Fundamentals of Inertial Aiding

April 17, 2024

IEEE AESS Virtual Distinguished Lecture

Michael Braasch, Ph.D., P.E., F.IEEE, F.ION
Emeritus Professor of Electrical Engineering
Ohio University Avionics Engineering Center
Distinguished Lecturer, IEEE AESS

OHIO UNIVERSITY

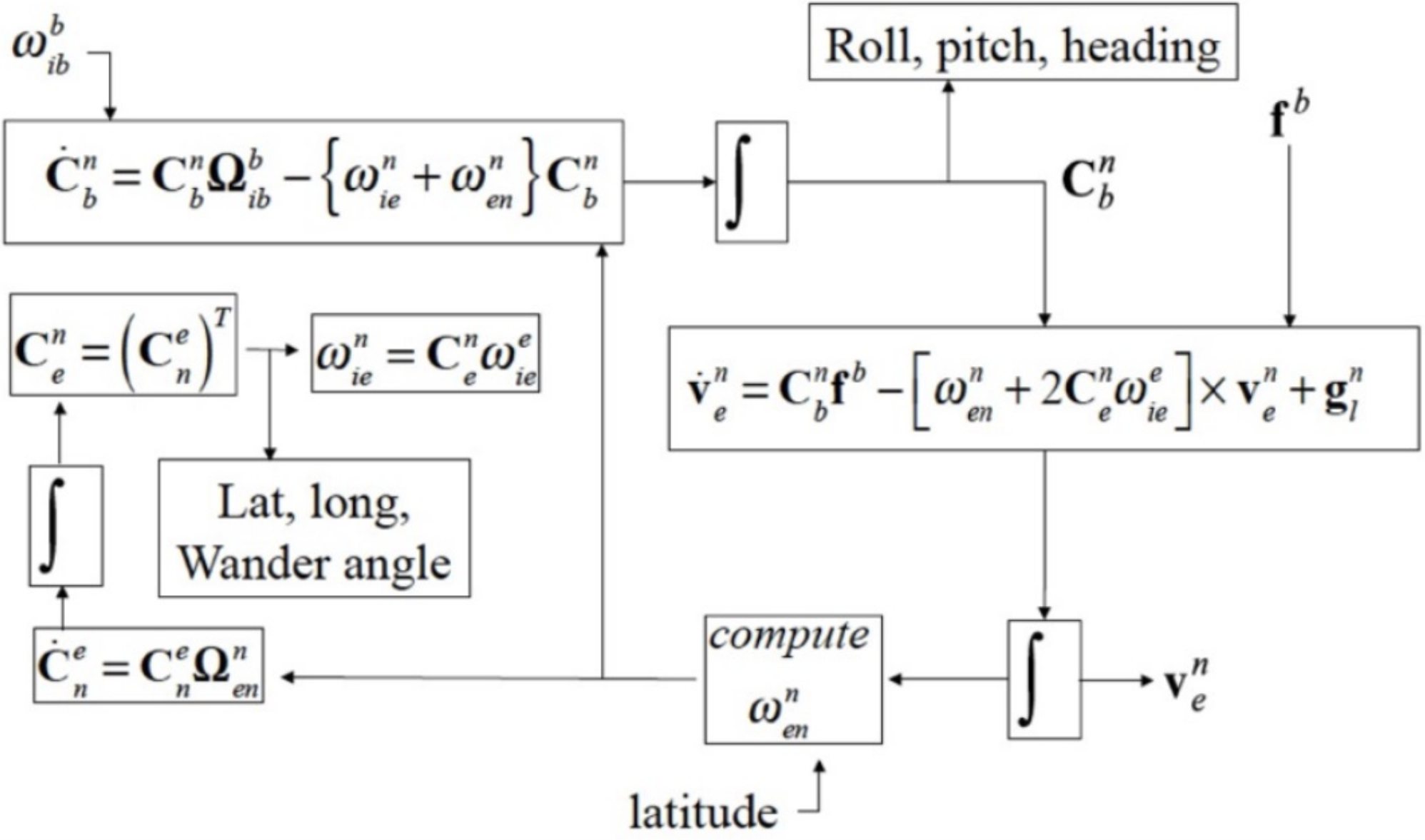
Why Inertial?

- One sensor suite determines position, velocity and attitude
- Immune to interference and jamming
- High data rates (~ 50 – 1000 Hz)
- Low data latencies (nearly instantaneous)
- Relatively noise-less in the short term
- Poor long-term accuracy: not stable, errors drift/grow with time

Two Solutions to Long-Term Error Growth

- One: Live with it (fly from New York to London and you're only off by about 10 miles when you enter British airspace [nav-grade IMU assumed])
- Two: Integrate an external/independent source of position to correct ('aid') the INS
- Assuming the second choice is preferred, what is the optimal way to perform the integration? *Will the integration accommodate a variety of aiding sources?*

Strapdown Processing (continous-time)



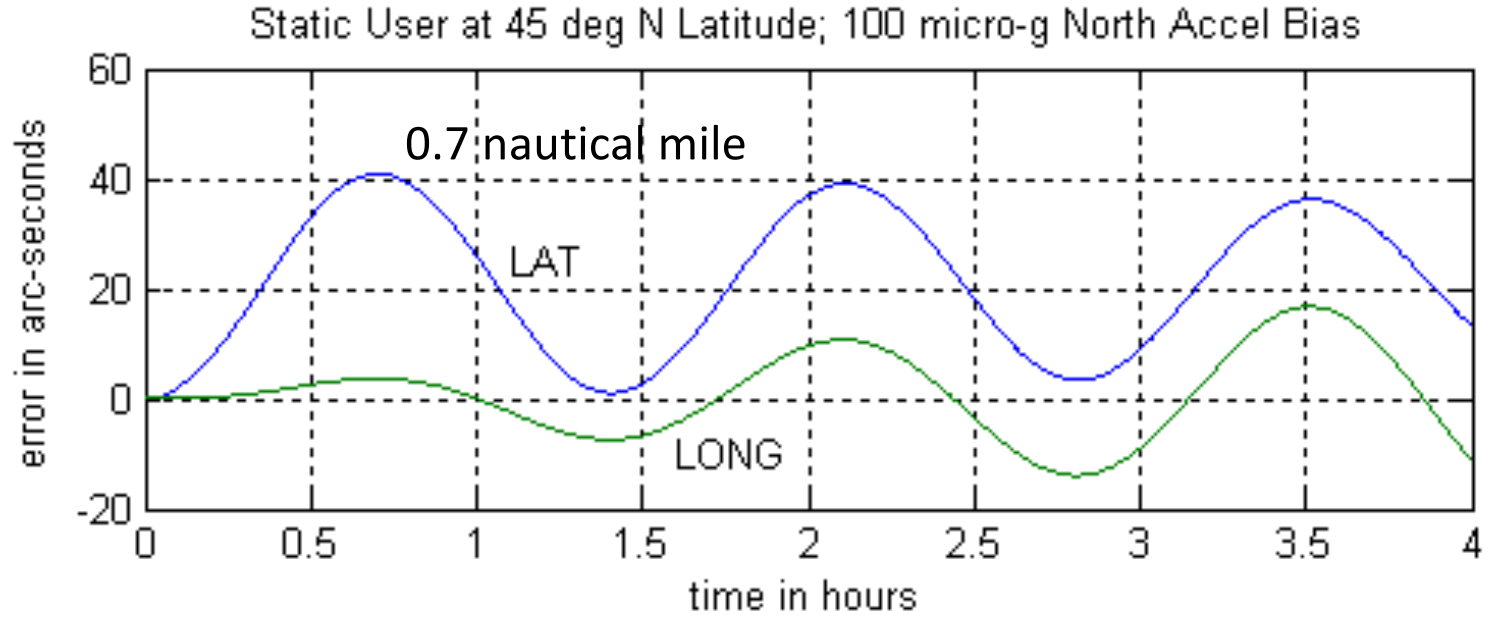
INS Error Characteristics

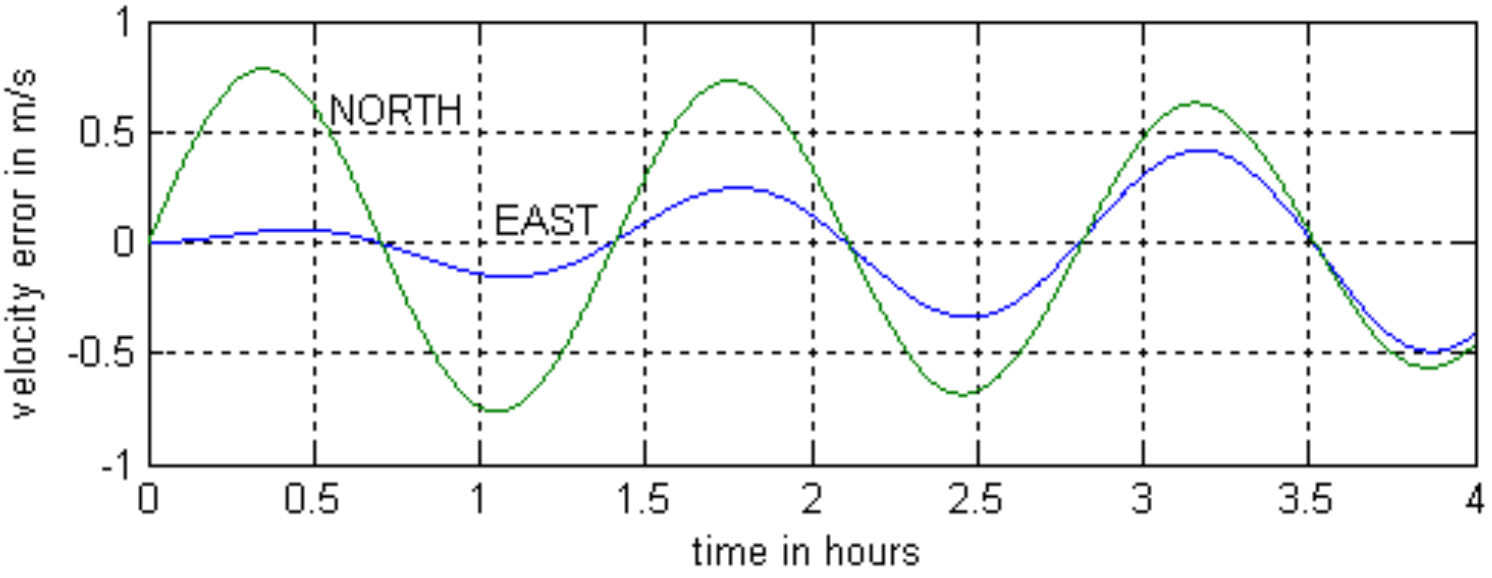
- In an aided-inertial system, we seek to estimate and correct the INS errors (position, velocity, attitude, sensor errors)
- To do this optimally, we need to characterize (i.e., mathematically model) these errors

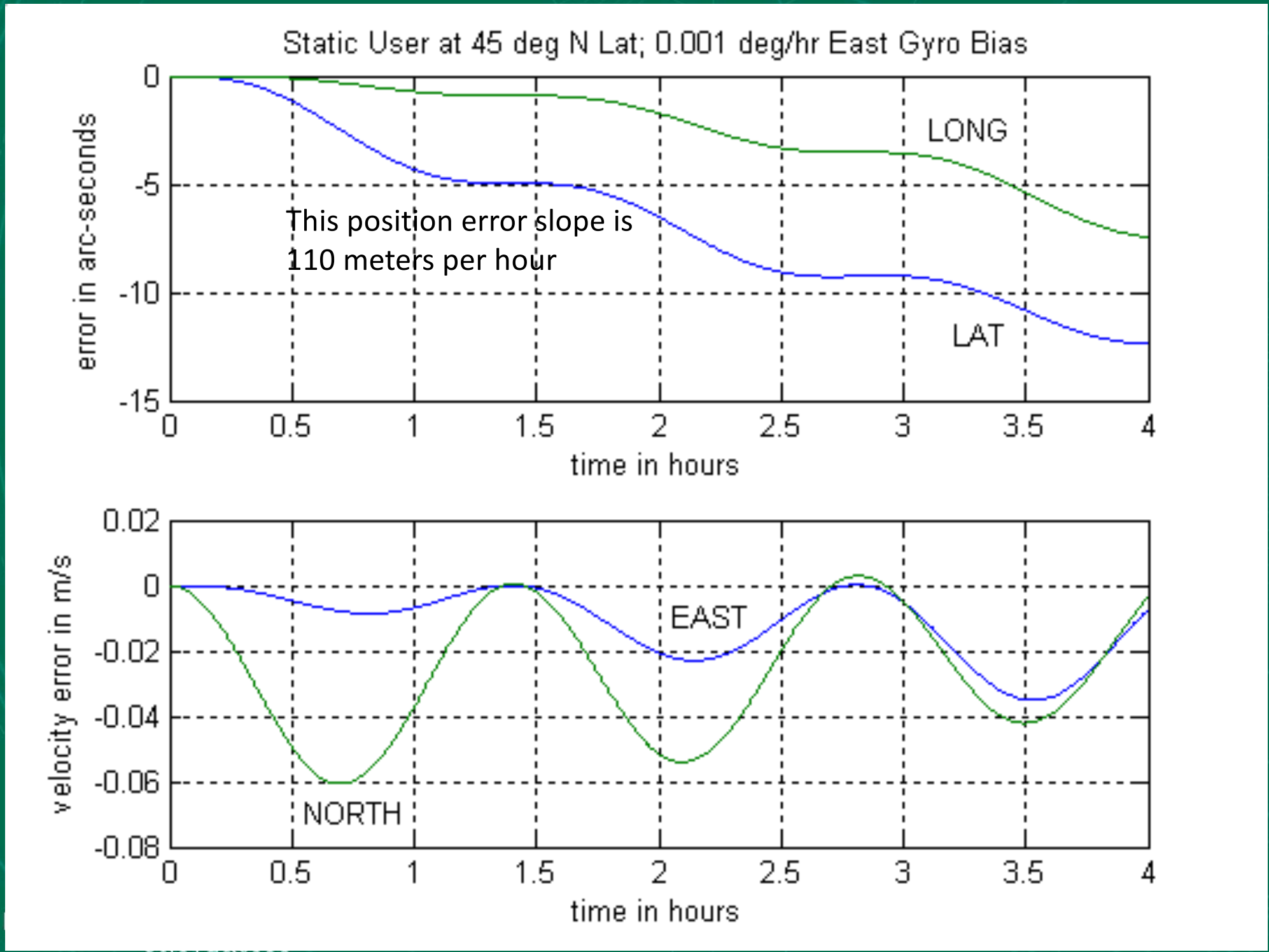
INS Errors

Broad categories

- Imperfect mounting of sensors in the box
- Imperfect mounting of the box in the vehicle
- Initialization errors
- Sensor errors
- Gravity model errors
- Computational errors


Static User at 45 deg N Latitude; 100 micro-g North Accel Bias
0.7 nautical mile
LAT
LONG
error in arc-seconds
time in hours
60
40
20
0
-20
0
0.5
1
1.5
2
2.5
3
3.5
4


NORTH
EAST
velocity error in m/s
time in hours
1
0.5
0
-0.5
-1
0
0.5
1
1.5
2
2.5
3
3.5
4


Static User at 45 deg N Lat; 0.001 deg/hr East Gyro Bias
error in arc-seconds
0
-5
-10
-15
LONG
LAT
This position error slope is
110 meters per hour
0
0.5
1
1.5
2
2.5
3
3.5
4
time in hours
velocity error in m/s
0.02
0
-0.02
-0.04
-0.06
-0.08
EAST
NORTH
0
0.5
1
1.5
2
2.5
3
3.5
4
time in hours

Short-Term Error Growth from Groves

100 micro-g per root-Hz

0.1 deg per root-hour

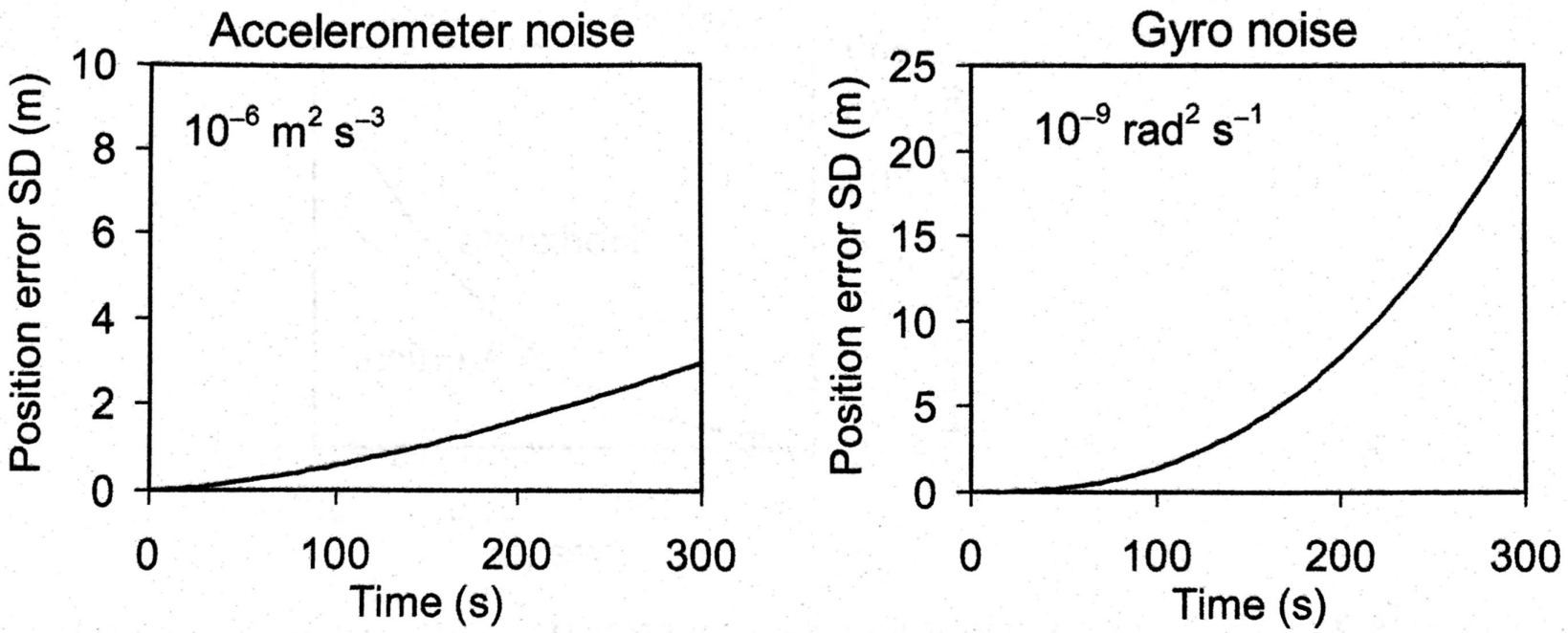
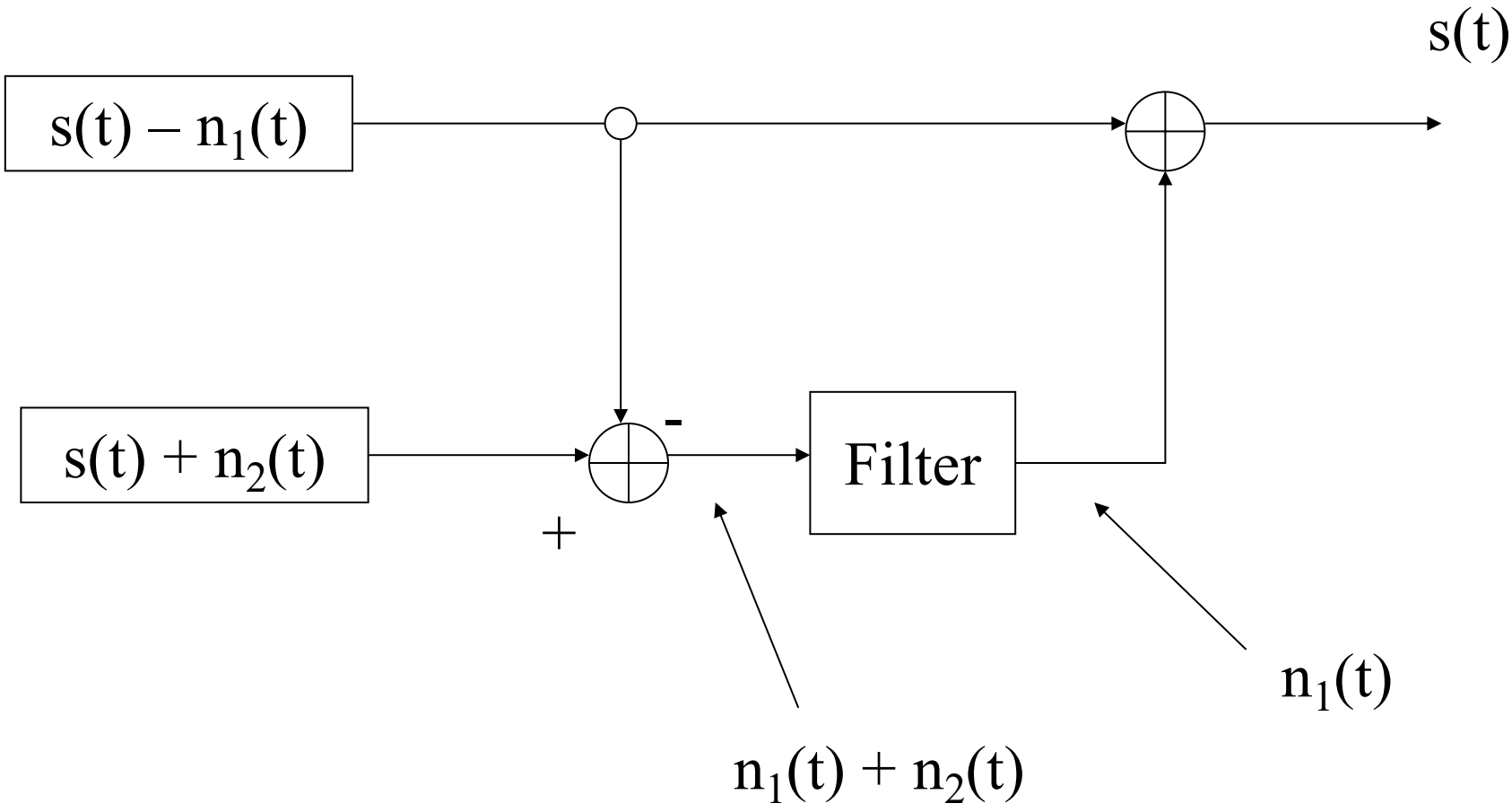


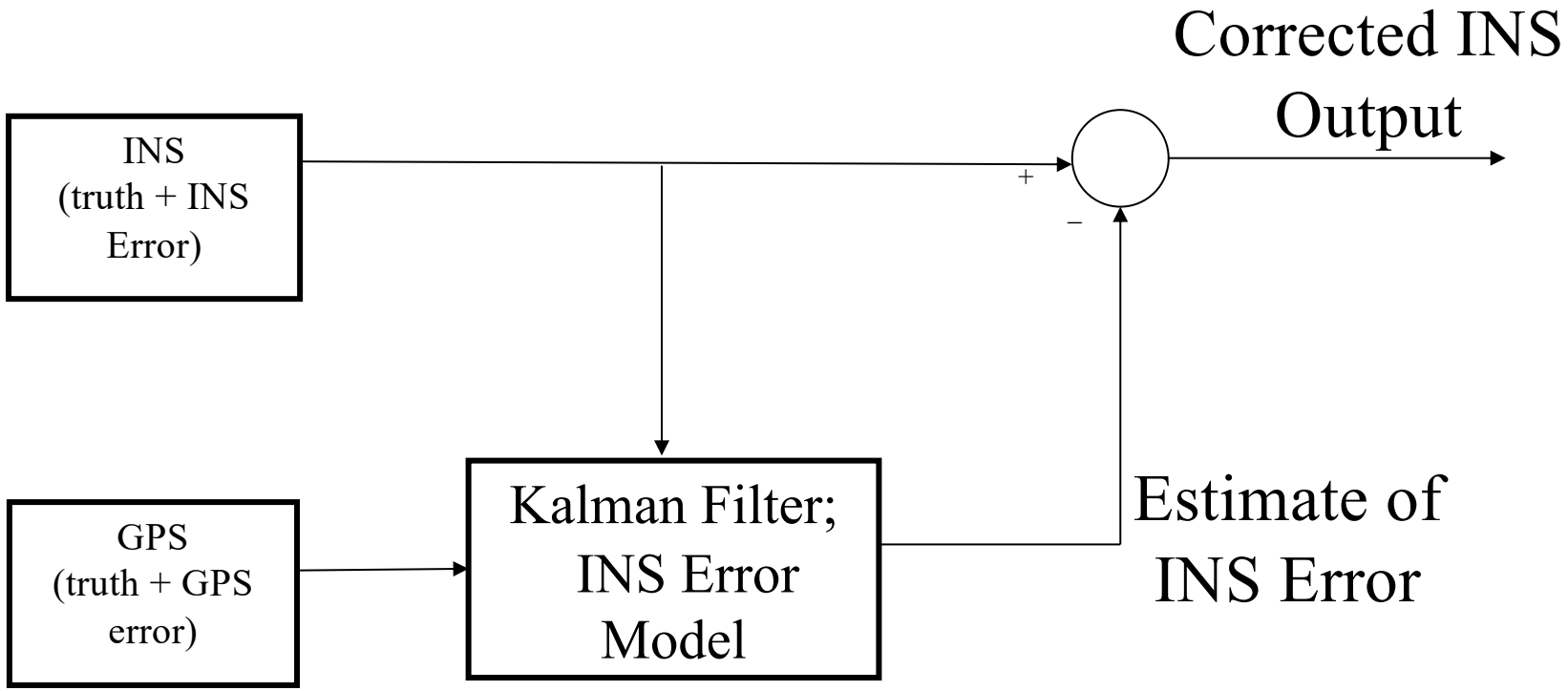
Figure 5.20 Short-term straight-line position error standard deviation growth per axis due to inertial sensor noise.

So How Do We Get Rid of the Drift?

Complementary Filtering



Feedforward GPS-Aided INS



Predictions/Measurements/Estimates

- The fundamental concept in *recursive* filtering (i.e., recursive estimation) theory is the following:

Estimates are weighted combinations of predictions and measurements

$$\text{Estimate} = \text{WeightFactor}_{pred} * \text{Prediction} + \text{WeightFactor}_{meas} * \text{Measurement}$$

Estimation of the Mean (Revisited)

- Recall the LSE of the mean: $\hat{\boldsymbol{\mu}} = \frac{1}{N}\sum_{i=1}^{N} X_i$
- Recursively:

$$\hat{x}_1 = \frac{1}{1} y_1$$

$$\hat{x}_2 = \frac{1}{2}\left(y_1 + y_2\right) = \frac{2-1}{2} y_1 + \frac{1}{2} y_2 = \left(\frac{2-1}{2}\right)\hat{x}_1 + \frac{1}{2} y_2$$

$$\hat{x}_3 = \frac{1}{3}\left(y_1 + y_2 + y_3\right) = \frac{1}{3}\left[2 \bullet \frac{1}{2}\left(y_1 + y_2\right) + y_3\right]$$

$$= \frac{3-1}{3}\hat{x}_2 + \frac{1}{3} y_3 \qquad \therefore \hat{x}_n = \frac{n-1}{n}\hat{x}_{n-1} + \frac{1}{n} y_n$$

Recursive Mean Estimation (cont'd)

- The recursive estimation has the form mentioned earlier:

$$\hat{x}_n = \frac{n-1}{n}\hat{x}_{n-1} + \frac{1}{n}y_n$$

estimate

Weighting factors

prediction

measurement

Recursive Mean Estimation (cont'd)

- Note the weighting factors are not constant:

$$\hat{x}_1 = \frac{1-1}{1}\hat{x}_0 + \frac{1}{1}y_1 = 0 \bullet \hat{x}_0 + 1 \bullet y_1$$

$$\hat{x}_\infty = \frac{\infty-1}{\infty}\hat{x}_{\infty-1} + \frac{1}{\infty}y_\infty = 1 \bullet \hat{x}_{\infty-1} + 0 \bullet y_\infty$$

Recursive Mean Estimation (cont'd)

- At the first update, 100% weight is given to the measurement; if there was any a priori knowledge available from which to make a prediction, it is thrown away as the weighting factor on the 'prediction' $\hat{x}_0$ is set to zero
- As time goes on, each successive measurement receives less and less weight and the 'prediction' receives more and more

Bucket-Full of Resistors Example

- Earlier we mentioned the simple example of an unknown resistor and three ohmmeters
- Assuming we only had the measurements from the three meters, the best estimate would simply be the average (i.e., the sample mean)
- However, what if we knew more about situation?

Bucket-Full of Resistors (cont'd)

- Reference: Roger M. du Plessis, "Poor Man's Explanation of Kalman Filtering," North American Aviation, Autonetics Division, Anaheim, CA, June 1967 (currently available from Taygeta).
- Assume we have the following information:
  - Resistor is labeled as being nominally 100 ohms with a 1% tolerance
  - The RMS error on each ohmmeter is 3 ohms

Bucket-Full of Resistors (cont'd)

- Least-squares estimation (sample mean in this case) assumes you only have the measurement data and nothing else; any other information (if available) is thrown away
- Intuitively, it seems perfectly reasonable that we should be able to do a better job of estimation if we take into account the additional information (nominal resistor value and tolerance, ohmmeter accuracy)

Introducing: KALMAN!!!

- The answer, of course, is yes and we do it through the use of the Kalman Filter
- The least squares recursion has the form:

$$\hat{x}_n = \frac{n-1}{n}\hat{x}_{n-1} + \frac{1}{n}y_n$$

- Kalman has the following form:

$$\hat{x}_n = \hat{x}_n^- + K_n\left(y_n - H_n\hat{x}_n^-\right)$$

The Scalar Kalman Update Equation

$$\hat{x}_n = \hat{x}_n^- + K_n\left(y_n - H_n\hat{x}_n^-\right)$$

Current estimate

Current prediction

Kalman Gain

Current measurement

Data matrix (scalar = 1 for the resistor problem)

Prediction & Measurement

- We need to manipulate the equation a bit to get it into our form of measurement and prediction
- Since $H_n = 1$ in our case:

$$\hat{x}_n = \hat{x}_n^- + K_n\left(y_n - \hat{x}_n^-\right)$$

$$= \hat{x}_n^- + K_n y_n - K_n \hat{x}_n^-$$

$$= \left(1 - K_n\right)\hat{x}_n^- + K_n y_n$$

Kalman versus Least Squares

- Least Squares: $\hat{x}_n = \frac{n-1}{n}\hat{x}_{n-1} + \frac{1}{n}y_n$
- Kalman: $\hat{x}_n = (1-K_n)\bar{\hat{x}}_n + K_n y_n$
- K_n ranges from 0 to 1 thus the Kalman gain determines the weighting on the prediction and the measurement
- In order for Kalman to do better than Least Squares, K_n must embody the additional information known about the problem

The Kalman Gain

- The equation for the scalar Kalman Gain (stated without proof):

$$K_n = \frac{P_n^- H_n^T}{H_n P_n^- H_n^T + R_n}$$

  - where: P_n^- is the prediction error variance
  - R_n is the measurement error variance
  - For the resistor example, $H_n = 1$, thus:

$$K_n = \frac{P_n^-}{P_n^- + R_n}$$

Understanding the Kalman Gain

- In order to appreciate the influence of the Kalman gain, consider what happens as the prediction error varies:

$$\lim_{P_n^- \to 0} K_n = \lim_{P_n^- \to 0} \frac{P_n^-}{P_n^- + R_n} = 0$$

$$\lim_{P_n^- \to \infty} K_n = \lim_{P_n^- \to \infty} \frac{P_n^-}{P_n^- + R_n} = 1$$

- Thus when we have high confidence in our prediction, K_n approaches 0 and estimate = prediction:

$$\hat{x}_n = (1 - K_n)\hat{x}_n^- + K_n y_n = \hat{x}_n^-$$

Understanding the Kalman Gain (cont'd)

- Conversely, when we have low confidence in our prediction, K_n approaches 1 and estimate = measurement:

$$\hat{x}_n = (1 - K_n)\hat{x}_n^- + K_n y_n = y_n$$

- In actual operation, of course, our predictions lie somewhere between perfect and horrible, thus K_n strikes just the right balance between prediction and measurement

Relating to the Resistor Example

- For the resistor example, we stated the Kalman Gain was given by:

$$K_n = \frac{P_n^-}{P_n^- + R_n}$$

- And specifically:

P_1^- = (1 ohm)^2 = 1 ohm^2 = square of the tolerance (note: at the first instant of time, our best prediction is simply the nominal resistor value)

R_n = (3 ohm)^2 = 9 ohm^2 = square of the rms meter error (in this case R is actually a constant so it is not a function of n)

Computing Prediction Error Variance

- We need the prediction error variance P_n^- and the measurement error variance R_n in order to compute the Kalman gain
- We are assuming that our measurement error variance is a constant
- However, if the Kalman filter is doing its job right, its predictions should improve over time
- Thus we need an expression to compute P_n^- over time

Prediction Error Variance (cont'd)

- It turns out that we get our predictions from our estimates, so first we must compute the estimation error variance:

$$P_n = \left(I - K_n H_n\right) P_n^-$$

- However, we are assuming the true value of the resistor is a constant (i.e., it does not change no matter how many measurements we take). Thus there are no 'system' dynamics.
- As a result, for a STATIC system, the prediction error variance is equal to the estimation error variance:

$$P_{n+1}^- = P_n$$

Estimation Error Variance

- Let's check to see if the expression makes sense. Since $H_n = 1$ in this example: $P_n = \left(I - K_n\right) P_n^-$
- Also for this example: $K_n = \dfrac{P_n^-}{P_n^- + R_n}$
- Thus:

$$P_n = \left(I - \frac{P_n^-}{P_n^- + R_n} \right) P_n^-$$

$$= \frac{P_n^- \left(P_n^- + R_n\right)}{P_n^- + R_n} - \frac{P_n^- P_n^-}{P_n^- + R_n} = \frac{P_n^- R_n}{P_n^- + R_n}$$

Example Estimation Error Variance

- So for this example: $$P_n = \frac{\bar{P}_n R_n}{\bar{P}_n + R_n}$$
- If the measurement error is much larger than the estimation error, then the Kalman filter will rely primarily on the prediction and the estimation error variance is approximately the prediction error variance; the converse is also true

$$If\ R_n >> \bar{P}_n, Then\ P_n \approx \frac{\bar{P}_n R_n}{R_n} = \bar{P}_n$$

Filter Summary for this Example

- Step 0: Determine the initial prediction and the prediction error variance. For this case:

$$\hat{x}_1^- = 100 \text{ ohms} \qquad P_1^- = 1 \text{ ohms}^2$$

- Step 1: Compute the Kalman gain: $K_n = \dfrac{P_n^- H_n^T}{H_n P_n^- H_n^T + R_n}$ (recall $H_n = 1$ for this example)
- Step 2: Take a measurement (y_n) and compute a current estimate:

$$\hat{x}_n = \hat{x}_n^- + K_n\left(y_n - H_n \hat{x}_n^-\right)$$

Summary continued

- Step 3: Compute the estimation error variance:

$$P_n = \left(I - K_n H_n\right) P_n^-$$

- Step 4: Predict ahead to the next moment in time:

$$\hat{x}_{n+1}^- = \hat{x}_n$$

Again, since this 'system' is static, the best prediction is simply the previous estimate

Summary (cont'd)

- Step 5: Compute the prediction error variance

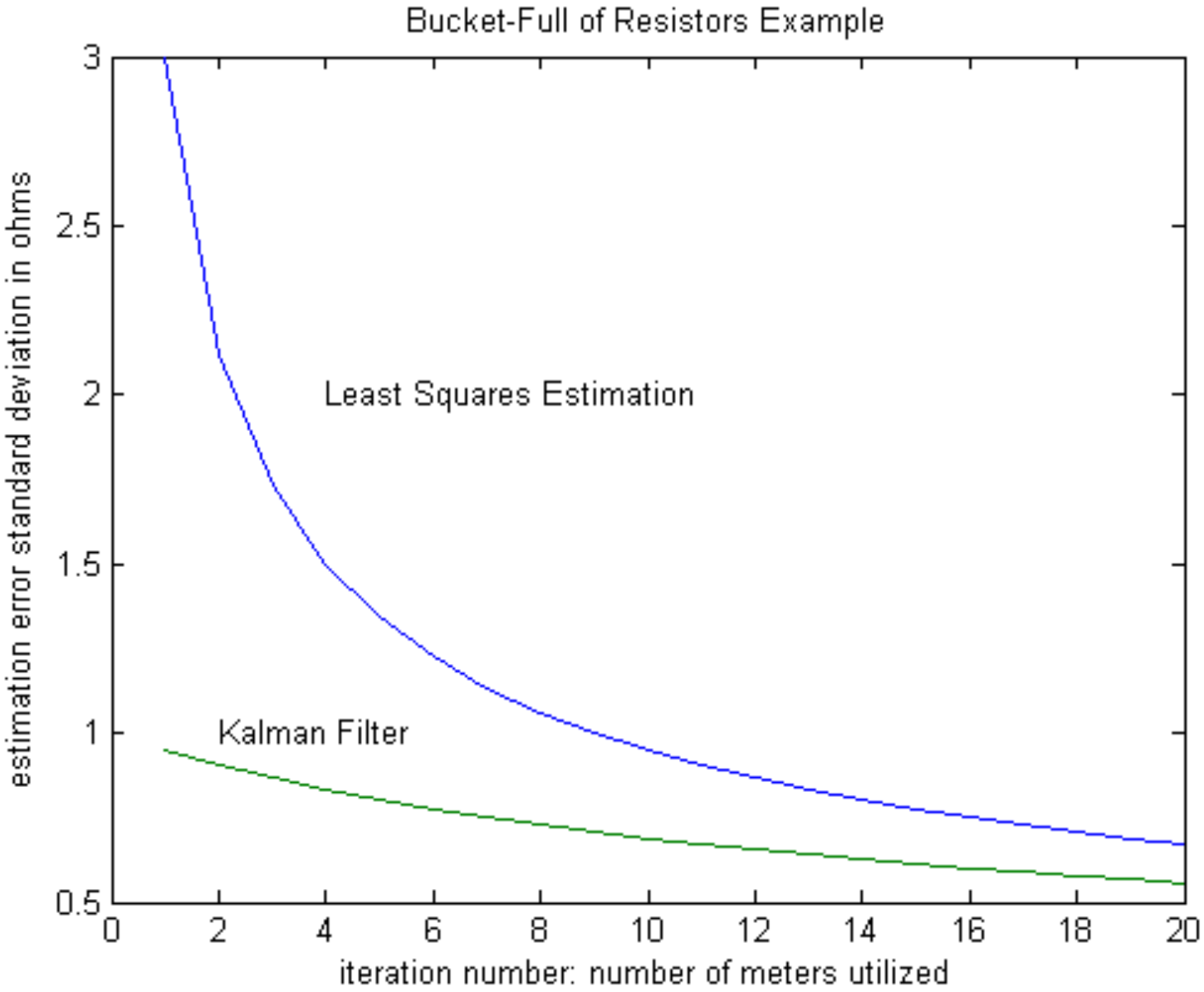
$$P_{n+1}^{-} = P_n$$

This follows since we set our prediction equal to our variance

- Step 6: Go back to the top of the loop (i.e., go to step 1 and do it all again)

Theoretical Comparison

- The next plot shows the comparison of theoretical performance between Least Squares and Kalman
- Since the standard deviation of the estimation error is plotted, the results are known as ensemble statistics
  - That is, the plots represent the standard deviation of the estimation errors obtained from an infinite number of Monte Carlo trials
  - Thus, on a particular trial, the Kalman may or may not perform this well. However, ON AVERAGE, Kalman will perform better than Least Squares



Kalman Algorithm Summary

Step 0: Choose $\hat{x}_1^-$ and determine P_1^-

Step 1: $K_k = P_k^- H_k^T \left(H_k P_k^- H_k^T + R_k \right)^{-1}$

Step 2: $\hat{x}_k = \underline{\hat{x}}_k^- + K_k \left(\underline{Z}_k - H_k \underline{\hat{x}}_k^- \right)$

Step 3: $P_k = (I - K_k H_k) P_k^-$ Estimation error covariance

Step 4: $\underline{\hat{x}}_{k+1}^- = \phi_k \underline{\hat{x}}_k$ prediction

Step 5: $P_{k+1}^- = \phi_k P_k \phi_k^T + Q_k$ Prediction error covariance

Step 6: Go to Step1

INS Error Modeling

- In the typical case of aided-INS, the Kalman Filter *is estimating INS errors* (i.e., position, velocity and attitude errors)
- The estimated errors are then subtracted from the INS output in order to correct them
- If they are small, the INS *errors* can be modeled *linearly* so the core Kalman Filter framework is satisfied
- Feedback is used to keep the errors small

Psi-Angle INS Error Model

$$\underline{\delta \dot{R}}^t = -\underline{\omega}_{et}^t \times \underline{\delta R}^t + \underline{\delta V}^t$$

$$\underline{\delta \dot{V}}^c = C_b^c \underline{\delta f}^b + \underline{\delta g}^c - \underline{\psi}^p \times \underline{f}^c - \left(2\underline{\omega}_{ie}^c + \underline{\omega}_{ec}^c\right) \times \underline{\delta V}^c$$

$$\underline{\dot{\psi}}^p = -\underline{\omega}_{ip}^p \times \underline{\psi}^p - C_b^p \underline{\delta \omega}_{ib}^b$$

References

Siouris, G., Aerospace Avionics Systems – A Modern Synthesis, Academic Press, San Diego, 1993.

Benson, D., "A Comparison of Two Approaches to Pure-Inertial and Doppler-Inertial Error Analysis," IEEE Transactions on Aerospace and Electronic Systems, Vol. AES-11, No. 4, July 1975.

INS Error State-Space Model

$$\begin{bmatrix} \delta\dot{\underline{R}} \\ \delta\dot{\underline{V}} \\ \dot{\underline{\psi}} \end{bmatrix} = \begin{bmatrix} \left[-\underline{\omega}_{ec}^{c}\times\right] & I & 0 \\ W & \left[-\left(2\underline{\omega}_{ie}^{c}+\underline{\omega}_{ec}^{c}\right)\times\right] & \left[\underline{f}^{b}\times\right] \\ 0 & 0 & \left[-\left(\underline{\omega}_{ie}^{c}+\underline{\omega}_{ec}^{c}\right)\times\right] \end{bmatrix} \begin{bmatrix} \delta\underline{R} \\ \delta\underline{V} \\ \underline{\psi} \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 \\ 0 & C_b^c & 0 \\ 0 & 0 & -C_b^c \end{bmatrix} \begin{bmatrix} 0 \\ \delta\underline{f}^{b} \\ \delta\underline{\omega}_{ib}^{b} \end{bmatrix}$$

where: '0' is a 3x3 matrix of zeros, 'I' is a 3x3 identity matrix, W is

$$\begin{bmatrix} -\frac{g_0}{R_0} & 0 & 0 \\ 0 & -\frac{g_0}{R_0} & 0 \\ 0 & 0 & \frac{2g_0}{R_0+h} \end{bmatrix}$$

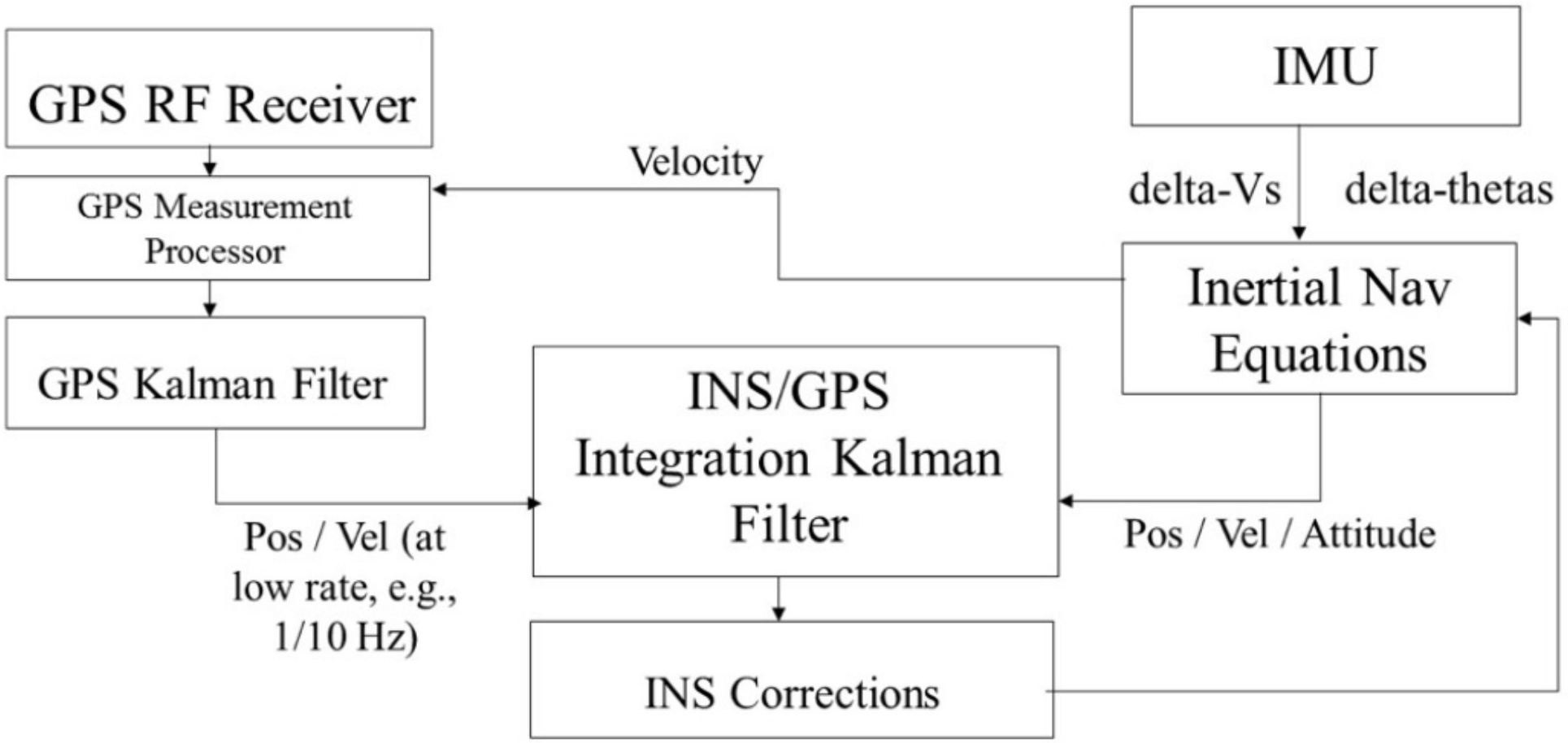
In practice, the sensor biases are not known and must be estimated. The continuous-time form of the system equation is:

$$\begin{bmatrix} \underline{\delta\dot{R}} \\ \underline{\delta\dot{V}} \\ \underline{\dot{\psi}} \\ \underline{\delta\dot{f}^b} \\ \underline{\delta\dot{\omega}^b_{ib}} \end{bmatrix} = \begin{bmatrix} \left[-\underline{\omega}^c_{ec}\times\right] & I & 0_{3x3} & 0_{3x3} & 0_{3x3} \\ W & \left[-\left(2\underline{\omega}^c_{ie}+\underline{\omega}^c_{ec}\right)\times\right] & \left[\underline{f}^b\times\right] & C^c_b & 0_{3x3} \\ 0_{3x3} & 0_{3x3} & \left[-\left(\underline{\omega}^c_{ie}+\underline{\omega}^c_{ec}\right)\times\right] & 0_{3x3} & -C^c_b \\ 0_{3x3} & 0_{3x3} & 0_{3x3} & \frac{-1}{\tau_{acc}}I & 0_{3x3} \\ 0_{3x3} & 0_{3x3} & 0_{3x3} & 0_{3x3} & \frac{-1}{\tau_{gyr}}I \end{bmatrix} \begin{bmatrix} \underline{\delta R} \\ \underline{\delta V} \\ \underline{\psi} \\ \underline{\delta f^b} \\ \underline{\delta\omega^b_{ib}} \end{bmatrix} + \begin{bmatrix} 0_{3x1} \\ 0_{3x1} \\ 0_{3x1} \\ \underline{\eta}_{acc} \\ \underline{\eta}_{gyr} \end{bmatrix}$$

Discretized form is the state-transition matrix

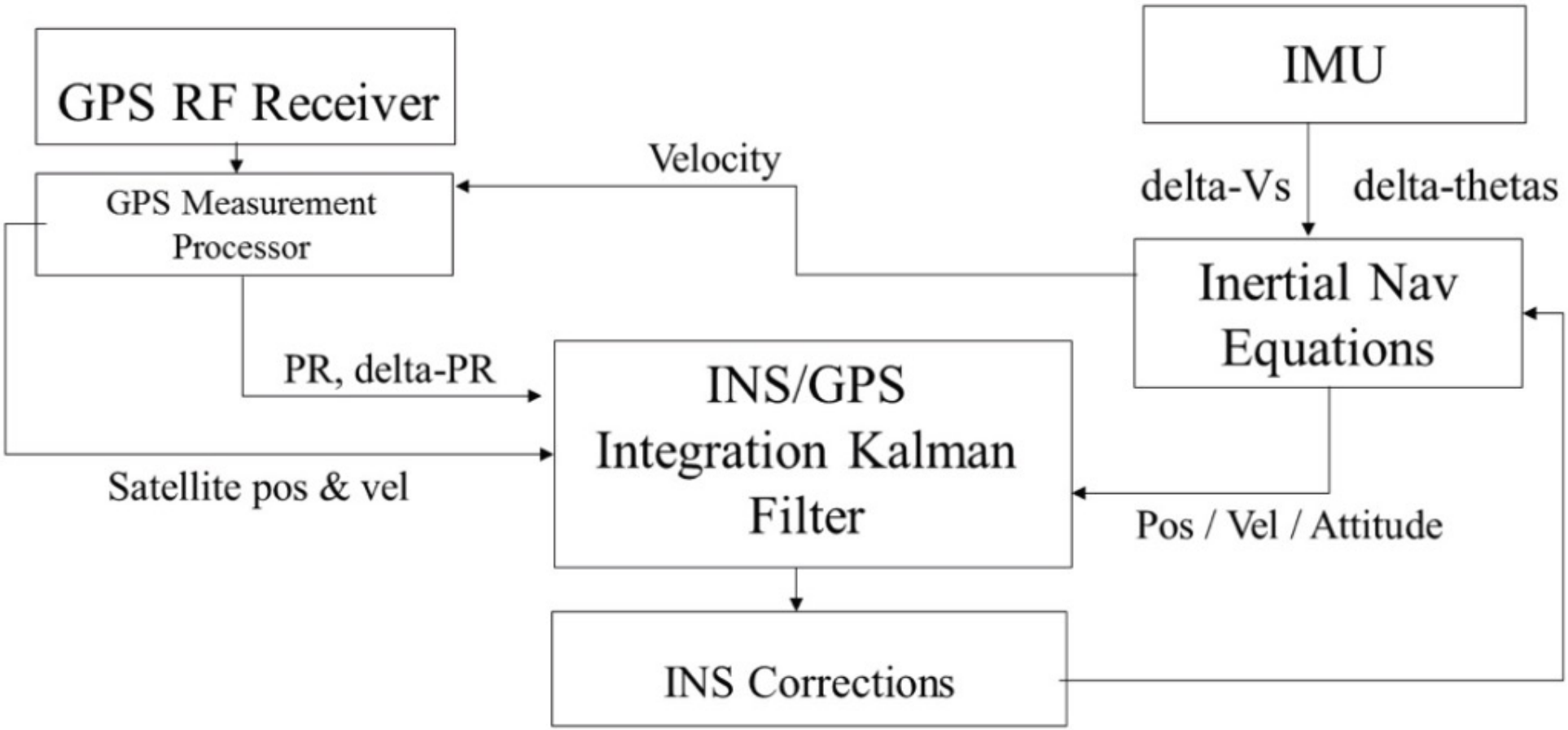
Covariance of this vector is Q

Loosely-Coupled GNSS/INS



Adapted from: Groves, P., Principles of GNSS, Inertial and Multisensor Integrated Navigation Systems, 2008.

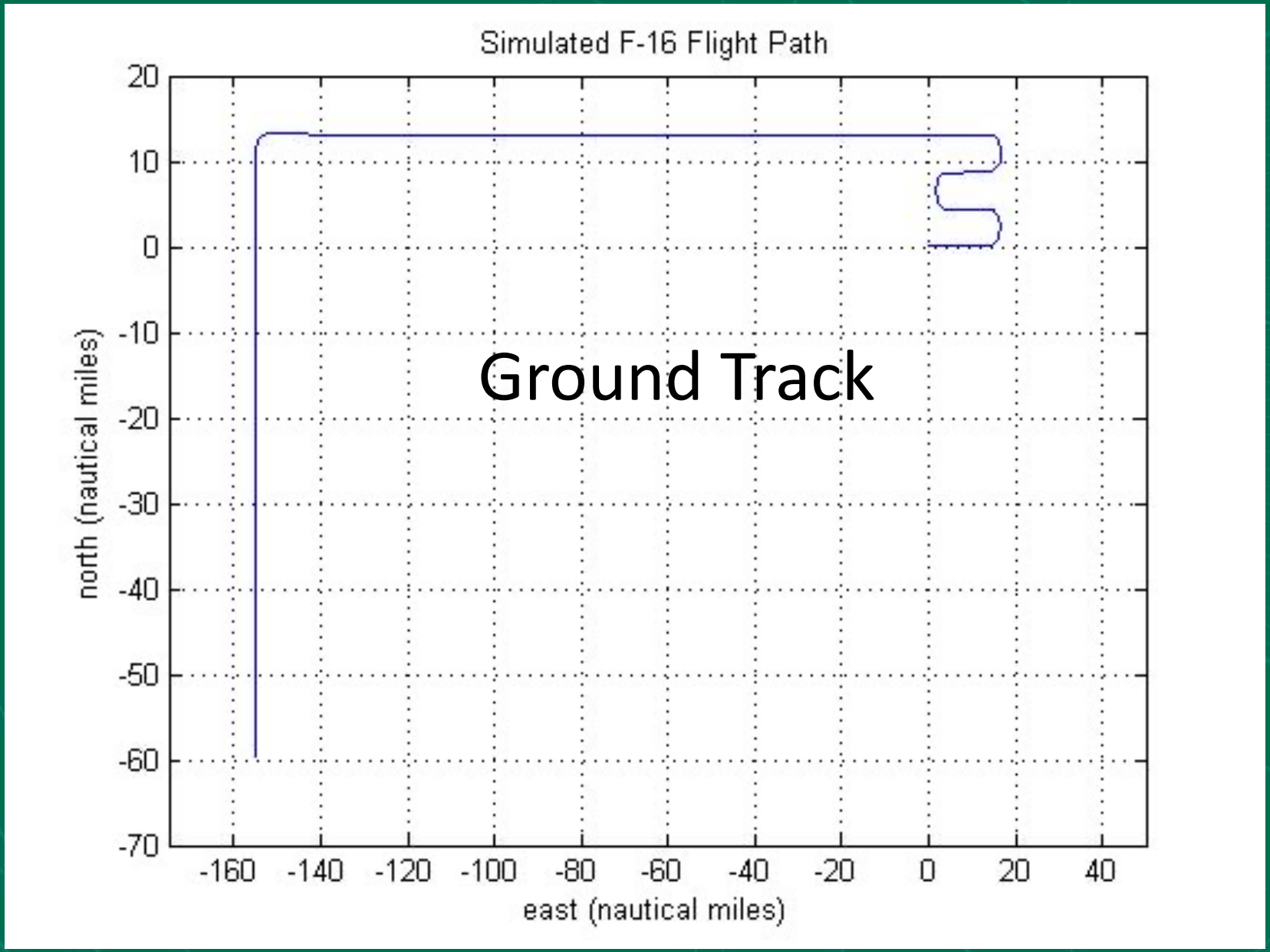
Tightly-Coupled GNSS/INS Integration

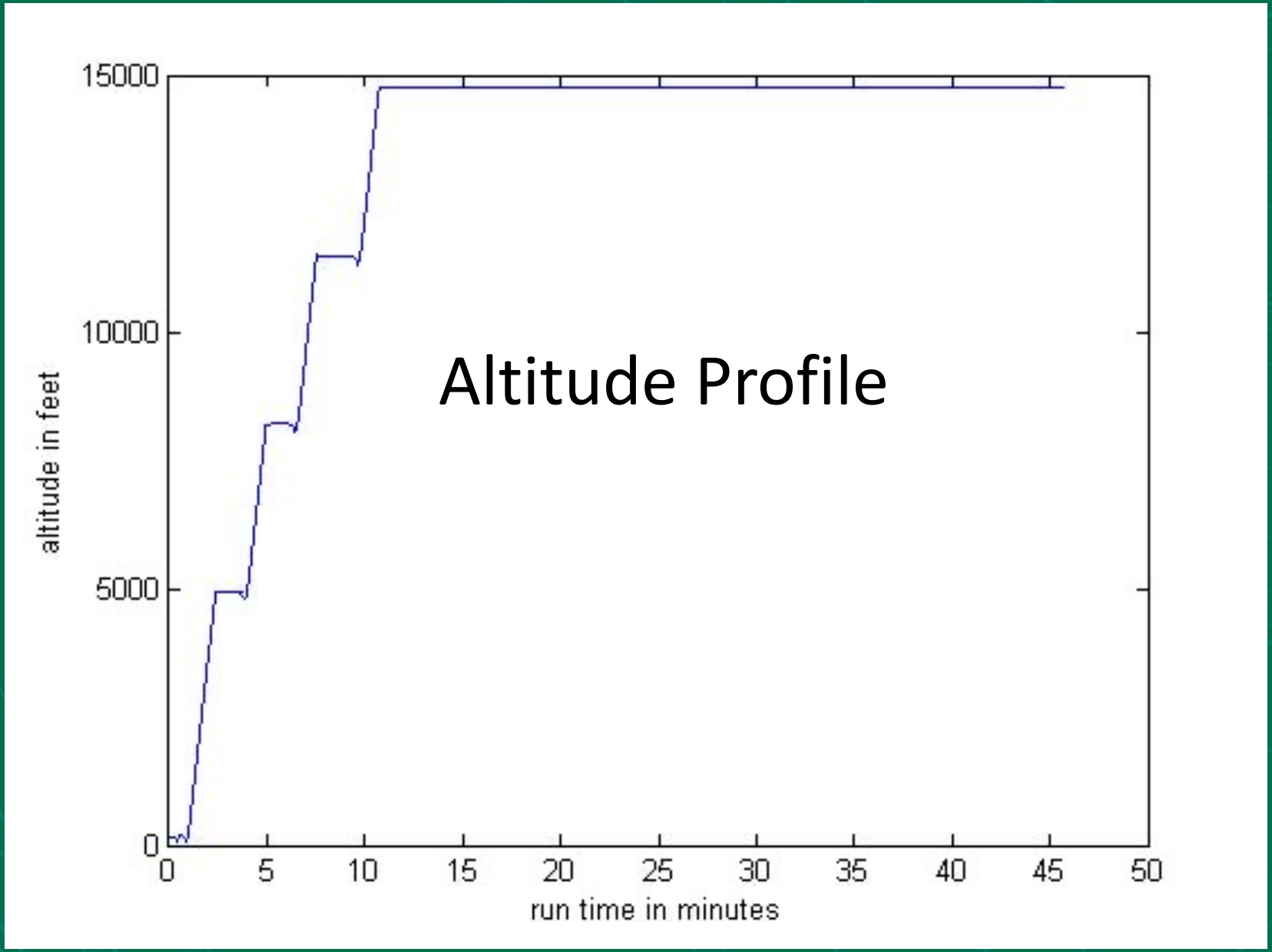


Adapted from: Groves, P., Principles of GNSS, Inertial and Multisensor Integrated Navigation Systems, 2008.

Simple Loosely-Coupled Example

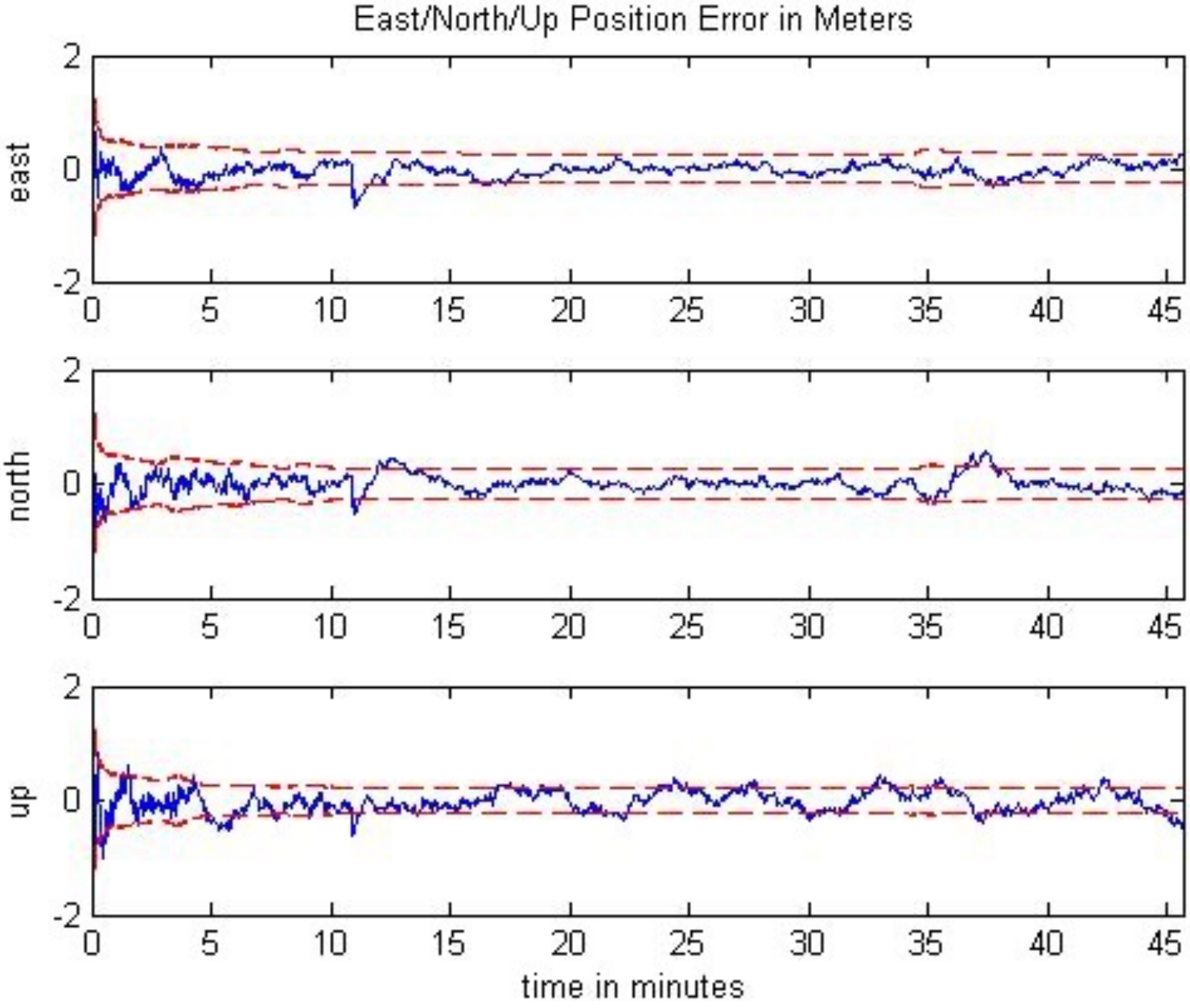
- Simulation of an F-16 trajectory
- Some s-turns as it gradually climbs up to an altitude of about 15,000 feet
- Thence proceeding due west for about 25 minutes and then a turn to the south
- Cruise speed is 389 knots (nautical miles per hour)

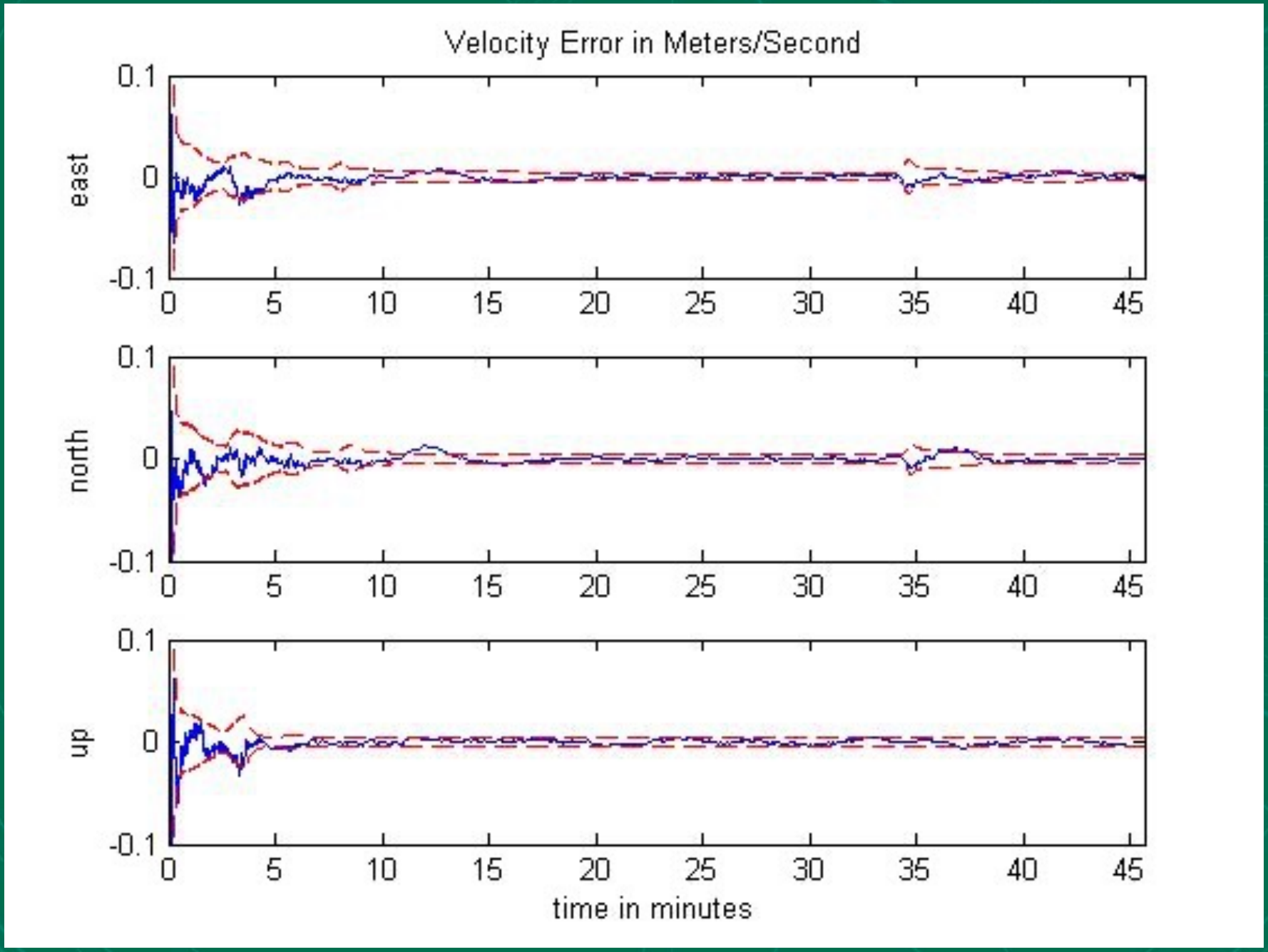

Simulated F-16 Flight Path
Ground Track
north (nautical miles)
east (nautical miles)
20
10
0
-10
-20
-30
-40
-50
-60
-70
-160
-140
-120
-100
-80
-60
-40
-20
0
20
40

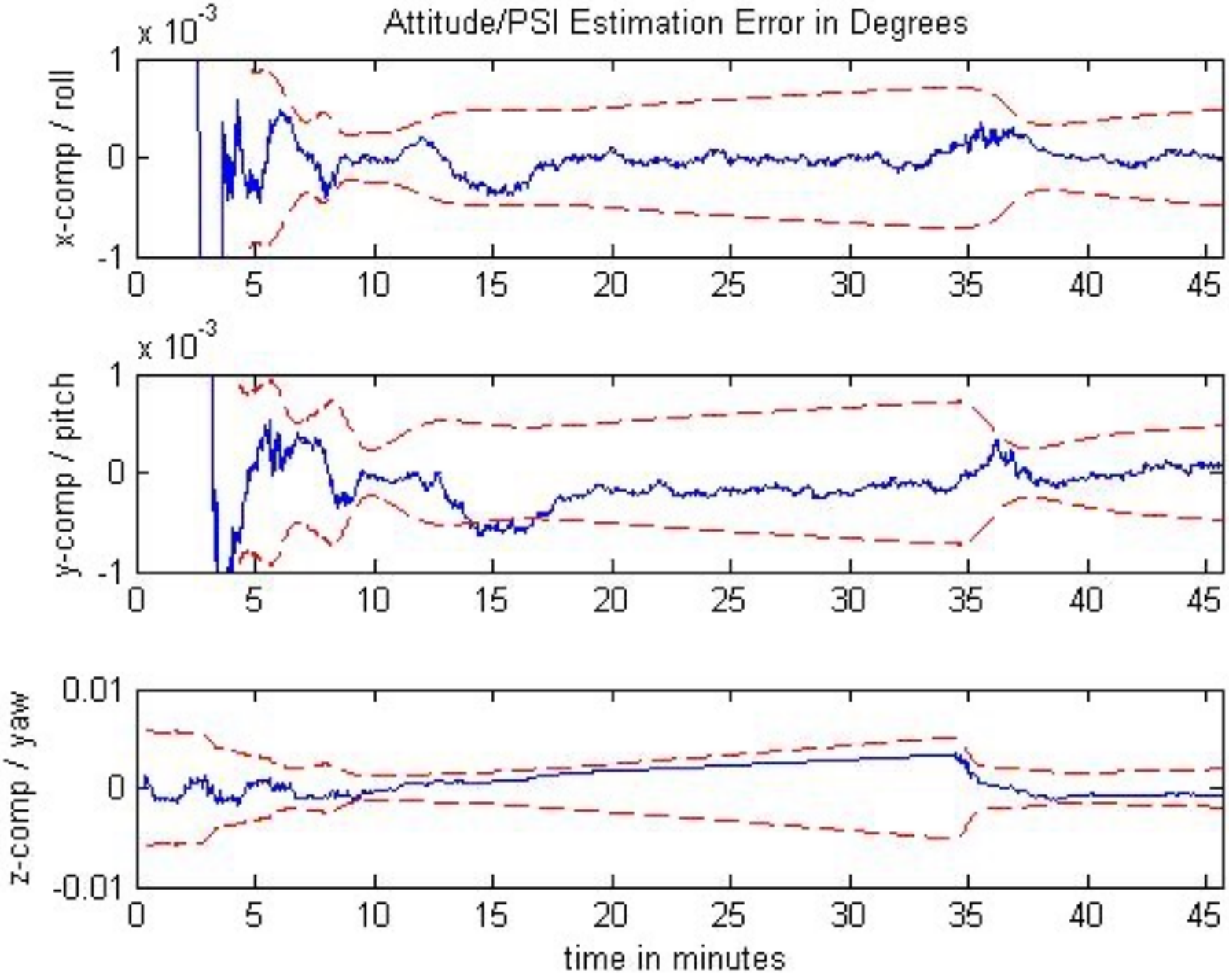

Altitude Profile
15000
10000
5000
0
altitude in feet
0
5
10
15
20
25
30
35
40
45
50
run time in minutes

Modeling the Nav-grade INS

- Accelerometers biases: ~ 50 micro-gs
- Gyro biases: ~ 0.01 degrees-per-hour
- Initial velocity error: 2 centimeters-per-second (north and east axes)
- Initial attitude determination error: 0.006 degrees (each axis)


East/North/Up Position Error in Meters
east
north
up
2
0
-2
0
5
10
15
20
25
30
35
40
45
time in minutes


Velocity Error in Meters/Second
east
north
up
0.1
0
-0.1
0
5
10
15
20
25
30
35
40
45
time in minutes


Attitude/PSI Estimation Error in Degrees
x 10^-3
x-comp / roll
y-comp / pitch
z-comp / yaw
time in minutes

Regarding State Observability …

In a loosely-coupled GPS-aided INS, the basic observable is the difference between GPS-derived position and INS-derived position

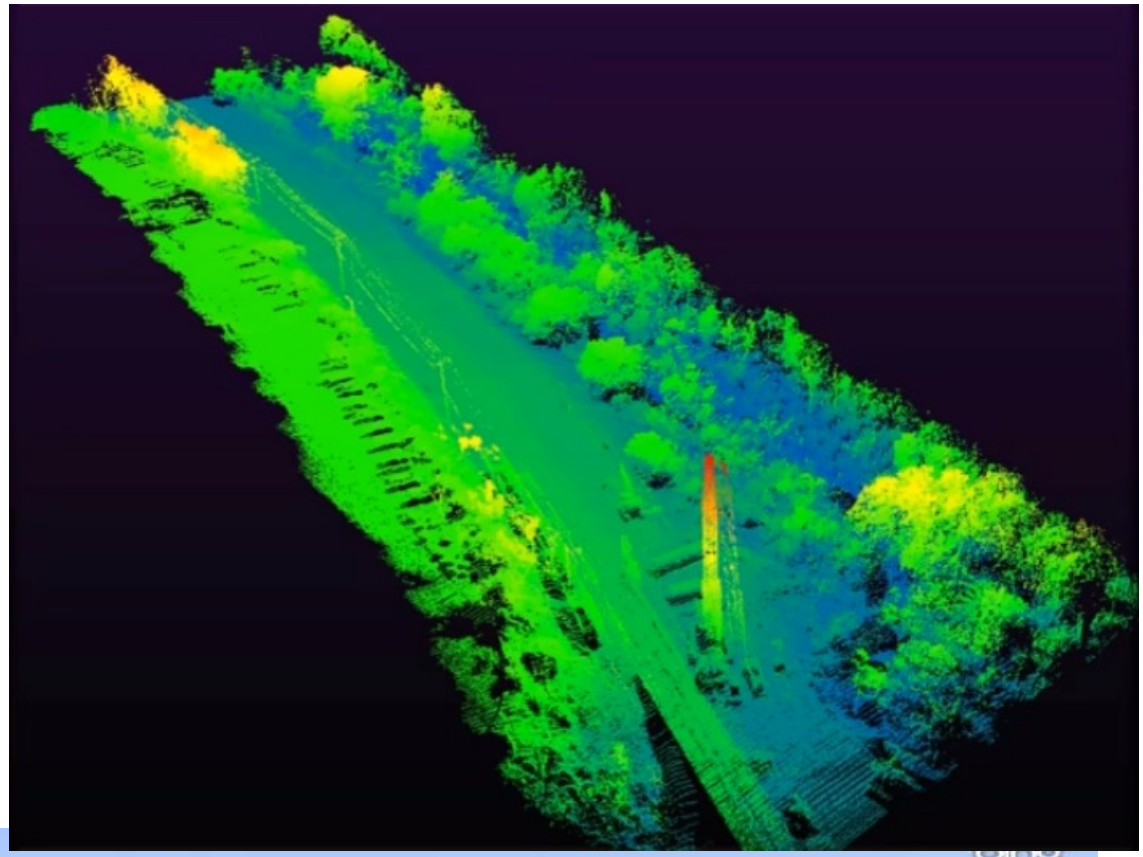
How does attitude error couple into position error?

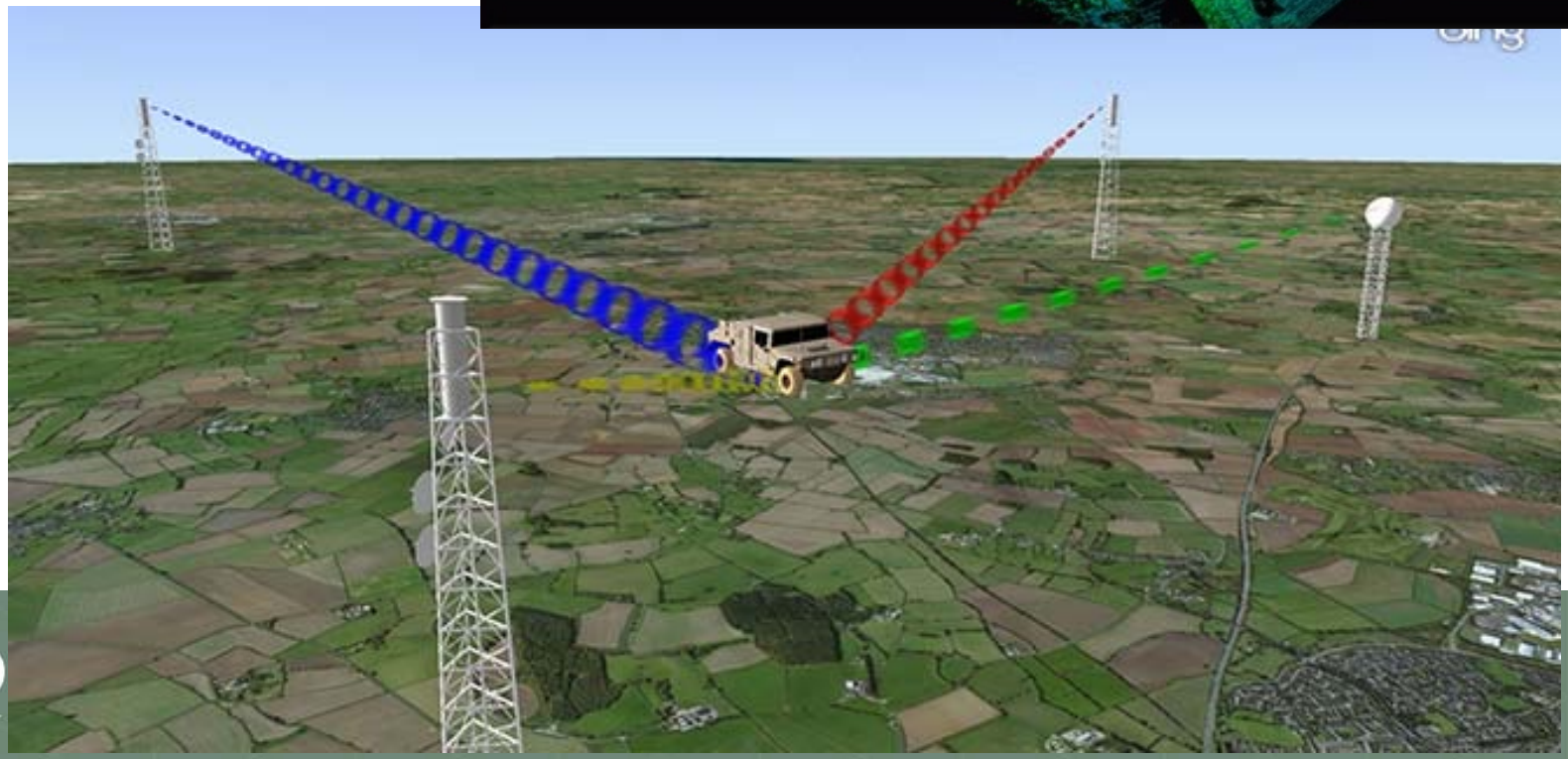
$$\underline{\delta \dot{R}}^t = -\underline{\omega}_{et}^t \times \underline{\delta R}^t + \underline{\delta V}^t$$

$$\underline{\delta \dot{V}}^c = C_b^c \underline{\delta f}^b + \underline{\delta g}^c - \underline{\psi}^p \times \underline{f}^c - \left(2\underline{\omega}_{ie}^c + \underline{\omega}_{ec}^c\right) \times \underline{\delta V}^c$$

$$\underline{\dot{\psi}}^p = -\underline{\omega}_{ip}^p \times \underline{\psi}^p - C_b^p \underline{\delta \omega}_{ib}^b$$

The Kalman framework is readily extensible to a variety of aiding sources





OHIO UNIVERSITY

Aiding sources may come and go, but the inertial measurement unit will remain

QUESTIONS?

Ohio University Convocation Center - Athens