ATEP SYS12525

Day 2

Radar Trackers and Applications for SAADS

October 26, 1999

http://www.jameskbeard.com

ATEP SYS12525

Radar Trackers and Applications for SAADS

October 26, 1999

Topic 8: System Modeling

Sensor Systems Engineering for the 21st Century

Basic Concept

- Consider Position and Velocity in a State Vector:

$$\underline{x} = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} \text{< Position >} \\ \text{< Velocity >} \end{bmatrix}$$

- Position and Velocity Extrapolation Over Time Interval T:

$$\underline{x}(t+T) = \Phi(T) \cdot \underline{x}(t), \;\; \Phi(T) = \begin{bmatrix} 1 & T \\ 0 & 1 \end{bmatrix}$$

Notation

$\underline{x}$ State Vector

$\underline{f}(\underline{x})$ Vector of functions of the elements of $\underline{x}$

$\underline{u}(t)$ Known input that drives $\dot{\underline{x}}$

L Matrix mapping $\underline{u}(t)$ into $\dot{\underline{x}}$

$\underline{w}(t)$ Vector of noise inputs, covariance Q

G Matrix mapping $\underline{w}(t)$ into $\dot{\underline{x}}$

State Space System Modeling

- Continuous Formulation

$$\frac{d\underline{x}}{dt} = \underline{f}(\underline{x}) + L \cdot \underline{u}(t) + G \cdot \underline{w}(t),\ \ \mathrm{Cov}\{\underline{w}\} = Q$$

$$\underline{x}(t) = \Phi(t, t_0) \cdot \underline{x}(t_0) + \int_{t_0}^{t} \Phi(t, \tau)\big(L \cdot \underline{u}(\tau) + G \cdot \underline{w}(\tau)\big) d\tau$$

$$\frac{d}{dt}\Phi(t, t_0) = F(t) \cdot \Phi(t, t_0),\ \ F(t) = \frac{\partial \underline{f}(\underline{x})}{\partial \underline{x}},\ \ \Phi(t_0, t_0) = I$$

- Discrete Formulation

$$\underline{x}_{k+1} = \Phi \cdot \underline{x}_k + \Lambda \cdot \underline{u}_k + \Gamma \cdot \underline{w}_k$$

Continuous Formulation

- Real World
  - Most Things are Continuous
    - » Aircraft Position
    - » Sensor Position
  - Discrete Formulation Approximation Follows from Continuous Formulation
    - » Sampling Continuous Case (as in Gelb pp. 66-67)
    - » Directly Approximating the Continuous Case
- Simple to Use with Discrete Kalman Filter

State Transition Matrix

- Time Derivative of Matrix Superposition Integral

$$\underline{x}(t) = \Phi(t, t_0) \cdot \underline{x}(t_0) + \int_{t_0}^{t} \Phi(t, \tau)\left(L \cdot \underline{u}(\tau) + G \cdot \underline{w}(\tau)\right) d\tau$$

- Result

$$\dot{\underline{x}}(t) = F \cdot \Phi(t, t_0) \cdot \underline{x}(t_0) + \Phi(t, t)\left(L \cdot \underline{u}(t) + G \cdot \underline{w}(t)\right)$$

- Shows
  - Matrix Superposition Integral is State Propagation Equation
  - State Transition Matrix $\Phi(t,t_0)$ is General Homogeneous Solution to State Equation

Special Cases

- For F(t) Constant

$$\Phi(t, t_0) = \exp(F \cdot (t - t_0))$$

- For $t - t_0$ Small

$$\Phi(t, t_0) \approx I + F(t_1) \cdot (t - t_0), \;\; t_0 < t_1 < t$$

- For Constant Velocity Case

$$F = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}, \;\; \Phi(t - t_0) = \begin{bmatrix} 1 & t - t_0 \\ 0 & 1 \end{bmatrix}$$

Matrix Superposition Integral

- Generalizes Linear System Model
  - Inhomogeneities; $L \cdot \underline{u}(t)$
  - Noisy Influences; $G \cdot \underline{w}(t)$
- Similar to Scalar Superposition Integral
- Leads to General Continuous State Equation
- Necessary for Next Step
  - Covariance Propagation
  - Continuous and Discrete Formulations

Covariance Propagation Equation

- Given
  - Random Vector $\underline{x}$ with Covariance P_x
  - Functional Relationship $\underline{y}(\underline{x})=\underline{f}(\underline{x})$
- From the Taylor Expansion

$$\underline{y} = \underline{f}(\underline{x}_0) + F \cdot (\underline{x} - \underline{x}_0) + O\left((\underline{x} - \underline{x}_0)^2\right), \; F = \frac{\partial \underline{f}(\underline{x})}{\partial \underline{x}}$$

- The Mean and Covariance of $\underline{y}$ are Approximately

$$\underline{\bar{y}} \approx \underline{f}(\underline{\bar{x}}), \; P_y \approx F \cdot P_x \cdot F^T$$

State Covariance Propagation

- State Propagation Equation (Gelb, p. 76)

$$\underline{x}_{k+1} = \Phi_k \cdot \underline{x}_k + \Lambda_k \cdot \underline{u}_k + \Gamma_k \cdot \underline{w}_k$$

- Covariance of Both Sides Gives

$$P_{k+1} = \Phi_k \cdot P_k \cdot \Phi_k^T + \Gamma_k \cdot Q_k \cdot \Gamma_k^T, \ Q_k = Cov\left\{\underline{w}_k\right\}$$

- Subtracting P_k from P_{k+1} and Taking Limit
  - Gelb, page 77
  - Continuous Covariance Propagation Equation

$$\dot{P} = F \cdot P + P \cdot F^T + G \cdot Q \cdot G^T$$

Markov Processes

- Definition

A Markov Process is a sequence of Gaussian zero-mean random numbers y_i which can be produced by uncorrelated, constant variance Gaussian noise passed through a recursive filter. If the order of the filter is N, the sequence is called Markov-N.

- Example

$$y_{i+1} = a \cdot y_i + b \cdot n_i,\;\; n_i \in G(0,1),\;\; \left\langle x_i \cdot x_j \right\rangle = \delta_{i,j}$$

- Steady State Variance (Noise Gain)

$$\left\langle y^2 \right\rangle = \frac{b^2}{1-a^2} \cdot \left\langle n^2 \right\rangle$$

Noise Gain Through Filter

- Compare DC Gain to Noise Gain

$$\langle y \rangle = \frac{b}{1-a} \cdot \langle x \rangle$$

- Power Ratio, Noise to DC

$$Noise\,Gain = \frac{1+a}{1-a} = \frac{1}{\tanh\left(\frac{T}{2\tau}\right)}$$

- Correlation Time is ▲

Vector Markov Process

- A Markov-N Process
  - Can Be Represented by a Markov-1 Process
  - Involving an N-Vector

$$\underline{y}_{k+1} = A \cdot \underline{y}_k + B \cdot \underline{n}_k$$

- Steady State Covariance

$$P_\infty = \sum_{i=0}^{\infty} A^i \cdot B \cdot B^T \cdot A^{iT}$$

- Converges if All Characteristic Values of A are Inside the Unit Circle (Why?)

From Gelb Pages 51 – 56

- Time Domain Emphasized
  - – Recent Work
  - – Frequency Domain Presented in Class
- N^{th} Order Differential Equation
  - – Matrix Representation – Companion Form
  - – Relationship with Block Diagram
- Mechanical System Example

From Gelb Pages 57 – 63

- State Transition Matrix
  - A More General Form for the State Equation is

$$\underline{\dot{x}}(t) = \underline{f}\left(\underline{x}(t), t\right),\ \underline{x}(t_0) = \underline{x}_0$$

  - The Equivalent State Transition Matrix is

$$\Phi(t, t_0) = \frac{\partial \underline{x}(t)}{\partial \underline{x}_0}$$

- Does This Work for Gelb's Cases?

From Gelb, Pages 63 – 67

- Matrix Superposition Integral

$$\underline{x}(t) = \Phi(t,t_0)\cdot \underline{x}_0 + \int_{t_0}^{t} \Phi(t,\tau)\cdot L(\tau)\cdot \underline{u}(\tau)\cdot d\tau$$

- Note for General Case

$$\frac{d}{dt}\Phi(t,t_0) = \frac{\partial \dot{\underline{x}}_H}{\partial \underline{x}_0} = \frac{\partial \underline{f}(\underline{x}_H)}{\partial \underline{x}_0} = F(t)\cdot\Phi(t,t_0)$$

$$F(t) = \frac{\partial \underline{f}(\underline{x}_H(t),t)}{\partial \underline{x}_H}$$

From Gelb, Pages 72 – 78

- Covariance Propagation Equation
  - Noise in True State Treated
  - Estimation Not Considered
- Noise in True State
  - Driven by “Plant Noise”
  - Covariance Q
- Applicability to Our Generalization
  - Holds for Discrete Case
  - Linearized Approximation for Continuous Case

From Gelb Problems 3-1 and 3-2, page 97

- Linear Superposition Integral
  - Shows Linear Superposition Integral is Solution
  - Shows Steady State Solution
- Significance
  - Superposition Integral Provides Useful Approximations
  - Steady State Covariance Matrix Used in Prediction Modeling

Solution to Linear Variance Equation

- The Equation

$$P(t) = \Phi(t,t_0)P(t_0)\Phi(t,t_0)^T + \int_{t_0}^{t} \Phi(t,\tau)G(\tau)Q(\tau)G^T(\tau)\Phi(t,\tau)\cdot d\tau$$

- Key Derivative

$$\frac{d}{dt}\left(\Phi(t,t_0)\cdot A\cdot\Phi(t,t_0)^T\right)$$

$$= F\cdot\Phi(t,t_0)\cdot A\cdot\Phi(t,t_0)^T$$

$$+\Phi(t,t_0)\cdot A\cdot\Phi(t,t_0)^T\cdot F^T$$

- Use Leibniz's Rule and Chain Rule

$$\frac{d}{dt}\int_{a(t)}^{b(t)} f(x,t)dx = \dot{b}f(b,t) - \dot{a}f(a,t) + \int_{a(t)}^{b(t)}\left(\frac{\partial}{\partial t}f(x,t)\right)dx$$

Steady State Covariance

- Linear Variance Equation
  - Use Special Case

$$\dot{P} = F \cdot P + P \cdot F^T + Q$$

  - Set Time Derivative to Zero
  - Use Closed Form

$$\Phi(t, t_0) = \exp(F \cdot (t - t_0))$$

  - Take Limit
- What are Conditions on Existence?

Radar Trackers and Applications for SAADS

October 26, 1999

Topic 9: Vector Rotation with Quaternions

Sensor Systems Engineering for the 21st Century

Coordinate Systems Definition

- Inertial Coordinate System Examples
  - – Local North, East, Down
  - – Either
    - » Rotates With Earth, or
    - » Non-Rotating Earth Model
  - – Earth Centered Inertial Coordinates (ECIC)
- Airframe
  - – Nose, Right Wing, Down
  - – Rotates With Airframe

From Raytheon ATEP SYS12525 for SAADS Day 2, October 26, 1999

Coordinate Axes

- Inertial Coordinates -- Axes Are

$$\underline{i} = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}, \quad \underline{j} = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}, \quad \underline{k} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$$

- Airframe Coordinates
  - Axes are Rows of A Matrix
  - A Matrix is [$\underline{i}$, $\underline{j}$, $\underline{k}$] in Airframe Coordinates

Coordinate Changes

- Reference -- Quaternion Report, pp. 24-32
- The Euler Sequence of Angles From Inertial to Non-Inertial Coordinates
  - Yaw, or Azimuth First
  - Pitch, or Elevation Angle Next
  - Roll, or Rotation About X Axis Last
- Euler Sequence of Angles from Non-Inertial to Inertial Coordinates
  - First Roll, Then Pitch, and Yaw Last
- Operation $\underline{r}'=A\cdot\underline{x}$ or $q\cdot\underline{r}\cdot q^*$ Rotates from Inertial to Airframe Coordinates

Rotation Matrix

$$A = A_R \cdot A_P \cdot A_Y$$

$$A_R = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\phi & \sin\phi \\ 0 & -\sin\phi & \cos\theta \end{bmatrix} \text{(roll)}$$

$$A_P = \begin{bmatrix} \cos\gamma & 0 & -\sin\gamma \\ 0 & 1 & 0 \\ \sin\gamma & 0 & \cos\gamma \end{bmatrix} \text{(pitch)}$$

$$A_y = \begin{bmatrix} \cos\psi & \sin\psi & 0 \\ -\sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \text{(yaw)}$$

Rotation Quaternion

$$q = q_R \cdot q_P \cdot q_Y$$

$$q_R = \cos\left(\frac{\phi}{2}\right) + \underline{i} \cdot \sin\left(\frac{\phi}{2}\right), \;\; \underline{i} = \text{ inertial x axis}$$

$$q_P = \cos\left(\frac{\gamma}{2}\right) + \underline{j} \cdot \sin\left(\frac{\gamma}{2}\right), \;\; \underline{j} = \text{ inertial y axis}$$

$$q_P = \cos\left(\frac{\psi}{2}\right) + \underline{k} \cdot \sin\left(\frac{\psi}{2}\right), \;\; \underline{k} = \text{ inertial z axis}$$

Synthesizing Quaternions: FORTRAN Code

```
subroutine qsynth(roll,pitch,yaw,q) !Quaternion synthesis
c Inputs:
c roll  Roll, radians
c pitch Pitch, radians
c yaw   Yaw, radians
c Output:
c q(4)  Rotation quaternion, inertial to airframe coordinates
c Coordinate systems: NWU inertial, nose-left-up airframe
c Algorithm: q=qr*qp*qy,
c   qr=cos(roll/2)+i*sin(roll/2)
c   qp=cos(pitch/2)+j*sin(roll/2)
c   qy=cos(yaw/2)+k*sin(yaw/2)
      implicit none
      double precision roll,pitch,yaw,q(4),qr(2),qp(2),qy(2),qi(4)
 ...
```

Synthesizing Quaternions: FORTRAN Code (2 of 2)

```
...
     qr(1)=cos(.5d0*roll) !Store quaternion factors
     qr(2)=sin(.5d0*roll) !Use two locations for factors
     qp(1)=cos(.5d0*pitch)
     qp(2)=sin(.5d0*pitch)
     qy(1)=cos(.5d0*yaw)
     qy(2)=sin(.5d0*yaw)
     qi(1)=qr(1)*qp(1) !Store qi=qr*qp
     qi(2)=qp(1)*qr(2)
     qi(3)=qr(1)*qp(2)
     qi(4)=qr(2)*qp(2)
     q(1)=qi(1)*qy(1)-qi(4)*qy(2) !Compute output q=qi*qy
     q(2)=qi(2)*qy(1)+qi(3)*qy(2)
     q(3)=qi(3)*qy(1)-qi(2)*qy(2)
     q(4)=qi(4)*qy(1)+qi(1)*qy(2)
     return
     end
```

Rotating Vectors With Quaternions

- Using Finished Equation

$$q = q_0 + \underline{v}$$

$$\underline{r}' = q \cdot \underline{r} \cdot q^* = \left(q_0^2 - \left|\underline{v}^2\right|\right) \cdot \underline{r} + 2 \cdot q_0 \cdot \underline{v} \times \underline{r} + 2 \cdot \left(\underline{v}^T \cdot \underline{r}\right) \cdot \underline{v}$$

- Using Intermediate Products

$$\begin{aligned}
\underline{r}' &= \left(q_0 + \underline{v}\right) \cdot \underline{r} \cdot \left(q_0 - \underline{v}\right) \\
&= \left(q_0 + \underline{v}\right) \cdot \left(\underline{v}^T \cdot \underline{r} + q_0 \cdot \underline{r} + \underline{v} \times \underline{r}\right) \\
&= q_0^2 \cdot \underline{r} + 2 \cdot q_0 \cdot \underline{v} \times \underline{r} + \left(\underline{v}^T \cdot \underline{r}\right) \cdot \underline{v} + \underline{v} \times \left(\underline{v} \times \underline{r}\right)
\end{aligned}$$

Rotation with Quaternions: FORTRAN Code

```
subroutine quarot(q,v,vr) !Quaternion rotation of a vector
c Inputs:
c q(4)  Rotation quaternion
c v     Vector to be rotated
c Output: vr=q*v*conj(q)
c Algorithm: q=q0+w, w is vector part
c vr=(q0^2-(wT*w))*v + 2*q0*(w X v) + 2*(wT*v)*w
      implicit none
      integer i
      double precision q(4),v(3),vr(3),vtemp1(3),
     + temp1,dot,temp2,temp3
      call crossp(q(2),v,vtemp1) !Begin with cross product
      temp1=q(1)**2 !Find q0**2 minus squared length of w, q0**2-(wT*w)
      do i=2,4
        temp1=temp1-q(i)**2
        enddo
      temp2=2.d0*q(1)
      temp3=2.d0*dot(v,q(2))
      do i=1,3 !Combine temporary vectors as per algorithm
        vr(i)=temp1*v(i)+temp2*vtemp1(i)+temp3*q(i+1)
        enddo
      return
      end
```

ATEP SYS12525

Radar Trackers and Applications for SAADS

October 26, 1999

Topic 10: The Alpha-Beta Filter

Sensor Systems Engineering for the 21st Century

Optimizing the Alpha Beta Filter

- Notation
  - States
    - » True target state at time t_i $\underline{x}_i$
    - » Estimated target state at time t_i $\hat{\underline{x}}_i$
    - » Extrapolated target state at time t_i from data available up to time t_{i-1} $\tilde{\underline{x}}_i$
  - Plant Noise
    - » Perturbation of $\underline{x}$ from t_{i-1} to t_i $\underline{u}_i$
    - » Covariance of $\underline{u}_i$ Q

Notation (Continued)

- Measurements
  - Vector of measurements available at time t_i $\underline{y}_i$
  - Random noise in the measurements $\underline{v}_i$
  - Covariance of $\underline{v}_i$ R_i
- Mapping
  - Target states to the measurements H_i
  - Tracker Gain K_i
  - State Transition Matrix from t_{i-1} to t_i Φ_i

More Notation

- State Covariances
  - Covariance of $\underline{\tilde{x}}_i - \underline{x}_i$ $\quad \tilde{P}_i$
  - Covariance of $\underline{\hat{x}}_i - \underline{x}_i$ $\quad P_i$
- Target Motion with Noise Perturbation

$$\underline{x}_i = \Phi_i \cdot \underline{x}_{i-1} + \underline{u}_i$$

- Measurement Model

$$\underline{y}_i = H_i \cdot \underline{x}_i + \underline{v}_i$$

Still More Notation

- Extrapolation of State Vector Estimate from t_{i-1} to t_i

$$\underline{\tilde{x}}_i = \Phi_i \cdot \underline{\hat{x}}_{i-1}$$

- Update of Estimate

$$\begin{aligned}\underline{\hat{x}}_i &= \underline{\tilde{x}}_i + K_i \cdot \left(\underline{y}_i - H_i \cdot \underline{\tilde{x}}_i \right) \\ &= \left(I - K_i \cdot H_i \right) \cdot \Phi_i \cdot \underline{\hat{x}}_{i-1} + K \cdot \underline{y}_i\end{aligned}$$

Covariance of Estimate

- Extrapolated States

$$\tilde{P}_i = \Phi_i \cdot P_{i-1} \cdot \Phi_i^T + Q_i$$

- Updated State Vector

$$P_i = \left(I - K_i \cdot H_i\right) \cdot \tilde{P}_i \cdot \left(I - K_i \cdot H_i\right)^T + K_i \cdot R_i \cdot K_i^T$$

How to Minimize of P_i

- Define a Cost Function
  - Trace of P_i is
    - » Scalar Cost Function
    - » Gradients are Simple to Compute
    - » Visualization
      - Box Containing Localization Ellipsoid
      - Distance from Center to Corner
  - Determinant of P_i
    - » Square of Volume of Box
    - » Gives Same Answer
- Show Result Using the Trace

Optimum Gain

- Gradient of Trace Follows Gelb, page 23

$$\frac{\partial \operatorname{tr}(P_i)}{\partial K} = -2 \cdot (I - K_i \cdot H_i) \cdot \tilde{P}_i \cdot H_i^T + 2 \cdot K_i \cdot R_i = 0$$

- Solving for K_i

$$K_i \cdot \left(H_i \cdot \tilde{P}_i \cdot H_i^T + R_i \right) = \tilde{P}_i \cdot H_i^T$$

- Gain for Minimum Variance

$$K_i = \tilde{P}_i \cdot H_i^T \cdot \left(H_i \cdot \tilde{P}_i \cdot H_i^T + R_i \right)^{-1}$$

Radar Trackers and Applications for SAADS

October 26, 1999

Topic 11: The Kalman Filter

Sensor Systems Engineering for the 21st Century

Using K_i to Find P_i

- The General Equation for P_i

$$
\begin{aligned}
P &= \left(I - K \cdot H\right) \cdot \tilde{P} \cdot \left(I - K \cdot H\right)^T + K \cdot R \cdot K^T \\
&= \left(I - K \cdot H\right) \cdot \tilde{P} - \tilde{P} \cdot H^T \cdot K^T \\
&\quad + K \cdot H \cdot \tilde{P} \cdot H^T \cdot K^T + K \cdot R \cdot K^T \\
&= \left(I - K \cdot H\right) \cdot \tilde{P} - \tilde{P} \cdot H^T \cdot K^T \\
&\quad + K \cdot \left(H \cdot \tilde{P} \cdot H^T + R\right) \cdot K^T
\end{aligned}
$$

- Using the Optimum K_i

$$P = \left(I - K \cdot H\right) \cdot \tilde{P}$$

Developing Alternative Forms for P

- Substituting the Optimal K in the Last Equation

$$P = \tilde{P} - \tilde{P} \cdot H^T \left(H \cdot \tilde{P} \cdot H^T + R \right)^{-1} \cdot H \cdot \tilde{P}$$

- Matrix Inversion Lemma

$$\left(A + B \cdot C^{-1} \cdot D \right)^{-1}$$

$$= A^{-1} - A^{-1} \cdot B \cdot \left(C + D \cdot A^{-1} \cdot B \right)^{-1} \cdot D \cdot A^{-1}$$

Alternative Form for P

- Form for Inverse of P

$$P^{-1} = \tilde{P}^{-1} + H^T \cdot R^{-1} \cdot H$$

- Alternative Form for Filter Gain (Gelb, page 112)

$$K = P \cdot H^T \cdot R^{-1}$$

- Interesting Footnote; if used, usually causes numerical instability in the recursion!

$$P \cdot \tilde{P}^{-1} = I - K \cdot H$$

Summary of Simple Kalman Filter

- Models
  - State Vector

$$\underline{x} = \Phi \cdot \underline{x}(-) + \underline{w},\ \ \mathrm{Cov}(\underline{w}) = Q$$

  - Measurements

$$\underline{y} = H\underline{x} + \underline{v},\ \ \mathrm{Cov}(\underline{v}) = R$$

- Initialization

$$\underline{x}(-) = \underline{x}(0),\ \ P(-) = P(0)$$

Implementation of Kalman Filter

- Extrapolation to Current Time
  - States

$$\underline{\tilde{x}} = \Phi \cdot \underline{\hat{x}}(-)$$

  - Covariance

$$\tilde{P} = \Phi \cdot P(-) \cdot \Phi^T + Q$$

- Kalman Gain

$$K = \tilde{P} \cdot H^T \cdot \left(H \cdot \tilde{P} \cdot H^T + R \right)$$

Simple Kalman Update

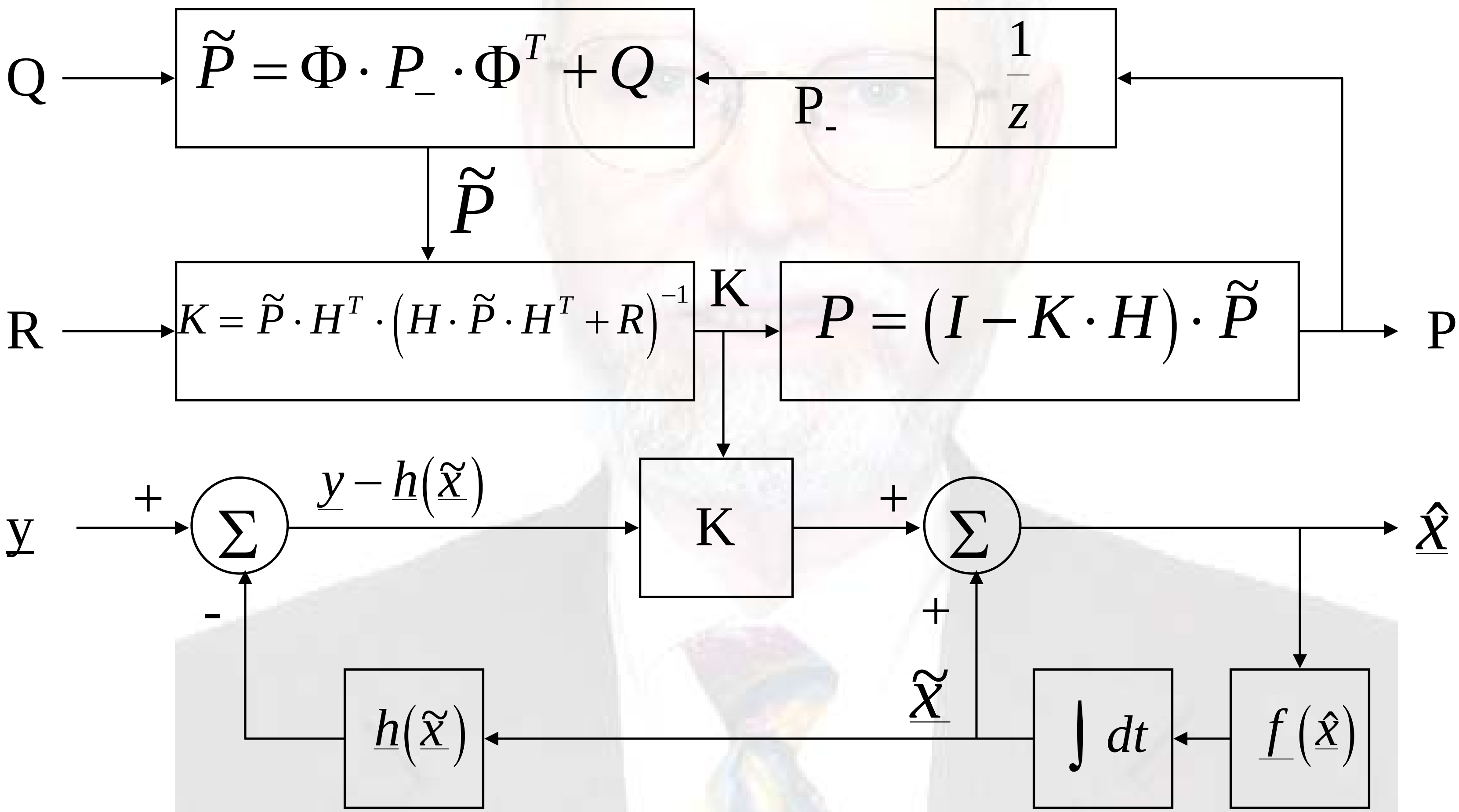
- State Vector

$$\underline{\hat{x}} = \underline{\tilde{x}} + K \cdot \left(\underline{y} - H \cdot \underline{\tilde{x}} \right)$$

- Covariance Matrix

$$\begin{aligned} P &= \left(I - K \cdot H \right) \cdot \tilde{P} \\ &= \left(I - K \cdot H \right) \cdot \tilde{P} \cdot \left(I - K \cdot H \right)^T + K \cdot R \cdot K^T \\ &= \left[\tilde{P}^{-1} + H^T \cdot R^{-1} \cdot H \right]^{-1} \end{aligned}$$

Kalman Filter Data Flow



Steady State Covariance

- Linear Variance Equation
  - Use Special Case

$$\dot{P} = F \cdot P + P \cdot F^T + Q$$

  - Set Time Derivative to Zero
  - Use Closed Form

$$\Phi(t, t_0) = \exp\left(F \cdot (t - t_0)\right)$$

  - Take Limit
- What are Conditions on Existence?

ATEP SYS12525

Radar Trackers and Applications for SAADS

October 26, 1999

Topic 12: Tracker Architectures

Sensor Systems Engineering for the 21st Century

High Level Tracker

- Only Needed When
  - Mid Level Tracker Can't Meet Requirements
  - Higher Accuracy or Longer Integration Time Needed than General Purpose Kalman Filter can Provide
- Design for Single Requirement
  - Accuracy vs. Integration Time
  - Often a Batch

Types of Tracker Estimators

- Ad Hoc
  - Simple Averaging
  - Linear Regression
  - Least Squares
  - Alpha-Beta
  - Weighted Last Squares
- All Have Their Legitimate Application
  - Minimum Complexity to Meet Requirements

Estimator Types (Continued)

- Kalman Filter
- Other Statistically Based Estimators
  – Bayesian Mean
  – Maximum Likelihood
  – Hypotheses Testing
  – Median, Centiles
  – Kolmogorov-Smirnov
  – Other

Best Candidates for High Level Tracker Batch Estimators

- Method of Maximum Likelihood
  - Very Simple, General, Powerful
  - Sufficient, Consistent
  - Always Asymptotically Unbiased, Efficient
- Bayesian Weighted Least Squares
  - Uses *A Priori* Information
  - Similar Otherwise to Maximum Likelihood
- Maximum A Priori

Lessons Learned

- Mapping
  - Estimate is Same
    - » Variance
    - » Standard Deviation
  - Maximum Likelihood Always Maps (Why?)
- The Variance is Biased
  - Estimator is Nonlinear
  - Asymptotically Unbiased, Efficient
  - Bias is Correctable

Summary

- Estimation Theory Examples
  - Classical Mean and Variance with MLE
  - Bivariate Regression
  - Multivariate Regression
- Singer Process Noise Model
- Extended Kalman Filters
  - Differences
  - Approximations

Summary (Concluded)

- Adaptive Kalman Filter Derivation
  - Augmented State Vector
  - Example
- Tuning Kalman Filters
  - Simplification of Derivation
  - File of the Week
- Batch Estimators
  - Maximum Likelihood
  - Others

Tracker Architecture

- Defined by Functional Flowdown
  - Tracker Manager
    - » Supports System Requirements
    - » Highest Level Tracker CSC
  - Other Tracker Functions
    - » Organized by Tracker Manager
    - » Lower Level CSC's
- Application and Requirements Driven

Two Tracker Architectures

- Conventional Tracker Architecture When
  - High Update Rate
  - Low Target Density in Measurement Space
- Multiple Hypothesis Tracker When
  - High False Alarm Rate
  - Low Update Rate
  - High Target Density
- Conventional Trackers Today, MHT to Follow

The Tracker Manager

- Inputs
  - Signal Processor Detection Data
  - INS Data
  - Operator Inputs
  - Command and Control Inputs and Data
- Outputs
  - Track Files
  - Display and Control Support
  - Command and Control Support

Sequence of Operations

- Process Detection Data
  - Reformat Information for Tracker Use
  - Compute Variances of Measurements
- Associate Detections with Track Files
- Perform Track File Maintenance
  - Update
  - Maintain Track Quality Score Functions
  - Initiate, Drop Tracks
  - Bifurcate, Merge Tracks

Differences Between Trackers

- Use of Detection Data
  - Only Once per Detection
  - Or, As Many Times as Association Indicates
- Bifurcation of Track Files
  - Split a Track File into Two Track Files
  - When You Have Two Updates in One Dwell
  - Or, When an Association is Specious
- Merging Not Always Necessary

Merging Tracks

- Multiple Track File Updates from One Detection
  - Useful in Tracking Aircraft in Close Formation
  - Dual Tracks of a Single Target are Possible
- How to Do It
  - Treat Both State Vectors and Covariances as “Measurements”
  - Estimate a Single State Vector

Bifurcating Tracks

- Nearly Always Necessary
  - – Aircraft in Close Formation
    - » May Become Resolved by Radar
    - » Aircraft Flight Paths May Diverge
  - – Aircraft may Fire Missile
  - – Range Gate Pulloff (RGPO) Jamming
- How to Do It
  - – Detections “Walk Away” from Track
  - – Two Detections Want to Associate

Track Quality Score Functions

- Used to Decide When to Drop Tracks
- Simple
  - Time in Track
  - Number of Consecutive “Misses”
- Statistically Based
  - Localization Accuracy from Covariance Matrix
  - Likelihood Ratio Based on Target Model
    - » Number Crunching Available from Association Process
    - » Core Score Function for Multiple Hypothesis

Other Tracker Functions

- Generating Alerts
  - Missile Firing Warning
  - ECCM Operator Inputs
- Data Fusion
  - Measurements from
    - More than One Sensor on the Platform
    - Other Platforms
  - Association and Update from Multiple Sensors

ATEP SYS12525

Radar Trackers and Applications for SAADS

October 26, 1999

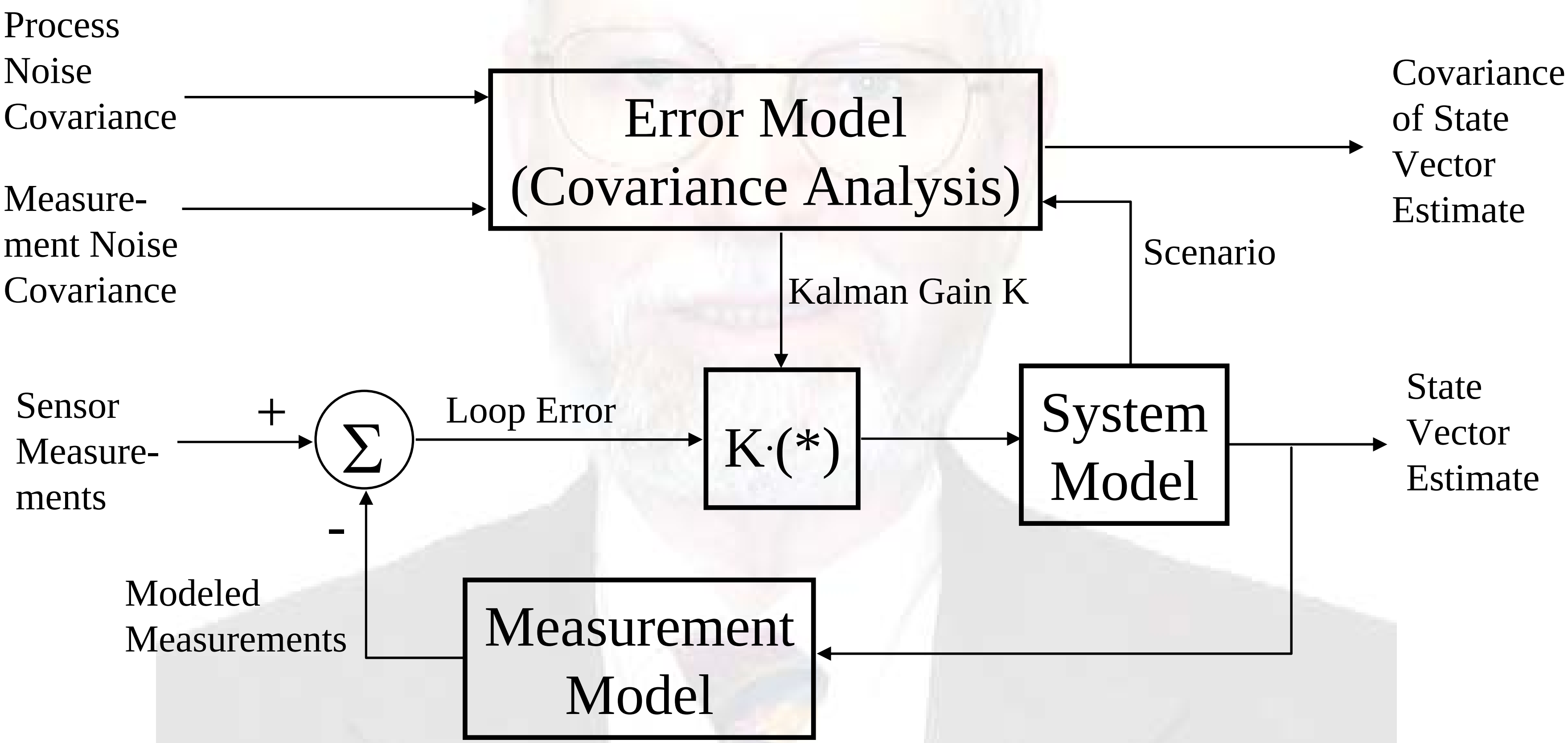
Topic 13: Kalman Tracking Filters

Sensor Systems Engineering for the 21st Century

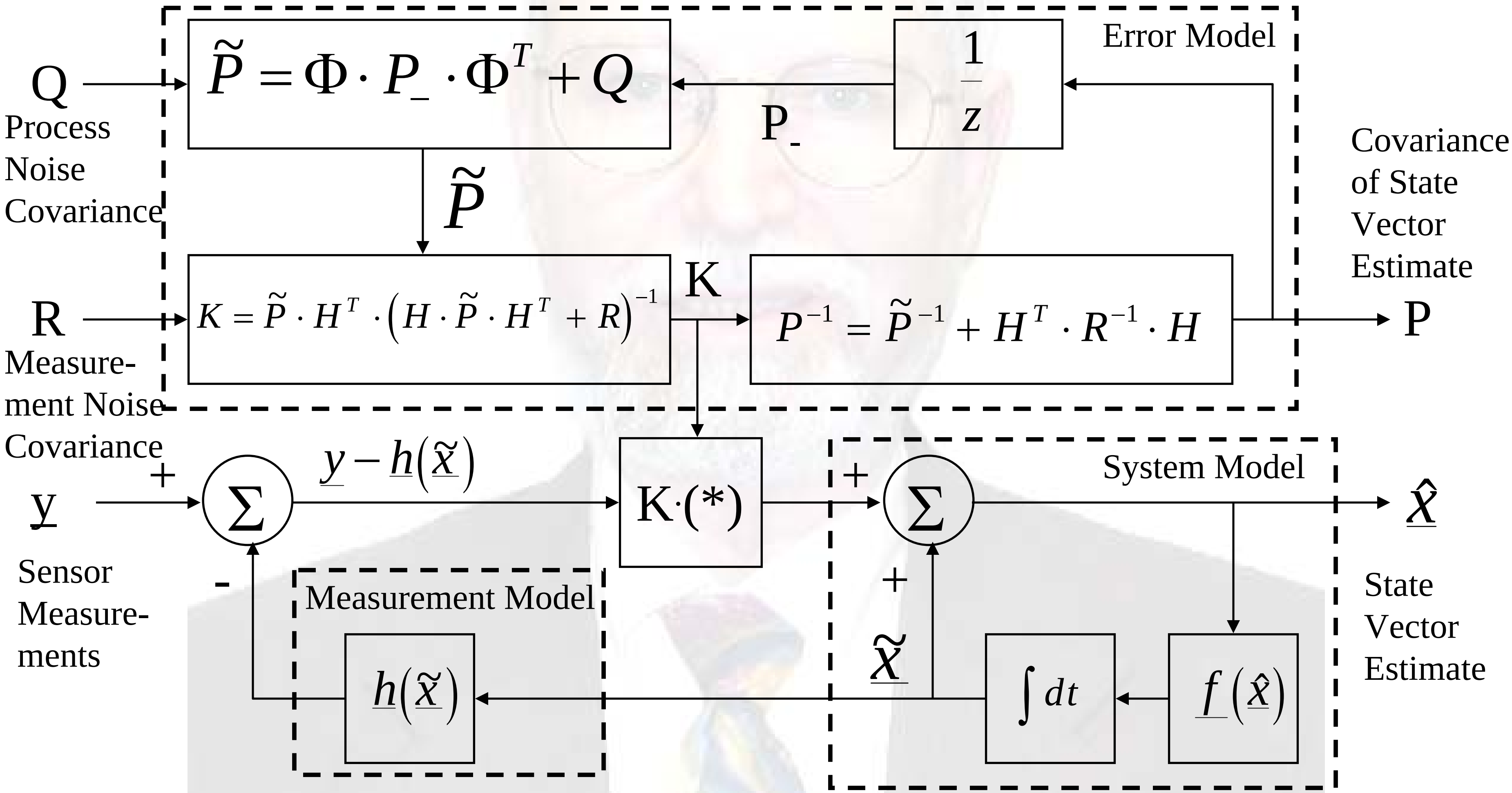
Tracking Filters

- Alpha Beta
  - Simple Two State
  - Already Studied
- Alpha Beta Gamma
  - Three State Extension
  - Still Limited to Range Measurements Only
- Kalman Filters

Kalman Filter Concept



Kalman Filter Data Flow



Use of Kalman Filters in Trackers

- Function: Track File Maintenance
- Begin With Track File Requirements
  - Support Association of Detections to Track Files
  - Store State of Target Track
    - » First Detection or Initialized
    - » Other Track Quality Indicators
    - » Displays and Controls Information
  - Support Range and Doppler Resolve

Track Filter Architecture

- Support Association
  - Use Simple Two-State
    - » Range (Ambiguous)
    - » Doppler (Ambiguous)
    - » Azimuth
    - » Elevation
  - Use Ambiguous or Unambiguous Range and Doppler for Association
- Support Estimation Functions Separately

Tiered Track Filters

- Lowest Level
  - – Support Association
  - – Support Range and Doppler Resolve
- Mid Level
  - – Support Real Time Displays and Controls
  - – Support Command and Control
- High Level
  - – Support Fire Control
  - – Support Platform Survivability

The Track Filters

- Low Level
  - Very Simple
  - Adaptive Kalman Filter
- Mid Level
  - Requirements Driven Design
  - Adaptive Square Root Kalman Filter
- High Level
  - Batch Estimator for Cramer-Rao Bound Performance
  - Traceable to Method of Maximum Likelihood

Square Root Filters

- Three Principal Types
  - Square Root Covariance - Potter
  - UDUT Factorization - Agee, Turner
  - Square Root Information - Agee, Turner, Carlson, Bierman
- All are Algebraically Equivalent to EKF
  - Extrapolation, Update Algorithms Differ
  - Track File Storage of Covariance is Different

Recommendations

- Use Special Adaptive Two or Three State
- Use SRIF or UDUT for Four or More States
- UDUT
  - Standard Kalman Format
  - Computation Requirements Low
- SRIF
  - High Performance even with Observability Problems
  - Computation Requirements Low

The Special Adaptive Filters

- Several Varieties
  - Two State, Upgrade of Alpha Beta Tracker
  - Two State for Chirped Pulses
  - Three State Upgrade of Alpha Beta Gamma, Chirped Pulses
  - Etc.
- Divide, Not Subtract
  - In Kalman Gain
  - In Covariance Update
- Seen Only In
  - This Course
  - Some East Coast Raytheon Trackers Since 1979
- Known as “Snake Oil Trackers”

Two State Snake Oil Tracker

- Kalman Gain (Range Measurement Only)

$$K = \frac{1}{\tilde{p}_{11} + R} \cdot \begin{bmatrix} \tilde{p}_{11} \\ \tilde{p}_{12} \end{bmatrix}$$

$$\tilde{P} = \begin{bmatrix} p_{11} + 2Tp_{12} + T^2 p_{22} + q_{11} & p_{12} + Tp_{22} \\ p_{12} + Tp_{22} & p_{22} + q_{22} \end{bmatrix}$$

- Covariance Update

$$P = \frac{1}{1 + \frac{\tilde{p}_{11}}{R}} \cdot \begin{bmatrix} \tilde{p}_{11} & \tilde{p}_{12} \\ \tilde{p}_{12} & \tilde{p}_{22} + \frac{D}{R} \end{bmatrix}, \quad D = \tilde{p}_{11} \cdot \tilde{p}_{22} - \tilde{p}_{12}^2$$

Initialization From Data

- From First Hit

$$\underline{x}(t_0) = \begin{bmatrix} y_0 \\ 0 \end{bmatrix}, \quad P = \begin{bmatrix} R_0 & 0 \\ 0 & < \text{Large} > \end{bmatrix}$$

- From Second Hit (Full Initialization using MLE)

$$\underline{x}(t_1) = \begin{bmatrix} y_1 \\ \dfrac{y_1 - y_0}{T} \end{bmatrix}, \quad P(t_1) = \begin{bmatrix} R_1 & \dfrac{R_1}{T} \\ \dfrac{R_1}{T} & \dfrac{R_0 + R_1}{T^2} \end{bmatrix}$$

Adaptive Process Noise

- Adaptation by Estimation of Process Noise Matrix Q as an Unknown
  - Gelb, pp. 316-320
  - Two papers by R.K. Mehra in IEEE AES in 1970 and 1971
- Modifications for Simplicity and Practicality
  - Use Assumed Form for Process Noise Covariance Matrix
  - Simplify Equations
  - Apply Estimate to Current Update

The Innovations Sequence

- The Kalman Filter Loop Error Data

$$\underline{e} = \underline{y} - \underline{h}(\underline{x}),\ \ \mathrm{Cov}\{\underline{e}\} = E = H \cdot \tilde{P} \cdot H^T + R$$

$$g = \underline{e}^T \cdot E^{-1} \cdot \underline{e} \quad \text{is chi-square}$$

- Important Properties
  - Uncorrelated Update to Update (Innovations Sequence)
  - Sensitive to Errors in System Model
  - Provides Observability of Q and R

Application of Adaptive Process Noise

- Theoretical Approach
  - Use Cross Correlations of the Innovations Sequence Between Updates
  - Estimate Q or R, or Both
- Heuristic Approaches
  - Assume Form for Q with Magnitude Unknown
  - Use g to Estimate Magnitude
- Simplest Approach: Use $|\underline{e}|^2$ Instead of g

Assumed Forms for Q

- Simplest Basic Format for Q

$$Q = \begin{bmatrix} q_{11} & 0 \\ 0 & q_{22} \end{bmatrix} \cdot \left| \underline{e} \right|^2$$

- Options
  - Highest Accuracy: $q_{11} = 0$
  - Fastest Maneuver Detection: $q_{11} > 0$
- Tuning
  - Select q_{22} to Scale Process Noise
  - Minimize q_{11}

Other Forms of Adaptive Kalman Filters

- Estimate Plant Noise Level
  - Use Form for Plant Noise:

$$Q = \left(\left(\underline{y} - \underline{h}(\tilde{x}) \right)^T \cdot W \cdot \left(\underline{y} - \underline{h}(\tilde{x}) \right) \right) \cdot Q_0$$

  - Tuning Matrices
    - » Diagonal Matrix W is Weights for Measurements
    - » Diagonal Q_0 is Magnitude and Form Information for Q
- Adaptive Bandwidth Feature
  - Tunable for High Performance with Steady Targets
  - Opens Up when Target Maneuvers

Two State for Chirp

- Measurement Sensitivity Matrix

$$H = \begin{bmatrix} c & 1 \end{bmatrix}$$

  – Parameter c is Chirp Rate Over Center Frequency

- Kalman Gain

$$K = \frac{1}{q + R} \cdot \begin{bmatrix} \tilde{p}_{11} \cdot c + \tilde{p}_{12} \\ \tilde{p}_{12} \cdot c + \tilde{p}_{22} \end{bmatrix},$$

$$q = \tilde{p}_{11} \cdot c^2 + 2 \cdot \tilde{p}_{12} \cdot c + \tilde{p}_{22}$$

Two State Chirp (Continued)

- Covariance Update

$$P = \frac{1}{1+\frac{q}{R}} \cdot \begin{bmatrix} \tilde{p}_{11} + \frac{D}{R} & \tilde{p}_{12} - \frac{D}{R} \cdot c \\ \tilde{p}_{12} - \frac{D}{R} \cdot c & \tilde{p}_{22} + \frac{D}{R} \cdot c^2 \end{bmatrix}$$

Three State Snake Oil Tracker

- Upgrade of Alpha Beta Gamma Tracker
- Measurement Sensitivity Matrix

$$H = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix}$$

- Kalman Gain

$$K = \frac{1}{\tilde{p}_{11} + R} \cdot \begin{bmatrix} \tilde{p}_{11} \\ \tilde{p}_{12} \\ \tilde{p}_{13} \end{bmatrix}$$

Three State Covariance Update

$$P = \frac{1}{1+\frac{\tilde{p}_{11}}{R}} \cdot \begin{bmatrix} \tilde{p}_{11} & \tilde{p}_{12} & \tilde{p}_{13} \\ \tilde{p}_{12} & \tilde{p}_{22} + \frac{a_{33}}{R} & \tilde{p}_{23} - \frac{a_{23}}{R} \\ \tilde{p}_{13} & \tilde{p}_{23} - \frac{a_{23}}{R} & \tilde{p}_{33} + \frac{a_{22}}{R} \end{bmatrix},$$

a_{ij} are elements of $A = \left|\tilde{P}\right| \cdot \tilde{P}^{-1}$

Three State for Chirp

- Measurement Sensitivity Matrix

$$H = \begin{bmatrix} c & 1 & 0 \end{bmatrix}$$

- Kalman Gain

$$K = \frac{1}{q + R} \cdot \begin{bmatrix} \tilde{p}_{11} \cdot c + \tilde{p}_{12} \\ \tilde{p}_{12} \cdot c + \tilde{p}_{22} \\ \tilde{p}_{13} \cdot c + \tilde{p}_{23} \end{bmatrix},$$

$$q = H \cdot \tilde{P} \cdot H^T = \tilde{p}_{11} \cdot c^2 + 2 \cdot \tilde{p}_{12} \cdot c + \tilde{p}_{22}$$

Three State Chirp (Continued)

- Covariance Update Algorithm Based On

$$P = \tilde{P} \cdot \begin{bmatrix} a_{11} + \frac{a_{33}}{R} \cdot c^2 & a_{12} + \frac{a_{33}}{R} \cdot c & a_{13} \\ a_{12} + \frac{a_{33}}{R} \cdot c & a_{22} + \frac{a_{33}}{R} & a_{23} \\ a_{13} & a_{23} & a_{33} \end{bmatrix}^{-1}$$

Three State Chirp (Continued)

- Matrix Inversion Lemma

$$P = \left(\tilde{P}^{-1} + H^T \cdot R^{-1} \cdot H\right)^{-1}$$

$$= \tilde{P} - \frac{1}{H \cdot \tilde{P} \cdot H^T + R} \cdot \left(\tilde{P} \cdot H^T\right) \cdot \left(\tilde{P} \cdot H^T\right)^T$$

- Computational From Is ...

$$P = \frac{1}{1+\frac{q}{R}} \cdot \left(\tilde{P} + \frac{a_{33}}{R} \cdot \begin{bmatrix} 1 & -c & 0 \\ -c & c^2 & 0 \\ 0 & 0 & 0 \end{bmatrix} + \frac{a_{23} \cdot c - a_{13}}{R} \cdot \begin{bmatrix} 0 & 0 & 1 \\ 0 & 0 & -c \\ 1 & -c & 0 \end{bmatrix} + \frac{a_{22} \cdot c^2 - 2 \cdot a_{12} \cdot c + a_{11}}{R} \cdot \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \right)$$

Derivation of
The Adaptive Kalman Filter

- System Model

$$\dot{\underline{x}} = \underline{f}\left(\underline{x}\right) + G \cdot \underline{w}, \quad \underline{x}\left(t_0\right) = \underline{x}_0$$

- Measurement Model

$$\underline{y} = \underline{h}\left(\underline{x}\right) + \underline{v}$$

- System Transition Matrix

$$\Phi\left(t, t_0\right) = \frac{\partial \underline{x}\left(t\right)}{\partial \underline{x}_0}$$

Extended Kalman Filter Notation Revisited (Concluded)

- Measurement Sensitivity Matrix

$$H = \frac{\partial \underline{h}(\underline{x})}{\partial \underline{x}}, \quad \underline{x} = \underline{\tilde{x}}$$

- Other
  - Kalman Gain Unchanged
  - Covariance Extrapolation Unchanged
  - Covariance Update Unchanged
  - All are Approximations

Adaptive Kalman Filter Reformulation

- Augment State Vector
  - Plant Noise Scaling Parameter is in States
  - Add New Measurement
- Augmented State Model

$$\underline{x}_a = \begin{bmatrix} \underline{x}_1 \\ x_2 \\ \underline{x}_3 \end{bmatrix}, \begin{bmatrix} \dot{\underline{x}}_1 \\ \dot{x}_2 \\ \dot{\underline{x}}_3 \end{bmatrix} = \begin{bmatrix} \underline{f}\left(\underline{x}_1\right) + \sqrt{x_2} \cdot \underline{x}_3 \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ \varepsilon \cdot w_2 \\ \underline{w}_3 \end{bmatrix}$$

Augmented Measurements

- Measurement Model

$$\underline{y}_a = \begin{bmatrix} \underline{h}_1(\underline{x}_1) \\ \underline{e}_1 \cdot \underline{e}_1^T - Trace\{E\} \end{bmatrix} + \begin{bmatrix} R_1 & 0 \\ 0 & R_2 \end{bmatrix}$$

$$\underline{e}_1 = \underline{y}_1 - \underline{h}_1(\underline{\tilde{x}}_1), \;\; E = H_1 \cdot \tilde{P}_{11} \cdot H_1^T + R_1$$

- Arbitrary New Measurement

$$y_2 = \underline{e}_1^T \cdot \underline{e}_1 - Trace\left\{H_1 \cdot \tilde{P}_{11} \cdot H_1^T + R_1\right\}$$

$$\tilde{P}_{11} = \Phi_{11} \cdot P_{11-} \cdot \Phi_{11}^T + x_2 \cdot Q_0$$

The New Measurement

- Measurement Sensitivity Matrix

$$H_2 = -Trace\left\{H_1 \cdot Q_0 \cdot H_1^T\right\}$$

- Variance of Measurement

$$R_2 = 2 \cdot Trace\left\{E^2\right\} = \sum_{i,j} e_{ij}^2$$

- See Handout Write-up for Derivation

The New Measurement

- Covariance Extrapolation (Scalar)

$$\tilde{P}_{22} = P_{22-} + \varepsilon^2$$

- Update (All Quantities are Scalars)

$$K_2 = \tilde{P}_{22} \cdot H_2^T \cdot \left(H_2 \cdot \tilde{P}_{11} \cdot H_2^T + R_2 \right)^{-1}$$

$$\tilde{x}_2 = \hat{x}_{2-}, \;\; \hat{x}_2 = \tilde{x}_2 + K_2 \cdot \left(\underline{e}_1^T \cdot \underline{e}_1 - Trace\{E\} \right)$$

$$P_{22} = \left(1 - K_2 \cdot H_2\right) \cdot \tilde{P}_{22} = \left(\tilde{P}_{22}^{-1} + H_2 \cdot R_2^{-1} \cdot H_2 \right)^{-1}$$

Summary

- Adaptive Kalman Filter is an EKF
- Formulation is Not Unique
  - Second Measurement is Arbitrary
  - Memo from 1981 is More General than Example
  - Mehra’s Paper Estimates Entire Q Matrix
- It’s a Tool for Adding Robustness

Tuning Kalman Filters

- Tips
  - Covariance Propagates from Velocity States to Position States through Extrapolation Equation
  - Position Covariance Does Not Propagate
  - Plant Noise: Less is Better
  - Don’t Solve Problems with Plant Noise
- General Principles
  - Use Two Tiers of Trackers
  - The requirements of the top and bottom tier are different
  - Tune them Separately

General Principles

- Tune Low Level Trackers for Robustness
  - Uncoupled Two State Trackers of Measurements
  - Tune Adaptive Plant Noise for Unexpected Target Behavior
- Tune Mid Level Trackers for Performance
  - Use Only as Many States as You Can Observe Well
  - Trade Off Robustness for Performance

Use Monte Carlo Simulations to Prove Tracker Designs

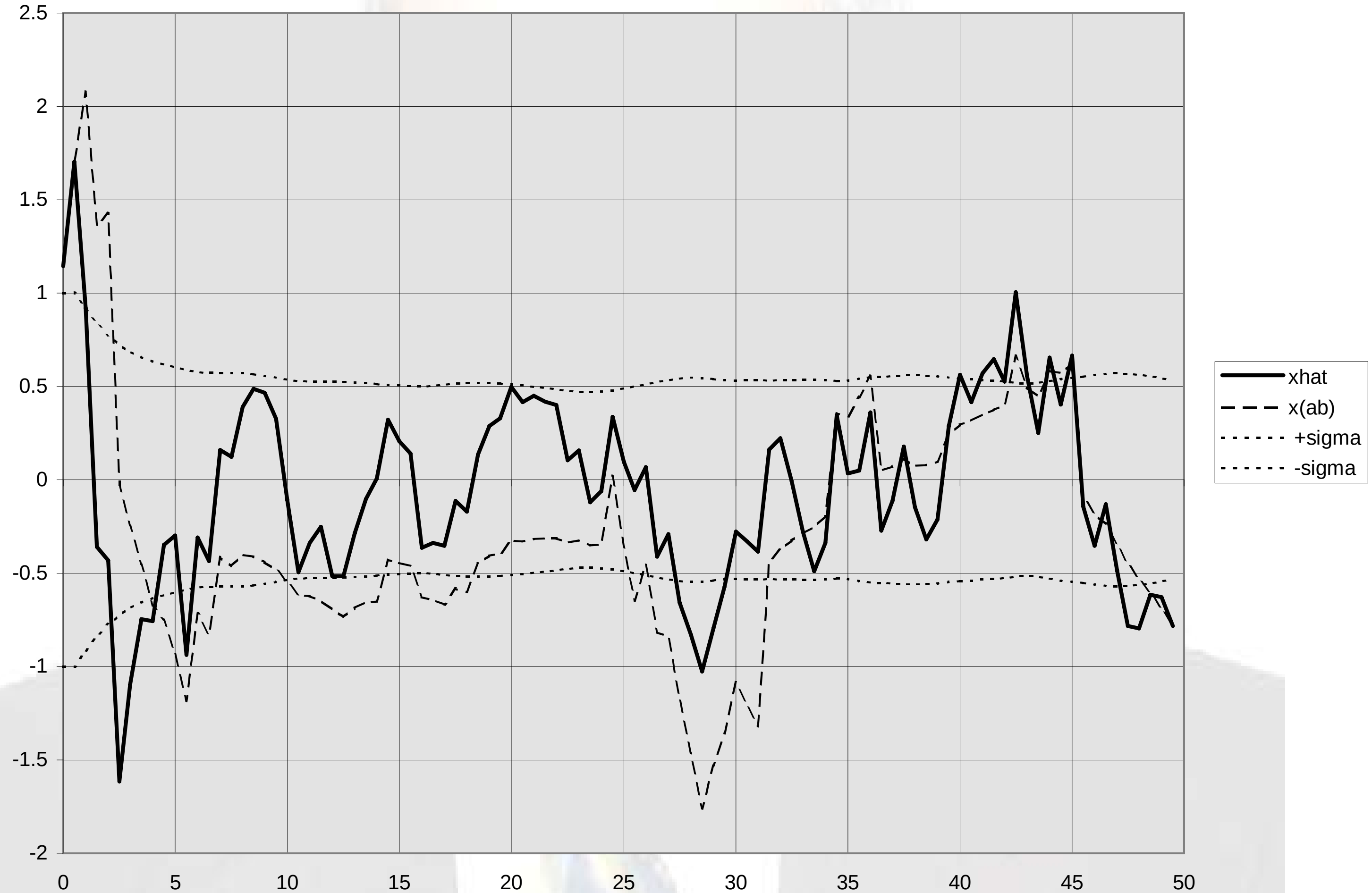
- Once Tuning Produces a Design, Perform a Monte Carlo Simulation
- During Run, Save for Each Update Time
  - Sum of Estimates for Each State
  - Sum of Squares of Estimates for Each State
  - Maximum and Minimum of Estimates for Each State
- Compute and Plot Summary
  - Mean of Each State
  - Variance and Extreme Values of Each State

File of the Week

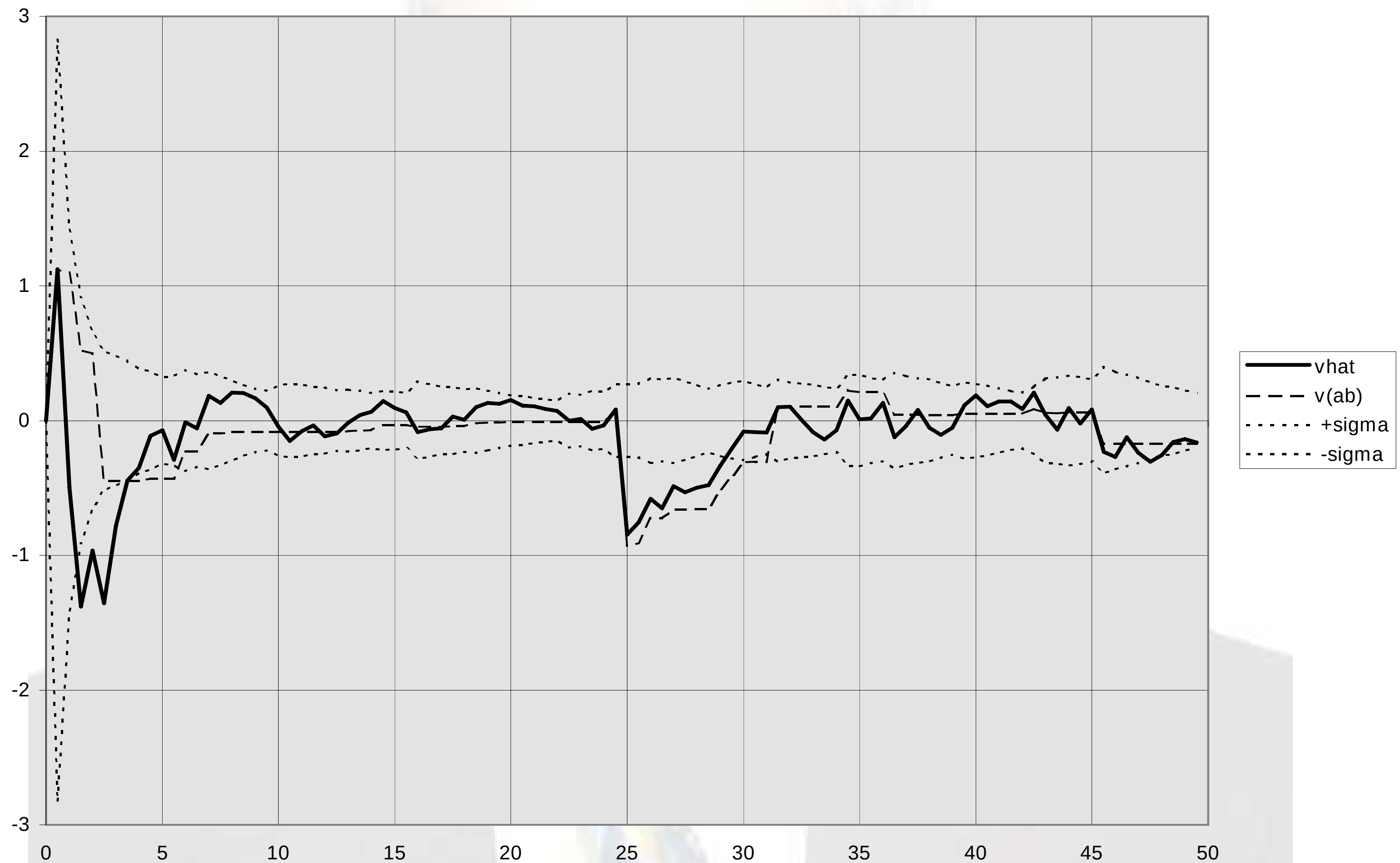
- Two Modules
  - Two Trackers
    - » Adaptive “Snake Oil” Two State
    - » Adaptive Alpha Beta
  - Random Number Generator
- Operations
  - Scenario with Abrupt Velocity Step
  - Adaptive Tuning Parameters on Sheet1
  - Plots Position and Velocity Errors

These “File of the Week” handouts are Excel files that are not portable across versions. Thus these files are omitted.

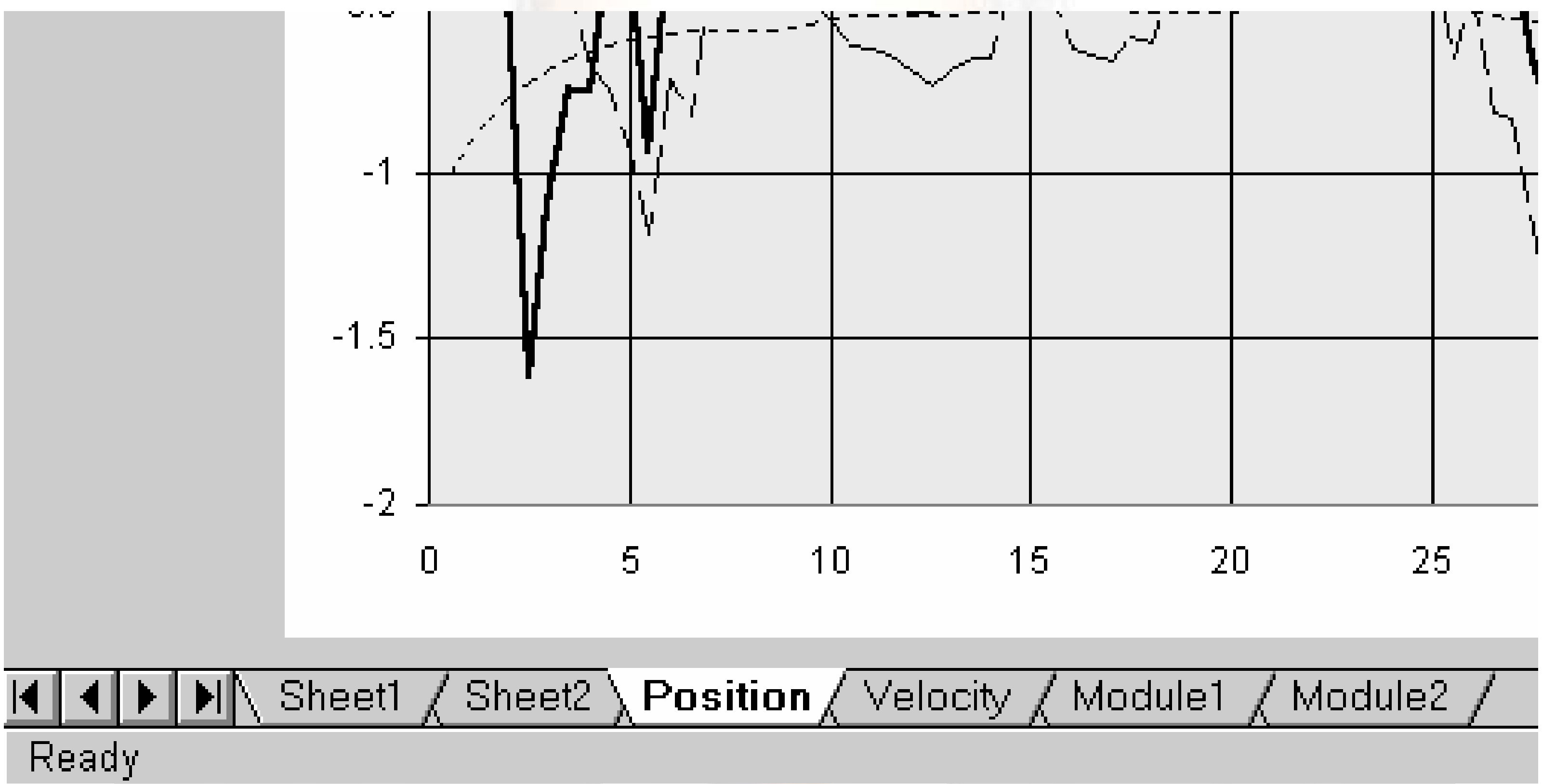
Position State Display



Velocity State Display



Module Tabs


-1
-1.5
-2
0
5
10
15
20
25
Sheet1
Sheet2
Position
Velocity
Module1
Module2
Ready

ATEP SYS12525

Radar Trackers and Applications for SAADS

October 26, 1999

Topic 14: Process Noise Modeling

Sensor Systems Engineering for the 21st Century

System Process Noise

- Used in
  - Most Raytheon Air to Air Trackers
  - Discoverer II Space to Ground Trackers
  - AMSTE Air to Ground Trackers
  - Others
- Types
  - Singer Variance Model
  - Nearly Constant Acceleration
  - Nearly Constant Velocity
- Reference: Singer, R. A., “Estimating Optimal Tracking Filter Performance for Manned Maneuvering Targets, “AES-6, pp. 473-483, July 1970

Markov System Model

- System Model

$$\underline{x} = \begin{bmatrix} \text{position} \\ \text{velocity} \\ \text{acceleration} \end{bmatrix}, \quad \dot{\underline{x}} = F \cdot \underline{x} + \underline{w}$$

- Constant Matrix F

$$F = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & -\rho \end{bmatrix}$$

- Mentioned in Gelb, Problem 3-6 p. 98

System Transition Matrix

$$F = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & -\rho \end{bmatrix}, F^n = \begin{bmatrix} 0 & 0 & (-\rho)^{n-2} \\ 0 & 0 & (-\rho)^{n-1} \\ 0 & 0 & (-\rho)^n \end{bmatrix}$$

$$\exp(F \cdot t) = \begin{bmatrix} 1 & t & f_2(t) \\ 0 & 1 & f_1(t) \\ 0 & 0 & f_0(t) \end{bmatrix} = \begin{bmatrix} 1 & t & \frac{t^2}{2} + \cdots \\ 0 & 1 & t + \cdots \\ 0 & 0 & \exp(-\rho \cdot t) \end{bmatrix}$$

$$f_k(t) = \frac{\exp(-\rho \cdot t) - \sum_{p=0}^{k-1} \frac{(-\rho \cdot t)^p}{p!}}{(-\rho)^k} = \frac{t^k}{k!} + \ldots$$

Linear Variance Equation

- Equation (Gelb, p. 77)

$$\dot{P} = F \cdot P + P \cdot F^T + Q$$

- Solution (Gelb problem 3-1, p. 97)

$$P(t) = \Phi(t - t_0) \cdot P(t_0) \cdot \Phi^T(t - t_0) + \int_{t_0}^{t} \Phi(t - \tau) \cdot Q \cdot \Phi^T(t - \tau) \cdot d\tau$$

$$\Phi(t - t_0) = \exp(F \cdot (t - t_0)), \quad Q = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & q_{33} \end{bmatrix}$$

Equivalent Process Noise for Discrete Covariance Update

- Discrete Covariance Update

$$P = \Phi \cdot P \cdot \Phi^T + \Gamma$$

- Process Noise Equation

$$\Gamma\left(t - t_0\right) = \int_{t_0}^{t} \Phi\left(t - \tau\right) \cdot Q \cdot \Phi^T\left(t - \tau\right) \cdot d\tau$$

$$= q_{33} \cdot \int_{t_0}^{t} \begin{bmatrix} f_2^{\,2}(\tau) & f_1(\tau) \cdot f_2(\tau) & f_0(\tau) \cdot f_2(\tau) \\ f_1(\tau) \cdot f_2(\tau) & f_1^{\,2}(\tau) & f_0(\tau) \cdot f_1(\tau) \\ f_0(\tau) \cdot f_2(\tau) & f_0(\tau) \cdot f_1(\tau) & f_0^{\,2}(\tau) \end{bmatrix} \cdot d\tau$$

Nearly Constant Acceleration

$$\Phi(t-t_0) = \begin{bmatrix} 1 & t-t_0 & \frac{(t-t_0)^2}{2} \\ 0 & 1 & t-t_0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$\Gamma(t-t_0) = q_{33} \cdot \begin{bmatrix} \frac{(t-t_0)^5}{20} & \frac{(t-t_0)^4}{8} & \frac{(t-t_0)^3}{6} \\ \frac{(t-t_0)^4}{8} & \frac{(t-t_0)^3}{3} & \frac{(t-t_0)^2}{2} \\ \frac{(t-t_0)^3}{6} & \frac{(t-t_0)^2}{2} & t-t_0 \end{bmatrix}$$

Modified Singer

- No Acceleration State

$$\Phi(t-t_0)=\begin{bmatrix}1 & f_1(t-t_0)\\ 0 & f_0(t-t_0)\end{bmatrix}$$

- Process Noise Equation

$$\Gamma(t-t_0)=\int_{t_0}^{t}\Phi(t-\tau)\cdot Q\cdot\Phi^T(t-\tau)\cdot d\tau$$

$$=q_{22}\cdot\int_{t_0}^{t}\begin{bmatrix}f_1^{\,2}(\tau) & f_0(\tau)\cdot f_1(\tau)\\ f_0(\tau)\cdot f_1(\tau) & f_0^{\,2}(\tau)\end{bmatrix}\cdot d\tau$$

Nearly Constant Velocity

- Modified Singer for $\rho = 0$

$$\Phi(t - t_0) = \begin{bmatrix} 1 & t - t_0 \\ 0 & 1 \end{bmatrix}$$

- Discrete Formulation Process Noise

$$\Gamma(t - t_0) = q_{22} \cdot \begin{bmatrix} \frac{(t - t_0)^3}{3} & \frac{(t - t_0)^2}{2} \\ \frac{(t - t_0)^2}{2} & t - t_0 \end{bmatrix}$$

Terms of Gamma Matrix

$$\int_{t_0}^{t} f_0^{\,2}(\tau)\cdot d\tau = \frac{1-\exp\left(-2\rho\left(t-t_0\right)\right)}{2\rho}$$

$$\int_{t_0}^{t} f_0(\tau)\cdot f_1(\tau)\cdot d\tau = \frac{-\exp\left(-2\rho\left(t-t_0\right)\right)+2\exp\left(-\rho\left(t-t_0\right)\right)-1}{2\rho^2}$$

$$\int_{t_0}^{t} f_1^{\,2}(\tau)\cdot d\tau = \frac{-\exp\left(-2\rho\left(t-t_0\right)\right)+4\exp\left(-\rho\left(t-t_0\right)\right)+2\rho\left(t-t_0\right)-3}{2\rho^3}$$

Higher Order Terms

$$\int_{t_0}^{t} f_0(\tau)\cdot f_2(\tau)\cdot d\tau$$

$$= \frac{-\exp\left(-2\rho\left(t-t_0\right)\right)-2\cdot\rho\left(t-t_0\right)\exp\left(-\rho\left(t-t_0\right)\right)+1}{2\rho^3}$$

$$\int_{t_0}^{t} f_1(\tau)\cdot f_2(\tau)\cdot d\tau =$$

$$\frac{-\exp\left(-2\rho\left(t-t_0\right)\right)+2\cdot\left(1-\rho\left(t-t_0\right)\right)\cdot\exp\left(-\rho\left(t-t_0\right)\right)-\left(1-\rho\left(t-t_0\right)\right)^2}{2\rho^4}$$

Last Term

$$6\rho^5 \int_{t_0}^{t} f_2^2(\tau) \cdot d\tau =$$

$$-3\exp\left(-2\rho\left(t-t_0\right)\right) - 12\exp\left(-\rho\left(t-t_0\right)\right)\cdot\rho\left(t-t_0\right)$$

$$+2\cdot\left(\rho\left(t-t_0\right)\right)^3 - 6\cdot\left(\rho\left(t-t_0\right)\right)^2 + 6\cdot\rho\left(t-t_0\right) + 3$$

Covariance Mapping to 9 States

$$\Phi(dt)=\begin{bmatrix} I & I\cdot dt & I\cdot f_2(dt) \\ 0 & I & I\cdot f_1(dt) \\ 0 & 0 & I\cdot f_0(dt) \end{bmatrix},\ \ Q=\begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & Q_{33} \end{bmatrix}$$

$$\Gamma(t-t_0)=\int_{t_0}^{t}\Phi(t-\tau)\cdot Q\cdot\Phi^T(t-\tau)\cdot d\tau$$

$$=q_{33}\cdot\int_{t_0}^{t}\begin{bmatrix} Q_{33}\cdot f_2^{\,2}(\tau) & Q_{33}\cdot f_1(\tau)\cdot f_2(\tau) & Q_{33}\cdot f_0(\tau)\cdot f_2(\tau) \\ Q_{33}\cdot f_1(\tau)\cdot f_2(\tau) & Q_{33}\cdot f_1^{\,2}(\tau) & Q_{33}\cdot f_0(\tau)\cdot f_1(\tau) \\ Q_{33}\cdot f_0(\tau)\cdot f_2(\tau) & Q_{33}\cdot f_0(\tau)\cdot f_1(\tau) & Q_{33}\cdot f_0^{\,2}(\tau) \end{bmatrix}\cdot d\tau$$

Interpreting Process Noise

- The Defining Equation for Acceleration Noise

$$\dddot{x} = -\rho \cdot \ddot{x} + w, \ \left\langle w^2 \right\rangle = q_{33}$$

- Physical Interpretation
  - Gelb, pp. 42-45, 81-84
  - Acceleration Noise Autocorrelation and Power Spectrum

$$\phi(\tau) = \frac{q_{33}}{2\rho} \cdot \exp(-\rho \cdot |\tau|), \ \Phi(\omega) = \frac{q_{33}}{\omega^2 + \rho^2}$$

Example 1 -- Singer Model

- Process Noise Specification
  - RMS Acceleration Noise Amplitude of c g's
  - Time Constant of T seconds
- Solve for q_{33}:

$$\frac{q_{33}}{2\cdot\rho} = \left(c\cdot\left(9.80665\,\frac{\text{meters}}{\text{second}^2}\right)\right)^2,\ \ \rho = \frac{1}{T}$$

- Physical Units of q_{33} are meters^2/second^5
- Can Be Combined with Adaptive Techniques

Example 2 -- No Acceleration State

- Process Noise Specification
  - RMS Velocity Noise Amplitude of k knots
  - Time Constant of T Seconds
- Solve for q_{22}:

$$\frac{q_{22}}{2\cdot\rho}=\left(k\cdot\left(\frac{1852}{3600}\frac{\text{meters}}{\text{second}}\right)\right)^2,\quad \rho=\frac{1}{T}$$

- Physical Units of q_{22} are meters^2/second^3
- Can be Combined with Adaptive Techniques

Examples of Q_{33}

- Independent Process Noise Variances Along Three Orthogonal Axes
  - Airframe coordinates -- longitudinal, lateral, and down
  - Ground vehicle -- longitudinal, lateral
  - Ground vehicle on road -- longitudinal only
- Variances qa_i, axes $\underline{ua}_i$

$$Q_{33} = \sum_i qa_i \cdot \underline{ua}_i \cdot \underline{ua}_i^T$$