One Dimensional INS Design Example

Jay A. Farrell *

March 16, 2021

Abstract

Many students try to jump into very challenging Kalman filter designs without ever developing a strong understand of a simple design. This often leads to frustration. A curse of the Kalman Filter is that it is about five equation, see the Appendix, which seem very easy to understand and implement; however, those five equations are built on a set of models, parameters, and theoretical ideas, which if not accurate and understood will lead to unsatisfactory implementations.

This document presents the basic ideas of aided inertial navigation using a simplified one-dimensional example using IMU data. These basic ideas include the system kinematic equations, the INS implementation, the error model, quantification of the model parameters, estimation of the model errors by a Kalman filter, and correction of the INS state.

Special attention is paid to the development of the state transition model for the Kalman filter. The simplified one-dimensional model removes various complicating factors that arise in a full INS (e.g., nonlinearity and attitude), to enable the reader to focus on the basic INS and Kalman filtering ideas. Understanding the basics is critical before moving to more complicated scenarios.

The article is intended to come along with at least one IMU data set. The intent is as follows:

- Readers complete the design themselves and implement it in Matlab or Python testing their implementation using the provided data.
- The solution is provided so that readers can get help when they get stuck. Please try to solve the problem on your own. You miss the valuable learning and implementation experience if you simply read the solution.

If you are a UCR student, please contact me with questions. I am providing this document and data as an educational service.

If you find errors in the document or implementation, please do let me know.

1 Notation

In this document, the equality symbol '$=$' will be used in its normal sense. It is used in models that are used for analysis, but which cannot or need not be implemented in software solutions. The symbol '$\doteq$' will be used to indicate computations that will be implemented in the software. This distinction is meant to help people new to the field to distinguish between aided INS implementation and the analysis of that implementation.

Boldface symbols represent matrices or vectors.

2 Problem Statement

A sensor is constrained to move in a single direction while measuring its acceleration in that direction. The system does not rotate. The task is to estimate the position and velocity of the sensor in that one direction as a function of time t. At certain instants of time T_k for $k = 1, 2, 3, \ldots$, measurements of the sensor position are available.

The sensor is an Inertial Measurement Unit (IMU). The IMU provides acceleration measurements $\tilde{u}(t_i)$ for $i = 0, 1, 2, \ldots$ where $t_i = i\tau$ and $\tau = \frac{1}{f_s}$, The IMU was selected so that the sampling rate f_s is at lease twice the bandwidth of the motion of the sensor. It will typically be the case that $(T_k - T_{k-1}) >> \tau$, so that there are many IMU measurements between position measurements.

3 INS Approach and Background

This section develops the time differential equations for the state vector of the Inertial Navigation System (INS).

3.1 Kinematic Model

The position $p(t)$, velocity $v(t)$, and acceleration $a(t)$ are related to each other by the kinematic model

$$\dot{p}(t) = v(t) \quad (1)$$

$$\dot{v}(t) = a(t). \quad (2)$$

It is important to understand the difference between kinematic and dynamic models. The kinematic model is pristine, without any sources of error and no parameters that might be inaccurate.

A dynamic model would account for how the acceleration a was affected by position, velocity, and external inputs. A dynamic model is necessary for control system design and analysis. A kinematic model is used for Inertial Navigation System (INS) design and analysis.

*† Professor at the Dept. of Electrical and Computer Engineering, UC Riverside. {farrell}@ece.ucr.edu.

3.1.1 IMU Model

In this article, the accelerometer measurement $\tilde{u} \in \Re$ will be modeled as

$$\tilde{u}(t) = a(t) - z(t) \tag{3}$$

where $a(t)$ is the acceleration and $z(t)$ represents additive measurement errors. One challenge in INS implementation is characterizing the IMU errors in a manner that is compatible with the solution approach (i.e., as a state space model with finite dimension).

For the present example, assume that

$$\begin{aligned} z(t) &= b(t) + n(t) && (4)\\ \dot{b}(t) &= \omega(t). && (5) \end{aligned}$$

This is one of the simplest IMU error models. In this model, $n(t)$ is white Gausian measurement noise with power spectral density $Q_n = \sigma_n^2$, $\omega(t)$ is white Gaussian measurement noise with power spectral density $Q_\omega = \sigma_\omega^2$. Such models are discussed in Section 4.4.2-4.6 in [1]. The symbol $n(t)$ is referred to as the velocity random walk error component. The symbol $b(t)$ is often referred to as the sensor bias, it represents time correlated measurement errors.

Do Problem 5.1 described on p. 5.

3.1.2 System State Space Model

From eqns. (1-2) and (5), the continuous-time state-space model is

$$\dot{\boldsymbol{x}}(t) = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \boldsymbol{x}(t) + \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} a(t) + \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \omega(t) \tag{6}$$

where the state vector is $\boldsymbol{x}(t) = [p(t), v(t), b(t)]^\top$.

Note the following points:

- The state vector includes the kinematic state $[p(t), v(t)]$ and the sensor calibration factor $b(t)$. These are the quantities that will be estimated. By estimating IMU sensor calibration factors, such as $b(t)$, the INS accuracy will be enhanced.
- The symbol $n(t)$ does not appear in eqn. (6), because the sensor measurement error $z(t)$ has no effect on the time evolution of the actual state vector $\boldsymbol{x}(t)$.
- Nothing in the above corresponds to an INS computation. It is only system modeling.

3.2 Aiding Sensor Model

In this example, the aiding sensor measures position at $T = 20$ second intervals. The sensor measurement is modeled as

$$y(kT) = p(kT) + \eta(kT) \tag{7}$$

where $\eta(t)$ is white Gaussian noise with variance σ_η^2. To simplify notation, the document will use the following symbols: $y_k = y(kT)$, $p_k = p(kT)$, and $\eta_k = \eta(kT)$. In matrix form, eqn. (7) can be written as

$$y(kT) = \mathbf{H}\,\boldsymbol{x}_k + \eta_k \tag{8}$$

with $\mathbf{H} = [1, 0, 0]$.

3.3 INS Implementation

Let $\hat{p}(t)$, $\hat{v}(t)$, and $\hat{b}(t)$ represent the INS estimate of the position, velocity, and IMU bias at time t.

Based on eqns. (1-2) and taking the expected value of both sides of eqn. (5), the continuous-time INS model is

$$\begin{aligned} \dot{\hat{p}}(t) &= \hat{v}(t) && (9)\\ \dot{\hat{v}}(t) &= \hat{a}(t) && (10)\\ \dot{\hat{b}}(t) &= 0 && (11) \end{aligned}$$

where $\hat{a}(t) = \tilde{u}(t) + \hat{b}(t)$. Because the IMU measurements are taken at the discrete-time intervals $t_i = i\tau$, these equations are implemented in discrete-time, for example:

$$\begin{aligned} \hat{p}_{i+1} &\doteq \hat{p}_i + \hat{v}_i\tau + 0.5\hat{a}_i\tau^2 && (12)\\ \hat{v}_{i+1} &\doteq \hat{v}_i + \hat{a}_i\tau && (13)\\ \hat{b}_{i+1} &\doteq \hat{b}_i && (14) \end{aligned}$$

with $\hat{a}_i \doteq \tilde{u}_i + \hat{b}_i$. The discrete-time INS state estimate vector is $\hat{\boldsymbol{x}}_i = [\hat{p}_i, \hat{v}_i, \hat{b}_i]^\top$. Eqns. (12-14) are computed at the IMU rate f_s.

In matrix state space form eqns. (12-14) become

$$\hat{\boldsymbol{x}}_{i+1} = \boldsymbol{\Phi}_i\,\hat{\boldsymbol{x}}_i + \mathbf{G}_i\,\tilde{\mathbf{u}}_i \tag{15}$$

with $\boldsymbol{\Phi}_i = \begin{bmatrix} 1 & \tau & 0.5\,\tau^2 \\ 0 & 1 & \tau \\ 0 & 0 & 1 \end{bmatrix}$, and $\mathbf{G}_i = \begin{bmatrix} 0.5\,\tau^2 \\ \tau \\ 0 \end{bmatrix}$.

Remark 3.1 *The state space model in eqn. (15) is not appropriate for Kalman filter design, because it does not match eqns. (52). Eqn. (15) depends on the signal $a(t)$, through $\tilde{u}$. The signal a is not a known deterministic signal, nor is it a signal that can be modeled stochastically with known Gaussian distribution. Therefore, the Kalman filter cannot yet be applied, additional manipulations are required.* △

Remark 3.2 *Eqns. (12-14) are computed at the IMU rate f_s. The subscript 'i' is being used to denote the INS state variables accumulated at the IMU rate. Assuming that $\frac{T}{\tau} = L$ is an integer, there will be L INS iterations between any two consecutive aiding measurements: $T = L\,\tau$. Starting at time zero, the INS updates occur at $t_i = i\,\tau$ for $i = 0, 1, \ldots, L-1$. The INS uses $\tilde{u}_{L-1} = \tilde{u}((L-1)\tau)$ to compute*

$$\hat{\boldsymbol{x}}_i\big|_{i=L} = \hat{\boldsymbol{x}}(L\tau) = \hat{\boldsymbol{x}}(T) = \hat{\boldsymbol{x}}_k\big|_{k=1}.$$

The discussion of the Kalman filter will use the subscript k to denote signals at the time of the aiding measurements: $\boldsymbol{x}_k =$

$\boldsymbol{x}(kT)$, $\mathbf{y}_k = \mathbf{y}(kT)$, *etc. Symbols such as* $\boldsymbol{x}_3$ *will not be used, as it is unclear whether it means* $\boldsymbol{x}(3\tau)$ *or* $\boldsymbol{x}(3T)$. *The subscript 'i' will always refer to the INS time step* τ. *The subscript 'k' will always refer to the Kalman Filter time step* T. △

Once $\hat{\mathbf{p}}(kT)$ becomes available at $t = kT$, using the expected value of eqn. (7), the INS can predict the sensor measurement as

$$\hat{y}(kT) = \hat{p}(kT). \tag{16}$$

In matrix form, eqn. (16) can be written as

$$\hat{y}(kT) = \mathbf{H}\,\hat{\boldsymbol{x}}_k \tag{17}$$

with $\mathbf{H} = [1,\, 0,\, 0]$.

Do Problem 5.2 described on p. 5.

3.4 Error State Model

This section develops a model for the time evolution of the error in an INS implementation. The model is in state space form with stochastic inputs having known statistics, as is appropriate for development of an optimal state estimator (i.e., Kalman filter).

Assuming that the IMU is sampled fast enough that the acceleration can be considered constant over each sampling period, the discrete-time version of eqns. (1-2) and (5) is

$$\begin{aligned} p_{i+1} &= p_i + v_i\tau + 0.5a_i\tau^2 & (18)\\ v_{i+1} &= v_i + a_i\tau & (19)\\ b_{i+1} &= b_i + \omega_i & (20) \end{aligned}$$

where ω_i is a white Gaussian discrete-time noise process with second-order statistics equivalent to those of $\omega(t)$ at the IMU sampling instants.

The discrete-time error state vector is

$$\delta\boldsymbol{x}_i = \boldsymbol{x}_i - \hat{\boldsymbol{x}}_i. \tag{21}$$

Differencing eqns. (18-20) with eqns. (12-14), the discrete-time difference equations for the error state simplify to

$$\begin{aligned} \delta p_{i+1} &= \delta p_i + \delta v_i\tau + 0.5\delta b_i\tau^2 + 0.5n_i\tau^2 & (22)\\ \delta v_{i+1} &= \delta v_i + \delta b_i\tau + n_i\tau & (23)\\ \delta b_{i+1} &= \delta b_i + \omega_i & (24) \end{aligned}$$

where n_i is the discrete-time Guassian white noise process equivalent to $n(t)$. These equations can be written in the matrix state-space form

$$\delta\boldsymbol{x}_{i+1} = \boldsymbol{\Phi}_i\,\delta\boldsymbol{x}_i + \boldsymbol{\Gamma}_i\,\boldsymbol{\omega}_i \tag{25}$$

where $\boldsymbol{\omega}_i = [n_i, \omega_i]^\top$ is Gaussian white noise with covariance $\mathbf{Q}_d$ and

$$\boldsymbol{\Phi}_i \doteq \begin{bmatrix} 1 & \tau & 0.5\tau^2 \\ 0 & 1 & \tau \\ 0 & 0 & 1 \end{bmatrix}, \quad \boldsymbol{\Gamma}_i \doteq \begin{bmatrix} 0.5\tau^2 & 0 \\ \tau & 0 \\ 0 & 1 \end{bmatrix},$$

and $\mathbf{Q}_d \doteq \begin{bmatrix} \sigma_{n_i}^2 & 0 \\ 0 & \sigma_{\omega_i}^2 \end{bmatrix}$. For more complex problems, methods to compute $\boldsymbol{\Phi}$ and $\mathbf{Q}_d$ from $\mathbf{F}$, $\mathbf{G}$, and $\mathbf{Q}$ are discussed in Sect. 4.7 of [1].

The aiding sensor prediction error, called the *residual* measurement, is defined to be

$$r_k = y_k - \hat{y}_k. \tag{26}$$

Substituting in eqns. (8) and (17), the residual can be modeled as

$$r_k = \mathbf{H}\delta\boldsymbol{x}_k + \eta_k. \tag{27}$$

Note that eqn. (25) is not useful directly and cannot be integrated, since neither $\delta\boldsymbol{x}_i$ nor $\boldsymbol{\omega}_i$ are known. Nonetheless, the analysis of the error growth that results from this equation is key to the design of the optimal state estimator, as eqn. (25) does match the form of eqn. (52) as required for the design of the Kalman filter.

One last issue is that eqn. (25) accounts for error growth over one IMU time period, while eqn. (27) occurs after $N = \frac{T}{\tau}$ IMU time cycles. The error accumulation over these N IMU cycles is discussed in the next section.

Do Problem 5.3 described on p. 5.

3.5 Covariance Propagation

The uncertainty in the state estimate $\hat{\boldsymbol{x}}_i$ is characterized by the covariance matrix of the error state:

$$cov(\hat{\boldsymbol{x}}_i) = cov(\delta\boldsymbol{x}_i) = \mathbf{P}_i. \tag{28}$$

Using eqn. (25), Section 4.65 in [1] shows that given an initial value $\mathbf{P}_0$ the error covariance matrix is propagated through time at the IMU rate as

$$\mathbf{P}_{i+1} \doteq \boldsymbol{\Phi}_i\mathbf{P}_i\boldsymbol{\Phi}_i^\top + \boldsymbol{\Gamma}_i\mathbf{Q}_d\boldsymbol{\Gamma}_i^\top. \tag{29}$$

Eqn. (29) needs to be propagated Tf_s times (i.e., at each IMU sampling instant) between aiding measurements (occurring with period T).

Dropping the time subscript for this paragraph, to simplify the notation, the structure of $\mathbf{P}$ is

$$\mathbf{P} = \begin{bmatrix} \sigma_p^2 & \rho_{pv}\sigma_p\sigma_v & \rho_{pb}\sigma_p\sigma_b \\ \rho_{pv}\sigma_p\sigma_v & \sigma_v^2 & \rho_{vb}\sigma_v\sigma_b \\ \rho_{pb}\sigma_p\sigma_b & \rho_{vb}\sigma_v\sigma_b & \sigma_b^2 \end{bmatrix} \tag{30}$$

where σ_*^2 represents the covariance of variable * and ρ_{ab} is the cross-correlation between variable a and b. Therefore, once $\mathbf{P}$ is computed at any time, the error covariance of any element of the state vector can be computed. The only assumption for this to be valid is that the IMU error model of eqns. (3-5) and its parameters Q_n and Q_ω must be accurate.

Do Problem 5.4 described on p. 5.

3.6 INS Implementation Summary

This section summarizes the INS implementation equations over the time interval $t \in [t_{k-1}, t_k]$, assuming the initial condition $\boldsymbol{x}_{k-1} \sim N(\hat{\boldsymbol{x}}_{k-1}, \mathbf{P}_{k-1})$. At the IMU rate f_s, as each IMU measurement $\tilde{u}_i$ arrives:

INS state propagation: The INS computes eqns. (12-13).

INS state error covariance propagation: The INS computes the error state covariance matrix using eqn. (29).

After incorporating all f_s IMU measurements in $[t_{k-1}, t_k)$, the INS has effectively implemented the Kalman filter's task, as stated in eqns. (54-55), of propagating the state estimate and its error covariance matrix between the times of the last and the current aiding measurements.

This computed INS state vector $\hat{\boldsymbol{x}}$ is communicated at any sub-multiple of f_s to whichever other systems might be relying on it: control, planning, etc.

4 Kalman Filter Implementation

As stated in Section 3.6, the INS has already implemented the time propagation step of the Kalman filter; therefore, this section discusses the Kalman filter measurement update. For the measurement update, the Kalman filter prior is the result of the last step of the INS time propagation over the last interval (see Remark 3.2):

$$\hat{\boldsymbol{x}}_k^- = E\langle \boldsymbol{x}_k | y_{k-1}, \ldots, y_1 \rangle = \hat{\boldsymbol{x}}_i \text{ and } \mathbf{P}_k^- = \mathbf{P}_i \text{ with } i = T\, f_s. \tag{31}$$

Therefore, $\hat{\boldsymbol{x}}_k^- \sim N(\boldsymbol{x}_i, \mathbf{P}_k^-)\big|_{i=T\,f_s}$. Eqn. (17) is used to compute the prior output:

$$\hat{y}_k^- \doteq \mathbf{H}\, \hat{\boldsymbol{x}}_k^-. \tag{32}$$

4.1 KF Implementation

The measurement residual is computed as

$$r_k \doteq y_k - \hat{y}_k^-. \tag{33}$$

The covariance of the residual is

$$S_k \doteq \mathbf{H}\mathbf{P}_k^- \mathbf{H}^\top + \sigma_\eta^2. \tag{34}$$

The Kalman measurement update is

$$\begin{aligned} \mathbf{K}_k &\doteq \mathbf{P}_k^- \mathbf{H}^\top (S_k)^{-1} && (35)\\ \mathbf{P}_k^+ &\doteq \mathbf{P}_k^- - \mathbf{K}_k(\mathbf{H}\,\mathbf{P}_k^-) && (36)\\ \boldsymbol{x}_k^+ &\doteq \boldsymbol{x}_k^- + \mathbf{K}_k\, r_k. && (37) \end{aligned}$$

The posterior state $\boldsymbol{x}_k^+$ and covariance $\mathbf{P}_k^-$ are now available as initial conditions for the next set of INS computations as summarized in Section 3.6.

4.2 KF Clarification

It is important to clarify an issue that is not obvious in the presentation of Section 4.1. The Kalman filter was actually designed for the error model. Therefore, eqn. (37) should actually be

$$\delta\boldsymbol{x}_k^+ = \delta\boldsymbol{x}_k^- + \mathbf{K}_k\, r_k. \tag{38}$$

Due to eqn. (31), the prior has distribution $\delta\boldsymbol{x}_k^- = \mathbf{0}$; therefore, eqn. (38) would reduce to

$$\delta\boldsymbol{x}_k^+ = \mathbf{K}_k\, r_k. \tag{39}$$

Using eqn. (21), the corrected INS state estimate is

$$\begin{aligned} \boldsymbol{x}_k^+ &= \boldsymbol{x}_k^- + \delta\boldsymbol{x}_k^+ && (40)\\ \boldsymbol{x}_k^+ &= \boldsymbol{x}_k^- + \mathbf{K}_k\, r_k, && (41) \end{aligned}$$

which matches eqn. (37). After correcting the INS state as in eqn. (37), then $\boldsymbol{x}_k^+ = E\langle \boldsymbol{x}_k | y_k, \ldots, y_1 \rangle$ and $\delta\boldsymbol{x}_k^+ = \mathbf{0}$.

4.3 KF Residual Sequence

If the INS and Kalman filter are properly implemented with correct parameter settings, then the residual should be zero mean, white, and have covariance S_k. The designer should always check the covariance sequence to ensure that this is the case. If it is not, then debugging is required.

Designing of the Kalman filter should not involve any 'tuning' of the parameters in $\mathbf{Q}_d$. These parameters relate to the performance of the IMU and should be determined from the specification parameters of the IMU that are supplied by the manufacturer.

After reading the material in this section, complete Problem 5.5 on page 6.

5 Reader Exercises

Test your understanding of the above materials by completing the following.

Problem 5.1 Unzip the data files and see the README.txt. for a data description.

Analyze the basic data to ensure that it looks as expected. Note the following:

- Find a time segment during which the IMU is stationary. Look closely at the measured acceleration during this time. You are checking reasonableness. A real IMU would have a spec sheet and you would analyze the measured IMU data relative to that specification.
- The time vector in each data structure is the received time at the CPU, not the measurement time at the sensor. Analyze the delta time between measurements. Does it meet the specification?
- The sensor data can be thought of as coming from 20 second long experiments starting at $t = 20$. During each 20 second interval, the IMU has some 1-D motion, but it is stationary at a known position at

$$t_k = k20.$$

Check the data to ensure that this is true.

Problem 5.2 Implement a pure INS (no aiding).

1. Starting at $t = 0$ with position, velocity, and bias equal to zero (i.e., $\hat{x}(0) = [0, 0, 0]^\top$), use eqns. (12-14) to integrate the INS state over the interval for which the IMU data is available.
2. Use the data for $t \in [1, 20]s$ to estimate the bias:

$$\bar{b} = \frac{-1}{20} \int_0^{20} \tilde{u}(t)dt.$$

3. Starting at $t = 20$ with position and velocity equal to zero, and the bias equal to the value estimated in Step 2 (i.e., $\hat{x}(20) = [0, 0, \bar{b}]^\top$); use equations (12-14) to integrate the INS state over the interval for which the IMU data is available.
4. Estimate the initial condition for the bias by any other method you wish and repeat the INS integration to compare performance.

Problem 5.3

1. Analyze the difference between eqns. (12-14) and (18-20) to derive eqns. (22-24).
2. Use eqns (8), (17), and (26), to derive eqn. (27).
3. Assume that the IMU may have a scale factor error. In this case, eqn. (3) would be modified to:

$$\tilde{u}(t) = (1 - s)a(t) - z(t) \tag{42}$$

where s is a small constant with random initial value.

   (a) Show that δa defined as $\delta a = a - \hat{a}$ has the linearized model:

$$\delta a = \frac{\hat{a}}{1 - \hat{s}}\delta s + \frac{1}{1 - \hat{s}}\delta b + \frac{1}{1 - \hat{s}}\eta. \tag{43}$$

   (b) Revise the error model to account for the scale factor error δs as an additional state variable.

Problem 5.4 In this problem treat the time interval $t \in [1, 20]s$ as 19 different data sets that are each two seconds long.

The IMU is at rest at the origin during this entire time interval; therefore, any deviation of the position and velocity portion of the state estimate from 0 is INS state error. In this problem, you will generate 19 instances of the INS state estimation error and compare that with the computed error state covariance.

1. For each two second interval $t \in [\ell, \ell + 2)$ for $\ell = 0, \ldots, 18$, complete the following steps:

   (a) Average the $N = f_s$ samples of acceleration data for $t \in [\ell, \ell + 1)$ to estimate the bias $\hat{b}_\ell$ and its covariance $[\mathbf{P}]_{33} = \sigma_b^2$. Note that for the values of $\sigma_b^2 = \frac{var(n)}{N}$ should be essentially the same for each averaging interval.

   (b) At $t = \ell + 1$, define the initial state estimate to be $\hat{\boldsymbol{x}} = [0, 0, \hat{b}_\ell]^\top$, with initial covariance matrix

$$\mathbf{P} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & \sigma_b^2 \end{bmatrix}.$$

   Use the IMU data for $t \in [\ell + 1, \ell + 2)$ and eqns. (12-14) to integrate the state estimate over this one second interval. Call this one second long trajectory

$$\mathbf{X}^l = [\boldsymbol{x}(\ell{+}1), \boldsymbol{x}(\ell{+}1{+}\tau), \boldsymbol{x}(\ell{+}1{+}2\tau), \ldots, \boldsymbol{x}(\ell{+}2)].$$

   For all 19 values of ℓ, plot the position and velocity states from $\mathbf{X}^\ell$ versus the integration time,

$$\begin{aligned} \bar{t} &= t - (\ell + 1) && (44) \\ &= [0, 1, 2, \ldots, N]\tau. && (45) \end{aligned}$$

   This common time axis facilitates the subsequent analysis.

2. Use eqn. (29) to propagate the error state covariance matrix $\mathbf{P}_i$ as a function of the integration time $\bar{t}_i = i\tau$. Assume that $\sigma_n = 1 \times 10^{-3} \frac{m/s}{\sqrt{s}}$ and $\sigma_w = 1 \times 10^{-5} \frac{m/s^2}{\sqrt{s}}$ Note that this sequence will be essentially the same for each value of ℓ, so only needs to be computed once. From $\mathbf{P}_i$ extract $\sigma_p(i)$ and $\sigma_v(i)$ using eqn. (30). Plot $\pm\sigma_p(i)$ on the position graph from Step 2 and $\pm\sigma_v(i)$ on the velocity graph from Step 2.

3. If the values of σ_n^2 and σ_ω^2 are defined accurately, then about 68% of trajectory points will be within the $\pm 1\sigma$ envelope. Does this approximately hold for your implementation?

4. For this simple problem, the error covariance can be found in closed form as

$$P_p(\bar{t}) = \frac{\sigma_n^2}{3}\bar{t}^3 + \frac{\sigma_b^2}{4}\bar{t}^4 + \frac{\sigma_\omega^2}{20}\bar{t}^5 \quad (46)$$

$$P_v(\bar{t}) = \sigma_n^2 \bar{t} + \sigma_b^2 \bar{t}^2 + \frac{\sigma_\omega^2}{3}\bar{t}^3. \quad (47)$$

Check that all the units in each term of these equations work out correctly. Each term can be plotted separately to determine which error source is dominant over a one second window.

Problem 5.5 Implement the Kalman filter to correct the INS state estimate at the times $t = k\,20$. Analyze the residuals relative to their expected standard deviation. For this exercise let $\sigma_\eta = 0.01m$.

6 Solutions

This solution works with simulated data. For this data set, the known position at the stationary instants is $p(T_k) = 0$.

6.1 Solution for Problem 5.1

This section analyzes the raw accelerometer data to ensure reasonableness of the data and checks any assumptions.

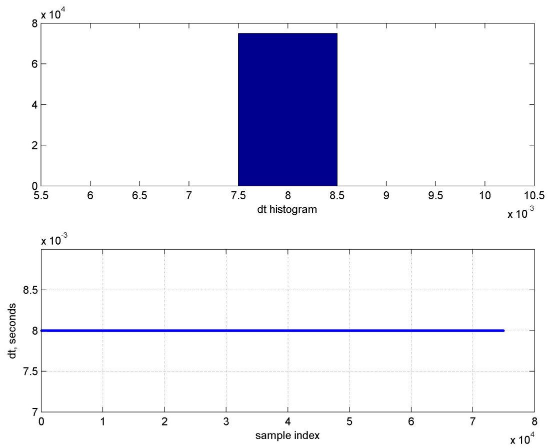

Figure 1: Analysis of the IMU delta-time δt between samples.

Fig. 1 shows a histogram and time series for the delta time between IMU samples. Because this is simulation data it is perfect with $\delta t = \tau = \frac{1}{f_s}$ for all samples. For actual data this is not typically the case. For actual data there will actually be two different δt values: the value reported by the IMU and the value reported by the CPU. The CPU includes communication and processing latency. The INS integration process should use the δt reported by the IMU, not the CPU.

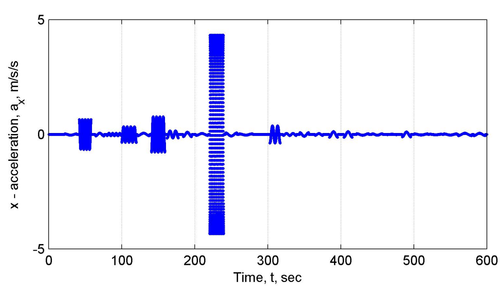

Figure 2: Measured acceleration $\tilde{u}(t_i)$ for $t_i \in [1, 600]$.

Fig. 2 plots the measured acceleration versus time. The maximum acceleration is less than $1\,g = 9.8m/s/s$. The analyst must ensure that the IMU measurement range is (more than) sufficient for the expected operating scenarios. Several periods of oscillatory motion are obvious. At this scale, it is challenging to observe anything more specific.

Fig. 3 shows the measured acceleration for the first twenty seconds of data. During this time interval, the sensor is stationary, allowing a few interesting observations. First, the readings are biased below zero by about $0.01m/s/s$. Second, the standard deviation is 0.001 m/s/s. Third, other than the bias, there are no obvious correlations in the data. The correlation sequence can be plotted in Matlab using the 'xcorr' function.

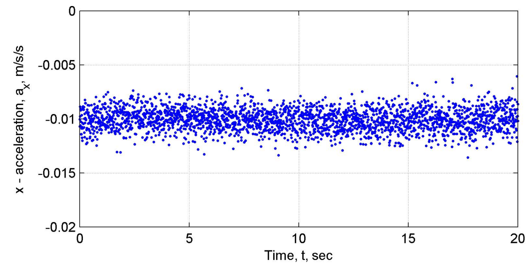

Figure 3: Measured acceleration $\tilde{u}(t_i)$ for $t_i \in [1, 20]$.

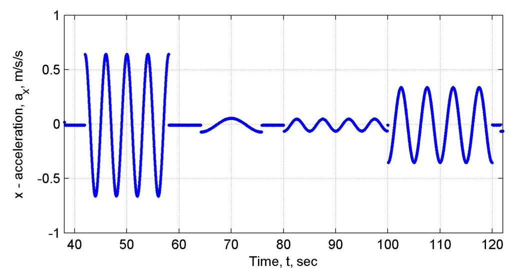

Figure 4: Measured acceleration $\tilde{u}(t_i)$ for $t_i \in [38, 122]$.

Fig. 4 shows the measured acceleration over four intervals of motion. Each interval of motion involves oscillation and returns to its original position with zero velocity at $T_k = k\,20$. The FFT of this signal is shown in Fig. 5. The majority of the signal power is below 10 Hz, which is viewed as the bandwidth of the system; therefore, the assumption that the Nyquist frequency $f_N = f_s/2 = 62.5Hz$ should be significantly higher than the system bandwidth is verified.

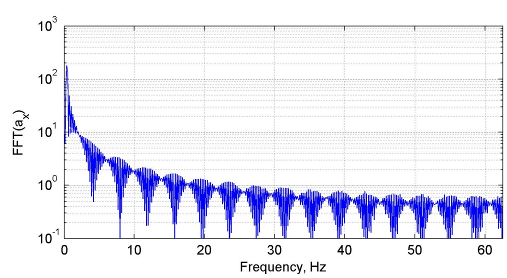

Figure 5: FFT($\tilde{u}(t_i)$) for $t_i \in [38, 122]$.

Since there are no problems with the data set or the assumptions, the analysis and implementation proceed to the next step.

In the real-time implementation, care must be taken at start-up to ensure that the buffer is cleared and that the sensor has time to turn on and start sending data.

6.2 Solution for Problem 5.2

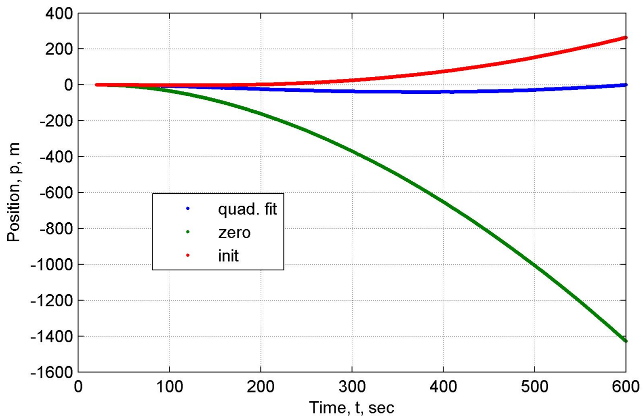

Figure 6: INS computed position without aiding for three different initial conditions for the bias.

Fig. 6 displays the INS computed position for three different estimates of the bias. The green curve initializes the bias $\hat{b}_{20} = 0$. Over the 600 second interval of integration, the position error grows to -1428m. The red curve initializes the bias to the average over the first 20 seconds: $\hat{b}_{20} = \bar{b}$. Over the 600 second interval of integration, the position error grows to 265m, which is better, but not very great. The blue curve initializes the bias to a value (explained in the next paragraph) that causes the final position to be zero. Even though the blue curve has a correct final value, its maximum position error near $t = 380s$ is approximately 39 m.

Consider the green curve again, its near parabolic shape can be used to estimate an effective value of the bias over the full time interval. Using the model that the position error due to integration of a bias over an interval T should be $\hat{p}(T) = 0.5bT^2$ yields

$$\hat{b} = 2p/T^2 = 2(1428.5)/(580)^2 \ m/s/s. \tag{48}$$

This is the bias value used to compute the blue curve. Note that there is no constant bias value that is correct. The bias is time-varying and must be estimated in real-time.

Fig. 7 shows portions of the position and velocity trajectories computed by the INS using the bias estimate from eqn. (48). The top subplot shows the INS computed velocity at all times. This plot is included to show that the motion of the IMU (i.e., the oscillations) is superimposed on another signal. To visualize that signal, the middle subplot shows the IMU computed velocity at the times $T_k = k\,20s$, which are those times at which we know from the design of the experiment that the IMU is in fact stationary. These are the velocity errors. Similarly, the third subplot shows the position errors, which are the values of the INS computed position at T_k, when we know that the IMU is sitting at the origin. Comparing these two error plots, you should be able to see that the position error is the integral of the velocity error and the point at which the position error has maximum magnitude is the point at which the velocity error is zero.

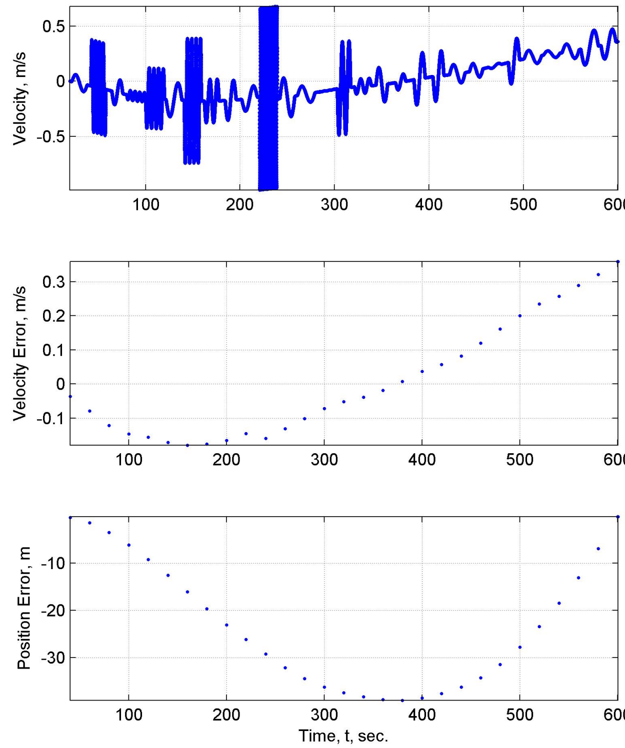

Figure 7: INS trajectory for initial bias estimate -0.0135 m/s/s.

Because this experiment was designed with long periods when the sensor is stationary, to maintain accuracy the interested reader could compute bias estimates periodically during those stationary time intervals. Such methods tuned to a specific experiment are fun to play with, but are ad-hoc and non-optimal. Subsequently, we will investigate general purpose optimal methods.

The purpose of these graphs are to illustrate a few key points:

- The INS is an integrative process. The information being integrated include errors in initial condition and from the IMU; therefore error accumulates in the kinematic state integrated by the IMU. These errors accumulate slowly and can be accurately modeled.
- Because the initial condition and IMU errors are random, the error model will be built using stochastic models.
- Estimation of the IMU calibration factors (e.g., the bias) is important.
- The bias is time varying. Even if accurately estimated over a time interval, its value will change versus time resulting in velocity and position errors; therefore, its value must be estimated persistently.

6.3 Solution for Problem 5.3

For Part 3, assume that $\dot{s}(t) = 0$ and $s(0) \sim N(0, P_s(0))$. The estimator will provide $\hat{s} = E\langle s(t)|y(\tau)\,\forall\tau < t\rangle$ and $\hat{b} = E\langle b(t)|y(\tau)\,\forall\tau < t\rangle$. The scale factor error is defined as $\delta s(t) = s(t) - \hat{s}(t)$. Taking the expected value of both sides of eqn. (42) yields

$$E\langle \tilde{u}(t)|y(\tau)\,\forall\tau < t\rangle = (1 - \hat{s}(t))a(t) - \hat{b}. \tag{49}$$

Based on eqn. (49), given a measurement $\tilde{u}(t)$, the acceleration would be computed as

$$\hat{a}(t) = \frac{1}{1-\hat{s}(t)}\left(\tilde{u}(t) + \hat{b}\right). \tag{50}$$

To find the error model, substitute in eqns. (42) and (4), then simplify

$$\hat{a}(t) = \frac{1}{1-\hat{s}(t)}\left((1-s)a(t) - z(t) + \hat{b}\right)$$
$$\hat{a}(t) = \frac{1}{1-\hat{s}(t)}\left((1-s)a(t) - (b(t)+n(t)) + \hat{b}(t)\right)$$
$$\hat{a}(t) = \frac{1-s}{1-\hat{s}(t)}a(t) + \frac{1}{1-\hat{s}(t)}\left(\hat{b}(t) - b(t) - n(t)\right).$$

The following will utilize the definitions: $\delta b(t) = b(t) - \hat{b}(t)$, and $\delta a(t) = a(t) - \hat{a}(t)$.

$$\hat{a}(t) = \frac{1-(\delta s(t) + \hat{s}(t))}{1-\hat{s}(t)}a(t) - \frac{1}{1-\hat{s}(t)}\left(\delta b(t) + n(t)\right)$$
$$\hat{a}(t) = a(t) - \frac{a(t)}{1-\hat{s}(t)}\delta s(t) - \frac{1}{1-\hat{s}(t)}\left(\delta b(t) + n(t)\right)$$
$$\hat{a}(t) = a(t) - \frac{\hat{a}(t)+\delta a(t)}{1-\hat{s}(t)}\delta s(t) - \frac{1}{1-\hat{s}(t)}\left(\delta b(t) + n(t)\right)$$

which yields

$$\delta a(t) = \frac{\hat{a}(t)}{1-\hat{s}(t)}\delta s(t) + \frac{1}{1-\hat{s}(t)}\delta b(t) + \frac{1}{1-\hat{s}(t)}\eta(t). \tag{51}$$

after dropping the second order product term with $\delta a(t)\delta s(t)$.

Note that the coefficients in this equation can be found directly as the partial derivative of $\hat{a}$ with respect to the calibration parameter. For example, $\frac{\partial}{\partial \hat{b}}\hat{a} = \frac{1}{1-\hat{s}}$.

6.4 Solution for Problem 5.4

Fig. 8 shows the 19 error simulations, each starting with position and velocity at zero and bias initialized to the average sensor reading over the last one second. The 19 values of the bias are shown in the third figure from the top. The value of σ_b was approximately $1.0 \times 10^{-4} m/s/s$. The position and velocity error trajectories are shown in the top two plots.

Fig. 9 shows the same 19 position and velocity error trajectories plotted versus integration time (i.e., a common axis). The subplots of this figure also show a wide black curve that shows $\pm\sigma_p$ and $\pm\sigma_v$ as extracted from the $\mathbf{P}_k$ computed using eqn. (29) and extracted using the structure of $\mathbf{P}$ as defined in eqn. (30).

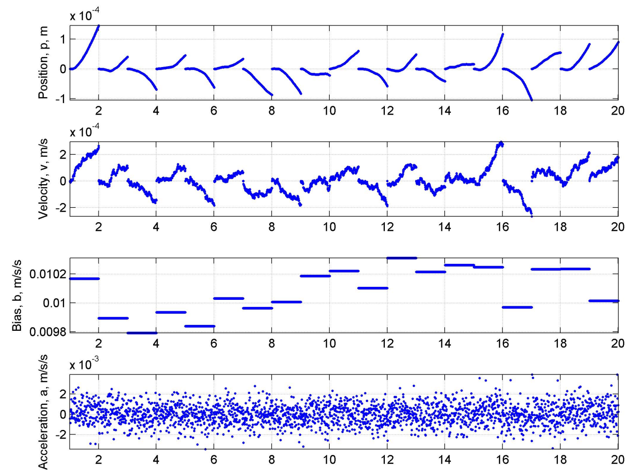

Figure 8: Nineteen examples of INS errors each integrated over one second. Each example is plotted in its own color, starting with position and velocity equal to zero and bias equal to the average over the previous one second.

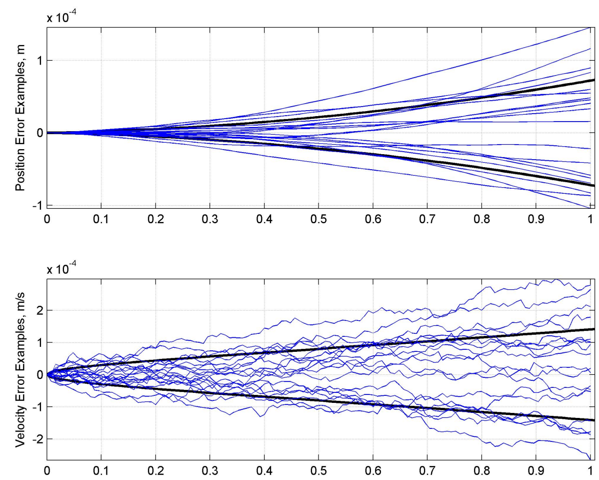

Figure 9: The same 19 INS error examples as in Fig.8, plotted with a common time axis. Each subplot is overlaid with a wide black graph of $\pm\sigma$ where sigma is extracted from the error covariance matrix $\mathbf{P}$ computed by eqn. (29).

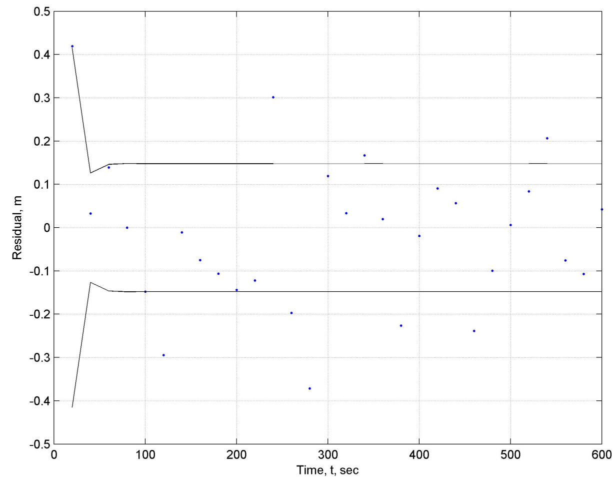

Figure 10: Kalman filter residual r_k and the envelope of its residual $\pm\sigma_{r_k}$ where $\sigma_{r_k} = \sqrt{S_k}$.

6.5 Solution for Problem 5.5

Fig. 10 plots the Kalman filter residuals versus time as dots, along with the computed standard deviation of the residuals as a solid black curve. The residual properties (zero mean, uncorrelated in time, standard deviation about right) look acceptable. In steady-state, the position error after 20 seconds of uncorrected motion is expected to be about 0.13m.

Figs 11 and 12 show magnified views of the INS state during two time periods. The blue curves represent the Kalman filter corrected INS estimate of the state vector. The black curves are that state estimate plus and minus the state error standard deviation as extracted from the diagonal elements of $\mathbf{P}$ as explained relative to eqn. (30).

Note that:

- The actual motion of the IMU is clearly evident in the position and velocity plots.
- Error does accumulate during each 20 second interval of INS integration between Kalman filter corrections.
- The high frequency IMU measurement noise, see Fig. 4, is effectively removed by the INS integration process. Integration is low pass in nature.
- The INS error accumulation is effectively removed by the KF error estimation, which includes calibration of the IMU bias. The very large error growth due to the bias, see Fig. 6, has been removed.
- The Kalman filter corrections to the state and updates to the covariance are clearly observed at $t_k = k20$.
- The magnitude of the state corrections is similar to the quantification of that error by the standard deviations extracted from the $\mathbf{P}$ matrix.

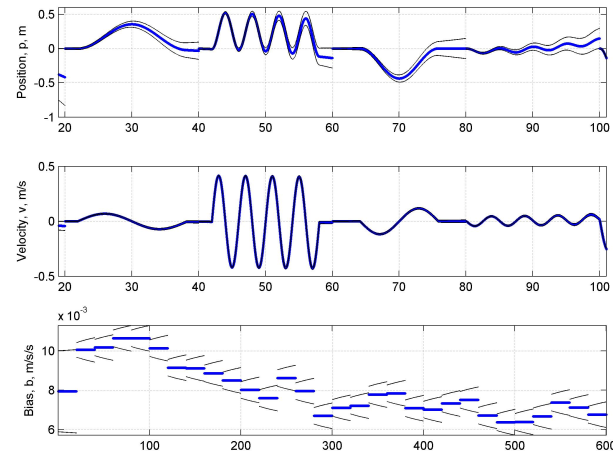

Figure 11: Kalman filter corrected INS estimate of the state vector (blue). State estimate plus and minus the state error standard deviation (black).

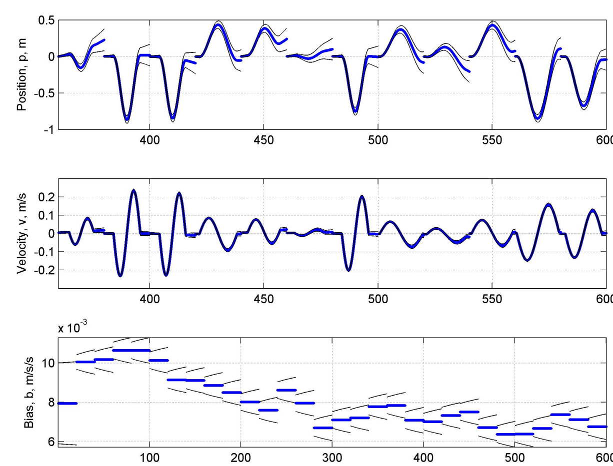

Figure 12: Kalman filter corrected INS estimate of the state vector (blue). State estimate plus and minus the state error standard deviation (black).

A The Discrete Time Kalman Filter

The Kalman filter assumes a linear system model:

$$\mathbf{z}_{k+1} = \boldsymbol{\Phi}_k \, \mathbf{z}_k + \mathbf{B}_k \, \mathbf{u}_k + \boldsymbol{\Gamma}_k \, \boldsymbol{\omega}_k \quad (52)$$

$$\mathbf{y}_k = \mathbf{H}_k \, \mathbf{z}_k + \boldsymbol{\eta}_k. \quad (53)$$

with $\boldsymbol{\Phi}_k$, $\mathbf{B}_k$, $\boldsymbol{\Gamma}_k$, $\mathbf{H}_k$ all known; a known prior distribution $\mathbf{z}_0 \sim N(\hat{\mathbf{z}}_0, \mathbf{P}_0)$; a known deterministic input sequence $\mathbf{u}_k$; and, two zero mean, white Gaussian random sequences $\boldsymbol{\omega}_k$ and $\boldsymbol{\eta}_k$ with known covariance matrix sequences $\mathbf{Q}_k$ and $\mathbf{R}_k$. In addition, it is assumed that $\boldsymbol{\omega}_k$ and $\boldsymbol{\eta}_k$ are uncorrelated with each other and with $\mathbf{z}_0$.

For a system with the model in the previous paragraph, the objective is to estimate the state sequence $\mathbf{z}_k$ optimally, using only the current and past measurements $\{\mathbf{y}_k\}_{j=1}^{k}$.

The Kalman filter [2,3] solves this problem, see also Chapter 5 in [1]. The Kalman filter has two types of updates.

Time Propagation: The time propagation step transports the state estimate between the times of two measurements. The equations are:

$$\mathbf{P}_k^- \doteq \boldsymbol{\Phi}_{k-1} \, \mathbf{P}_{k-1}^+ \, \boldsymbol{\Phi}_{k-1}^\top + \boldsymbol{\Gamma}_{k-1} \mathbf{Q}_{k-1} \boldsymbol{\Gamma}_{k-1}^\top \quad (54)$$

$$\hat{\mathbf{z}}_k^- \doteq \boldsymbol{\Phi}_{k-1} \, \hat{\mathbf{z}}_{k-1}^+ + \mathbf{B}_{k-1} \, \mathbf{u}_{k-1}. \quad (55)$$

Measurement Update: The measurement update step corrects the state estimate to account for the new infomation in the measurement $\mathbf{y}_k$. The equations are:

$$\mathbf{K}_k \doteq \mathbf{P}_k^- \mathbf{H}_k^\top (\mathbf{H}_k \mathbf{P}_k^- \mathbf{H}_k^\top + \mathbf{R}_k)^{-1} \quad (56)$$

$$\mathbf{P}_k^+ \doteq \mathbf{P}_k^- - \mathbf{K}_k \mathbf{H}_k \, \mathbf{P}_k^- \quad (57)$$

$$\hat{\mathbf{z}}_k^+ \doteq \hat{\mathbf{z}}_k^- + \mathbf{K}_k \, \mathbf{r}_k. \quad (58)$$

The derivation of these equations can be found in the many books on optimal estimation. Those same books will present various forms of the Kalman filter. All forms yield the same state estimates in theory, but have different computational and numeric properties.

Note that both the state estimate and the covariance matrix have two values at each measurement time: $\hat{\mathbf{z}}_k^+$ and $\hat{\mathbf{z}}_k^-$ and $\mathbf{P}_k^-$ and $\mathbf{P}_k^+$. The superscript minus sign represents the values before including $\mathbf{y}_k$. The superscript plus sign represents the values after including $\mathbf{y}_k$.

The above form of the equations assumes that the measurements $\hat{\mathbf{y}}(t)$ are equally spaced at $t_k = kT$. Equally spaced measurements is not a requirement of the Kalman filter. It just simplifies the presentation of the algorithm. Asynchronous measurements are straightforward to incorporated, as long as the time propagation step can transport the state estimate and covariance matrix from the time of the last measurement to the time of the most recent measurement.

B Data Description

This pdf file should come with a zipped data set. It contains the following vaiables: t is the vector of measurement times; *acc* is the vector of acceleration measurements.

This is data created via simulation. It intent is to correspond to the situation where an experimenter moves a 1-d accelerometer in the direction of its sensitive axis, without rotation. Therefore, the continuous-time kinematic model is given by eqns. (1-2).

That experimenter ensures that the sensor is stationary at its original location for $t = kT$ with $T = 20$ seconds. Therefore, defining $y_k = y(kT)$,

$$y_k = p_k + \eta_k, \text{ for } k = 0, 1, 2, \ldots.$$

See eqn. (8). The measurement noise corresponds to the experimenters placement accuracy, perhaps 0.001 m.

References

[1] J. A. Farrell, *Aided Navigation: GPS with High Rate Sensors.* McGraw Hill, 2008.

[2] R. E. Kalman, "A New Approach to Linear Filtering and Prediction Problems," *Transactions of the ASME–Journal of Basic Engineering*, vol. 82, no. D, pp. 35–45, 1960.

[3] R. E. Kalman and R. S. Bucy, "A new approach to linear filtering and prediction theory," *ASME Journal of Basic Engineering, Series D*, vol. 83, pp. 95–108, 1961.