# Statistical Orbit Determination

Lecture 5

January 31, 2018

State Noise Compensation

# Filter Saturation

# Filter Saturation

As a sufficiently large amount of observations are processed, the covariance asymptotically approaches zero

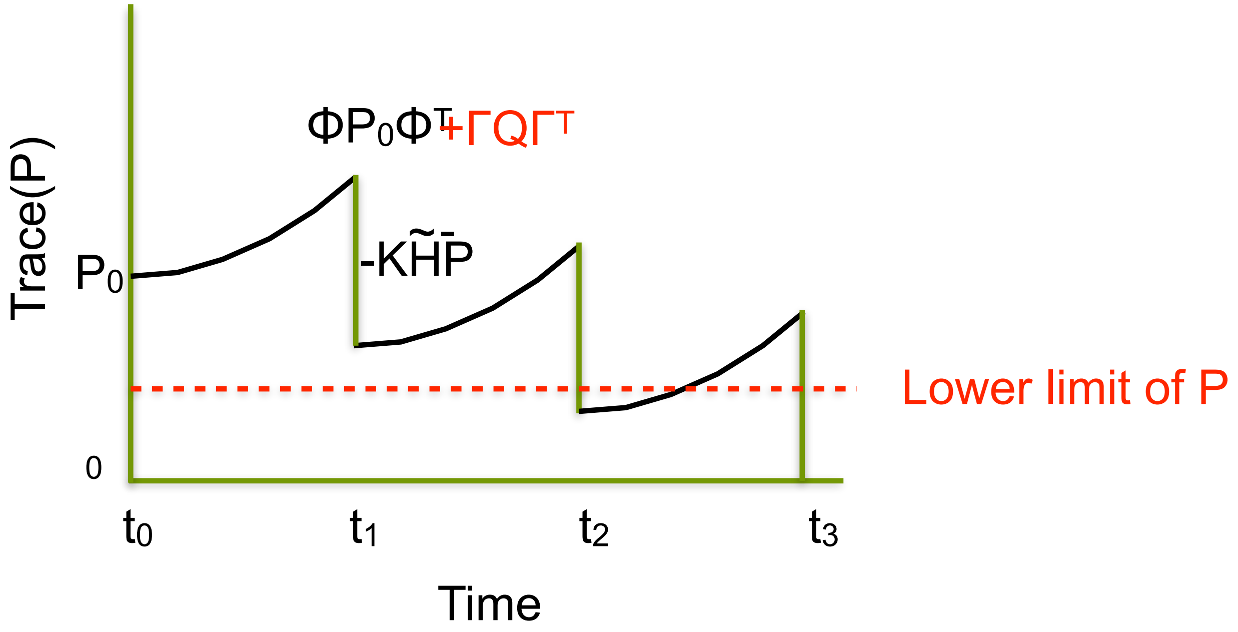
$$P \rightarrow 0 \Rightarrow K \rightarrow 0$$

$$\hat{x} = \bar{x} - \cancel{K(y - H\bar{x})}$$

Filter becomes insensitive to further observations

# Saturation and State Noise Compensation

- Filter thinks it has already determined the state with much confidence
- Becomes insensitive to observations




- SNC: Compensate for modeling errors by adding process noise to acceleration equations
- Prevents saturation by adding process noise uncertainty to covariance time update

CCAR

ORCCA

# State Noise Compensation

# State Noise Compensation Algorithm

Differential equations of motion given by

$$\dot{\vec{X}}(t) = F(\vec{X}, t) + B(t)\vec{u}(t)$$

Assume u(t) is white Gaussian process noise

$$E[\vec{u}(t)] = 0, E[\vec{u}(t_i)\vec{u}(t_j)^T] = Q(t_i)\delta_{ij}$$

Process noise covariance matrix

B(t) maps process noise into state derivatives

Assume $Q(t_i)$ is constant

# State Noise Compensation Algorithm

Linearize Diff. Eq. about reference trajectory:

$$\dot{\vec{X}}(t) = \dot{\vec{X}}^*(t) + \frac{\partial F(\vec{X}(t))}{\partial \vec{X}(t)}^* \left( \vec{X}(t) - \vec{X}^*(t) \right) + B(t)\vec{u}(t)$$

Define:

$$\vec{x}(t) \equiv \vec{X}(t) - \vec{X}^*(t) \qquad A(t) \equiv \frac{\partial F(\vec{X}(t))}{\partial \vec{X}(t)}^*$$

Propagation of state deviation including process noise (for a linear system):

$$\dot{\vec{x}}(t) = A(t)\vec{x}(t) + B(t)\vec{u}(t)$$

$$\vec{x}_{k+1} = \Phi(t_{k+1}, t_k)\vec{x}_k + \Gamma(t_{k+1}, t_k)\vec{u}_k$$

# State Noise Compensation Algorithm

Time update for estimation error covariance matrix:

$$\bar{P}_{k+1} = \Phi(t_{k+1}, t_k) P_k \Phi^T(t_{k+1}, t_k) + \Gamma(t_{k+1}, t_k) Q \Gamma^T(t_{k+1}, t_k)$$

where $\Gamma(t_{k+1}, t_k)$ is given by:

Evaluate Γ

$$\Gamma(t_{k+1}, t_k) = \int_{t_k}^{t_{k+1}} \Phi(t_{k+1}, \tau) B(\tau) d\tau$$

Assume process noise only added to accelerations

$$\dot{\vec{x}} = \begin{bmatrix} \dot{x} & \dot{y} & \dot{z} & \ddot{x} & \ddot{y} & \ddot{z} \end{bmatrix}^T, \vec{u} = \begin{bmatrix} u_{\ddot{X}} & u_{\ddot{Y}} & u_{\ddot{Z}} \end{bmatrix}^T, B = \begin{bmatrix} 0_{3x3} \\ I_{3x3} \end{bmatrix}$$

B(τ) constant, $\Phi(t_{k+1}, \tau)$ given by:

$$\Phi(t_{k+1}, \tau) = \begin{bmatrix} \frac{\partial \vec{r}(t_{k+1})}{\partial \vec{r}(\tau)} & \frac{\partial \vec{r}(t_{k+1})}{\partial \dot{\vec{r}}(\tau)} \\ \frac{\partial \dot{\vec{r}}(t_{k+1})}{\partial \vec{r}(\tau)} & \frac{\partial \dot{\vec{r}}(t_{k+1})}{\partial \dot{\vec{r}}(\tau)} \end{bmatrix}$$

Simplify notation:

$$\Phi(t_{k+1}, \tau) = \begin{bmatrix} \phi_1(\tau) & \phi_2(\tau) \\ \phi_3(\tau) & \phi_4(\tau) \end{bmatrix}$$

# State Noise Compensation Algorithm

$$\Phi(t_{k+1}, \tau)B(\tau) = \begin{bmatrix} \phi_1(\tau) & \phi_2(\tau) \\ \phi_3(\tau) & \phi_4(\tau) \end{bmatrix} \begin{bmatrix} 0_{3x3} \\ I_{3x3} \end{bmatrix} = \begin{bmatrix} \phi_2(\tau) \\ \phi_4(\tau) \end{bmatrix}$$

From definition of Γ:

$$\Gamma(t_{k+1}, t_k) = \int_{t_k}^{t_{k+1}} \begin{bmatrix} \phi_2(\tau) \\ \phi_4(\tau) \end{bmatrix} d\tau$$

How to determine Γ?
Can numerically integrate
or
approximate the integral

# State Noise Compensation Algorithm

## Expand state transition sub-matrix:

$$\phi_2(\tau) = \frac{\partial \vec{r}(t_{k+1})}{\partial \dot{\vec{r}}(\tau)} = \begin{bmatrix} t_{k+1} - \tau & \cancel{\frac{\partial X(t_{k+1})}{\partial \dot{Y}(\tau)}} & \cancel{\frac{\partial X(t_{k+1})}{\partial \dot{Z}(\tau)}} \\ \frac{\partial Y(t_{k+1})}{\partial \dot{X}(\tau)} & \frac{\partial Y(t_{k+1})}{\partial \dot{Y}(\tau)} & \frac{\partial Y(t_{k+1})}{\partial \dot{Z}(\tau)} \\ \frac{\partial Z(t_{k+1})}{\partial \dot{X}(\tau)} & \frac{\partial Z(t_{k+1})}{\partial \dot{Y}(\tau)} & \frac{\partial Z(t_{k+1})}{\partial \dot{Z}(\tau)} \end{bmatrix}$$

Assume $\dot{X}(\tau)$ is constant over time btw. observations

(dense tracking data)

## Approximate:

$$X(t_{k+1}) = \dot{X}(\tau)\,[t_{k+1} - \tau] \qquad \frac{\partial X(t_{k+1})}{\partial \dot{X}(\tau)} = t_{k+1} - \tau$$

Assumed $t_{k+1} - \tau$ is small, and $X(t_{k+1})$ is negligibly affected by $\dot{Y}(\tau)$ and $\dot{Z}(\tau)$

# State Noise Compensation Algorithm

Applying same assumptions to $Y(t_{k+1})$ and $Z(t_{k+1})$:

$$\phi_2(\tau) \cong \begin{bmatrix} t_{k+1}-\tau & 0 & 0 \\ 0 & t_{k+1}-\tau & 0 \\ 0 & 0 & t_{k+1}-\tau \end{bmatrix}$$

$$\Gamma(t_{k+1}, t_k) = \int_{t_k}^{t_{k+1}} \begin{bmatrix} \phi_2(\tau) \\ \phi_4(\tau) \end{bmatrix} d\tau$$

Now approximate other portion of $\Gamma$:

$$\phi_4(t_{k+1}, \tau) = \frac{\partial \dot{\vec{r}}(t_{k+1})}{\partial \dot{\vec{r}}(\tau)}$$

Again, assumed velocity is constant over short time intervals

$$\phi_4(t_{k+1}\tau) \cong I$$

3x3 Identity Matrix

CCAR

ORCCA

# State Noise Compensation Algorithm

Substitute $\Phi_2$ and $\Phi_4$ into $\Gamma$ integral:

$$\Gamma(t_{k+1}, t_k) = \int_{t_k}^{t_{k+1}} \begin{bmatrix} t_{k+1}-\tau & 0 & 0 \\ 0 & t_{k+1}-\tau & 0 \\ 0 & 0 & t_{k+1}-\tau \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} d\tau$$

Define: $\Delta t \equiv t_{k+1} - t_k$

$$\Gamma(t_{k+1}, t_k) = \begin{bmatrix} \frac{\Delta t^2}{2} & 0 & 0 \\ 0 & \frac{\Delta t^2}{2} & 0 \\ 0 & 0 & \frac{\Delta t^2}{2} \\ \Delta t & 0 & 0 \\ 0 & \Delta t & 0 \\ 0 & 0 & \Delta t \end{bmatrix} = \Delta t \begin{bmatrix} \frac{\Delta t}{2} I_{3x3} \\ I_{3x3} \end{bmatrix}$$

Map process noise at time $t_k$ into state at time $t_{k+1}$

# State Noise Compensation Algorithm

Assume Q is band diagonal

$$Q = \begin{bmatrix} \sigma^2_{\ddot{X}} & 0 & 0 \\ 0 & \sigma^2_{\ddot{Y}} & 0 \\ 0 & 0 & \sigma^2_{\ddot{Z}} \end{bmatrix}$$

Process noise contribution to estimation error covariance at time update $t_{k+1}$ is given by:

$$\Gamma(t_{k+1}, t_k) Q \Gamma^T(t_{k+1}, t_k) = \Delta t^2 \begin{bmatrix} \frac{\Delta t}{2} I_{3x3} \\ I_{3x3} \end{bmatrix} \begin{bmatrix} \sigma^2_{\ddot{X}} & 0 & 0 \\ 0 & \sigma^2_{\ddot{Y}} & 0 \\ 0 & 0 & \sigma^2_{\ddot{Z}} \end{bmatrix} \begin{bmatrix} \frac{\Delta t}{2} I_{3x3} & I_{3x3} \end{bmatrix}$$

# State Noise Compensation Algorithm

$$\Gamma(t_{k+1},t_k)Q\Gamma^T(t_{k+1},t_k) = \Delta t^2 \begin{bmatrix} \frac{\Delta t^2}{4}\sigma^2_{\ddot{X}} & 0 & 0 & \frac{\Delta t}{2}\sigma^2_{\ddot{X}} & 0 & 0 \\ 0 & \frac{\Delta t^2}{4}\sigma^2_{\ddot{Y}} & 0 & 0 & \frac{\Delta t}{2}\sigma^2_{\ddot{Y}} & 0 \\ 0 & 0 & \frac{\Delta t^2}{4}\sigma^2_{\ddot{Z}} & 0 & 0 & \frac{\Delta t}{2}\sigma^2_{\ddot{Z}} \\ \frac{\Delta t}{2}\sigma^2_{\ddot{X}} & 0 & 0 & \sigma^2_{\ddot{X}} & 0 & 0 \\ 0 & \frac{\Delta t}{2}\sigma^2_{\ddot{Y}} & 0 & 0 & \sigma^2_{\ddot{Y}} & 0 \\ 0 & 0 & \frac{\Delta t}{2}\sigma^2_{\ddot{Z}} & 0 & 0 & \sigma^2_{\ddot{Z}} \end{bmatrix}$$

- This represents contribution to estimation error covariance from uncertainty in accelerations acting on the system
- Magnitude of σ's should correspond to magnitude of acceleration uncertainty
    - Might be easier to define in RIC frame (ex: drag)

$$Q_{ECI} = \gamma^T Q_{RIC} \gamma$$

- If magnitudes unknown, determine σ's through trial and error

# SNC Flow Chart