Optimal Estimation Methods

(Lecture 17 – Batch State Estimation: 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

Continuous-Time (i)

- True system and measurements

$$\frac{d}{dt}\mathbf{x}(t) = F(t)\,\mathbf{x}(t) + B(t)\,\mathbf{u}(t) + G(t)\,\mathbf{w}(t)$$
$$\tilde{\mathbf{y}}(t) = H(t)\,\mathbf{x}(t) + \mathbf{v}(t)$$

  - Forward filter is the same as the forward Kalman filter

$$\frac{d}{dt}\hat{\mathbf{x}}_f(t) = F(t)\,\hat{\mathbf{x}}_f(t) + B(t)\,\mathbf{u}(t) + K_f(t)[\tilde{\mathbf{y}}(t) - H(t)\,\hat{\mathbf{x}}_f(t)]$$
$$K_f(t) = P_f(t)\,H^T(t)\,R^{-1}(t)$$
$$\frac{d}{dt}P_f(t) = F(t)\,P_f(t) + P_f(t)\,F^T(t) - P_f(t)\,H^T(t)\,R^{-1}(t)\,H(t)\,P_f(t) + G(t)\,Q(t)\,G^T(t)$$

Continuous-Time (ii)

- Backward filter

$$\frac{d}{dt}\hat{\mathbf{x}}_b(t) = F(t)\,\hat{\mathbf{x}}_b(t) + B(t)\,\mathbf{u}(t) + K_b(t)[\tilde{\mathbf{y}}(t) - H(t)\,\hat{\mathbf{x}}_b(t)]$$

$$K_b(t) = P_b(t)\,H^T(t)\,R^{-1}(t)$$

$$\frac{d}{dt}P_b(t) = F(t)\,P_b(t) + P_b(t)\,F^T(t) - P_b(t)\,H^T(t)\,R^{-1}(t)\,H(t)\,P_b(t) + G(t)\,Q(t)\,G^T(t)$$

- These equations much be integrated backwards in time
  - It is convenient to set $\tau = T - t$, where T is the terminal time of the data interval
  - Since $d\mathbf{x}/dt = -d\mathbf{x}/d\tau$, writing the truth state equation in terms of τ gives

$$\frac{d}{d\tau}\mathbf{x}(t) = -F(t)\,\mathbf{x}(t) - B(t)\,\mathbf{u}(t) - G(t)\,\mathbf{w}(t)$$

Continuous-Time (iii)

- Therefore, the backward filter equations can be written in terms of τ by replacing $F(t)$ with $-F(t)$, $B(t)$ with $-B(t)$, and $G(t)$ with $-G(t)$, which leads to

$$\frac{d}{d\tau}\hat{\mathbf{x}}_b(t) = -F(t)\,\hat{\mathbf{x}}_b(t) - B(t)\,\mathbf{u}(t) + K_b(t)[\tilde{\mathbf{y}}(t) - H(t)\,\hat{\mathbf{x}}_b(t)] \tag{1}$$

$$K_b(t) = P_b(t)\,H^T(t)\,R^{-1}(t)$$

$$\begin{aligned}\frac{d}{d\tau}P_b(t) &= -F(t)\,P_b(t) - P_b(t)\,F^T(t)\\ &\quad - P_b(t)\,H^T(t)\,R^{-1}(t)\,H(t)\,P_b(t) + G(t)\,Q(t)\,G^T(t)\end{aligned} \tag{2}$$

- From this point forward whenever $d/d\tau$ is used, this will denote a backward differentiation $\frac{d}{dt}$ - forward
- Note that if $F(t)$ is stable going forward in time, then $-F(t)$ is stable going backward in time
- The continuous-time smoother combination of the forward and backward state estimates follows exactly from the discrete-time equivalent

Continuous-Time (iv)

- The continuous-time equivalent of smoother covariance is simply given by

$$\boxed{P(t) = \left[P_f^{-1}(t) + P_b^{-1}(t)\right]^{-1}} \qquad (3)$$

- The continuous-time equivalent of the smoother state estimate is simply given by

$$\boxed{\hat{\mathbf{x}}(t) = P(t)\,\left[P_f^{-1}(t)\,\hat{\mathbf{x}}_f(t) + P_b^{-1}(t)\,\hat{\mathbf{x}}_b(t)\right]}$$

- Boundary Conditions
  - Since at time $t = T$ the smoother estimate must be the same as the forward Kalman filter
  - This clearly requires the following conditions

$$\hat{\mathbf{x}}(T) = \hat{\mathbf{x}}_f(T)$$
$$P(T) = P_f(T)$$

Continuous-Time (v)

- From Eq. (3) the covariance condition at the terminal time can only be satisfied when $P_b^{-1}(T) = 0$
  - Therefore, $P_b(t)$ is not finite at the terminal time
  - To overcome this difficulty, consider taking the time derivative of $P_b^{-1}(t)\, P_b(t) = I$, which gives

$$\left[\frac{d}{d\tau} P_b^{-1}(t)\right] P_b(t) + P_b^{-1}(t) \left[\frac{d}{d\tau} P_b(t)\right] = 0$$

  - Rearranging yields

$$\left[\frac{d}{d\tau} P_b^{-1}(t)\right] = -P_b^{-1}(t) \left[\frac{d}{d\tau} P_b(t)\right] P_b^{-1}(t)$$

  - Substituting Eq. (2) gives

$$\begin{aligned}\frac{d}{d\tau} P_b^{-1}(t) &= P_b^{-1}(t)\, F(t) + F^T(t)\, P_b^{-1}(t) \\ &\quad - P_b^{-1}(t)\, G(t)\, Q(t)\, G^T(t)\, P_b^{-1}(t) + H^T(t)\, R^{-1}(t)\, H(t)\end{aligned} \qquad (4)$$

with boundary condition $P_b^{-1}(T) = 0$

Continuous-Time (vi)

- Even with the inverse of $P_b(t)$ computed directly now, Eq. (3) still requires two matrix inverses
  - Overcome by using the matrix inversion lemma on Eq. (3), giving

$$\boxed{P(t) = P_f(t) - P_f(t)\,P_b^{-1}(t)[I + P_f(t)\,P_b^{-1}(t)]^{-1}P_f(t)}$$

  - Only one inverse is now required
- The final time boundary condition on the backwards state estimate is still unknown
- Define the following variable

$$\hat{\boldsymbol{\chi}}_b(t) \equiv P_b^{-1}(t)\,\hat{\mathbf{x}}_b(t)$$

where $\hat{\boldsymbol{\chi}}_b(T) = \mathbf{0}$ since $P_b^{-1}(T) = 0$
- Differentiating gives

$$\left[\frac{d}{d\tau}\hat{\boldsymbol{\chi}}_b(t)\right] = \left[\frac{d}{d\tau}P_b^{-1}(t)\right]\hat{\mathbf{x}}_b(t) + P_b^{-1}(t)\left[\frac{d}{d\tau}\hat{\mathbf{x}}_b(t)\right]$$

Continuous-Time (vii)

- Substituting Eqs. (1) and (4) gives

$$\boxed{\begin{aligned}\frac{d}{d\tau}\hat{\boldsymbol{\chi}}_b(t) = {} & \left[F(t) - G(t)\,Q(t)\,G^T(t)\,P_b^{-1}(t)\right]^T\hat{\boldsymbol{\chi}}_b(t) \\ & - P_b^{-1}(t)\,B(t)\,\mathbf{u}(t) + H^T(t)\,R^{-1}(t)\,\tilde{\mathbf{y}}(t)\end{aligned}}$$

- Recall the discrete-time smoother state equation

$$\hat{\mathbf{x}}_k = [I - K_k]\hat{\mathbf{x}}_{f\,k}^{+} + P_k\hat{\boldsymbol{\chi}}_{b\,k}^{-}$$

- Its continuous-time equivalent is given by

$$\boxed{\hat{\mathbf{x}}(t) = [I - K(t)]\hat{\mathbf{x}}_f(t) + P(t)\,\hat{\boldsymbol{\chi}}_b(t)}$$

where the gain is given by

$$K(t) \equiv P_f(t)\,P_b^{-1}(t)[I + P_f(t)\,P_b^{-1}(t)]^{-1}$$

Continuous-Time Summary (i)

Model	$\frac{d}{dt}\mathbf{x}(t) = F(t)\,\mathbf{x}(t) + B(t)\,\mathbf{u}(t) + G(t)\,\mathbf{w}(t),\ \mathbf{w}(t) \sim N(\mathbf{0}, Q(t))$ $\tilde{\mathbf{y}}(t) = H(t)\,\mathbf{x}(t) + \mathbf{v}(t),\ \mathbf{v}(t) \sim N(\mathbf{0}, R(t))$ spectral densities
Forward Covariance	$\frac{d}{dt}P_f(t) = F(t)\,P_f(t) + P_f(t)\,F^T(t)$ $-P_f(t)\,H^T(t)\,R^{-1}(t)H(t)\,P_f(t)$ $+G(t)\,Q(t)\,G^T(t),$ $P_f(t_0) = E\{\tilde{\mathbf{x}}_f(t_0)\,\tilde{\mathbf{x}}_f^T(t_0)\}$
Forward Filter	$\frac{d}{dt}\hat{\mathbf{x}}_f(t) = F(t)\,\hat{\mathbf{x}}_f(t) + B(t)\,\mathbf{u}(t)$ $+P_f(t)\,H^T(t)\,R^{-1}(t)[\tilde{\mathbf{y}}(t) - H(t)\,\hat{\mathbf{x}}_f(t)],\quad \hat{\mathbf{x}}_f(t_0) = \hat{\mathbf{x}}_{f0}$

Continuous-Time Summary (ii)

Backward Covariance	$\frac{d}{d\tau} P_b^{-1}(t) = P_b^{-1}(t)\, F(t) + F^T(t)\, P_b^{-1}(t)$ $- P_b^{-1}(t)\, G(t)\, Q(t)\, G^T(t)\, P_b^{-1}(t)$ $+ H^T(t)\, R^{-1}(t)\, H(t), \quad P_b^{-1}(T) = 0$
Backward Filter	$\frac{d}{d\tau} \hat{\boldsymbol{\chi}}_b(t) = \left[F(t) - G(t)\, Q(t)\, G^T(t)\, P_b^{-1}(t)\right]^T \hat{\boldsymbol{\chi}}_b(t)$ $- P_b^{-1}(t)\, B(t)\, \mathbf{u}(t) + H^T(t)\, R^{-1}(t)\, \tilde{\mathbf{y}}(t), \quad \hat{\boldsymbol{\chi}}_b(T) = 0$
Gain	$K(t) = P_f(t)\, P_b^{-1}(t)\left[I + P_f(t)\, P_b^{-1}(t)\right]^{-1}$
Covariance	$P(t) = [I - K(t)] P_f(t)$
Estimate	$\hat{\mathbf{x}}(t) = [I - K(t)]\hat{\mathbf{x}}_f(t) + P(t)\, \hat{\boldsymbol{\chi}}_b(t)$

Steady State Smoother (i)

- For autonomous systems, at steady state we have

$$P_b^{-1} F + F^T P_b^{-1} - P_b^{-1} G\, Q\, G^T P_b^{-1} + H^T R^{-1} H = 0$$

  - As before, form the following Hamiltonian matrix

$$\mathcal{H} \equiv \begin{bmatrix} -F & G\,Q G^T \\ H^T R^{-1} H & F^T \end{bmatrix}$$

  - Take an eigenvalue/eigenvector decomposition

$$\mathcal{H} = \begin{bmatrix} W_{11} & W_{12} \\ W_{21} & W_{22} \end{bmatrix} \begin{bmatrix} \Lambda & 0 \\ 0 & -\Lambda \end{bmatrix} \begin{bmatrix} W_{11} & W_{12} \\ W_{21} & W_{22} \end{bmatrix}^{-1}$$

    where Λ is a diagonal matrix of the n eigenvalues in the right-half plane, and W_{11}, W_{21}, W_{12}, and W_{22} are block element of the eigenvector matrix

Steady State Smoother (ii)

- Using the same approach as before the steady-state covariance for the update is given by

$$\boxed{P_b^{-1} = W_{21} W_{11}^{-1}}$$

- Compute the steady-state forward covariance and gain from before
- Then the steady-state gain for the backwards filter can be computed, as well as the steady-state smoother gain and covariance

$$K = P_f \, P_b^{-1} [I + P_f \, P_b^{-1}]^{-1}$$
$$P = [I - K] P_f$$

RTS Smoother (i)

- Rauch, Tung and Striebel (RTS) form begins by taking the backwards time-derivative of

$$P^{-1}(t) = P_f^{-1}(t) + P_b^{-1}(t)$$

which gives

$$\frac{d}{d\tau} P^{-1}(t) = -P_f^{-1}(t) \left[\frac{d}{d\tau} P_f(t) \right] P_f^{-1}(t) + \frac{d}{d\tau} P_b^{-1}(t)$$

using $dP_f/\,dt = -dP_f/\,d\tau$ gives

$$\frac{d}{d\tau} P^{-1}(t) = P_f^{-1}(t) \left[\frac{d}{dt} P_f(t) \right] P_f^{-1}(t) + \frac{d}{d\tau} P_b^{-1}(t)$$

- Substitute the following

$$\begin{aligned} \frac{d}{d\tau} P_b^{-1}(t) &= P_b^{-1}(t)\, F(t) + F^T(t)\, P_b^{-1}(t) \\ &\quad - P_b^{-1}(t)\, G(t)\, Q(t)\, G^T(t)\, P_b^{-1}(t) + H^T(t)\, R^{-1}(t)\, H(t) \end{aligned}$$

RTS Smoother (ii)

and

$$\frac{d}{dt} P_f(t) = F(t)\, P_f(t) + P_f(t)\, F^T(t) - P_f(t)\, H^T(t)\, R^{-1}(t)\, H(t)\, P_f(t) + G(t)\, Q(t)\, G^T(t)$$

to give

$$\frac{d}{d\tau} P^{-1}(t) = P_f^{-1}(t)\, F(t) + F^T(t)\, P_f^{-1}(t) + P_f^{-1}(t)\, G(t)\, Q(t)\, G^T(t)\, P_f^{-1}(t) + P_b^{-1}(t)\, F(t) + F^T(t)\, P_b^{-1}(t) - P_b^{-1}(t)\, G(t)\, Q(t)\, G^T(t)\, P_b^{-1}(t)$$

- Next substitute $P_b^{-1}(t) = P^{-1}(t) - P_f^{-1}(t)$ to give

$$\frac{d}{d\tau} P^{-1}(t) = P^{-1}(t)\, F(t) + F^T(t)\, P^{-1}(t) + P_f^{-1}(t)\, G(t)\, Q(t)\, G^T(t)\, P_f^{-1}(t) - \left[P^{-1}(t) - P_f^{-1}(t)\right] G(t)\, Q(t)\, G^T(t) \left[P^{-1}(t) - P_f^{-1}(t)\right]$$

RTS Smoother (iii)

- Substituting the following relation

$$\left[\frac{d}{d\tau} P^{-1}(t)\right] = P^{-1}(t) \left[\frac{d}{dt} P(t)\right] P^{-1}(t)$$

  and then multiplying both sides of the resulting expression by $P(t)$ yields

$$\boxed{\begin{aligned} \frac{d}{dt} P(t) = &\left[F(t) + G(t)\, Q(t)\, G^T(t)\, P_f^{-1}(t)\right] P(t) \\ &+ P(t) \left[F(t) + G(t)\, Q(t)\, G^T(t)\, P_f^{-1}(t)\right]^T - G(t)\, Q(t)\, G^T(t) \end{aligned}}$$

- Since $P_b^{-1}(T) = 0$, then this equation is integrated backward in time with the boundary condition $P(T) = P_f(T)$
- This form clearly has significant computational advantages over integrating the backward filter covariance
  - The smoother covariance is calculated directly without the need to first calculate the backward filter covariance

RTS Smoother (iv)

- For autonomous systems, at steady state we have

$$\boxed{0 = \left[F + G\,Q\,G^T\,P_f^{-1}\right]P + P\left[F + G\,Q\,G^T\,P_f^{-1}\right]^T - G\,Q\,G^T}$$

  - Reduces down to an algebraic Lyapunov equation, which is a linear equation
    - Can be used to find the steady-state value of P
  - Note what happens when $Q = 0$
    - No viable solution for P is possible
    - Again re-enforces that no smoothing is possible without process noise

- To derive the smoother state equation begin with

$$P^{-1}(t)\hat{\mathbf{x}}(t) = P_f^{-1}(t)\,\hat{\mathbf{x}}_f(t) + \hat{\boldsymbol{\chi}}_b(t)$$

  - Taking the time derivative gives

$$P^{-1}(t)\left[\frac{d}{dt}\hat{\mathbf{x}}(t)\right] + \left[\frac{d}{dt}P^{-1}(t)\right]\,\hat{\mathbf{x}}(t)$$
$$= P_f^{-1}(t)\,\left[\frac{d}{dt}\hat{\mathbf{x}}_f(t)\right] + \left[\frac{d}{dt}P_f^{-1}(t)\right]\,\hat{\mathbf{x}}_f(t) + \frac{d}{dt}\hat{\boldsymbol{\chi}}_b(t)$$

  - Use the following relations

$$\left[\frac{d}{dt}P_f^{-1}(t)\right] = -P_f^{-1}(t)\,\left[\frac{d}{dt}P_f(t)\right]\,P_f^{-1}(t)$$
$$\left[\frac{d}{dt}P^{-1}(t)\right] = -P^{-1}(t)\,\left[\frac{d}{dt}P(t)\right]\,P^{-1}(t)$$

RTS Smoother (vi)

to give

$$P^{-1}(t)\left[\frac{d}{dt}\hat{\mathbf{x}}(t)\right] = P^{-1}(t)\left[\frac{d}{dt}P(t)\right]P^{-1}(t)\,\hat{\mathbf{x}}(t) + P_f^{-1}(t)\left[\frac{d}{dt}\hat{\mathbf{x}}_f(t)\right] - P_f^{-1}(t)\left[\frac{d}{dt}P_f(t)\right]P_f^{-1}(t)\,\hat{\mathbf{x}}_f(t) + \frac{d}{dt}\hat{\boldsymbol{\chi}}_b(t)$$

- Substituting all expressions and after considerable algebraic manipulations yields

$$\frac{d}{dt}\hat{\mathbf{x}}(t) = F(t)\,\hat{\mathbf{x}}(t) + B(t)\,\mathbf{u}(t) + G(t)\,Q(t)\,G^T(t)\,P_f^{-1}(t)\,[\hat{\mathbf{x}}(t) - \hat{\mathbf{x}}_f(t)]$$

- Simply use $d/dt\,[\hat{\mathbf{x}}(t)] = -d/d\tau\,[\hat{\mathbf{x}}(t)]$ to obtain

$$\boxed{\frac{d}{d\tau}\hat{\mathbf{x}}(t) = -F(t)\,\hat{\mathbf{x}}(t) - B(t)\,\mathbf{u}(t) - G(t)\,Q(t)\,G^T(t)\,P_f^{-1}(t)\,[\hat{\mathbf{x}}(t) - \hat{\mathbf{x}}_f(t)]}$$

- This integrated backward in time with the boundary condition

$$\hat{\mathbf{x}}(T) = \hat{\mathbf{x}}_f(T)$$

RTS Smoother Summary (i)

Model	$\frac{d}{dt}\mathbf{x}(t) = F(t)\,\mathbf{x}(t) + B(t)\,\mathbf{u}(t) + G(t)\,\mathbf{w}(t),\ \mathbf{w}(t) \sim N(\mathbf{0}, Q(t))$ $\tilde{\mathbf{y}}(t) = H(t)\,\mathbf{x}(t) + \mathbf{v}(t),\ \mathbf{v}(t) \sim N(\mathbf{0}, R(t))$
Forward Covariance	$\frac{d}{dt}P_f(t) = F(t)\,P_f(t) + P_f(t)\,F^T(t)$ $-P_f(t)\,H^T(t)\,R^{-1}(t)H(t)\,P_f(t)$ $+G(t)\,Q(t)\,G^T(t),$ $P_f(t_0) = E\{\tilde{\mathbf{x}}_f(t_0)\,\tilde{\mathbf{x}}_f^T(t_0)\}$
Forward Filter	$\frac{d}{dt}\hat{\mathbf{x}}_f(t) = F(t)\,\hat{\mathbf{x}}_f(t) + B(t)\,\mathbf{u}(t)$ $+P_f(t)\,H^T(t)\,R^{-1}(t)[\tilde{\mathbf{y}}(t) - H(t)\,\hat{\mathbf{x}}_f(t)], \quad \hat{\mathbf{x}}_f(t_0) = \hat{\mathbf{x}}_{f0}$

RTS Smoother Summary (ii)

Smoother Covariance	$\frac{d}{d\tau}P(t) = -[F(t) + G(t)\,Q(t)\,G^T(t)\,P_f^{-1}(t)]P(t)$ $-P(t)[F(t) + G(t)\,Q(t)\,G^T(t)\,P_f^{-1}(t)]^T$ $+G(t)\,Q(t)\,G^T(t), \quad P(T) = P_f(T)$
Smoother Estimate	$\frac{d}{d\tau}\hat{\mathbf{x}}(t) = -F(t)\,\hat{\mathbf{x}}(t) - B(t)\,\mathbf{u}(t)$ $-G(t)\,Q(t)\,G^T(t)\,P_f^{-1}(t)\,[\hat{\mathbf{x}}(t) - \hat{\mathbf{x}}_f(t)]\,, \quad \hat{\mathbf{x}}(T) = \hat{\mathbf{x}}_f(T)$

- Note how much simpler the backwards equations are compared to the other smoother
- The covariance expression is not needed either to obtain the state estimate

RTS Smoother Stability (i)

- As before, we consider only the homogenous part to prove stability

$$\frac{d}{d\tau}\hat{\mathbf{x}}(t) = -[F(t) + G(t)\,Q(t)\,G^T(t)\,P_f^{-1}(t)]\,\hat{\mathbf{x}}(t) \qquad (1)$$

  - Consider the following candidate Lyapunov function

$$V[\hat{\mathbf{x}}(t)] = \hat{\mathbf{x}}^T(t)\,P_f^{-1}(t)\,\hat{\mathbf{x}}(t)$$

  - Take the time derivative

$$\frac{d}{d\tau}V[\hat{\mathbf{x}}(t)] = \left[\frac{d}{d\tau}\hat{\mathbf{x}}(t)\right]^T P_f^{-1}(t)\,\hat{\mathbf{x}}(t) + \hat{\mathbf{x}}^T(t)\left[\frac{d}{d\tau}P_f^{-1}(t)\right]\hat{\mathbf{x}}(t)$$
$$+\,\hat{\mathbf{x}}^T(t)\,P_f^{-1}(t)\left[\frac{d}{d\tau}\hat{\mathbf{x}}(t)\right]$$

RTS Smoother Stability (ii)

- Now use $dP_f^{-1}/d\tau = -dP_f^{-1}/dt$

$$\frac{d}{d\tau}V[\hat{\mathbf{x}}(t)] = \left[\frac{d}{d\tau}\hat{\mathbf{x}}(t)\right]^T P_f^{-1}(t)\,\hat{\mathbf{x}}(t) - \hat{\mathbf{x}}^T(t)\left[\frac{d}{dt}P_f^{-1}(t)\right]\hat{\mathbf{x}}(t)$$
$$+\,\hat{\mathbf{x}}^T(t)\,P_f^{-1}(t)\left[\frac{d}{d\tau}\hat{\mathbf{x}}(t)\right]$$

- Next use

$$\left[\frac{d}{dt}P_f^{-1}(t)\right] = -P_f^{-1}(t)\left[\frac{d}{dt}P_f(t)\right]P_f^{-1}(t)$$

to give

$$\frac{d}{dt}P_f^{-1}(t) = -P_f^{-1}(t)\,F(t) - F^T(t)\,P_f^{-1}(t)$$
$$-\,P_f^{-1}(t)G(t)\,Q(t)\,G^T(t)P_f^{-1}(t) + H^T(t)\,R^{-1}(t)\,H(t)\,P_f(t)$$

- Substitute this equation and Eq. (1) into the derivative of the candidate Lyapunov function

RTS Smoother Stability (iii)

$$\frac{d}{d\tau}V[\hat{\mathbf{x}}(t)] = -\hat{\mathbf{x}}^T(t)[\cancel{F^T(t)\,P_f^{-1}(t)} + P_f^{-1}(t)\,G(t)\,Q(t)\,G^T(t)\,P_f^{-1}(t)]\,\hat{\mathbf{x}}(t)$$
$$+\,\hat{\mathbf{x}}^T(t)[\cancel{P_f^{-1}(t)}\,F(t) + \cancel{F^T(t)\,P_f^{-1}(t)} + P_f^{-1}(t)\cancel{G(t)\,Q(t)}\,G^T(t)P_f^{-1}(t)$$
$$-\,H^T(t)\,R^{-1}(t)\,H(t)\,P_f(t)]\,\hat{\mathbf{x}}(t)$$
$$-\,\hat{\mathbf{x}}^T(t)\,[\cancel{P_f^{-1}(t)}\,F(t) + P_f^{-1}(t)\,G(t)\,\cancel{Q(t)\,G^T}(t)\,P_f^{-1}(t)]\,\hat{\mathbf{x}}(t)$$

- Reduces down to

$$\frac{d}{d\tau}V[\hat{\mathbf{x}}(t)] = -\hat{\mathbf{x}}^T(t)\left[H^T(t)\,R^{-1}(t)H(t) + P_f^{-1}(t)\,G(t)\,Q(t)\,G^T(t)\,P_f^{-1}(t)\right]\hat{\mathbf{x}}(t)$$

- Clearly, if $R(t)$ is positive definite and $Q(t)$ is at least positive semi-definite, then the Lyapunov condition is satisfied $(\dot{V}(x) \le 0)$
  - So the continuous-time RTS smoother is stable
  - Similar conditions as in the Kalman filter

Example (i)

Covariance $\simeq$ $\begin{pmatrix}\text{Spectral} \\ \text{density}\end{pmatrix}$ (Δt)

$Q_k \simeq Q(t)\,\Delta t$

- Simple first-order system

$$\dot{x}(t) = f\,x(t) + w(t)$$
$$y(t) = x(t) + v(t)$$

where f is a constant, and the spectral densities of $w(t)$ and $v(t)$ are given by q and r, respectively

- The steady-state smoother covariance is given by solving

$$0 = 2[f + q\,p_f^{-1}(t)]\,p(t) - q \qquad \text{(Lyapunov)}$$

which gives

$$p = \frac{q}{2(f + q\,p_f^{-1})}$$

- The steady-state forward covariance is given by solving

$$0 = r^{-1}p_f^2 - 2fp_f - q \qquad \text{(Algebraic ricatta)}$$

which gives (positive root only)

$$p_f^{-1} = \frac{r^{-1}}{a + f}, \quad a \equiv \sqrt{f^2 + r^{-1}q}$$

Example (ii)

- Substituting this into the p equation yields

$$p = \frac{q}{2a}$$

- The steady-state backward covariance is given by solving

$$0 = q\,p_b^{-2} - 2\,f\,p_b^{-1} - r^{-1}$$

which gives (positive root only)

$$p_b^{-1} = \frac{a+f}{q}$$

- Let's verify $p^{-1} = p_f^{-1} + p_b^{-1}$; substituting quantities requires

$$\begin{aligned}\frac{2a}{q} &= \frac{r^{-1}}{a+f} + \frac{a+f}{q} \\ &= \frac{r^{-1}q + (a+f)^2}{q(a+f)} \\ &= \frac{r^{-1}q/(a+f) + a + f}{q}\end{aligned}$$

Example (iii)

- Need to show $[r^{-1}q/(a+f)] + (a+f) = 2a$, or

$$r^{-1}q + (a+f)^2 = 2a(a+f)$$

$$r^{-1}q + f^2 + r^{-1}q + 2f\sqrt{f^2 + r^{-1}q} + f^2 = 2(f^2 + r^{-1}q) + 2f\sqrt{f^2 + r^{-1}q}$$

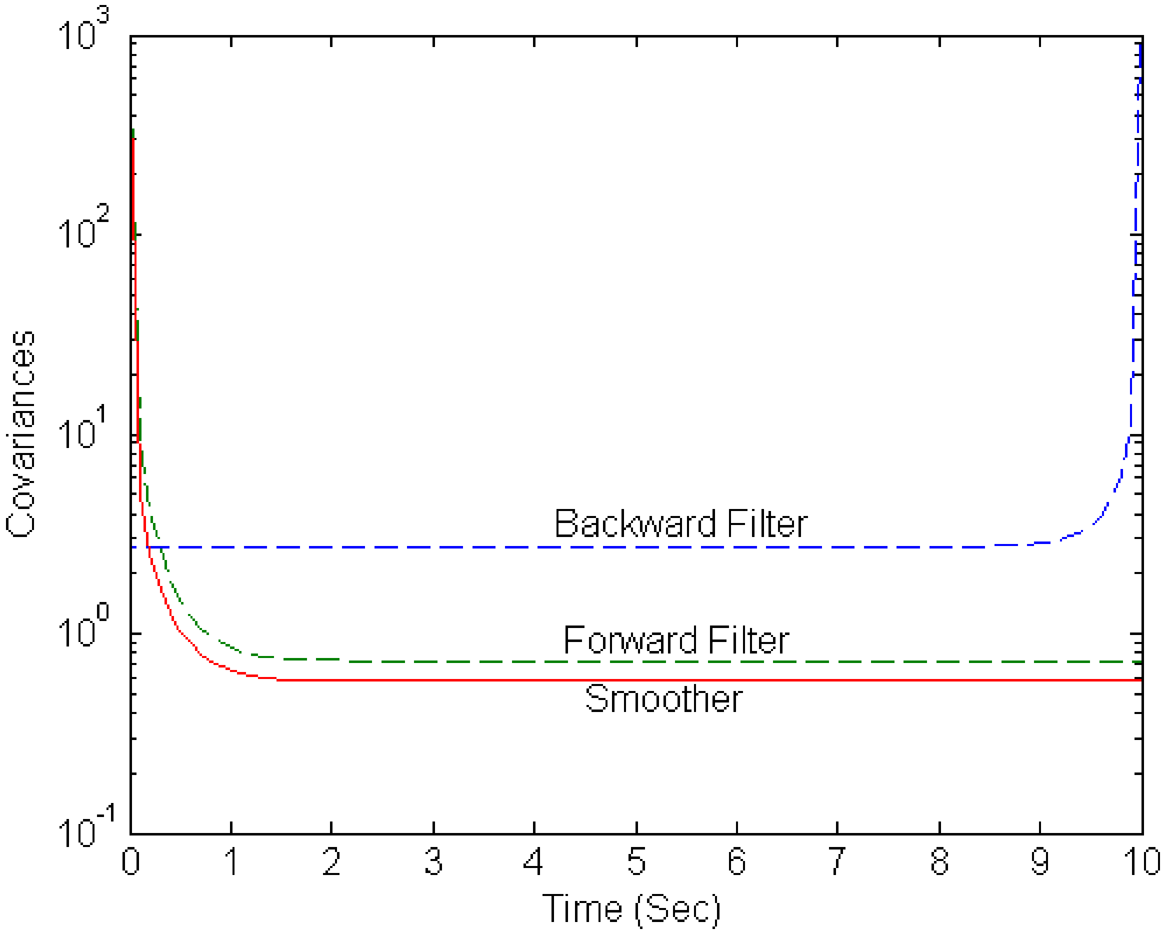
$$2(f^2 + r^{-1}q) + 2f\sqrt{f^2 + r^{-1}q} = 2(f^2 + r^{-1}q) + 2f\sqrt{f^2 + r^{-1}q}\checkmark$$

- An interesting aspect of the backward filter covariance is that it is zero when $q = 0$
  - Smoother covariance is equivalent to the forward filter covariance
  - Hence, for this case the smoother offers no improvements over the forward filter, which was discussed before
- Consider the following values: $f = -1$, $q = 2$, and $r = 1$
  - Values become $p_f = 0.7321$, $p_b = 2.7321$, and $p = 0.5774$
- An interesting case occurs when $f = 0$, which gives

$$\hat{x}(t) = \frac{1}{2}\left[\hat{x}_f(t) + \hat{x}_b(t)\right]$$

  - Using the steady-state smoother the optimal estimate of $x(t)$ is the average of the forward and backward filter estimates

Example (iv)



These plots are found by integrating the covariance equations

Note that the steady-state values match their analytical solutions

Example (v)

```
t=[0:0.1:10];
[t,pf]=ode23(@ric_forfun,t,1000);
[t,pbi]=ode23(@ric_backfun,t,0);

pb=pbi.^(-1);pb(1)=1000;
p=(pf.^(-1)+pbi).^(-1);

tt=[10:-0.1:0]'; % plots pb backwards

semilogy(tt,pb,'--',t,pf,'--',t,p)
set(gca,'Fontsize',12)
ylabel('Covariances')
xlabel('Time (Sec)')
set(gca,'xtick',[0 1 2 3 4 5 6 7 8 9 10])
text(3.8+0.4,3.5,'Backward Filter','Fontsize',12)
text(3.9+0.4,0.93,'Forward Filter','Fontsize',12)
text(4.1+0.4,0.47,'Smoother','Fontsize',12)
```

Example (vi)

```
function fun=ric_forfun(t,x);
% Forward One
f=-1;q=2;r=1;
fun=2*f*x-x^2*inv(r)+q;
```

```
function fun=ric_backfun(t,x);
% Backward One
f=-1;q=2;r=1;
fun=2*f*x-x^2*q+inv(r);
```

Nonlinear Smoothing (i)

- First step is to use the forward extended Kalman filter
  - Backward filter is not as straightforward as the extended Kalman filter though
    - This is due to the fact that we linearize the backward-time filter about the forward-time filter estimated trajectory, not the backward-time filter estimate trajectory
    - Hence, the linearized Kalman filter form will be used to derive the backward-time smoother, where the nominal (*a priori*) estimate is given by the forward-time extended Kalman filter
  - The derivation of the nonlinear smoother can be shown by using the same procedure leading to the forward/backward filters shown previously
    - However, we will only show the RTS version of this smoother, since it has clear advantages over the two filter solution

Nonlinear Smoothing (ii)

- First linearize $\mathbf{f}(\hat{\mathbf{x}}(t),\, \mathbf{u}(t),\, t)$ about $\hat{\mathbf{x}}_f(t)$
- Then, using $d\mathbf{x}/dt = -d\mathbf{x}/d\tau$ to denote the backward-time integration leads to

$$\boxed{\frac{d}{d\tau}\hat{\mathbf{x}}(t) = -\left[F(t) + K(t)\right]\left[\hat{\mathbf{x}}(t) - \hat{\mathbf{x}}_f(t)\right] - \mathbf{f}(\hat{\mathbf{x}}_f(t),\, \mathbf{u}(t),\, t)}$$

where

$$\boxed{K(t) \equiv G(t)\, Q(t)\, G^T(t)\, P_f^{-1}(t)}$$

$$F(t) \equiv \left.\frac{\partial \mathbf{f}}{\partial \mathbf{x}}\right|_{\hat{\mathbf{x}}_f(t),\, \mathbf{u}(t)}$$

- This equation must be integrated backward in time with a boundary condition of $\hat{\mathbf{x}}(T) = \hat{\mathbf{x}}_f(T)$
- Note that it is a linear equation, which allows us to use linear integration methods

Nonlinear Smoothing (iii)

- Smoother covariance is given as before with the linearized matrices

$$\boxed{\frac{d}{d\tau}P(t) = -\left[F(t) + K(t)\right]P(t) - P(t)\left[F(t) + K(t)\right]^T + G(t)\,Q(t)\,G^T(t)}$$

- This equation must also be integrated backward in time with a boundary condition of $P(T) = P_f(T)$
- Again note that it is a linear equation, which allows us to use linear integration methods

Nonlinear Smoothing (iv)

Process Noise Spectral Density

Model	$\frac{d}{d\tau}\mathbf{x}(t) = \mathbf{f}(\mathbf{x}(t),\, \mathbf{u}(t),\, t) + G(t)\,\mathbf{w}(t),\; \mathbf{w}(t) \sim N(\mathbf{0}, Q(t))$ $\tilde{\mathbf{y}}_k = \mathbf{h}(\mathbf{x}_k) + \mathbf{v}_k,\; \mathbf{v}_k \sim N(\mathbf{0}, R_k)$ Measurement noise Covariance
Forward Initialize	$\hat{\mathbf{x}}_f(t_0) = \hat{\mathbf{x}}_{f0}$ $P_{f0} = E\left\{\tilde{\mathbf{x}}_f(t_0)\,\tilde{\mathbf{x}}_f^T(t_0)\right\}$
Forward Gain	$K_{fk} = P_{fk}^- H_k^T(\hat{\mathbf{x}}_{fk}^-)[H_k(\hat{\mathbf{x}}_{fk}^-)P_{fk}^- H_k^T(\hat{\mathbf{x}}_{fk}^-) + R_k]^{-1}$ $H_k(\hat{\mathbf{x}}_{fk}^-) \equiv \left.\frac{\partial \mathbf{h}}{\partial \mathbf{x}}\right\rvert_{\hat{\mathbf{x}}_{fk}^-}$
Forward Update	$\hat{\mathbf{x}}_{fk}^+ = \hat{\mathbf{x}}_{fk}^- + K_{fk}[\tilde{\mathbf{y}}_k - \mathbf{h}(\hat{\mathbf{x}}_{fk}^-)]$ $P_{fk}^+ = [I - K_{fk}H_k(\hat{\mathbf{x}}_{fk}^-)]P_{fk}^-$
Forward Propagation	$\frac{d}{dt}\hat{\mathbf{x}}_f(t) = \mathbf{f}(\hat{\mathbf{x}}_f(t),\, \mathbf{u}(t),\, t)$ $\frac{d}{dt}P_f(t) = F(\hat{\mathbf{x}}_f(t),\, t)\,P_f(t) + P_f(t)\,F^T(\hat{\mathbf{x}}_f(t),\, t)$ $+G(t)\,Q(t)\,G^T(t)$ $F(\hat{\mathbf{x}}_f(t),\, t) \equiv \left.\frac{\partial \mathbf{f}}{\partial \mathbf{x}}\right\rvert_{\hat{\mathbf{x}}_f(t)}$

Nonlinear Smoothing (v)

Gain	$K(t) \equiv G(t)\, Q(t)\, G^T(t)\, P_f^{-1}(t)$
Smoother Covariance	$\frac{d}{d\tau} P(t) = -[F(t) + K(t)]P(t) - P(t)[F(t) + K(t)]^T$ $+G(t)\, Q(t)\, G^T(t), \quad P(T) = P_f(T)$
Smoother Estimate	$\frac{d}{d\tau}\hat{\mathbf{x}}(t) = -\,[F(t) + K(t)]\,[\hat{\mathbf{x}}(t) - \hat{\mathbf{x}}_f(t)]$ $-\mathbf{f}(\hat{\mathbf{x}}_f(t),\, \mathbf{u}(t),\, t), \quad \hat{\mathbf{x}}(T) = \hat{\mathbf{x}}_f(T)$

Example (i)

- Consider Van der Pol's equation

$$m\,\ddot{x} + 2\,c\,(x^2 - 1)\,\dot{x} + k\,x = 0$$

  - Convert to state space using $\mathbf{x} = \begin{bmatrix} x & \dot{x} \end{bmatrix}^T$

$$\dot{x}_1 = x_2$$
$$\dot{x}_2 = -2\,(c/m)(x_1^2 - 1)\,x_2 - (k/m)\,x_1$$

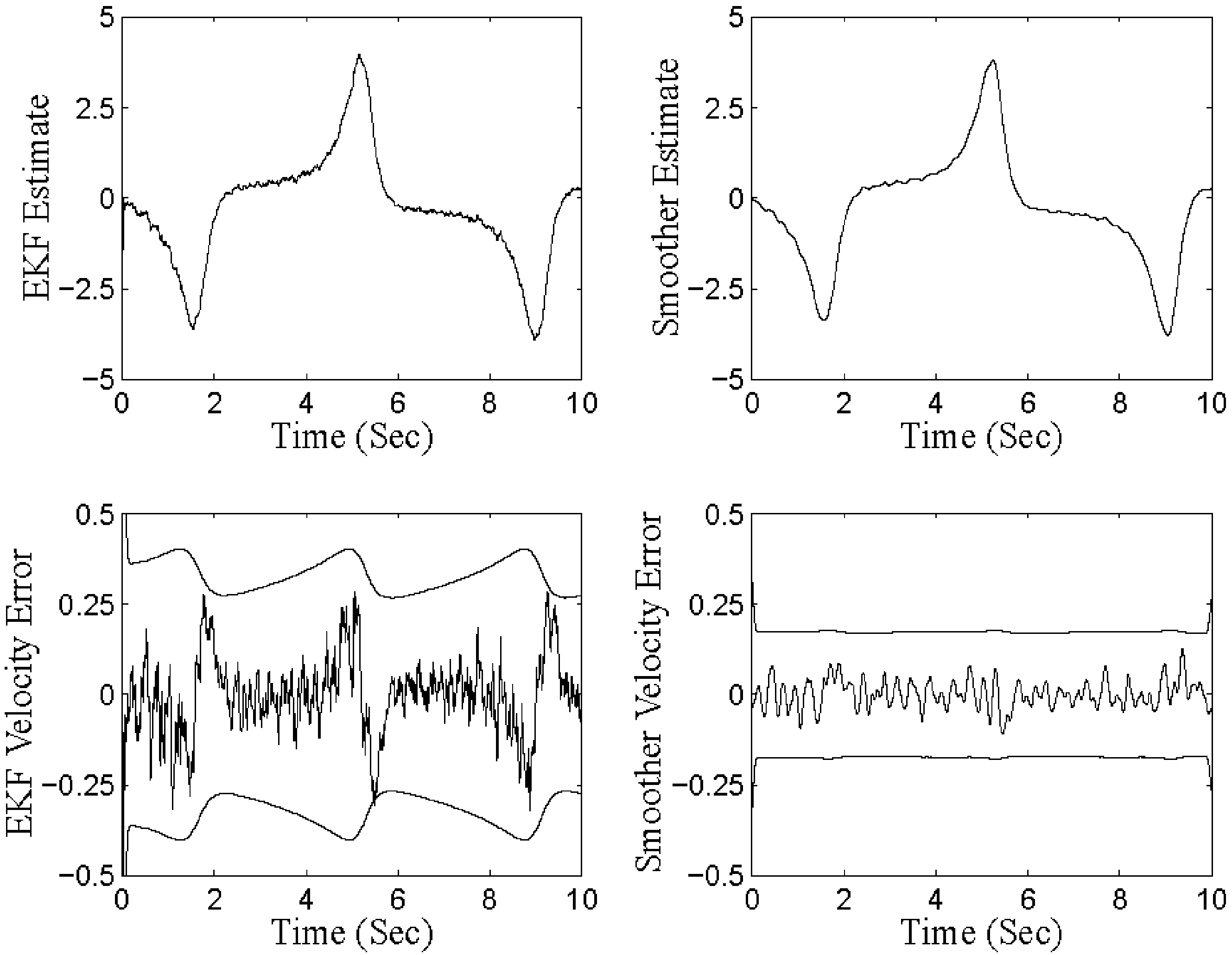
  - The measurement output is position only, so $H = [1 \;\; 0]$
  - Synthetic states are generated using $m = c = k = 1$, with an initial condition of $\mathbf{x}_0 = [1 \;\; 0]^T$
  - The sampling interval is at 0.01 second intervals and the measurement noise standard deviation is set to 0.01
  - The linearized model matrices are given by

$$F = \begin{bmatrix} 0 & 1 \\ -4\,(c/m)\,\hat{x}_1\hat{x}_2 - (k/m) & -2\,(c/m)\,(\hat{x}_1^2 - 1) \end{bmatrix}, \quad G = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

Example (ii)

- Note that no process noise (i.e., no error) is introduced into the first state
  - This is due to the fact that the first state is a kinematical relationship that is correct in theory and in practice (i.e., velocity is always the derivative of position)
- In the EKF the model parameters are assumed to be given by $m = 1$, $c = 1.5$, and $k = 1.2$, which introduces errors in the assumed system, compared to the true system
  - Overcome by tuning the process noise covariance matrix
- Since we know the truth we can tune Q until the estimate errors are within their respective 3σ bounds
  - A value of 0.2 is found to give good results
- Initial covariance is set to $P_0 = 1000\, I$

Example (iii)



Note how the covariance of the smoother "flattens" out versus the EKF

Example (iv)

```
% State and Initialize
dt=0.01;t=[0:dt:10]';m=length(t);
h=[1 0];r=0.01^2;
xe=zeros(m,2);x=zeros(m,2);p_cov=zeros(m,2);p_cov_s=zeros(m,2);
ym=zeros(m,1);
x0=[1;0];x(1,:)=x0';xe(1,:)=x0';
p0=1000*eye(2);p=p0;p_cov(1,:)=diag(p0)';
p_storep=zeros(m,4); p_storeu=zeros(m,4);
xs=zeros(m,2);

% Truth and Model Parameters
c=1;k=1;
cm=1.5;km=1.2;

% Process Noise (note: now continuous)
q=.2*[0 0;0 1];
```

Example (v)

```
% Main Forward Routine
for i=1:m-1;

% Truth and Measurements
f1=dt*polfun(x(i,:),c,k);
f2=dt*polfun(x(i,:)+0.5*f1',c,k);
f3=dt*polfun(x(i,:)+0.5*f2',c,k);
f4=dt*polfun(x(i,:)+f3',c,k);
x(i+1,:)=x(i,:)+1/6*(f1'+2*f2'+2*f3'+f4');
ym(i)=x(i,1)+sqrt(r)*randn(1);

% Kalman Update
gain=p*h'*inv(h*p*h'+r);
p=(eye(2)-gain*h)*p;
p_storeu(i,:)=[p(1,1) p(1,2) p(2,1) p(2,2)];
xe(i,:)=xe(i,:)+gain'*(ym(i)-xe(i,1));
```

Example (vi)

```
% State Propagation
f1=dt*polfun(xe(i,:),cm,km);
f2=dt*polfun(xe(i,:)+0.5*f1',cm,km);
f3=dt*polfun(xe(i,:)+0.5*f2',cm,km);
f4=dt*polfun(xe(i,:)+f3',cm,km);
xe(i+1,:)=xe(i,:)+1/6*(f1'+2*f2'+2*f3'+f4');

% Covariance Propagation
xdum=[p(1,1) p(1,2) p(2,1) p(2,2)];
f1=dt*polfun_cov(xdum,xe(i,:),cm,km,q);
f2=dt*polfun_cov(xdum+0.5*f1',xe(i,:),cm,km,q);
f3=dt*polfun_cov(xdum+0.5*f2',xe(i,:),cm,km,q);
f4=dt*polfun_cov(xdum+f3',xe(i,:),cm,km,q);
xdum=xdum+1/6*(f1'+2*f2'+2*f3'+f4');
p=[xdum(1) xdum(2);xdum(3) xdum(4)];

p_storep(i+1,:)=[p(1,1) p(1,2) p(2,1) p(2,2)];
p_cov(i+1,:)=diag(p)';

end
```

Example (vii)

```
% RTS Initialize
xs(m,:)=xe(m,:);
p_cov_s(m,:)=p_cov(m,:);
pb=[p_storep(m,1) p_storep(m,2)
    p_storep(m,3) p_storep(m,4)];

p_prop=[p_storep(m,1) p_storep(m,2)
        p_storep(m,3) p_storep(m,4)];
ddd_i=inv(h*p_prop*h'+r);
gain=p_prop*h'*ddd_i;
lam=-(h'*ddd_i*(ym(m)-xe(m,1)))';
```

Example (viii)

```
% Main Backward Routine
for i=m-1:-1:1
% Covariance
p_prop=[p_storep(i+1,1) p_storep(i+1,2);p_storep(i+1,3) p_storep(i+1,4)];p_propi=inv(p_prop);
% Backward State
f1=dt*polfun_b(xs(i+1,:),p_prop,xe(i+1,:),cm,km,q);
f2=dt*polfun_b(xs(i+1,:)+0.5*f1',p_prop,xe(i+1,:),cm,km,q);
f3=dt*polfun_b(xs(i+1,:)+0.5*f2',p_prop,xe(i+1,:),cm,km,q);
f4=dt*polfun_b(xs(i+1,:)+f3',p_prop,xe(i+1,:),cm,km,q);
xs(i,:)=xs(i+1,:)+1/6*(f1'+2*f2'+2*f3'+f4');

% Backward Covariance
xdum=[pb(1,1) pb(1,2) pb(2,1) pb(2,2)];
f1=dt*polfun_covb(xdum,xe(i+1,:),cm,km,q,p_propi);
f2=dt*polfun_covb(xdum+0.5*f1',xe(i+1,:),cm,km,q,p_propi);
f3=dt*polfun_covb(xdum+0.5*f2',xe(i+1,:),cm,km,q,p_propi);
f4=dt*polfun_covb(xdum+f3',xe(i+1,:),cm,km,q,p_propi);
xdum=xdum+1/6*(f1'+2*f2'+2*f3'+f4');pb=[xdum(1) xdum(2);xdum(3)
xdum(4)];p_cov_s(i,:)=diag(pb)';
end
```

Example (ix)

```
% 3-Sigma Outliers
sig3=p_cov.^(0.5)*3;
sig3_s=p_cov_s.^(0.5)*3;

% Plot Results
subplot(221)
plot(t,xe(:,2))
set(gca,'Fontsize',12);
axis([0 10 -5 5]);set(gca,'Xtick',[0 2 4 6 8 10]);set(gca,'Ytick',[-5 -2.5 0 2.5 5]);
xlabel('Time (Sec)')
ylabel('EKF Estimate')

subplot(222)
plot(t,xs(:,2))
set(gca,'Fontsize',12);
axis([0 10 -5 5]);set(gca,'Xtick',[0 2 4 6 8 10]);set(gca,'Ytick',[-5 -2.5 0 2.5 5]);
xlabel('Time (Sec)')
ylabel('Smoother Estimate')
```

Example (x)

```
subplot(223)
plot(t,xe(:,2)-x(:,2),t,sig3(:,2),t,-sig3(:,2))
set(gca,'Fontsize',12);
axis([0 10 -0.5 0.5]);set(gca,'Xtick',[0 2 4 6 8 10]);set(gca,'Ytick',[-0.5 -0.25 0 0.25 0.5]);
xlabel('Time (Sec)')
ylabel('EKF Velocity Error')

subplot(224)
plot(t,xs(:,2)-x(:,2),t,sig3_s(:,2),t,-sig3_s(:,2))
set(gca,'Fontsize',12);
axis([0 10 -0.5 0.5]);set(gca,'Xtick',[0 2 4 6 8 10]);set(gca,'Ytick',[-0.5 -0.25 0 0.25 0.5]);
xlabel('Time (Sec)')
ylabel('Smoother Velocity Error')
```

Example (xi)

```
function f=polfun(x,c,k)

% Function Routine for Van der Pol's Equation

f=[x(2);-2*c*(x(1)^2-1)*x(2)-k*x(1)];
```

```
function f=polfun_cov(x,xe,c,k,q)

% Function Routine for Van der Pol's Covariance Equation in Forward Integration

% State Matrix
fpart=[0 1;-4*c*xe(1)*xe(2)-k -2*c*(xe(1)^2-1)];

% Covariance
p=[x(1) x(2);x(3) x(4)];
pdot=fpart*p+p*fpart'+q;
f=[pdot(1,1);pdot(1,2);pdot(2,1);pdot(2,2)];
```

Example (xii)

```
function f=polfun_b(x,p,xf,c,k,q)

% Function Routine for Van der Pol's Equation in Backwards Integration

fpart=[0 1;-4*c*xf(1)*xf(2)-k -2*c*(xf(1)^2-1)];
f=-[xf(2);-2*c*(xf(1)^2-1)*xf(2)-k*xf(1)]-[q*inv(p)+fpart]*([x(1);x(2)]-[xf(1);xf(2)]);
```

```
function f=polfun_covb(x,xe,c,k,q,pfi)

% Function Routine for Van der Pol's Covariance Equation in Backward Integration

% State Matrix
fpart=[0 1;-4*c*xe(1)*xe(2)-k -2*c*(xe(1)^2-1)];

% Covariance
p=[x(1) x(2);x(3) x(4)];
pdot=-[fpart+q*pfi]*p-p*[fpart+q*pfi]'+q;
f=[pdot(1,1);pdot(1,2);pdot(2,1);pdot(2,2)];
```

Discrete Measurements (i)

- Handle discrete-time measurement through an “adjoint variable” $\boldsymbol{\lambda}$
  - The propagation equations are given by

$$\boxed{\frac{d}{d\tau}\boldsymbol{\lambda}(t) = F^T(t)\,\boldsymbol{\lambda}(t)}$$

$$\boxed{\frac{d}{d\tau}\Lambda(t) = F^T(t)\,\Lambda(t) + \Lambda(t)\,F(t)}$$

  where Λ is the covariance of $\boldsymbol{\lambda}$

  - The backward updates are given by

$$\boxed{\boldsymbol{\lambda}_k^- = \left[I - H_k^T(\hat{\mathbf{x}}_{f\,k}^-)\,K_{f\,k}^T\right]\boldsymbol{\lambda}_k^+ - H_k^T(\hat{\mathbf{x}}_{f\,k}^-)\,D_{f\,k}^{-1}\left[\tilde{\mathbf{y}}_k - \mathbf{h}_k(\hat{\mathbf{x}}_{f\,k}^-)\right]}$$

$$\boxed{\Lambda_k^- = \left[I - K_{f\,k}\,H_k(\hat{\mathbf{x}}_{f\,k}^-)\right]^T\Lambda_k^+\left[I - K_{f\,k}\,H_k(\hat{\mathbf{x}}_{f\,k}^-)\right] + H_k^T(\hat{\mathbf{x}}_{f\,k}^-)\,D_{f\,k}^{-1}\,H_k(\hat{\mathbf{x}}_{f\,k}^-)}$$

where

$$D_{f\,k} \equiv H_k(\hat{\mathbf{x}}_{f\,k}^-)\, P_{f\,k}^- H_k^T(\hat{\mathbf{x}}_{f\,k}^-) + R_k$$

- Note that in this formulation $\boldsymbol{\lambda}_k^-$ is used to denote the backward update just before the measurement is processed
- If $T \equiv t_N$ is an observation time, then the boundary conditions are given by

$$\boldsymbol{\lambda}_N^- = -H_N^T(\hat{\mathbf{x}}_{f\,N}^-)\, D_{f\,N}^{-1} \left[\tilde{\mathbf{y}}_N - \mathbf{h}_N(\hat{\mathbf{x}}_{f\,N}^-)\right]$$

$$\Lambda_N^- = H_T N^T(\hat{\mathbf{x}}_{f\,N}^-)\, D_{f\,N}^{-1} H_N(\hat{\mathbf{x}}_{f\,N}^-)$$

- If T is not an observation time, then the boundary conditions are zero
- Smoother state and covariance can be constructed via either the propagated or updated values of

$$\hat{\mathbf{x}}_k = \hat{\mathbf{x}}_{f\,k}^\pm - P_{f\,k}^\pm \boldsymbol{\lambda}_k^\pm$$

$$P_k = P_{f\,k}^\pm - P_{f\,k}^\pm \Lambda_k^\pm P_{f\,k}^\pm$$

Optimal Control Approach (i)

- Helps explain where the adjoint variable comes from
  - We'll only focus on the continuous-time version
    - Can derive the full nonlinear continuous-discrete version as well
  - Want to minimize the following loss function

$$J[\mathbf{w}(t)] = \frac{1}{2}\int_{t_0}^{t_N} \{[\tilde{\mathbf{y}}(t) - H(t)\,\mathbf{x}(t)]^T R^{-1}(t)\,[\tilde{\mathbf{y}}(t) - H(t)\,\mathbf{x}(t)]$$

$$+\mathbf{w}^T(t)\,Q^{-1}(t)\mathbf{w}(t)\}\;dt + \frac{1}{2}[\hat{\mathbf{x}}_f(t_0) - \mathbf{x}(t_0)]^T P_f^{-1}(t_0)[\hat{\mathbf{x}}_f(t_0) - \mathbf{x}(t_0)]$$

subject to the dynamic constraint

$$\frac{d}{dt}\mathbf{x}(t) = F(t)\,\mathbf{x}(t) + B(t)\,\mathbf{u}(t) + G(t)\,\mathbf{w}(t)$$

  - Note that we are treating $\mathbf{w}(t)$ as a "control input" here
    - It's really a random variable, not a deterministic one, which is assumed here
    - Still, we'll proceed with this assumption

Optimal Control Approach (ii)

- Use a Lagrange multiplier approach to derive the following two-point boundary value problem (TPBVP)

$$\frac{d}{dt}\hat{\mathbf{x}}(t) = F(t)\,\hat{\mathbf{x}}(t) + B(t)\,\mathbf{u}(t) + G(t)\,\mathbf{w}(t) \qquad \text{(1a)}$$

$$\frac{d}{dt}\boldsymbol{\lambda}(t) = -F^T(t)\,\boldsymbol{\lambda}(t) - H^T(t)\,R^{-1}(t)\,H(t)\,\hat{\mathbf{x}}(t) + H^T(t)\,R^{-1}(t)\,\tilde{\mathbf{y}}(t) \qquad \text{(1b)}$$

$$\mathbf{w}(t) = -Q(t)\,G^T(t)\,\boldsymbol{\lambda}(t) \qquad \text{(1c)}$$

where $\boldsymbol{\lambda}(t)$ is the vector of Lagrange multipliers

- The boundary conditions are given by

$$\boldsymbol{\lambda}(T) = \mathbf{0}$$

$$\hat{\mathbf{x}}(T) = \hat{\mathbf{x}}_f(T)$$

- Note that because of the boundary conditions, a forward-backward solution must be employed
  - Also note how $\boldsymbol{\lambda}(t)$ is related to the control input $\mathbf{w}(t)$
  - The vector $\boldsymbol{\lambda}(t)$ tells us how much “control effort” is required

Optimal Control Approach (iii)

- Substitute (1c) into (1a) to obtain

$$\frac{d}{dt}\hat{\mathbf{x}}(t) = F(t)\,\hat{\mathbf{x}}(t) + B(t)\,\mathbf{u}(t) - G(t)\,Q(t)\,G^T(t)\boldsymbol{\lambda}(t) \tag{2a}$$

$$\frac{d}{dt}\boldsymbol{\lambda}(t) = -F^T(t)\,\boldsymbol{\lambda}(t) - H^T(t)\,R^{-1}(t)\,H(t)\,\hat{\mathbf{x}}(t) + H^T(t)\,R^{-1}(t)\,\tilde{\mathbf{y}}(t) \tag{2b}$$

- Assume that the form of the solution is given by

$$\hat{\mathbf{x}}(t) = \hat{\mathbf{x}}_f(t) - P_f(t)\,\boldsymbol{\lambda}(t) \tag{3}$$

for some matrix $P_f(t)$

- Comparing this to the boundary conditions requires $\boldsymbol{\lambda}(T) = \mathbf{0}$ and

$$\boldsymbol{\lambda}(t_0) = P_f^{-1}(t_0)\left[\hat{\mathbf{x}}_f(t_0) - \hat{\mathbf{x}}(t_0)\right]$$

  - Not really needed for anything though

- Take the time derivative of Eq. (3) to obtain

$$\frac{d}{dt}\hat{\mathbf{x}}(t) = \frac{d}{dt}\hat{\mathbf{x}}_f(t) - \left[\frac{d}{dt}P_f(t)\right]\boldsymbol{\lambda}(t) - P_f(t)\left[\frac{d}{dt}\boldsymbol{\lambda}(t)\right]$$

Optimal Control Approach (iv)

- Substitute Eq. (2) to obtain

$$F(t)\,\hat{\mathbf{x}}(t) + B(t)\,\mathbf{u}(t) - G(t)\,Q(t)\,G^T(t)\boldsymbol{\lambda}(t)$$
$$-\frac{d}{dt}\hat{\mathbf{x}}_f(t) + \left[\frac{d}{dt}P_f(t)\right]\boldsymbol{\lambda}(t) - P_f(t)\,F^T(t)\,\boldsymbol{\lambda}(t)$$
$$-P_f(t)\,H^T(t)\,R^{-1}(t)\,H(t)\,\hat{\mathbf{x}}(t) + P_f(t)\,H^T(t)\,R^{-1}(t)\,\tilde{\mathbf{y}}(t) = \mathbf{0}$$

- Substitute Eq. (3) and collect terms to give

$$\left[\frac{d}{dt}P_f(t) - F(t)\,P_f(t) - P_f(t)\,F^T(t) + P_f(t)\,H^T(t)\,R^{-1}(t)\,H(t)\,P_f(t)\right.$$
$$\left. - G(t)\,Q(t)\,G^T(t)\right]\boldsymbol{\lambda}(t) + F(t)\,\hat{\mathbf{x}}_f(t) + B(t)\,\mathbf{u}(t)$$
$$+ P_f(t)\,H^T(t)\,R^{-1}(t)[\tilde{\mathbf{y}}(t) - H(t)\,\hat{\mathbf{x}}_f(t)] - \frac{d}{dt}\hat{\mathbf{x}}_f(t) = \mathbf{0}$$

- Avoiding the trivial solution of $\boldsymbol{\lambda}(t) = \mathbf{0}$ gives

$$\frac{d}{dt}P_f(t) = F(t)\,P_f(t) + P_f(t)\,F^T(t) - P_f(t)\,H^T(t)\,R^{-1}(t)\,H(t)\,P_f(t) \;+ G(t)\,Q(t)\,G^T(t)$$

$$\frac{d}{dt}\hat{\mathbf{x}}_f(t) = F(t)\,\hat{\mathbf{x}}_f(t) + B(t)\,\mathbf{u}(t) + K_f(t)[\tilde{\mathbf{y}}(t) - H(t)\,\hat{\mathbf{x}}_f(t)]$$

where

$$K_f(t) \equiv P_f(t)\,H^T(t)\,R^{-1}(t)$$

- This is exactly the forward-time Kalman filter!
- Solving Eq. (3) for $\boldsymbol{\lambda}(t)$ and substituting into Eq. (2a) gives

$$\frac{d}{dt}\hat{\mathbf{x}}(t) = F(t)\,\hat{\mathbf{x}}(t) + B(t)\,\mathbf{u}(t) + G(t)\,Q(t)\,G^T(t)\,P_f^{-1}(t)\,[\hat{\mathbf{x}}(t) - \hat{\mathbf{x}}_f(t)]$$

- This is exactly the RTS smoother equation!
- A similar approach can be used to derive the discrete-time and nonlinear RTS forms as well