k, \mathbf{X}_k) \\ &= p(\tilde{\mathbf{y}}_{k+1} | \mathbf{x}_{k+1})\, p(\mathbf{x}_{k+1} | \mathbf{x}_k)\, p(\tilde{\mathbf{Y}}_k, \mathbf{X}_k) \end{aligned}$$

Cramér-Rao Inequality (iii)

- Define the following variables

$$
\begin{aligned}
D_k^{11} &= -E\left\{\frac{\partial^2}{\partial \mathbf{x}_k \partial \mathbf{x}_k^T} \ln p(\mathbf{x}_{k+1}|\mathbf{x}_k)\right\} \\
D_k^{21} &= -E\left\{\frac{\partial^2}{\partial \mathbf{x}_k \partial \mathbf{x}_{k+1}^T} \ln p(\mathbf{x}_{k+1}|\mathbf{x}_k)\right\} = (D_k^{12})^T \\
D_k^{22} &= -E\left\{\frac{\partial^2}{\partial \mathbf{x}_{k+1} \partial \mathbf{x}_{k+1}^T} \ln p(\mathbf{x}_{k+1}|\mathbf{x}_k)\right\} \\
&\quad -E\left\{\frac{\partial^2}{\partial \mathbf{x}_{k+1} \partial \mathbf{x}_{k+1}^T} \ln p(\tilde{\mathbf{y}}_{k+1}|\mathbf{x}_{k+1})\right\}
\end{aligned}
$$

Cramér-Rao Inequality (iv)

- The quantity A_{k+1} can now be computed using

$$
\begin{aligned}
A_{k+1} &= -E\left\{\frac{\partial^2}{\partial \mathbf{X}_{k-1}\partial \mathbf{X}_{k-1}^T}\ln p(\tilde{\mathbf{Y}}_{k+1}, \mathbf{X}_{k+1})\right\} \\
&= -E\left\{\frac{\partial^2}{\partial \mathbf{X}_{k-1}\partial \mathbf{X}_{k-1}^T}\left[\ln p(\tilde{\mathbf{y}}_{k+1}|\mathbf{x}_{k+1}) + \ln p(\mathbf{x}_{k+1}|\mathbf{x}_k) + \ln p(\tilde{\mathbf{Y}}_k, \mathbf{X}_k)\right]\right\} \\
&= -E\left\{\frac{\partial^2}{\partial \mathbf{X}_{k-1}\partial \mathbf{X}_{k-1}^T}\ln p(\tilde{\mathbf{Y}}_k, \mathbf{X}_k)\right\} \\
&= A_k
\end{aligned}
$$

- Others are given by

$$
B_{k+1} = B_k, \quad C_{k+1} = C_k + D_k^{11}
$$

$$
E_{k+1} = D_k^{12}, \quad G_{k+1} = D_k^{22}, \quad L_{k+1} = 0
$$

Cramér-Rao Inequality (v)

- We now have

$$\mathcal{F}_{k+1} = \begin{bmatrix} A_k & B_k & 0 \\ B_k^T & C_k + D_k^{11} & D_k^{12} \\ 0 & D_k^{21} & D_k^{22} \end{bmatrix}$$

- The matrix by J_{k+1} can now be computed through

$$\begin{aligned} J_{k+1} &= D_k^{22} - \begin{bmatrix} 0 & D_k^{21} \end{bmatrix} \begin{bmatrix} A_k & B_k \\ B_k^T & C_k + D_k^{11} \end{bmatrix}^{-1} \begin{bmatrix} 0 \\ D_k^{12} \end{bmatrix} \\ &= D_k^{22} - D_k^{21}(C_k - B_k^T A_k^{-1} B_k + D_k^{11})^{-1} D_k^{12} \end{aligned}$$

- This can be simplified down to

$$\boxed{J_{k+1} = D_k^{22} - D_k^{21}(J_k + D_k^{11})^{-1} D_k^{12}}$$

- The initial J_0 is computed using

$$\boxed{J_0 = -E\left\{\frac{\partial^2}{\partial \mathbf{x}_0 \partial \mathbf{x}_0^T} \ln p(\mathbf{x}_0)\right\}}$$

$p(\mathbf{x}_0)$ is the initial density function

Cramér-Rao Inequality (vi)

- To achieve the Cramér-Rao bound we must show

$$J_k = (P_k^+)^{-1} \equiv \mathcal{P}_k^+$$

  - For simplicity we assume that Υ_k is given by the identity matrix and that Q_k^{-1} exists
  - Densities of interest for the Kalman filter are

$$p(\mathbf{x}_0) = \frac{1}{[\det(2\pi P_0)]^{1/2}} \exp\left[-\frac{1}{2}(\mathbf{x}_0 - \hat{\mathbf{x}}_0)^T P_0^{-1}(\mathbf{x}_0 - \hat{\mathbf{x}}_0)\right]$$

$$p(\mathbf{x}_{k+1}|\mathbf{x}_k) = \frac{1}{[\det(2\pi Q_k)]^{1/2}} \exp\left[-\frac{1}{2}(\mathbf{x}_{k+1} - \Phi_k\mathbf{x}_k)^T Q_k^{-1}(\mathbf{x}_{k+1} - \Phi_k\mathbf{x}_k)\right]$$

$$p(\tilde{\mathbf{y}}_{k+1}|\mathbf{x}_{k+1}) = \frac{1}{[\det(2\pi R_k)]^{1/2}} \times \exp\left[-\frac{1}{2}(\tilde{\mathbf{y}}_{k+1} - H_{k+1}\mathbf{x}_{k+1})^T R_k^{-1}(\tilde{\mathbf{y}}_{k+1} - H_{k+1}\mathbf{x}_{k+1})\right]$$

- We now have

$$D_k^{11} = \Phi_k^T Q_k^{-1} \Phi_k, \quad D_k^{21} = -Q_k^{-1} \Phi_k, \quad D_k^{22} = Q_k^{-1} + H_{k+1}^T R_{k+1}^{-1} H_{k+1}$$

- Substituting into J_{k+1} gives

$$J_{k+1} = Q_k^{-1} - Q_k^{-1} \Phi_k (J_k + \Phi_k^T Q_k^{-1} \Phi_k)^{-1} \Phi_k^T Q_k^{-1} + H_{k+1}^T R_{k+1}^{-1} H_{k+1}$$

- After some rearranging we can show that the inverse of P_k^+, denoted by $\mathcal{P}_k^+$, from the Kalman filter is given by

$$\mathcal{P}_{k+1}^+ = Q_k^{-1} - Q_k^{-1} \Phi_k (\mathcal{P}_k^+ + \Phi_k^T Q_k^{-1} \Phi_k)^{-1} \Phi_k^T Q_k^{-1} + H_{k+1}^T R_{k+1}^{-1} H_{k+1}$$

- Thus $J_k \equiv \mathcal{P}_k^+$

- This proves that the Kalman filter achieves the Cramér-Rao lower bound and thus is an *efficient* estimator

Also an unbiased estimator.

Relationship to Least Squares

- The Kalman filter can be derived using a least-squares type loss function

$$J = \frac{1}{2}(\hat{\mathbf{x}}_0 - \mathbf{x}_0)^T \mathcal{P}_0 (\hat{\mathbf{x}}_0 - \mathbf{x}_0) + \frac{1}{2}\sum_{i=1}^{k}(\tilde{\mathbf{y}} - H_i\hat{\mathbf{x}}_i)^T R_i^{-1}(\tilde{\mathbf{y}} - H_i\hat{\mathbf{x}}_i)$$

subject to

$$\hat{\mathbf{x}}_{i+1} = \Phi_i\,\hat{\mathbf{x}}_i + \Upsilon_i \mathbf{w}_i, \quad i = 1,\, 2,\, \ldots,\, k-1$$

- The derivation is very long, but it can be shown that minimizing this loss function subject to the constraint gives back the Kalman filter equations

Correlated Filter (i)

- Assume that $\mathbf{w}_{k-1}$ and $\mathbf{v}_k$ are now correlated, with

$$E\left\{\mathbf{w}_{k-1}\mathbf{v}_k^T\right\} = S_k$$

  - From before we have

$$\tilde{\mathbf{x}}_{k+1}^- = \Phi_k \tilde{\mathbf{x}}_k^+ - \Upsilon_k \mathbf{w}_k$$

  - Then since $\tilde{\mathbf{x}}_{k-1}^+$ is uncorrelated with $\mathbf{v}_k$ we have

$$E\left\{\tilde{\mathbf{x}}_k^- \mathbf{v}_k^T\right\} = E\left\{\left(\Phi_{k-1}\tilde{\mathbf{x}}_{k-1}^+ - \Upsilon_{k-1}\mathbf{w}_{k-1}\right)\mathbf{v}_k^T\right\} = -\Upsilon_{k-1}S_k$$

  - Covariance update equation now becomes (valid for any gain)

$$P_k^+ = E\{(I - K_k H_k)\overbrace{\cancel{\tilde{\mathbf{x}}_k^- \tilde{\mathbf{x}}_k^{-T}}}^{P_k^-}(I - K_k H_k)^T\} + E\{(I - K_k H_k)\overbrace{\cancel{\tilde{\mathbf{x}}_k^- \mathbf{v}_k^T}}^{-\Upsilon_{k-1}S_k} K_k^T\}$$

$$+ E\{K_k \overbrace{\cancel{\mathbf{v}_k \tilde{\mathbf{x}}_k^{-T}}}^{-S_k^T \Upsilon_{k-1}^T}(I - K_k H_k)^T\} + E\{K_k \overbrace{\cancel{\mathbf{v}_k \mathbf{v}_k^T}}^{R_k} K_k^T\}$$

$$= [I - K_k H_k] P_k^- [I - K_k H_k]^T + K_k R_k K_k^T$$
$$- [I - K_k H_k]\Upsilon_{k-1} S_k K_k^T - K_k S_k^T \Upsilon_{k-1}^T [I - K_k H_k]^T$$

Correlated Filter (ii)

- Minimizing its trace leads to

$$K_k = [P_k^- H_k^T + \Upsilon_{k-1} S_k][H_k P_k^- H_k^T + R_k + H_k \Upsilon_{k-1} S_k + S_k^T \Upsilon_{k-1}^T H_k^T]^{-1}$$

  - Obtain standard gain when $S_k = 0$
- Substituting gain into covariance expression gives

$$P_k^+ = [I - K_k H_k] P_k^- - K_k S_k^T \Upsilon_{k-1}^T$$

  - Obtain standard covariance update when $S_k = 0$
- Example of filter use is an aircraft flying through a field of complex turbulence
  - The effect of turbulence on the aircraft's acceleration is complex, but can easily be modeled as process noise on $\mathbf{w}_{k-1}$
  - Since any sensor mounted on an aircraft is also corrupted by turbulence, the measurement error $\mathbf{v}_k$ is correlated with the process noise $\mathbf{w}_{k-1}$
  - Hence, the filter formulation presented here can be used directly to estimate aircraft state quantities in the face of turbulence disturbance

Correlated Filter Summary

Model	$\mathbf{x}_{k+1} = \Phi_k \mathbf{x}_k + \Gamma_k \mathbf{u}_k + \Upsilon_k \mathbf{w}_k, \quad \mathbf{w}_k \sim N(\mathbf{0}, Q_k)$ $\tilde{\mathbf{y}}_k = H_k \mathbf{x}_k + \mathbf{v}_k, \quad \mathbf{v}_k \sim N(\mathbf{0}, R_k)$ $E\left\{\mathbf{w}_{k-1}\mathbf{v}_k^T\right\} = S_k$
Initialize	$\hat{\mathbf{x}}(t_0) = \hat{\mathbf{x}}_0$ $P_0 = E\left\{\tilde{\mathbf{x}}(t_0)\,\tilde{\mathbf{x}}^T(t_0)\right\}$
Gain	$K_k = [P_k^- H_k^T + \Upsilon_{k-1} S_k]$ $\times[H_k P_k^- H_k^T + R_k + H_k \Upsilon_{k-1} S_k + S_k^T \Upsilon_{k-1}^T H_k^T]^{-1}$
Update	$\hat{\mathbf{x}}_k^+ = \hat{\mathbf{x}}_k^- + K_k[\tilde{\mathbf{y}}_k - H_k \hat{\mathbf{x}}_k^-]$ $P_k^+ = [I - K_k H_k] P_k^- - K_k S_k^T \Upsilon_{k-1}^T$
Propagation	$\hat{\mathbf{x}}_{k+1}^- = \Phi_k \hat{\mathbf{x}}_k^+ + \Gamma_k \mathbf{u}_k$ $P_{k+1}^- = \Phi_k P_k^+ \Phi_k^T + \Upsilon_k Q_k \Upsilon_k^T$