Lecture

Kalman Filtering Example

EE 570: Location and Navigation

Lecture Notes Update on April 3, 2014

Aly El-Osery, Electrical Engineering Dept., New Mexico Tech
Stephen Bruder, Electrical & Computer Engineering, Embry-Riddle Aeronautical University

.1

1 Kalman Filter

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 the uncorrelated requirement of the system noise.

.2

2 State Augmentation

Correlated State Noise

Given a state space system

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

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

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

$$\dot{\vec{x}}_2(t) = F_2(t)\vec{x}_2(t) + G_2(t)\vec{w}_2(t)$$
$$\vec{w}_1(t) = H_2(t)\vec{x}_2(t)$$

.3

Correlated State Noise

Define a new augmented state

$$\vec{x}_{aug} = \begin{pmatrix} \vec{x}_1(t) \\ \vec{x}_2(t) \end{pmatrix} \tag{1}$$

therefore,

$$\dot{\vec{x}}_{aug} = \begin{pmatrix} \dot{\vec{x}}_1(t) \\ \dot{\vec{x}}_2(t) \end{pmatrix} = \begin{pmatrix} F_1(t) & G_1 H_2(t) \\ 0 & F_2 \end{pmatrix} \begin{pmatrix} \vec{x}_1(t) \\ \vec{x}_2(t) \end{pmatrix} + \begin{pmatrix} 0 \\ G_2(t) \end{pmatrix} \vec{w}_2(t) \tag{2}$$

and

$$\vec{y}(t) = \begin{pmatrix} H_1 & 0 \end{pmatrix} \begin{pmatrix} \vec{x}_1(t) \\ \vec{x}_2(t) \end{pmatrix} + \vec{v}(t) \tag{3}$$

.4

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

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

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

.5

Correlated Measurement Noise

Define a new augmented state

$$\vec{x}_{aug} = \begin{pmatrix} \vec{x}_1(t) \\ \vec{x}_2(t) \end{pmatrix} \tag{4}$$

therefore,

$$\dot{\vec{x}}_{aug} = \begin{pmatrix} \dot{\vec{x}}_1(t) \\ \dot{\vec{x}}_2(t) \end{pmatrix} = \begin{pmatrix} F_1(t) & 0 \\ 0 & F_2 \end{pmatrix} \begin{pmatrix} \vec{x}_1(t) \\ \vec{x}_2(t) \end{pmatrix} + \begin{pmatrix} G_1(t) & 0 \\ 0 & G_2(t) \end{pmatrix} \begin{pmatrix} \vec{w}(t) \\ \vec{v}_2 \end{pmatrix} \tag{5}$$

and

$$\vec{y}(t) = \begin{pmatrix} H_1 & H_2 \end{pmatrix} \begin{pmatrix} \vec{x}_1(t) \\ \vec{x}_2(t) \end{pmatrix} \tag{6}$$

.6

3 Example

Problem Statement

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.

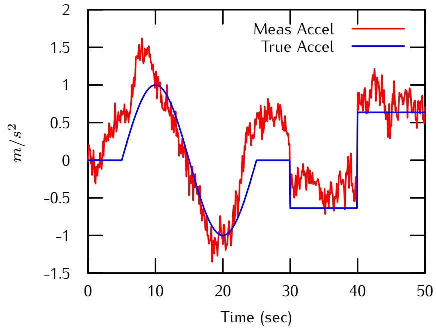
.7

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

.8

Input - Acceleration

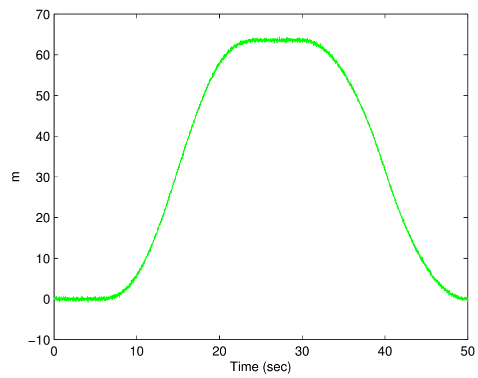


True Acceleration and Acceleration with Noise

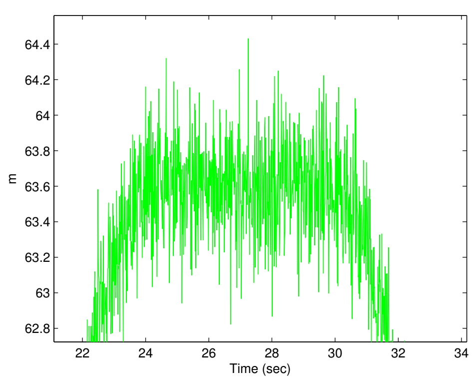
.9

Aiding Position Measurement

Absolute position measurment corrupted with noise



zoomed version



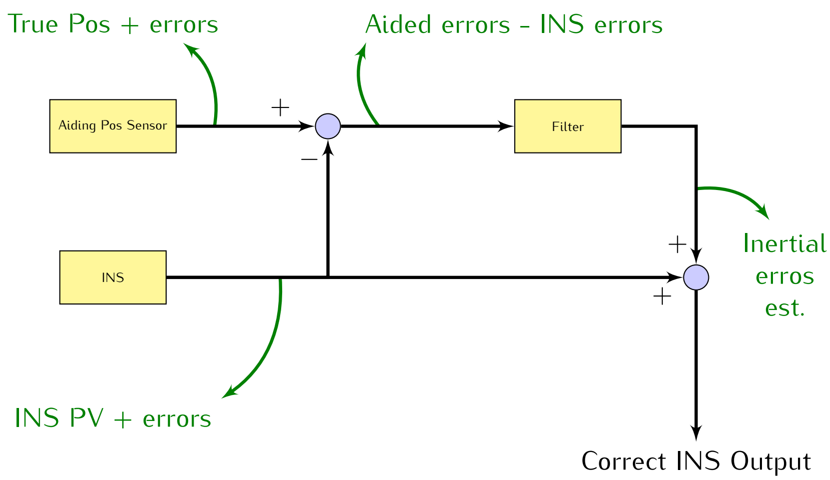
.10

Different Approaches

1. Clean up the noisy input to the system.
2. Use Kalamn filtering techiques with
   - A model of the system dyanmics (too ristrictive)
   - A model of the error dyanmics and correct the system output in
     - open-loop configuration, or
     - closed-loop configuration.

.11

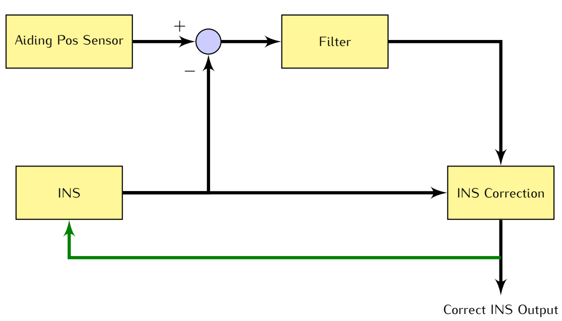
Open-Loop Integration



.12

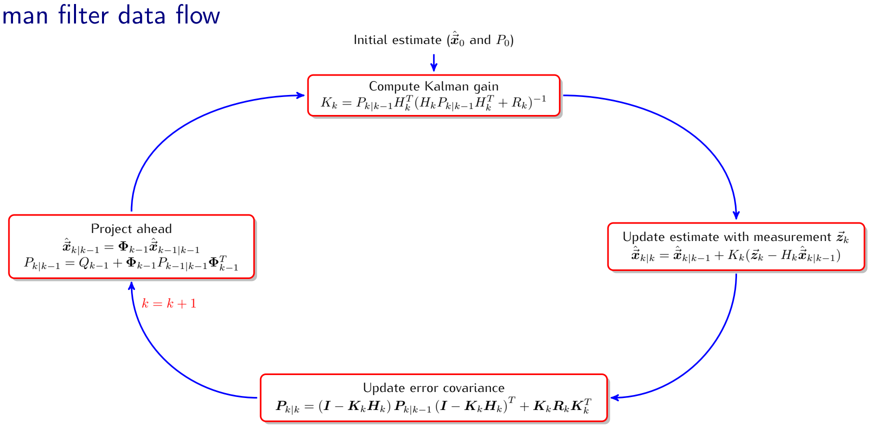
Closed-Loop Integration

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



.13

Kalman filter data flow



.14

Covariance Matrices

- State noise covariance matrix (continuous)

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

- State noise covariance matrix (discrete)

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

- Measurement noise covariance matrix

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

- Error covariance matrix

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

.15

System modeling

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

$$\begin{aligned} \dot{p}(t) &= v(t) \\ \dot{v}(t) &= a(t) \end{aligned} \tag{7}$$

where $a(t)$ is the input. .16

Measurement Model

Assuming that the accelerometer sensor measurement may be modeled as

$$\tilde{a}(t) = a(t) + b(t) + w_a(t) \tag{8}$$

and the bias is Markov, therefore

$$\dot{b}(t) = -\frac{1}{T_c} b(t) + w_b(t) \tag{9}$$

where both $w_a(t)$ and $w_b(t)$ are WGN. In addition we have the position measurment from which we can derive a delta position

$$\delta p(t) = p(t) - \hat{p}(t) \tag{10}$$

where $\hat{p}$ is the derived position by double integrating the measured acceleration. .17

Error Mechanization

Define the position and velocity error terms as

$$\begin{aligned} \delta \dot{p}(t) &= \dot{p}(t) - \dot{\hat{p}}(t) \\ &= v(t) - \hat{v}(t) \\ &= \delta v(t) \end{aligned} \tag{11}$$

and

$$\begin{aligned} \delta \dot{v}(t) &= \dot{v}(t) - \dot{\hat{v}}(t) \\ &= a(t) - \hat{a}(t) \\ &= -b(t) - w_a(t) \end{aligned} \tag{12}$$

where $b(t)$ is modeled as shown in Eq. 9 .18

First Order Markov Noise

State Equation

$$\dot{b}(t) = -\frac{1}{T_c}b(t) + w_b(t) \tag{13}$$

Autocorrelation Function

$$\mathbb{E}\{b(t)b(t+\tau)\} = \sigma^2 e^{-|\tau|/T_c} \tag{14}$$

where

$$\mathbb{E}\{w_b(t)w_b(t+\tau)\} = Q_b(t)\delta(t-\tau) \tag{15}$$

$$Q_b(t) = \frac{2\sigma_{BI}^2}{T_c} \tag{16}$$

and T_c is the correlation time and σ_{BI} is the bias instability. .19

State Space Formulation

$$\begin{aligned}\dot{\vec{x}}(t) &= \begin{pmatrix}\delta\dot{p}(t)\\ \delta\dot{v}(t)\\ \dot{b}(t)\end{pmatrix} = \begin{pmatrix}0 & 1 & 0\\ 0 & 0 & -1\\ 0 & 0 & -\frac{1}{T_c}\end{pmatrix}\begin{pmatrix}\delta p(t)\\ \delta v(t)\\ b(t)\end{pmatrix} + \begin{pmatrix}0 & 0 & 0\\ 0 & -1 & 0\\ 0 & 0 & 1\end{pmatrix}\begin{pmatrix}0\\ w_a(t)\\ w_b(t)\end{pmatrix}\\ &= F(t)\vec{x}(t) + G(t)\vec{w}(t)\end{aligned} \tag{17}$$

.20

Covariance Matrics

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

$$Q(t) = \begin{pmatrix}0 & 0 & 0\\ 0 & VRW^2 & 0\\ 0 & 0 & \frac{2\sigma_{BI}^2}{Tc}\end{pmatrix} \tag{18}$$

  make sure that the VRW and σ_{BI} are converted to have SI units.
- The measurement noise covariance matrix is $R = \sigma_p^2$

.21

Discretization

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

$$\vec{x}(k+1) = \Phi(k)\vec{x}(k) + \vec{w}_d \tag{19}$$

where

$$\Phi(k) \approx \mathcal{I} + Fdt \tag{20}$$

with the measurement equation

$$y(k) = H\vec{x} + w_p(k) = \delta p(k) + w_p(k) \tag{21}$$

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

$$\begin{aligned}Q_{k-1} \approx \frac{1}{2}[\Phi_{k-1}G(t_{k-1})Q(t_{k-1})G^T(t_{k-1}))\Phi_{k-1}^T +\\ G(t_{k-1})Q(t_{k-1})G^T(t_{k-1})]dt\end{aligned} \tag{22}$$

.22

Discrete First Order Markov Noise

State Equation

$$b_k = e^{-\frac{1}{T_c}dt} b_{k-1} + w_{bd,k-1} \tag{23}$$

System Covariance Matrix

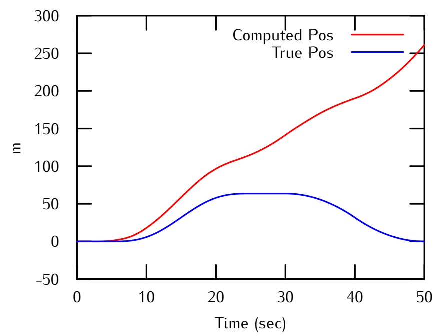
$$Q_{bd} = \sigma_{BI}^2 [1 - e^{-\frac{2}{T_c}dt}] \tag{24}$$

where w_{bd} is the discrete noise for the bias.

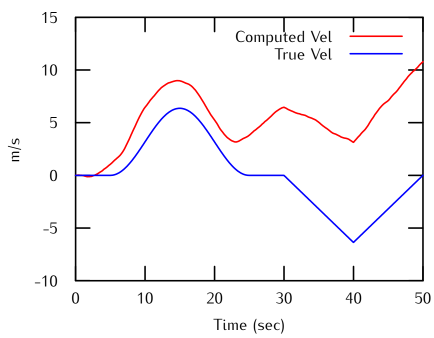
You should use Eqs. 23 and 24 to overwrite their values in Φ and Q_d since they don't need to be approximated. .23

Computed Position and Velocity

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

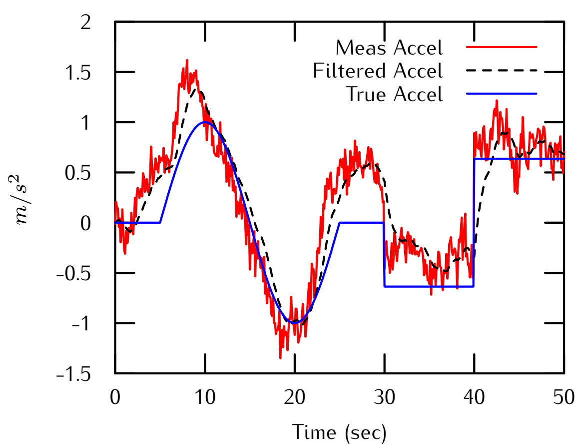


Position



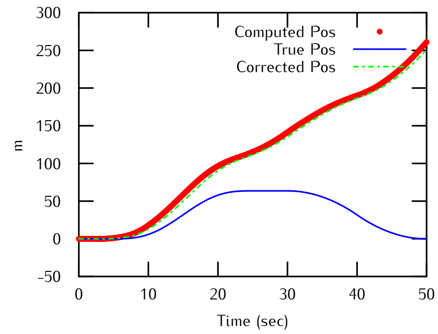
Velocity .24

Approach 1 — Filtered input Filtered Accel Measurement

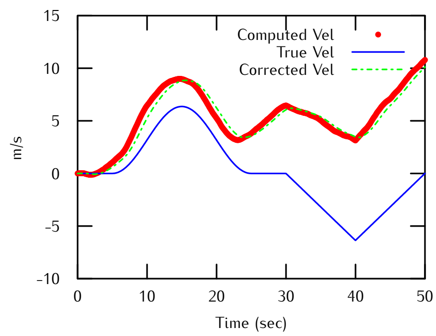


.25

Approach 1 — Filtered input Position and Velocity



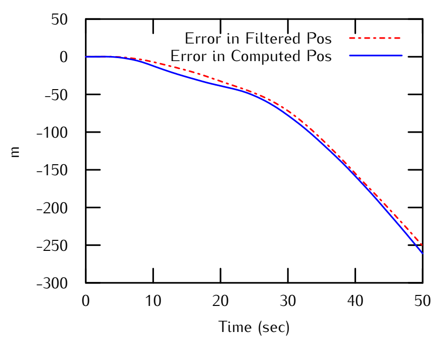
Position



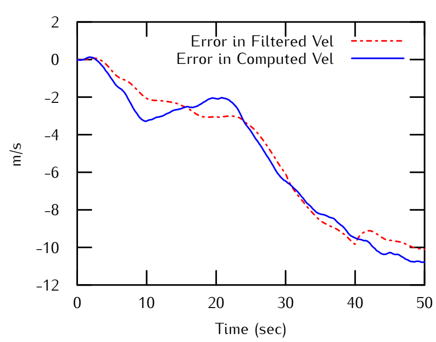
Velocity

.26

Approach 1 — Filtered input Position and Velocity Errors



Position Error



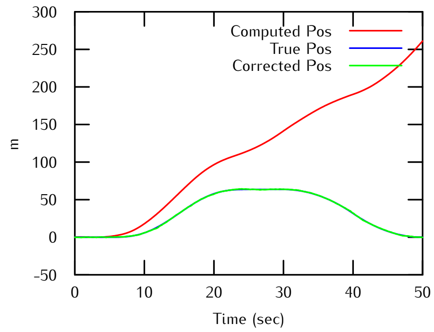
Velocity Error

.27

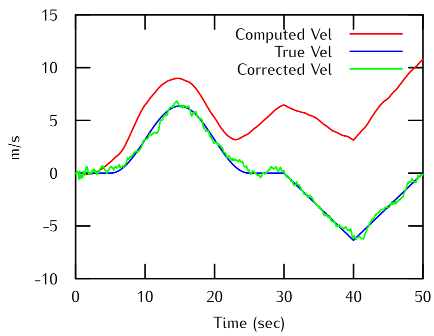
Approach 2 — Open-Loop Compensation Position and Velocity

Open-loop Correction

Best estimate = INS out (pos & vel) + KF est error (pos & vel)



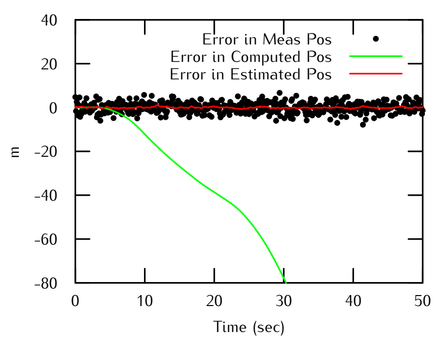
Position



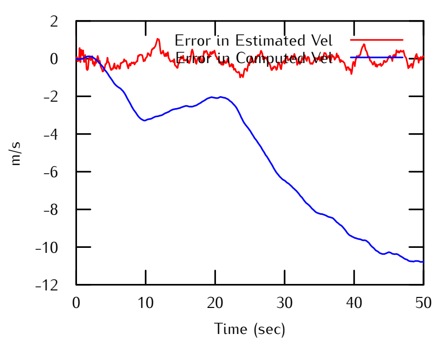
Velocity

.28

Approach 2 — Open-Loop Compensation Position and Velocity Errors



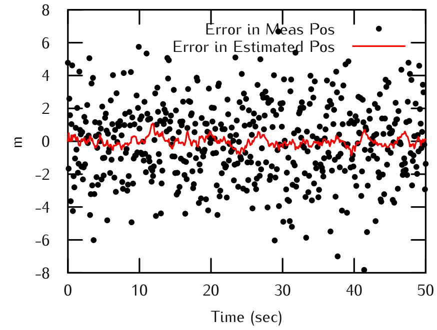
Position Error



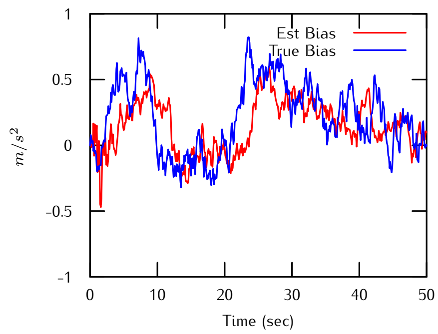
Velocity Error

.29

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



Position Error

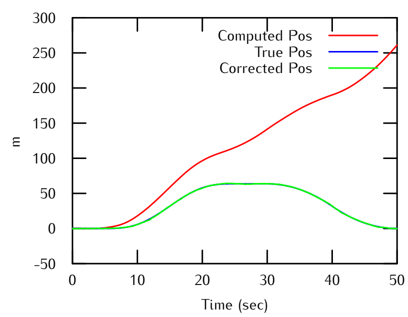


Bias

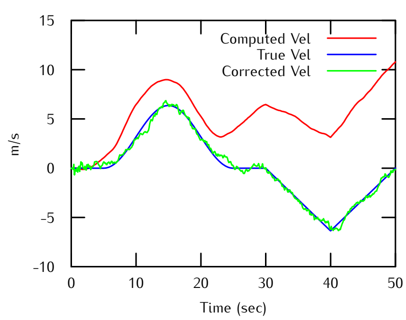
.30

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



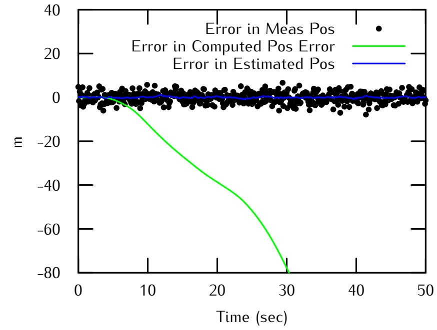
Position



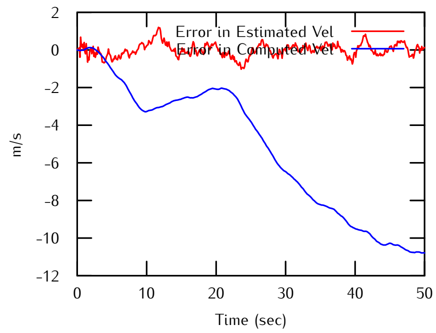
Velocity

.31

Approach 3 — Closed-Loop Compensation Position and Velocity Errors

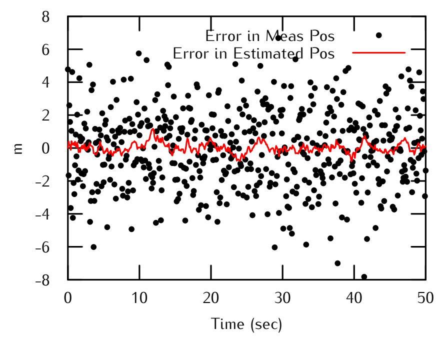


Position Error

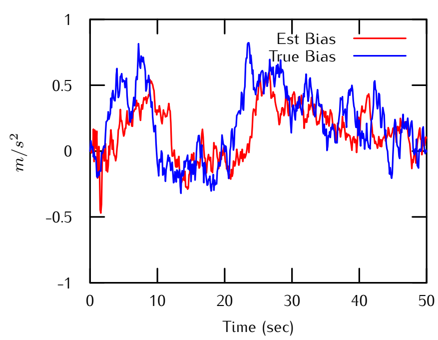


Velocity Error

Approach 3 — Closed-Loop Compensation Pos Error & Bias Estimate



Position Error



Bias