The Abdus Salam
International Centre
for Theoretical Physics

United Nations
Educational, Scientific and
Cultural Organization

IAEA
International Atomic Energy Agency

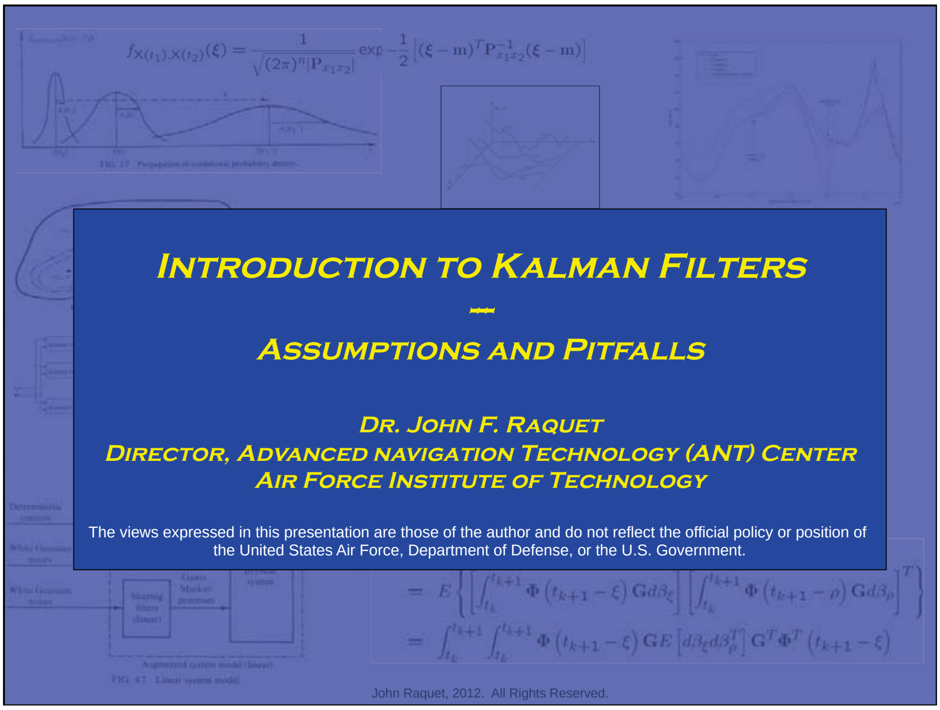
2458-2

Workshop on GNSS Data Application to Low Latitude Ionospheric Research

6 - 17 May 2013

Introduction to Kalman Filters

John F. Raquet
Air Force Institute of Technology
USA



INTRODUCTION TO KALMAN FILTERS

ASSUMPTIONS AND PITFALLS

DR. JOHN F. RAQUET
DIRECTOR, ADVANCED NAVIGATION TECHNOLOGY (ANT) CENTER
AIR FORCE INSTITUTE OF TECHNOLOGY

Kalman Filtering Overview

- Kalman filtering is an estimation approach that can be applied to navigation
  - Many other application areas
- Concepts to be covered
  - Information describing the system
    - State vector
    - Covariance matrix
  - Propagating state and covariance forward in time
  - Using measurements to update the state and covariance
- Assumptions/Limitations

Kalman Filtering: Information Describing the System (1/2)

- State vector
  - Set of variables that
    - Describe everything you want to know about the system
    - Include all of the information needed to determine how the system changes over time
    - Describe systematic errors in the measurements (anything that's not "noise")
  - Example: Hot air baloon

$$\mathbf{x} = \begin{bmatrix} x \\ y \\ z \\ \dot{x} \\ \dot{y} \\ \dot{z} \end{bmatrix}$$

$x, y, z =$ ENU position of balloon

$\dot{x}, \dot{y}, \dot{z} =$ ENU velocity of balloon

  - Does this describe what we want to know?
  - Does this describe how the system changes over time?
  - Would this be a good state vector for a fighter aircraft?
  - Altitude estimation example

Kalman Filtering: Information Describing the System (2/2)

- Covariance matrix
  - The covariance matrix basically describes how well the state is known
    - If the system only gives a state output, it's not that useful.
    - If it outputs the state and tells how accurate it is, then you have information that you can confidently act upon.
    - Hot air balloon example: the system state tells me that I'm 300 m above the ground descending at a rate of 10 m/sec.
      - Need to know covariance matrix as well.
        - » Case 1: Position accuracy = 10 m 1- σ, velocity accuracy = 1 m/sec 1-σ → probably not in danger until ~30 seconds
        - » Case 2: Position accuracy = 400 m 1- σ, velocity accuracy = 15 m/sec 1- σ → you could hit the ground any second!
  - How to interpret covariance matrix $P = E[(\underbrace{\hat{\mathbf{x}}}_{\text{Estimated state}} - \underbrace{\mathbf{x}}_{\text{True state}})(\hat{\mathbf{x}} - \mathbf{x})^T]$
    - Diagonal terms are the error variances of the estimated states
    - Off-diagonal terms are cross-covariances, describing the correlations of the errors between the states

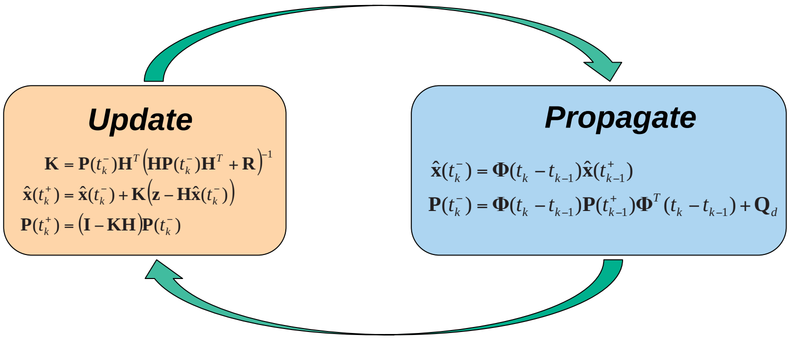
Kalman Filtering: Propagating Covariance and State Forward in Time

- State vector and covariance matrix can be propagated forward in time
  - If you know the current state estimate, you can determine the state estimate at a point in the future
  - If you know the current covariance matrix, you can determine the covariance matrix at a point in the future
  - Information about how the state and covariance changes over time is given in
    - Dynamics matrix $\mathbf{F}$: $\dot{\mathbf{x}} = \mathbf{F}\mathbf{x}$
    - State transition matrix $\mathbf{\Phi}$: $\mathbf{x}(t_1) = \mathbf{\Phi}(t_1 - t_0)\mathbf{x}(t_0)$
  - When propagating covariance forward in time, *process noise* is added to account for
    - Unmodeled dynamics
    - Unmodeled system inputs
    - Anything else that decreases the ability to predict the future state using the current state
  - Process noise increases uncertainty (i.e., larger covariance values)

Kalman Filtering: Measurement Updates

- A measurement gives information about the state values
  - Examples: GPS pseudorange (for position or clock bias) or Doppler (for velocity or clock drift)
- Effects of a measurement update
  - State values are adjusted to reflect the measurement
  - Covariance matrix is adjusted to reflect how well the state is known, now that the measurement is available
    - Measurements always decrease uncertainty (i.e., smaller covariance values)
- Measurement noise
  - Description of how precise the measurement is
  - The effect of measurement on state and covariance determined by tradeoff between
    - Measurement noise (how good the measurement is)
    - Covariance matrix (how well the state is known at this point)
- Relationship between measurement and states given by $\mathbf{H}$ matrix (same as least-squares)

The Kalman Filter Iteration



Update

$$\mathbf{K} = \mathbf{P}(t_k^-)\mathbf{H}^T\left(\mathbf{H}\mathbf{P}(t_k^-)\mathbf{H}^T + \mathbf{R}\right)^{-1}$$

$$\hat{\mathbf{x}}(t_k^+) = \hat{\mathbf{x}}(t_k^-) + \mathbf{K}\left(\mathbf{z} - \mathbf{H}\hat{\mathbf{x}}(t_k^-)\right)$$

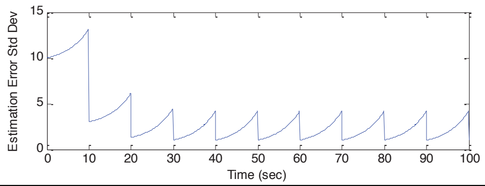
$$\mathbf{P}(t_k^+) = \left(\mathbf{I} - \mathbf{K}\mathbf{H}\right)\mathbf{P}(t_k^-)$$

Propagate

$$\hat{\mathbf{x}}(t_k^-) = \mathbf{\Phi}(t_k - t_{k-1})\hat{\mathbf{x}}(t_{k-1}^+)$$

$$\mathbf{P}(t_k^-) = \mathbf{\Phi}(t_k - t_{k-1})\mathbf{P}(t_{k-1}^+)\mathbf{\Phi}^T(t_k - t_{k-1}) + \mathbf{Q}_d$$

Example of Estimation Error Over Time



Measurement Model

Linear Measurement Model

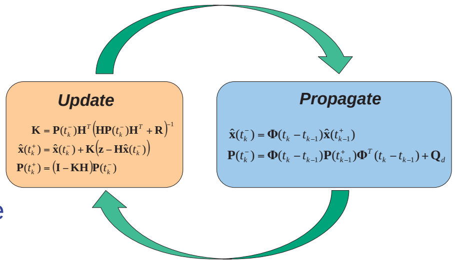
$$\mathbf{z} = \mathbf{H}\mathbf{x} + \mathbf{v}$$

measurement

sensitivity matrix

state

meas noise



Update

$$\mathbf{K} = \mathbf{P}(t_k^-)\mathbf{H}^T\left(\mathbf{H}\mathbf{P}(t_k^-)\mathbf{H}^T + \mathbf{R}\right)^{-1}$$

$$\hat{\mathbf{x}}(t_k^+) = \hat{\mathbf{x}}(t_k^-) + \mathbf{K}\left(\mathbf{z} - \mathbf{H}\hat{\mathbf{x}}(t_k^-)\right)$$

$$\mathbf{P}(t_k^+) = \left(\mathbf{I} - \mathbf{K}\mathbf{H}\right)\mathbf{P}(t_k^-)$$

Propagate

$$\hat{\mathbf{x}}(t_k^-) = \mathbf{\Phi}(t_k - t_{k-1})\hat{\mathbf{x}}(t_{k-1}^+)$$

$$\mathbf{P}(t_k^-) = \mathbf{\Phi}(t_k - t_{k-1})\mathbf{P}(t_{k-1}^+)\mathbf{\Phi}^T(t_k - t_{k-1}) + \mathbf{Q}_d$$

Measurement noise is described by the measurement noise covariance matrix:

$$\mathbf{R} = E[\mathbf{v}\mathbf{v}^T] = \begin{bmatrix} E[v_1^2] & E[v_1 v_2] & \cdots & E[v_1 v_n] \\ E[v_1 v_2] & E[v_2^2] & & \\ \vdots & & \ddots & \\ E[v_1 v_n] & & & E[v_n^2] \end{bmatrix} = \begin{bmatrix} \sigma_1^2 & \sigma_{12} & \cdots & \sigma_{1n} \\ \sigma_{12} & \sigma_2^2 & & \\ \vdots & & \ddots & \\ \sigma_{1n} & & & \sigma_n^2 \end{bmatrix}$$

- Key assumptions (in semi-nontechnical language)
  - Measurement errors $\mathbf{v}$ are Gaussian (follow a “bell curve”)
  - Measurement errors $\mathbf{v}$ are “white” (completely random from measurement to measurement)
  - Measurement model is linear

What if Measurements are Non-Linear

- Example of non-linear measurements: a range (distance) measurement (such as with GPS)
- Can use non-linear measurement model

Nonlinear

$$\mathbf{z} = \mathbf{h}(\mathbf{x}) + \mathbf{v}$$

Linear

$$\mathbf{z} = \mathbf{H}\mathbf{x} + \mathbf{v}$$

- Kalman filter is then modified to become an “Extended Kalman Filter” (EKF)
  - Requires linearization about the estimated solution
  - Because of this, an EKF is not, technically speaking, truly optimal like the KF
  - In many cases it would be “nearly optimal”—depends on the nature of the linearization

---

Dynamics Model

Discrete-Time Dynamics Model

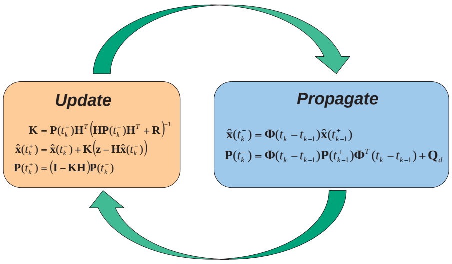
$$\mathbf{x}(t_k^-) = \mathbf{\Phi}\mathbf{x}(t_{k-1}^+) + \mathbf{w}_d$$

state after propagation

dynamics matrix

state before propagation

process noise



Update

$$\mathbf{K} = \mathbf{P}(t_k^-)\mathbf{H}^T\left(\mathbf{H}\mathbf{P}(t_k^-)\mathbf{H}^T + \mathbf{R}\right)^{-1}$$
$$\hat{\mathbf{x}}(t_k^+) = \hat{\mathbf{x}}(t_k^-) + \mathbf{K}\left(\mathbf{z} - \mathbf{H}\hat{\mathbf{x}}(t_k^-)\right)$$
$$\mathbf{P}(t_k^+) = \left(\mathbf{I} - \mathbf{K}\mathbf{H}\right)\mathbf{P}(t_k^-)$$

Propagate

$$\hat{\mathbf{x}}(t_k^-) = \mathbf{\Phi}(t_k - t_{k-1})\hat{\mathbf{x}}(t_{k-1}^+)$$
$$\mathbf{P}(t_k^-) = \mathbf{\Phi}(t_k - t_{k-1})\mathbf{P}(t_{k-1}^+)\mathbf{\Phi}^T(t_k - t_{k-1}) + \mathbf{Q}_d$$

Process noise is described by the measurement noise covariance matrix:

$$\mathbf{Q}_d = E[\mathbf{w}_d\mathbf{w}_d^T] = \begin{bmatrix} E[w_1^2] & E[w_1w_2] & \cdots & E[w_1w_n] \\ E[w_1w_2] & E[w_2^2] & & \\ \vdots & & \ddots & \\ E[w_1w_n] & & & E[w_n^2] \end{bmatrix}$$

- Key assumptions (in semi-nontechnical language)
  - Process noise $\mathbf{w}_d$ is Gaussian (follow a “bell curve”)
  - Process noise $\mathbf{w}_d$ is “white” (completely random from epoch to epoch)
  - Dynamics model is perfectly known

Initialization and “Time Constant” of a KF

- Things needed in order to initialize a filter
  - Initial state estimate
  - Initial covariance matrix
  - Measurement model(s)
  - Propagation model(s)
- Time constant (not meant in a precise, technical way)
  - Defines how long a measurement will affect the filter
  - In theory, every measurement will affect the filter for the rest of time
  - In practice, this may not be the case so much
    - Example: Case in which there is high propagation noise—old measurements are significantly “de-weighted” relative to new measurements
  - Warning: Even in a case where a filter has a “short” time constant (i.e., measurements lose impact fairly quickly), a large measurement error (blunder) can have a devastating impact

---

Kalman Filter Example: Hot Air Balloon

- Scenario: Want to estimate height of a hot air balloon on a windy day, starting at 800 m

- What I have
  - Radar altimeter to measure height above ground (assume ground height is known)
    - Meas error modeled as Gaussian with 2m standard deviation
  - Stochastic process model for how the wind affects the height of the balloon
  - Initial uncertainty modeled as Gaussian with standard deviation of 10m (height) and 1m/s (vertical velocity)
- What I want to know
  - Height estimate and standard deviation
  - Vertical velocity estimate and standard deviation

Stochastic Process Model

- State vector: $\mathbf{x} = \begin{bmatrix} h \\ \dot{h} \end{bmatrix}$
  - h: balloon height (m)
  - $\dot{h}$: baloon vertical velocity (m/s)
- Continuous time process model:

$$\dot{\mathbf{x}}(t) = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \mathbf{x}(t) + \mathbf{w}(t) \qquad E[\mathbf{w}(t)\mathbf{w}^T(t+\tau)] = Q\delta(\tau) \qquad Q = \begin{bmatrix} 0 & 0 \\ 0 & 0.01 \end{bmatrix}$$

Dirac delta

Process noise

Process noise matrix

Dynamics matrix F

- Discrete time process model: $\mathbf{x}(t_k) = \Phi(\Delta t)\mathbf{x}(t_{k-1}) + \mathbf{w}_d(t_k)$

$$\Phi(\Delta t) = e^{F\Delta t} = \begin{bmatrix} 1 & \Delta t \\ 0 & 0 \end{bmatrix} = \begin{bmatrix} 1 & 0.5 \\ 0 & 0 \end{bmatrix} \quad \text{for} \quad \Delta t = t_k - t_{k-1} = 0.5\,\text{sec}$$

$$E[\mathbf{w}_d(t_k)\mathbf{w}_d^T(t_k)] = Q_d = \begin{bmatrix} 0.0004 & 0.0013 \\ 0.0013 & 0.005 \end{bmatrix}$$

Kalman Filter Propagation Equations

- Propagate state: $\hat{\mathbf{x}}(t_k) = \Phi\hat{\mathbf{x}}(t_{k-1})$
- Propagate covariance: $P(t_k) = \Phi P(t_{k-1})\Phi^T + Q_d$
- Example
  - Initial conditions:

$$\hat{\mathbf{x}}(0) = \begin{bmatrix} 800 \\ 0 \end{bmatrix} \qquad P(0) = \begin{bmatrix} 10^2 & 0 \\ 0 & 1^2 \end{bmatrix}$$

  - First time step:

$$\hat{\mathbf{x}}(0.5) = \Phi\hat{\mathbf{x}}(0) = \begin{bmatrix} 1 & 0.5 \\ 0 & 0 \end{bmatrix}\begin{bmatrix} 800 \\ 0 \end{bmatrix} = \begin{bmatrix} 800 \\ 0 \end{bmatrix}$$

$$P(0.5) = \Phi P(0)\Phi^T + Q_d$$

$$= \begin{bmatrix} 1 & 0.5 \\ 0 & 0 \end{bmatrix}\begin{bmatrix} 100 & 0 \\ 0 & 1 \end{bmatrix}\begin{bmatrix} 1 & 0 \\ 0.5 & 0 \end{bmatrix} + \begin{bmatrix} 0.0004 & 0.0013 \\ 0.0013 & 0.005 \end{bmatrix}$$

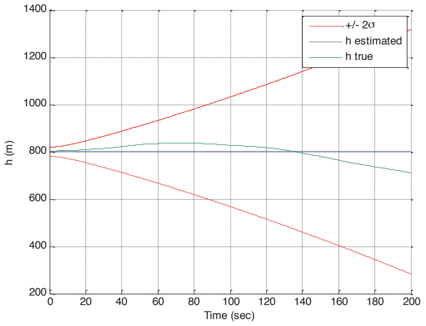
$$= \begin{bmatrix} 100.25 & 0.5 \\ 0.5 & 1 \end{bmatrix} + \begin{bmatrix} 0.0004 & 0.0013 \\ 0.0013 & 0.005 \end{bmatrix}$$

$$= \begin{bmatrix} 100.2504 & 0.5013 \\ 0.5013 & 1.005 \end{bmatrix} \longleftarrow$$

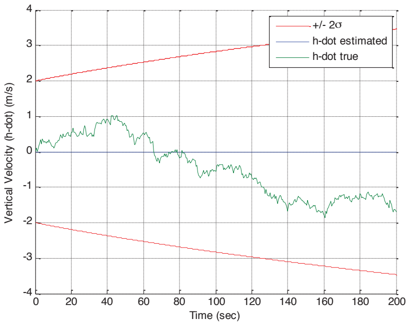
Notice that this is higher (more uncertainty) than when we started!

Propagation Example—No Measurements (Single Run)

h

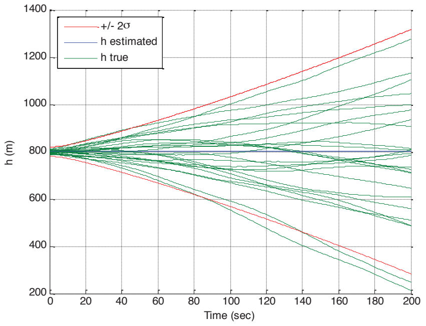


$\dot{h}$

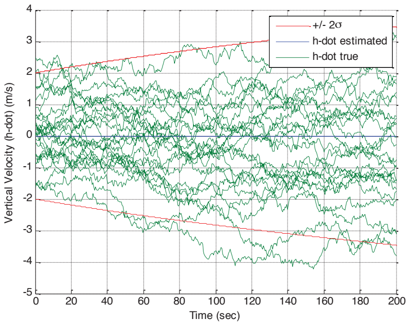


Propagation Example—No Measurements (25 Runs)

h



$\dot{h}$



Measurement Update at t=40 Sec

- At t=40 sec, a measurement of 820.97 m is taken (remember, meas error standard deviation is 2 m)
- Measurement model: $\mathbf{z} = \mathbf{Hx} + \mathbf{v}$ $\quad \mathbf{R} = E[\mathbf{vv}^T]$
  - In this case: $z = h_{meas} + v = \underbrace{\begin{bmatrix} 1 & 0 \end{bmatrix}}_{H} \underbrace{\begin{bmatrix} h \\ \dot{h} \end{bmatrix}}_{\mathbf{x}} + v \qquad R = E\left[v^2\right] = 4$

(Note that this is a scalar measurement)

- Step 1: Propagate up to measurement time: $t_k = 80\,\mathrm{sec}$

$$\hat{\mathbf{x}}(t_k) = \begin{bmatrix} 800 \\ 0 \end{bmatrix} \qquad P(t_k) = \begin{bmatrix} 1913.3 & 48 \\ 48 & 1.40 \end{bmatrix}$$

$\nwarrow \qquad \nearrow$

These are consistent with previous slides

Measurement Update at t=40 Sec

(continued)

- Step 2: Measurement update

State: $\hat{\mathbf{x}}(t_k) = \hat{\mathbf{x}}(t_k) + K\left[z(t_k) - H\hat{\mathbf{x}}(t_k)\right]$

Covariance: $P(t_k) = \left[I - KH\right]P(t_k)$

Kalman gain: $K = P(t_k)H^T\left[HP(t_k)H^T + R\right]^{-1}$

- In this case, since $H = \begin{bmatrix} 1 & 0 \end{bmatrix}$:

from previous slide: $P(t_k) = \begin{bmatrix} 1913.3 & 48 \\ 48 & 1.40 \end{bmatrix}$

$$K = \begin{bmatrix} \dfrac{P_{1,1}}{P_{1,1} + R} \\ \dfrac{P_{2,1}}{P_{1,1} + R} \end{bmatrix} = \begin{bmatrix} \dfrac{1913.3}{1913.3 + 4} \\ \dfrac{48}{3.9917 + 4} \end{bmatrix} = \begin{bmatrix} 0.998 \\ 0.025 \end{bmatrix}$$

$$\hat{\mathbf{x}}(t_k) = \hat{\mathbf{x}}(t_k) + K\left[z(t_k) - H\hat{\mathbf{x}}(t_k)\right]$$

$$= \begin{bmatrix} 800 \\ 0 \end{bmatrix} + \begin{bmatrix} 0.998 \\ 0.025 \end{bmatrix} \underbrace{\begin{bmatrix} 820.97 - 800 \end{bmatrix}}_{\text{residual}}$$

$$\hat{\mathbf{x}}(t_k) = \begin{bmatrix} 820.926 \\ 0.525 \end{bmatrix}$$

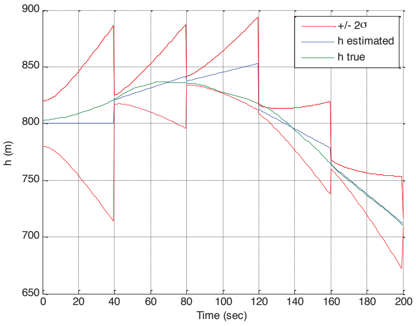
$$P(t_k) = \underbrace{\begin{bmatrix} 0.0021 & 0 \\ -0.0250 & 1 \end{bmatrix}}_{I-KH} \underbrace{\begin{bmatrix} 1913.3 & 48 \\ 48 & 1.40 \end{bmatrix}}_{P(t_k)} = \begin{bmatrix} 3.992 & 0.100 \\ 0.100 & 0.198 \end{bmatrix}$$

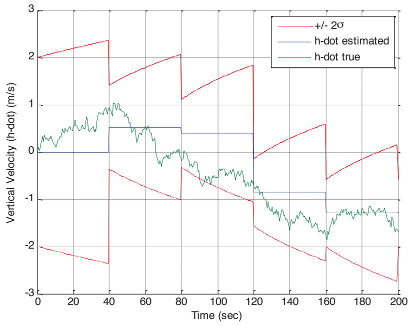
(Before measurement)

Filter Propagation/Measurement Incorporation Example

h

$\dot{h}$



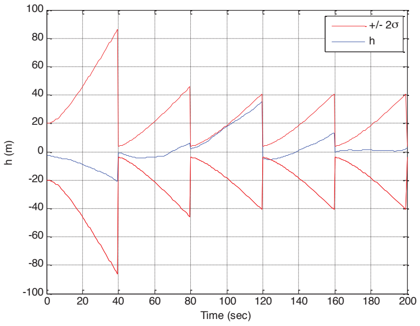


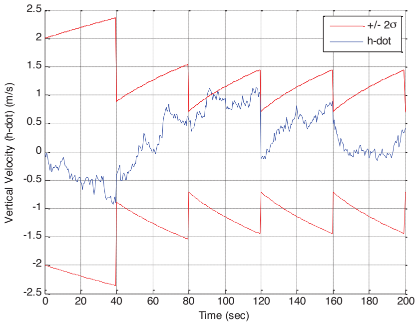
Errors in Filter Estimate for Same Case as Previous Slide

- These plots show the DIFFERENCE between the estimated state (blue on previous slide) and the true state (green on previous slide)

h

$\dot{h}$

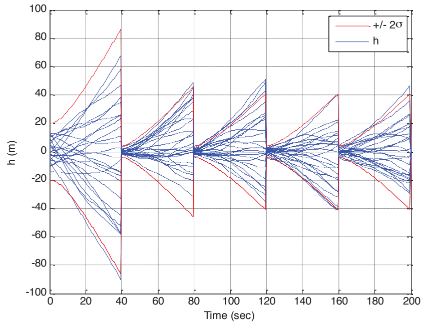


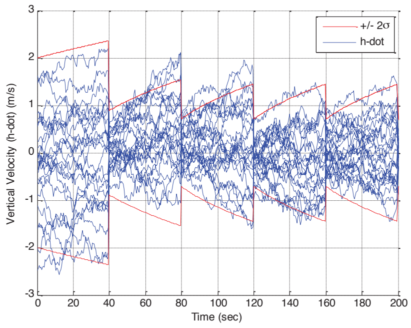


Errors in Filter Estimate
25 Monte Carlo Runs

h

$\dot{h}$





Modeling Errors

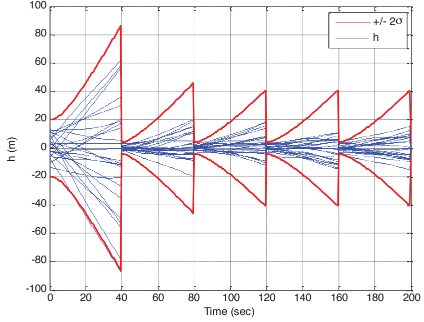
- In the previous example, the filter had a perfect model
- What happens if there is a process noise modeling error?
- Example:

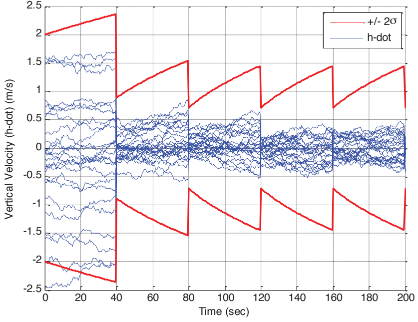
True: $Q = \begin{bmatrix} 0 & 0 \\ 0 & 0.001 \end{bmatrix}$

Modeled: $Q = \begin{bmatrix} 0 & 0 \\ 0 & 0.01 \end{bmatrix}$

h

$\dot{h}$



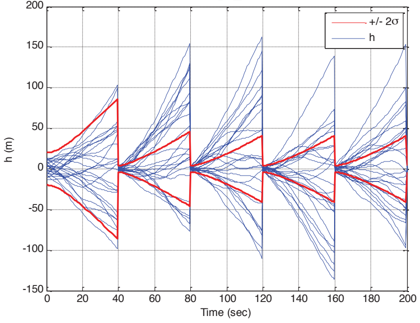


Another Modeling Error Example

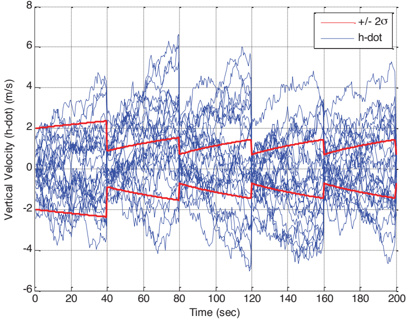
True: $Q = \begin{bmatrix} 0 & 0 \\ 0 & 0.1 \end{bmatrix}$

Modeled: $Q = \begin{bmatrix} 0 & 0 \\ 0 & 0.01 \end{bmatrix}$

h



$\dot{h}$



Other Comments on Example

- This was a simple example, but more complex examples (more states, more complicated measurement model) work the same way
- For GNSS systems, the H matrix is the same as the H matrix used for least-squares solutions
  - Measurement model is nonlinear, so Extended Kalman Filter (EKF) is used
- Kalman filter will give optimal results when all of its assumptions are met
  - Measurement errors are zero mean, white, Gaussian noise
  - Process noise (discrete-time) is zero mean, white, Gaussian noise
  - Measurement model and process model are known and correct
  - Measurements and process model are linear functions of the state
- If any of these are not met, it is not technically optical any more
  - However, it still may give “good” results
- Often, the modeling aspects of the problem are a more significant challenge than the filter itself

One Final Thought…

Life is a Kalman Filter

(really, it is!)