# Lecture

# Kalman Filtering Example

## EE 570: Location and Navigation

Lecture Notes Update on April 12, 2016

Aly El-Osery and Kevin Wedeward, Electrical Engineering Dept., New Mexico Tech
In collaboration with
Stephen Bruder, Electrical & Computer Engineering, Embry-Riddle Aeronautical University

.1

## 1 Kalman Filter

### Review: System Model

$$\dot{\vec{x}}(t) = F(t)\vec{x}(t) + G(t)\vec{w}(t) \tag{1}$$

$$\vec{y}(t) = H(t)\vec{x}(t) + \vec{v}(t) \tag{2}$$

**System Discretization**

$$\Phi_{k-1} = e^{F_{k-1}\tau_s} \approx \mathcal{I} + F_{k-1}\tau_s \tag{3}$$

where $F_{k-1}$ is the average of $F$ at times $t$ and $t-\tau_s$, and first order approximation is used. Leading to

$$\vec{x}_k = \Phi_{k-1}\ \vec{x}_{k-1} + \vec{w}_{k-1} \tag{4}$$

$$\vec{z}_k = H_k\ \vec{x}_k + \vec{v}_k \tag{5}$$

where $\Phi_{k-1}$ is $(n \times n)$ transition matrix relating $\vec{x}_{k-1}$ to $\vec{x}_k$, $H_k$ is $(m \times n)$ matrix provides noiseless connection between measurement and state vectors.

.2

### Review: Assumptions

- $\vec{w}_k$ and $\vec{v}_k$ are drawn from a Gaussian distribution, uncorrelated have zero mean and statistically independent.

$$\mathbb{E}\{\vec{w}_k\vec{w}_i^T\} = \begin{cases} Q_k & i = k \\ 0 & i \neq k \end{cases} \tag{6}$$

$$\mathbb{E}\{\vec{v}_k\vec{v}_i^T\} = \begin{cases} R_k & i = k \\ 0 & i \neq k \end{cases} \tag{7}$$

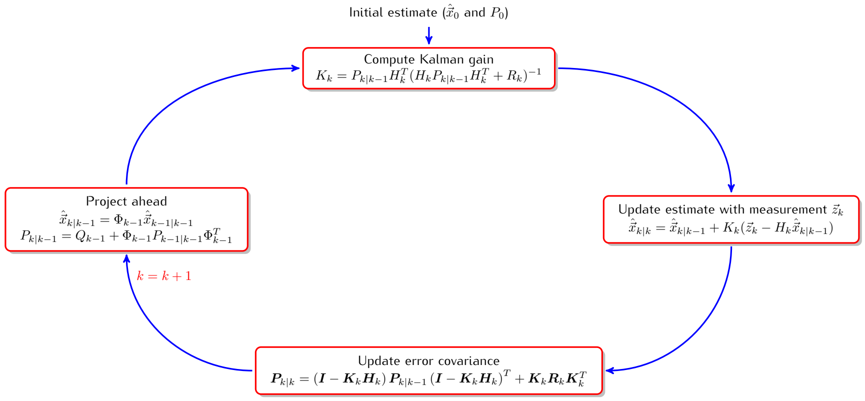
$$\mathbb{E}\{\vec{w}_k\vec{v}_i^T\} = \begin{cases} 0 & \forall i, k \end{cases} \tag{8}$$

- State covariance matrix

$$Q_{k-1} \approx \frac{1}{2}\left[\Phi_{k-1}G_{k-1}Q(t_{k-1})G_{k-1}^T\Phi_{k-1}^T + G_{k-1}Q(t_{k-1})G_{k-1}^T\right]\tau_s \tag{9}$$

.3

### Review: Kalman filter data flow




.4

### Remarks

- Kalman filter (KF) is optimal under the assumptions that the system is linear and the noise is uncorrelated
- Under these assumptions KF provides an unbiased and minimum variance estimate.
- If the Gaussian assumptions is not true, Kalman filter is biased and not minimum variance.
- If the noise is correlated we can augment the states of the system to maintain the uncorrelated requirement of the system noise.

.5

## 2 State Augmentation

### Correlated State Noise

Given a state space system

$$\dot{\vec{x}}_1(t) = F_1(t)\vec{x}_1(t) + G_1(t)\vec{w}_1(t)$$

$$\vec{y}_1(t) = H_1(t)\vec{x}_1(t) + \vec{v}_1(t)$$

As we have seen the noise $\vec{w}_1(t)$ may be non-white, e.g., correlated Gaussian noise, and as such may be modeled as

$$\dot{\vec{x}}_2(t) = F_2(t)\vec{x}_2(t) + G_2(t)\vec{w}_2(t)$$

$$\vec{w}_1(t) = H_2(t)\vec{x}_2(t)$$

.6

### Correlated State Noise

Define a new augmented state

$$\vec{x}_{aug} = \begin{pmatrix} \vec{x}_1(t) \\ \vec{x}_2(t) \end{pmatrix} \tag{10}$$

therefore,

$$\dot{\vec{x}}_{aug} = \begin{pmatrix} \dot{\vec{x}}_1(t) \\ \dot{\vec{x}}_2(t) \end{pmatrix} = \begin{pmatrix} F_1(t) & G_1H_2(t) \\ 0 & F_2(t) \end{pmatrix} \begin{pmatrix} \vec{x}_1(t) \\ \vec{x}_2(t) \end{pmatrix} + \begin{pmatrix} 0 \\ G_2(t) \end{pmatrix} \vec{w}_2(t) \tag{11}$$

and

$$\vec{y}(t) = \begin{pmatrix} H_1(t) & 0 \end{pmatrix} \begin{pmatrix} \vec{x}_1(t) \\ \vec{x}_2(t) \end{pmatrix} + \vec{v}_1(t) \tag{12}$$

.7

### Correlated Measurement Noise

Given a state space system

$$\dot{\vec{x}}_1(t) = F_1(t)\vec{x}_1(t) + G_1(t)\vec{w}(t)$$

$$\vec{y}_1(t) = H_1(t)\vec{x}_1(t) + \vec{v}_1(t)$$

In this case the measurement noise $\vec{v}_1$ may be correlated

$$\dot{\vec{x}}_2(t) = F_2(t)\vec{x}_2(t) + G_2(t)\vec{v}_2(t)$$

$$\vec{v}_1(t) = H_2(t)\vec{x}_2(t)$$

.8

### Correlated Measurement Noise

Define a new augmented state

$$\vec{x}_{aug} = \begin{pmatrix} \vec{x}_1(t) \\ \vec{x}_2(t) \end{pmatrix} \tag{13}$$

therefore,

$$\dot{\vec{x}}_{aug} = \begin{pmatrix} \dot{\vec{x}}_1(t) \\ \dot{\vec{x}}_2(t) \end{pmatrix} = \begin{pmatrix} F_1(t) & 0 \\ 0 & F_2(t) \end{pmatrix} \begin{pmatrix} \vec{x}_1(t) \\ \vec{x}_2(t) \end{pmatrix} + \begin{pmatrix} G_1(t) & 0 \\ 0 & G_2(t) \end{pmatrix} \begin{pmatrix} \vec{w}(t) \\ \vec{v}_2(t) \end{pmatrix} \tag{14}$$

and

$$\vec{y}(t) = \begin{pmatrix} H_1(1) & H_2(t) \end{pmatrix} \begin{pmatrix} \vec{x}_1(t) \\ \vec{x}_2(t) \end{pmatrix} \tag{15}$$

.9

# 3 Example

### Design Example

You are to design a system that estimates the position and velocity of a moving point in a straight line. You have:

1. an accelerometer corrupted with noise
2. an aiding sensor allowing you to measure absolute position that is also corrupted with noise.

.10

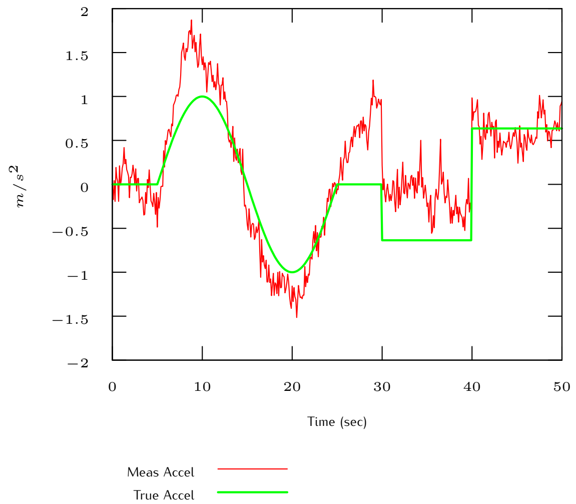
### Specification

- Sampling Rate Fs = 100Hz.
- Accelerometer specs
    1. VRW = $1mg/\sqrt{Hz}$.
    2. BI = $7mg$ with correlation time 6s.
- Position measurement is corrupted with WGN. $\sim \mathcal{N}(0, \sigma_p^2)$, where $\sigma_p = 2.5$m

.11

## Input - Acceleration

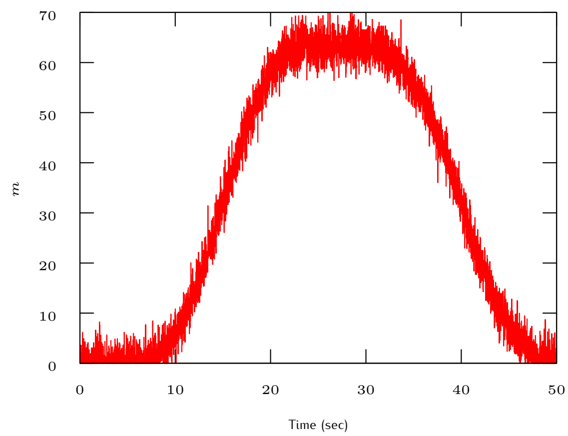
True Acceleration and Acceleration with Noise




.12

## Aiding Position Measurement

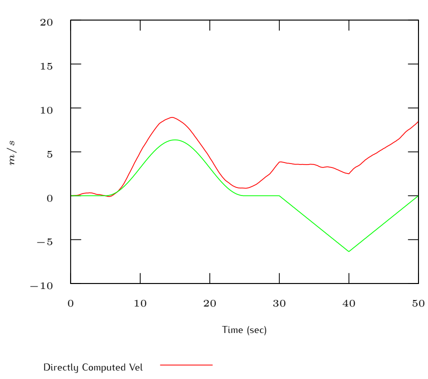
Absolute position measurement corrupted with noise



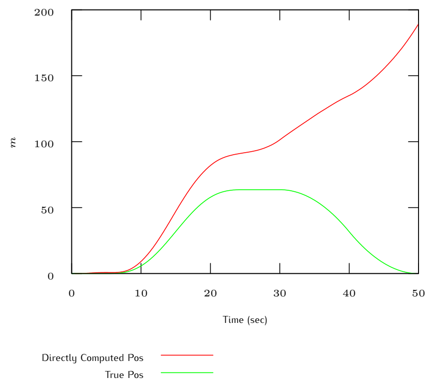

.13

## Computed Position and Velocity

Using only the acceleration measurement and an integration approach to compute the velocity, then integrate again to get position.




Velocity




Position

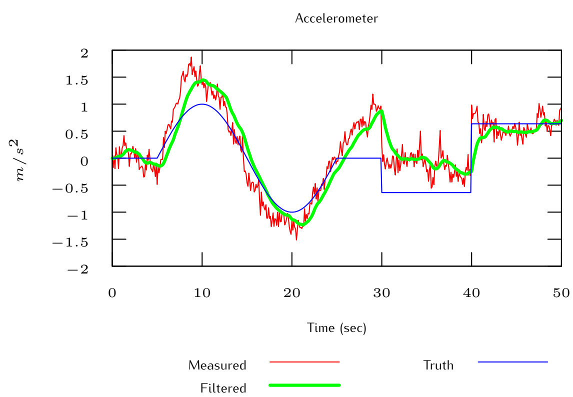
.14

## Different Approaches

1. Clean up the noisy input to the system by filtering
2. Use Kalman filtering techniques with
   - A model of the system dynamics (too restrictive)
   - A model of the error dynamics and correct the system output in
     - open-loop configuration, or
     - closed-loop configuration.

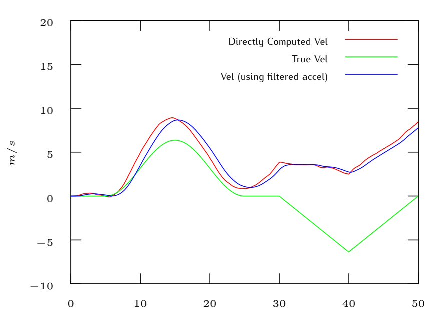
.15

## Approach 1 — Filtered input  Filtered Accel Measurement

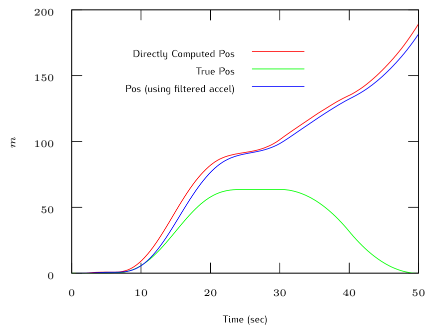



.16

## Approach 1 — Filtered input  Position and Velocity



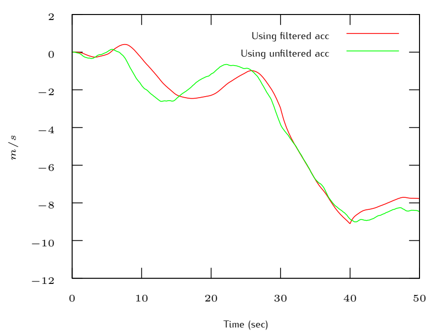

Velocity



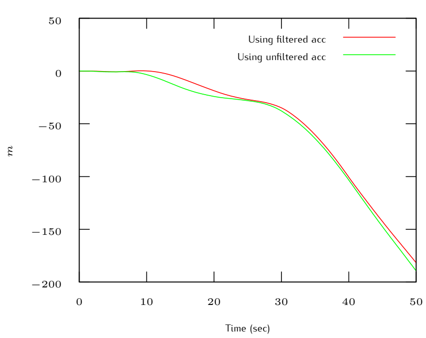

Position

.17

## Approach 1 — Filtered input Position and Velocity Errors



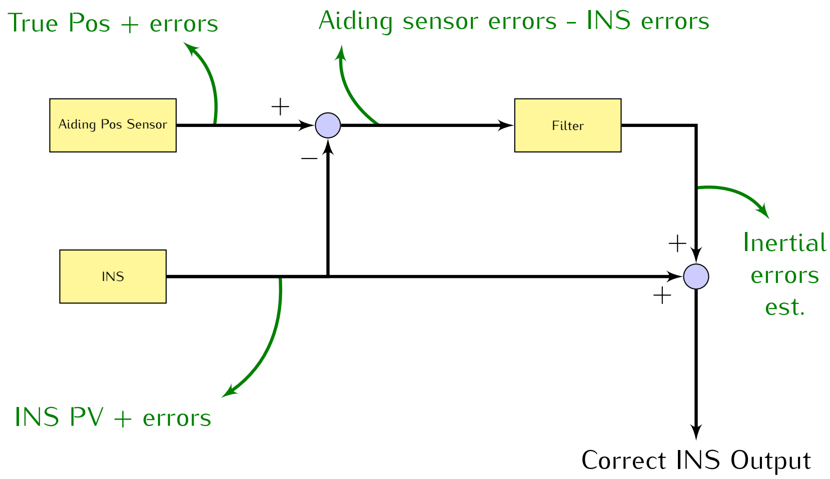

Velocity Error




Position Error

.18

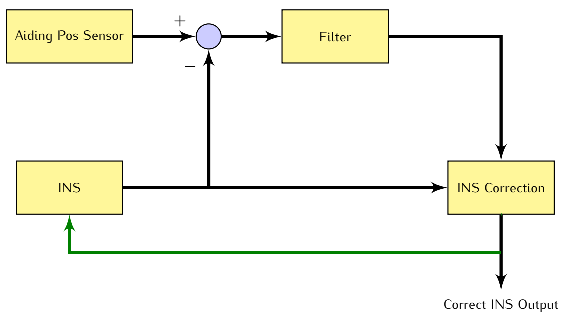
## Open-Loop Integration




.19

### Closed-Loop Integration

If error estimates are fedback to correct the INS mechanization, a reset of the state estimates becomes necessary.




.20

### Covariance Matrices

- State noise covariance matrix (continuous)

$$\mathbb{E}\{\vec{w}(t)\vec{w}^T(\tau)\} = Q(t)\delta(t-\tau)$$

- State noise covariance matrix (discrete)

$$\mathbb{E}\{\vec{w}_k\vec{w}_i^T\} = \begin{cases} Q_k & i = k \\ 0 & i \neq k \end{cases}$$

- Measurement noise covariance matrix

$$\mathbb{E}\{\vec{v}_k\vec{v}_i^T\} = \begin{cases} R_k & i = k \\ 0 & i \neq k \end{cases}$$

- Initial error covariance matrix

$$P_0 = \mathbb{E}\{(\vec{x}_0 - \hat{\vec{x}}_0)(\vec{x}_0 - \hat{\vec{x}}_0)^T\} = \mathbb{E}\{\vec{e}_0\hat{\vec{e}}_0^T\}$$

.21

### System Modeling

The position, velocity and acceleration may be modeled using the following kinematic model.

$$\begin{aligned} \dot{p}(t) &= v(t) \\ \dot{v}(t) &= a(t) \end{aligned} \tag{16}$$

where $a(t)$ is the input. Therefore, our estimate of the position is $\hat{p}(t)$ that is the double integration of the acceleration. .22

### Sensor Model

Assuming that the accelerometer sensor measurement may be modeled as

$$\tilde{a}(t) = a(t) + b(t) + w_a(t) \tag{17}$$

and the bias is Markov, therefore

$$\dot{b}(t) = -\frac{1}{T_c} b(t) + w_b(t) \tag{18}$$

where $w_a(t)$ and $w_b(t)$ are zero mean WGN with variances, respectively, $Fs \cdot VRW^2$

$$\mathbb{E}\{w_b(t) w_b(t+\tau)\} = Q_b(t)\delta(t-\tau) \tag{19}$$

$$Q_b(t) = \frac{2\sigma_{BI}^2}{T_c} \tag{20}$$

and $T_c$ is the correlation time and $\sigma_{BI}$ is the bias instability.

*Make sure that the $VRW$ and $\sigma_{BI}$ are converted to have SI units.* .23

### Error Mechanization

Define error terms as

$$\delta p(t) = p(t) - \hat{p}(t), \tag{21}$$

$$\begin{aligned} \delta \dot{p}(t) &= \dot{p}(t) - \dot{\hat{p}}(t) \\ &= v(t) - \hat{v}(t) \\ &= \delta v(t) \end{aligned} \tag{22}$$

and

$$\begin{aligned} \delta \dot{v}(t) &= \dot{v}(t) - \dot{\hat{v}}(t) \\ &= a(t) - \hat{a}(t) \\ &= -b(t) - w_a(t) \end{aligned} \tag{23}$$

where $b(t)$ is modeled as shown in Eq. 18 .24

### State Space Formulation

$$\begin{aligned} \dot{\vec{x}}(t) &= \begin{pmatrix} \delta \dot{p}(t) \\ \delta \dot{v}(t) \\ \dot{b}(t) \end{pmatrix} = \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & -1 \\ 0 & 0 & -\frac{1}{T_c} \end{pmatrix} \begin{pmatrix} \delta p(t) \\ \delta v(t) \\ b(t) \end{pmatrix} + \begin{pmatrix} 0 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 0 \\ w_a(t) \\ w_b(t) \end{pmatrix} \\ &= F(t)\vec{x}(t) + G(t)\vec{w}(t) \end{aligned} \tag{24}$$

.25

### Covariance Matrix

- The continuous state noise covariance matrix $Q(t)$ is

$$Q(t) = \begin{pmatrix} 0 & 0 & 0 \\ 0 & VRW^2 & 0 \\ 0 & 0 & \frac{2\sigma_{BI}^2}{Tc} \end{pmatrix} \quad (25)$$

- The measurement noise covariance matrix is $R = \sigma_p^2$, where $\sigma_p$ is the standard deviation of the noise of the absolute position sensor.

.26

### Discretization

Now we are ready to start the implementation but first we have to discretize the system.

$$\vec{x}(k+1) = \Phi(k)\vec{x}(k) + \vec{w}_d(k) \quad (26)$$

where

$$\Phi(k) \approx \mathcal{I} + Fdt \quad (27)$$

with the measurement equation

$$y(k) = H\vec{x} + w_p(k) = \delta p(k) + w_p(k) \quad (28)$$

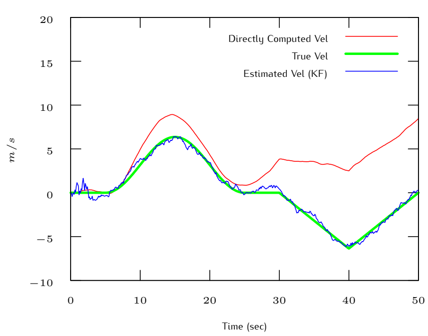
where $H = [1\ 0\ 0]$. The discrete $Q_d$ is approximated as

$$\begin{aligned} Q_{k-1} \approx \frac{1}{2}[\Phi_{k-1}G(t_{k-1})Q(t_{k-1})G^T(t_{k-1}))\Phi_{k-1}^T + \\ G(t_{k-1})Q(t_{k-1})G^T(t_{k-1})]dt \end{aligned} \quad (29)$$

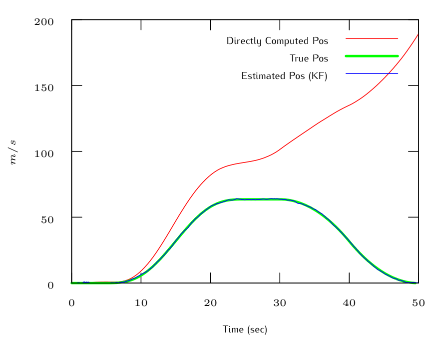
.27

### Approach 2 — Open-Loop Compensation  Position and Velocity

**Open-loop Correction**
Best estimate = INS out (pos & vel) + KF est error (pos & vel)



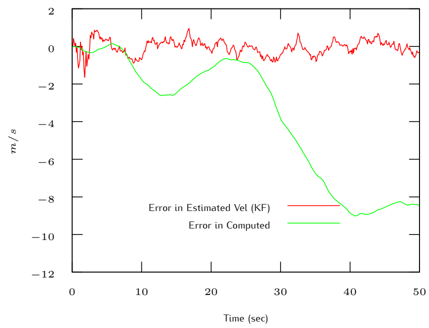

Velocity



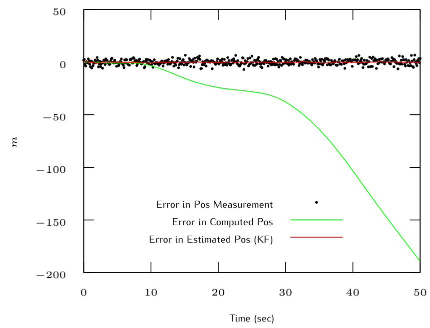

Position

.28

## Approach 2 — Open-Loop Compensation  Position and Velocity Errors



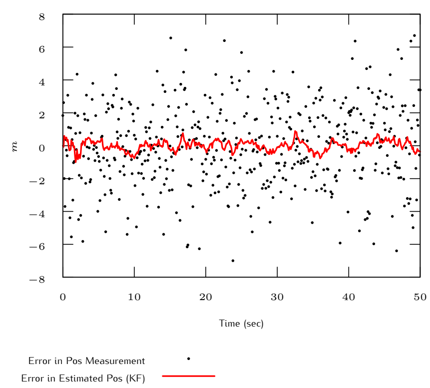

Velocity Error



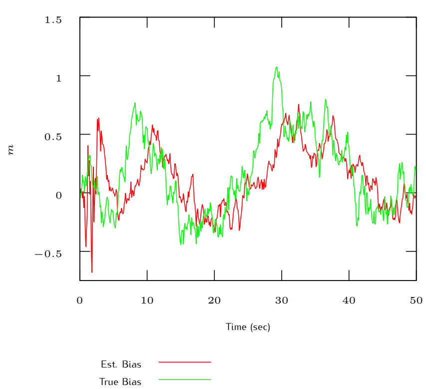

Position Error

.29

## Approach 2 — Open-Loop Compensation  Pos Error & Bias Estimate




Position Error

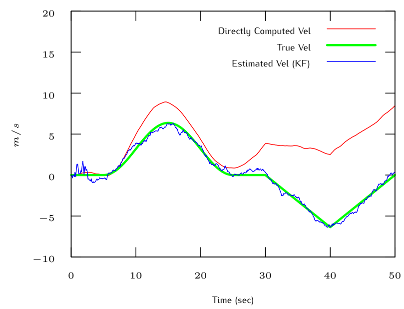



Bias

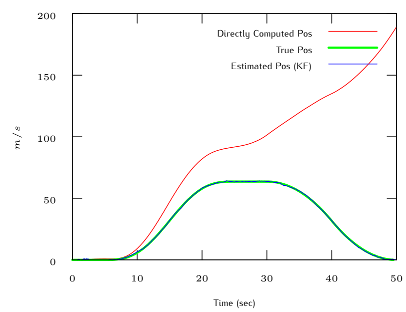
.30

## Approach 3 — Closed-Loop Compensation

**Closed-loop Correction**
Best estimate = INS out (pos,vel, & bias) + KF est error (pos, vel & bias) Use best estimate on next iteration of INS Accel estimate = accel meas - est bias Reset state estimates before next call to KF



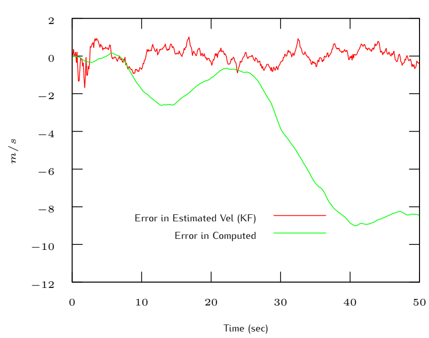

Velocity



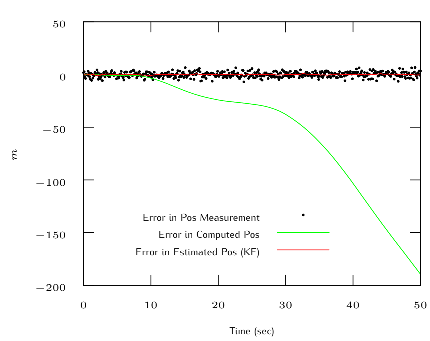

Position

.31

## Approach 3 — Closed-Loop Compensation Position and Velocity Errors

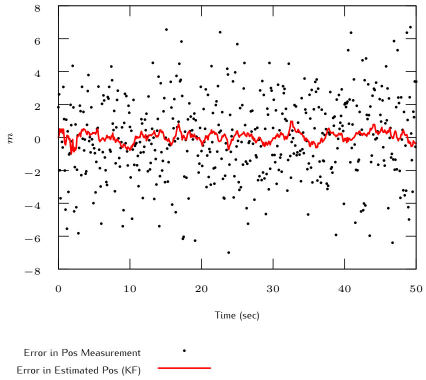



Velocity Error

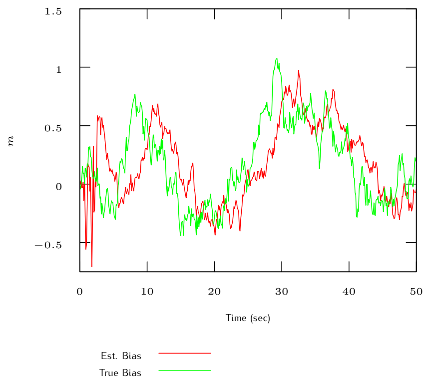



Position Error

.32




Position Error




Bias

.33