Research Article

INS/GPS for High-Dynamic UAV-Based Applications

Junchuan Zhou,¹ Stefan Knedlik,² and Otmar Loffeld¹

¹ Center for Sensor Systems (ZESS), University of Siegen, 57076 Siegen, Germany

² iMAR Navigation GmbH, 66386 St. Ingbert, Germany

Correspondence should be addressed to Junchuan Zhou, zhou@zess.uni-siegen.de

Received 1 August 2011; Accepted 5 December 2011

Academic Editor: Farid Melgani

Copyright © 2012 Junchuan Zhou et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

The carrier-phase-derived delta pseudorange measurements are often used for velocity determination. However, it is a type of integrated measurements with errors strongly related to pseudorange errors at the start and end of the integration interval. Conventional methods circumvent these errors with approximations, which may lead to large velocity estimation errors in high-dynamic applications. In this paper, we employ the extra states to “remember” the pseudorange errors at the start point of the integration interval. Sequential processing is employed for reducing the processing load. Simulations are performed based on a field-collected UAV trajectory. Numerical results show that the correct handling of errors involved in the delta pseudorange measurements is critical for high-dynamic applications. Besides, sequential processing can update different types of measurements without degrading the system estimation accuracy, if certain conditions are met.

1. Introduction

GPS receivers are widely used in navigation. However, the system performance largely depends on the signal environment, and the measurement update rate is low. This raises the need to integrate GPS with the inertial navigation system (INS) to have a robust continuous navigation solution regardless of the environment. Among several integration architectures, the tightly coupled INS/GPS integration is one of the most promising methods to fuse the GPS and INS data, where the code-derived pseudorange and carrier-phase-derived delta pseudorange measurements are often exploited. However, the delta pseudorange is an integrated measurement with errors strictly related to the pseudorange errors at the endpoints of the integration interval. In practice, various approximations are made to handle these errors. The weakest but often used approach is to simply consider the integral of velocity divided by time (an average value) as the instantaneous velocity measurement at the endpoint of the integration interval [1]. It may fulfill the accuracy requirements in static or low dynamic applications. Nevertheless, if the vehicle is maneuvering under high dynamics with low GPS data update rate, large velocity estimation

errors will appear. That is, the velocity errors will be strongly correlated to the accelerations and jerks involved in the trajectory [2]. In order to tackle this problem, we use delay states to “remember” the pseudorange errors at the start of the integration interval. Besides, for reducing the processing load, sequential processing is utilized to avoid the time consuming computation of matrix inversion when deriving the Kalman gain. Different types of measurements (e.g., integrated and nonintegrated measurements) are sequentially processed with no compromise made on their errors.

In the remainder of this paper, the content is organized as follows. In Section 2, the often used velocity determination method based on the delta pseudorange measurements is overviewed, and its problem is indicated. In Section 3, the approach based on the augmented state vector is proposed for tackling this problem. Moreover, sequential processing is utilized and the computational burden is analyzed. In Section 4, the INS/GPS tightly coupled system state space models are introduced. In Section 5, simulation tests are conducted. Numerical results are compared and analyzed. In the Appendix, the number of numerical operations involved in the calculation of the matrix inversion using the Gauss-Jordan elimination method is counted.

2. Velocity Determination

The delta pseudorange is formed by integrating the pseudorange rate over the filter measurement update interval. It is equal to the difference of pseudoranges measured at the endpoints of the filter update interval. Accordingly, the errors involved in delta pseudorange are strictly related to the corresponding pseudorange errors at the start and end of the integration interval, which are given in (1):

$$\Delta \rho_k = \int_{k-1}^k \delta \dot{\rho} dt = \delta \rho_k - \delta \rho_{k-1}, \quad (1)$$

where $\Delta \rho_k$ represents the delta pseudorange error measured at time instance k ; $\delta \dot{\rho}$ denotes the pseudorange rate error; $\delta \rho_k$ and $\delta \rho_{k-1}$ are the pseudorange errors at the two endpoints of the integration interval, which can be related to error states (i.e., position errors and receiver clock bias error) as

$$\begin{aligned} \delta \rho_k &= \hat{\rho}_k - \rho_k = \mathbf{H}_k \delta \mathbf{x}_k + \mathbf{v}_k, \\ \delta \rho_{k-1} &= \hat{\rho}_{k-1} - \rho_{k-1} = \mathbf{H}_{k-1} \delta \mathbf{x}_{k-1} + \mathbf{v}_{k-1}, \end{aligned} \quad (2)$$

where the error states are defined as $\delta \mathbf{x} = \hat{\mathbf{x}}_{\text{estimated}} - \mathbf{x}_{\text{true}}$; the $\mathbf{H}_k$ and $\mathbf{H}_{k-1}$ are the Jacobian matrices from the first-order linearization of the nonlinear pseudorange observation model at time instants of the start and end of the integration interval; $\mathbf{v}_k$ and $\mathbf{v}_{k-1}$ represent the zero-mean Gaussian white noises.

Substituting (2) into (1) yields

$$\delta \rho_k - \delta \rho_{k-1} = \mathbf{H}_k \delta \mathbf{x}_k - \mathbf{H}_{k-1} \delta \mathbf{x}_{k-1} + \mathbf{v}_{k,k-1}. \quad (3)$$

For clarity purpose, we will denote $\mathbf{v}_k^\phi$ to represent the cumulative effect of the carrier-phase-derived delta pseudorange measurement noises $\mathbf{v}_{k,k-1}$ from time instants $k-1$ to k . We divide (3) by the time interval $\Delta t = t_k - t_{k-1}$ and assume that this time interval is small. Thus, the changes in the Jacobian matrix $\mathbf{H}$ are negligible (i.e., $\mathbf{H}_{k-1} \approx \mathbf{H}_k$), and we arrive at

$$\begin{aligned} \frac{\delta \rho_k - \delta \rho_{k-1}}{\Delta t} &= \mathbf{H}_k \frac{(\delta \mathbf{x}_k - \delta \mathbf{x}_{k-1})}{\Delta t} + \mathbf{v}_k^\phi \\ \Rightarrow \delta \dot{\rho}_k &= \mathbf{H}_k \delta \dot{\mathbf{x}}_k + \mathbf{v}_k^\phi, \end{aligned} \quad (4)$$

where $\delta \dot{\mathbf{x}}_k$ contains the system velocity errors and receiver clock drift error.

This equation is often employed for velocity determination using delta pseudorange measurements. However, it relates the average information (i.e., $(\delta \rho_k - \delta \rho_{k-1})/\Delta t$) to the instantaneous velocity estimation errors at time instant k (i.e., $\delta \dot{\mathbf{x}}_k$). If the vehicle is maneuvering under high dynamics, this approximation will be very poor.

Therefore, we lay (4) aside. And from (2), we have

$$\begin{aligned} \delta \rho_k - \delta \rho_{k-1} &= (\hat{\rho}_k - \rho_k) - (\hat{\rho}_{k-1} - \rho_{k-1}) \\ &= (\hat{\rho}_k - \hat{\rho}_{k-1}) - (\rho_k - \rho_{k-1}). \end{aligned} \quad (5)$$

We denote $\hat{\mathbf{y}}_k^\phi = \hat{\rho}_k - \hat{\rho}_{k-1}$, $\mathbf{y}_k^\phi = \rho_k - \rho_{k-1}$ and $\delta \mathbf{y}_k^\phi = \hat{\mathbf{y}}_k^\phi - \mathbf{y}_k^\phi$. Thus, (3) can be reformulated as

$$\delta \mathbf{y}_k^\phi = \mathbf{H}_k \delta \mathbf{x}_k - \mathbf{H}_{k-1} \delta \mathbf{x}_{k-1} + \mathbf{v}_k^\phi. \quad (6)$$

Equation (6) relates the system current and delay error states to the delta pseudorange innovation vector. It can be used as the system observation model. However, this equation is not in the general format for applying a Kalman filter (KF). That is, the delay state term $\mathbf{H}_{k-1} \delta \mathbf{x}_{k-1}$ appears on the right side of the equation. In order to solve this problem, modifications are required. In this contribution, we augment the state vector to include the delay states. Details will be given in the next section.

3. Augmentation of the System State Vector

We augment the system state vector to include the delay states to “remember” the system estimation errors at the start of the integration interval. These delay states are explicitly estimated from the previous measurement update and do not evolve over time (e.g., [3]). Therefore, we should model them as random constants in the current measurement update. The augmented system model is

$$\begin{bmatrix} \delta \mathbf{x}_k \\ \delta \mathbf{x}_{d,k} \end{bmatrix} = \begin{bmatrix} \mathbf{\Phi}_{k,k-1} & \mathbf{O} \\ \mathbf{O} & \mathbf{I} \end{bmatrix} \begin{bmatrix} \delta \mathbf{x}_{k-1} \\ \delta \mathbf{x}_{d,k-1} \end{bmatrix} + \begin{bmatrix} \mathbf{w}_{k-1} \\ \mathbf{0} \end{bmatrix}, \quad (7)$$

where $\delta \mathbf{x}_{d,k}$ denotes the delay error state vector; $\mathbf{\Phi}_{k,k-1}$ is the transition matrix for current states; $\mathbf{I}$ is an identity matrix; $\mathbf{w}_{k-1}$ represents the zero-mean Gaussian white noises.

The state estimation error covariance and system process noise error covariance matrices are given in their partitioned forms as:

$$\mathbf{P}'_k = \begin{bmatrix} \mathbf{P}_k & \mathbf{P}_{cd,k} \\ \mathbf{P}_{dc,k} & \mathbf{P}_{d,k} \end{bmatrix}, \quad \mathbf{Q}'_k = \begin{bmatrix} \mathbf{Q}_k & \mathbf{O} \\ \mathbf{O} & \mathbf{O} \end{bmatrix}, \quad (8)$$

where $\mathbf{P}_k$ is the error covariance of $\delta \mathbf{x}_k$; $\mathbf{P}_{d,k}$ is the error covariance of $\delta \mathbf{x}_{d,k}$, which is explicitly estimated from the previous measurement update; $\mathbf{P}_{cd,k}$ and $\mathbf{P}_{dc,k}$ are the cross-error covariance matrices between the current error state vector $\delta \mathbf{x}_k$ and the delay error state vector $\delta \mathbf{x}_{d,k}$; $\mathbf{Q}_k$ is the process noise error covariance of $\mathbf{w}_k$; $\mathbf{O}$ is a zero matrix.

For the delta pseudorange observation model, it is given as

$$\delta \mathbf{y}_k^\phi = \begin{bmatrix} \mathbf{H}_k & -\mathbf{H}_{d,k} \end{bmatrix} \begin{bmatrix} \delta \mathbf{x}_k \\ \delta \mathbf{x}_{d,k} \end{bmatrix} + \mathbf{v}_k^\phi, \quad (9)$$

where $\mathbf{H}_k$ and $\mathbf{H}_{d,k}$ are the Jacobian matrices related to the time instants of the start and end of the integration interval; $\delta \mathbf{y}_k^\phi$ is the delta pseudorange innovation vector; $\mathbf{v}_k^\phi$ represents the measurement noise.

As compared with (6), in this arrangement, we actually consider the “delay states” as a part of the current states. Thus, all the terms are assumed to be related with the current time instant k .

Besides the delta pseudorange, the code-derived pseudorange measurements are also considered in the filter. For the pseudorange measurements, the observation model is

$$\hat{\rho}_k - \rho_k = \mathbf{H}_k \delta \mathbf{x}_k + \mathbf{v}_k^\rho. \quad (10)$$

Substituting $\delta \mathbf{y}_k^\rho = \hat{\rho}_k - \rho_k$ into (10), the overall system observation model can be derived in its partitioned form as

$$\begin{bmatrix} \delta \mathbf{y}_k^\rho \\ \delta \mathbf{y}_k^\phi \end{bmatrix} = \begin{bmatrix} \mathbf{H}_k & \mathbf{O} \\ \mathbf{H}_k & -\mathbf{H}_{d,k} \end{bmatrix} \begin{bmatrix} \delta \mathbf{x}_k \\ \delta \mathbf{x}_{d,k} \end{bmatrix} + \begin{bmatrix} \mathbf{v}_k^\rho \\ \mathbf{v}_k^\phi \end{bmatrix}, \quad (11)$$

with $\mathbf{R}_k' = \begin{bmatrix} \mathbf{R}_k^\rho & \mathbf{O} \\ \mathbf{O} & \mathbf{R}_k^\phi \end{bmatrix}$,

where $\mathbf{R}_k^\rho$ and $\mathbf{R}_k^\phi$ are the pseudorange and delta pseudorange measurement error covariance matrices, which are assumed to be uncorrelated with each other.

Thus, (7) and (11) comprise the new system propagation and observation models, which will be used in the INS/GPS integration system. In the following subsections, we will firstly apply the sequential processing to update different types of measurements based on the new developed system models. And then, its computational burden is analyzed with respect to the conventional batch processing.

3.1. Measurement Updates. In this contribution, we process the pseudorange and delta pseudorange measurements as two batches of data. In this way, the dimension of observation vector will be half of the case, if we process the data in one batch.

For sequential processing, if the measurement errors are uncorrelated, they can be processed one after another with zero-width time interval until all measurements are sequentially updated. Otherwise, decoupling of the correlated measurement errors must be made, and linear combinations of measurements should be conducted to yield a new set of measurements whose errors are uncorrelated. After sequentially updating the measurements, the final state estimates and the associated error covariance matrix will be the same as if the measurements are processed in one batch. In this way, the time-consuming calculation of the matrix inversion can be prevented (e.g., [4, 5]). The exact number of numerical operations involved in the matrix inversion is calculated in the Appendix using the Gauss-Jordan Elimination method (e.g., [6]).

3.1.1. Sequential Updates of Delta Pseudorange Measurements. The conventional batch measurement update of delta pseudorange measurements is performed as follows:

$$\begin{aligned} \begin{bmatrix} \mathbf{K}_k \\ \mathbf{K}_{d,k} \end{bmatrix} &= \begin{bmatrix} \mathbf{P}_k^- & \mathbf{P}_{cd,k}^- \\ \mathbf{P}_{dc,k}^- & \mathbf{P}_{d,k}^- \end{bmatrix} \begin{bmatrix} \mathbf{H}_k^T \\ -\mathbf{H}_{d,k}^T \end{bmatrix} \\ &\times \left[\begin{bmatrix} \mathbf{H}_k & -\mathbf{H}_{d,k} \end{bmatrix} \begin{bmatrix} \mathbf{P}_k^- & \mathbf{P}_{cd,k}^- \\ \mathbf{P}_{dc,k}^- & \mathbf{P}_{d,k}^- \end{bmatrix} \right. \\ &\quad \left. \times \begin{bmatrix} \mathbf{H}_k^T \\ -\mathbf{H}_{d,k}^T \end{bmatrix} + \mathbf{R}_k^\phi \right]^{-1}, \end{aligned}$$

$$\begin{aligned} \begin{bmatrix} \delta \hat{\mathbf{x}}_k^+ \\ \delta \hat{\mathbf{x}}_{d,k}^+ \end{bmatrix} &= \begin{bmatrix} \delta \hat{\mathbf{x}}_k^- \\ \delta \hat{\mathbf{x}}_{d,k}^- \end{bmatrix} + \begin{bmatrix} \mathbf{K}_k \\ \mathbf{K}_{d,k} \end{bmatrix} \\ &\times \left[\delta \mathbf{y}_k^\phi - \begin{bmatrix} \mathbf{H}_k & -\mathbf{H}_{d,k} \end{bmatrix} \begin{bmatrix} \delta \hat{\mathbf{x}}_k^- \\ \delta \hat{\mathbf{x}}_{d,k}^- \end{bmatrix} \right], \\ \begin{bmatrix} \mathbf{P}_k^+ & \mathbf{P}_{cd,k}^+ \\ \mathbf{P}_{dc,k}^+ & \mathbf{P}_{d,k}^+ \end{bmatrix} &= \left[\begin{bmatrix} \mathbf{I} & \mathbf{O} \\ \mathbf{O} & \mathbf{I} \end{bmatrix} - \begin{bmatrix} \mathbf{K}_k \\ \mathbf{K}_{d,k} \end{bmatrix} \begin{bmatrix} \mathbf{H}_k & -\mathbf{H}_{d,k} \end{bmatrix} \right] \\ &\times \begin{bmatrix} \mathbf{P}_k^- & \mathbf{P}_{cd,k}^- \\ \mathbf{P}_{dc,k}^- & \mathbf{P}_{d,k}^- \end{bmatrix}, \end{aligned} \quad (12)$$

where $\mathbf{K}_k$ is the Kalman gain for the current error state vector $\delta \mathbf{x}_k$, while $\mathbf{K}_{d,k}$ is the Kalman gain for the delay error state vector $\delta \mathbf{x}_{d,k}$.

As stated before, the delay error states will not be updated from current measurement update. Thus, we can set $\mathbf{K}_{d,k}$ to be a zero matrix. Hereby, (12) can be simplified as

$$\begin{aligned} \mathbf{K}_k &= \left[\mathbf{P}_k^- \mathbf{H}_k^T - \mathbf{P}_{cd,k}^- \mathbf{H}_{d,k}^T \right] \\ &\times \left[\begin{bmatrix} \mathbf{H}_k \mathbf{P}_k^- \mathbf{H}_k^T - \mathbf{H}_{d,k} \mathbf{P}_{dc,k}^- \mathbf{H}_k^T \\ -\mathbf{H}_k \mathbf{P}_{cd,k}^- \mathbf{H}_{d,k}^T + \mathbf{H}_{d,k} \mathbf{P}_{d,k}^- \mathbf{H}_{d,k}^T + \mathbf{R}_k^\phi \end{bmatrix} \right]^{-1} \\ \delta \hat{\mathbf{x}}_k^+ &= \delta \hat{\mathbf{x}}_k^- + \mathbf{K}_k \left[\delta \mathbf{y}_k^\phi - \mathbf{H}_k \delta \hat{\mathbf{x}}_k^- + \mathbf{H}_{d,k} \delta \hat{\mathbf{x}}_{d,k}^- \right], \quad (13) \\ \mathbf{P}_k^+ &= (\mathbf{I} - \mathbf{K}_k \mathbf{H}_k) \mathbf{P}_k^- + \mathbf{K}_k \mathbf{H}_{d,k} \mathbf{P}_{dc,k}^-, \\ \mathbf{P}_{cd,k}^+ &= (\mathbf{I} - \mathbf{K}_k \mathbf{H}_k) \mathbf{P}_{cd,k}^- + \mathbf{K}_k \mathbf{H}_{d,k} \mathbf{P}_{d,k}^-, \\ \mathbf{P}_{dc,k}^+ &= \left(\mathbf{P}_{dc,k}^- \right)^T, \\ \mathbf{P}_{d,k}^+ &= \left(\mathbf{P}_{d,k}^- \right)^T. \end{aligned}$$

It is worth mentioning that, in (13), we have not considered the measurement update of $\delta \hat{\mathbf{x}}_{d,k}$, but the uncertainty of $\delta \hat{\mathbf{x}}_{d,k}$ (i.e., $\mathbf{P}_{d,k}^-$, $\mathbf{P}_{dc,k}^-$, $\mathbf{P}_{cd,k}^-$) is accounted for in the derivation of the Kalman gain $\mathbf{K}_k$.

In sequential update of delta pseudorange measurements, the following parameters need to be firstly initialized. We denote $\mathbf{H}_k(i)$ and $\mathbf{H}_{d,k}(i)$ as the i th row of $\mathbf{H}_k$ and $\mathbf{H}_{d,k}$, respectively, and use $\mathbf{R}_k^\phi(i, i)$ to represent the i th main diagonal element of $\mathbf{R}_k^\phi$.

$$\mathbf{P}_k(0) = \mathbf{P}_k^-, \mathbf{P}_{cd,k}(0) = \mathbf{P}_{cd,k}^-, \mathbf{P}_{dc,k}(0) = \mathbf{P}_{dc,k}^-, \delta \hat{\mathbf{x}}_k(0) = \delta \hat{\mathbf{x}}_k^-,$$

$$\begin{aligned} \mathbf{H}_k &= \begin{bmatrix} \mathbf{H}_k(1) \\ \vdots \\ \mathbf{H}_k(m) \end{bmatrix}, \quad \mathbf{H}_{d,k} = \begin{bmatrix} \mathbf{H}_{d,k}(1) \\ \vdots \\ \mathbf{H}_{d,k}(m) \end{bmatrix}, \\ \mathbf{R}_k^\phi &= \begin{bmatrix} \mathbf{R}_k^\phi(1, 1) & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \mathbf{R}_k^\phi(m, m) \end{bmatrix}. \end{aligned} \quad (14)$$

From $i = 1$ to m , the measurements are updated sequentially as follows:

$$\begin{aligned}
 \mathbf{K}_k(i) &= \left[\mathbf{P}_k(i-1)\mathbf{H}_k(i)^T - \mathbf{P}_{cd,k}(i-1)\mathbf{H}_{d,k}(i)^T \right] \\
 &\quad \times \left[\mathbf{H}_k(i)\mathbf{P}_k(i-1)\mathbf{H}_k(i)^T \right. \\
 &\quad \left. - \mathbf{H}_{d,k}(i)\mathbf{P}_{dc,k}(i-1)\mathbf{H}_k(i)^T \right. \\
 &\quad \left. - \mathbf{H}_k(i)\mathbf{P}_{cd,k}(i-1)\mathbf{H}_{d,k}(i)^T \right. \\
 &\quad \left. + \mathbf{H}_{d,k}(i)\mathbf{P}_{d,k}\mathbf{H}_{d,k}(i)^T + \mathbf{R}_k^\phi(i, i) \right]^{-1}, \\
 \delta\hat{\mathbf{x}}_k(i) &= \delta\hat{\mathbf{x}}_k(i-1) + \mathbf{K}_k(i) \\
 &\quad \times \left[\delta\mathbf{y}_k^\phi(i) - \mathbf{H}_k(i)\delta\hat{\mathbf{x}}_k(i-1) + \mathbf{H}_{d,k}(i)\delta\hat{\mathbf{x}}_{d,k} \right], \\
 \mathbf{P}_k(i) &= [1 - \mathbf{K}_k(i)\mathbf{H}_k(i)]\mathbf{P}_k(i-1) + \mathbf{K}_k(i)\mathbf{H}_{d,k}(i)\mathbf{P}_{dc,k}(i-1), \\
 \mathbf{P}_{cd,k}(i) &= [1 - \mathbf{K}_k(i)\mathbf{H}_k(i)]\mathbf{P}_{cd,k}(i-1) + \mathbf{K}_k(i)\mathbf{H}_{d,k}(i)\mathbf{P}_{d,k}, \\
 \mathbf{P}_{dc,k}(i) &= \mathbf{P}_{cd,k}(i)^T. \tag{15}
 \end{aligned}$$

3.1.2. Sequential Updates of Pseudorange Measurements. The batch processing of pseudorange measurements is handled after we process the delta pseudorange measurements. The *a posteriori* estimates from the delta pseudorange measurement update (i.e., $\delta\hat{\mathbf{x}}_k^+$ and $\mathbf{P}_k^+$) are used here as the *a priori* estimates (i.e., $\delta\hat{\mathbf{x}}_k^-$ and $\mathbf{P}_k^-$).

$$\begin{aligned}
 \mathbf{K}_k &= \mathbf{P}_k^- \mathbf{H}_k^T (\mathbf{H}_k \mathbf{P}_k^- \mathbf{H}_k^T + \mathbf{R}_k^\rho)^{-1}, \\
 \delta\hat{\mathbf{x}}_k^+ &= \delta\hat{\mathbf{x}}_k^- + \mathbf{K}_k [\delta\mathbf{y}_k^\rho - \mathbf{H}_k \delta\hat{\mathbf{x}}_k^-], \\
 \mathbf{P}_k^+ &= (\mathbf{I} - \mathbf{K}_k \mathbf{H}_k) \mathbf{P}_k^-. \tag{16}
 \end{aligned}$$

For sequential processing, we define

$$\mathbf{H}_k = \begin{bmatrix} \mathbf{H}_k(1) \\ \vdots \\ \mathbf{H}_k(m) \end{bmatrix}, \quad \mathbf{R}_k^\rho = \begin{bmatrix} \mathbf{R}_k^\rho(1,1) & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \mathbf{R}_k^\rho(m,m) \end{bmatrix}. \tag{17}$$

From $i = 1$ to m , the measurements are sequentially updated as

$$\begin{aligned}
 \mathbf{K}_k(m+i) &= \mathbf{P}_k(m+i-1)\mathbf{H}_k(i)^T \\
 &\quad \times \left[\mathbf{H}_k(i)\mathbf{P}_k(m+i-1)\mathbf{H}_k(i)^T + \mathbf{R}_k^\rho(i, i) \right]^{-1}, \\
 \delta\hat{\mathbf{x}}_k(m+i) &= \delta\hat{\mathbf{x}}_k(m+i-1) + \mathbf{K}_k(m+i) \\
 &\quad \times \left[\delta\mathbf{y}_k^\rho(i) - \mathbf{H}_k(i)\delta\hat{\mathbf{x}}_k(m+i-1) \right], \\
 \mathbf{P}_k(m+i) &= [1 - \mathbf{K}_k(m+i)\mathbf{H}_k(i)]\mathbf{P}_k(m+i-1). \tag{18}
 \end{aligned}$$

After all the measurements are updated, the *a posteriori* estimates are calculated as

$$\delta\hat{\mathbf{x}}_k^+ = \delta\hat{\mathbf{x}}_k(m+m), \quad \mathbf{P}_k^+ = \mathbf{P}_k(m+m). \tag{19}$$

They are used as the delay states and covariance parameters in the next recursion

$$\delta\hat{\mathbf{x}}_{d,k+1} = \delta\hat{\mathbf{x}}_k(m+m), \quad \mathbf{P}_{d,k+1} = \mathbf{P}_k(m+m). \tag{20}$$

The whole process repeats until the end of the trajectory.

3.2. Computational Burden Comparisons. It is known that the sequential processing can save computational time. But how much can be exactly saved? In order to answer this question, the number of numerical operations involved in addition and multiplication should be counted separately. The method utilized in [7] has been employed in this paper, where subtraction is counted as addition and the division over scalar is considered as multiplication by the inverse of the scalar. The “ n ” represents the dimension of state vector and “ m ” is the dimension of the measurement vector. We denote the “ nn ” as an n -by- n square matrix, and “ $nm \times nn$ ” stands for an n -by- n matrix multiplies another n -by- n matrix. “ $n1$ ” means an n -dimensional vector, “1” means a scalar value. The numbers in the column of “Num.” represent the amount of operations involved in the measurement update of both the pseudorange and delta pseudorange measurements, while the numbers in the parentheses denote the operations encountered in the delta pseudorange measurement update only. For batch measurement update (i.e., (13) and (16)), the involved numerical operations are given in the left side of Table 1. For measurement update in sequence (i.e., (15) and (18)), they are given in the right side of Table 1. Table 2 summaries the overall conducted numerical operations involved in sequential and batch measurement updates.

The correctness of the calculations in Table 2 can be easily verified when we consider the case that there is only one satellite in view ($m = 1$). In this case, there should be no difference between the sequential and batch processing. We can prove this by assigning m to be 1. The batch processing returns $18n^2 - 2n$ for addition, and $18n^2 + 12n + 2$ for multiplication, which are obviously equal to the corresponding calculations in sequential processing.

Based on the calculations in Table 2, the relationships between the number of numerical operations and the dimension of the observation vector are illustrated in Figures 1 and 2, where the dimension of the state vector is 17 (i.e., $n = 17$).

Figure 1 shows that if more measurements are processed, more computational time can be saved from sequential processing. However, if the number of measurements is smaller than 4, no big difference can be observed. In this paper, we update the delta pseudorange and pseudorange as two batches of data. And hence, in the calculation, the dimension of observation vector is always smaller than 11 (i.e., $m \leq 11$), which saves a large number of numerical operations.

Figure 2 illustrates that the measurement update of the delta pseudorange requires much more numerical operations than the pseudorange measurement update due to the increased computational complexity in (15) as compared with (18). However, when compared with the batch measurement update (i.e., red curves in Figure 2), they are both saved.

4. Tightly Coupled INS/GPS State Space Models

The INS propagation model we use is often seen for low-cost microelectromechanical- (MEMS-) based IMU, which

TABLE 1: Numerical operations involved in sequential and batch measurement updates.

Operation	Num.	Add.	Multi.	Operation	Num.	Add.	Multi.
$nn \times nm$	3 (2)	$(n-1)nm$	mn^2	$nn \times n1$	3 (2) $\times m$	$n(n-1)$	n^2
$nm \times mn$	5 (4)	$(m-1)n^2$	mn^2	$n1 + n1$	3 (2) $\times m$	n	—
$nm + nm$	1 (1)	nm	—	$1n \times nn$	10 (8) $\times m$	$n(n-1)$	n^2
$nm \times mm$	2 (1)	$(m-1)mn$	nm^2	$1n \times n1$	8 (6) $\times m$	$n-1$	n
$mn \times nn$	10 (8)	$mn(n-1)$	mn^2	$1 + 1$	8 (6) $\times m$	1	—
$mn \times nm$	5 (4)	$(n-1)m^2$	nm^2	1×1	2 (1) $\times m$	—	1
$mm + mm$	5 (4)	m^2	—	$n1 \times 1$	4 (2) $\times m$	—	n
$nn + nn$	5 (4)	n^2	—	$n1 \times 1n$	5 (4) $\times m$	—	n^2
$mn \times n1$	3 (2)	$m(n-1)$	mn	$nn + nn$	5 (4) $\times m$	n^2	—
$m1 + m1$	3 (2)	m	—				
$nm \times m1$	2 (1)	$(m-1)n$	nm				
$n1 + n1$	2 (1)	n	—				
$\text{inv}(mm)$	2 (1)	$(3/2)m^3 - 2m^2 + (m/2)$	$(3/2)m^3 - (1/2)m$				

TABLE 2: The sum of additions and multiplications involved in batch and sequential processing.

Operations	Additions	Operations	Multiplications
Batch	$3m^3 + (7n-4)m^2 + (18n^2-9n+1)m$	Batch	$3m^3 + 7nm^2 + (18n^2+5n-1)m$
Sequential	$(18n^2-2n)m$	Sequential	$(18n^2+12n+2)m$

FIGURE 1: Computational burden analysis (addition versus multiplication) with varying number of measurements.

is also used as the system propagation model in INS/GPS integrated system. In terms of a low-cost IMU, the effects from the earth rotation cannot be observed. Hence, they are not considered in the model. Moreover, the transport rate and Coriolis terms have been neglected for simplicity. The

simplified propagation model in discrete time domain can be expressed in navigation frame as:

$$\begin{aligned}
 \mathbf{p}_{n,k+1} &= \mathbf{p}_{n,k} + \mathbf{v}_{n,k} \cdot \Delta t, \\
 \mathbf{v}_{n,k+1} &= \mathbf{v}_{n,k} + \left[\mathbf{R}_{b2n,k} \cdot (\tilde{\mathbf{f}}_{b,k} - \hat{\mathbf{f}}_{b,k}^{\text{error}} - \mathbf{w}_f) + \mathbf{g}_n \right] \cdot \Delta t, \\
 \boldsymbol{\psi}_{k+1} &= \boldsymbol{\psi}_k + \mathbf{E}_{b2n,k} \cdot (\tilde{\boldsymbol{\omega}}_{b,k} - \hat{\boldsymbol{\omega}}_{b,k}^{\text{error}} - \mathbf{w}_\omega) \cdot \Delta t,
 \end{aligned} \tag{21}$$

where k is the time instant; $\tilde{\mathbf{f}}_b$ is the measurement vector of the specific force; $\tilde{\boldsymbol{\omega}}_b$ represents the measurement vector of angular rate; $\mathbf{g}_n$ is the local gravity vector; $\boldsymbol{\psi}$ is the Euler angles; $\hat{\mathbf{f}}_{b,k}^{\text{error}}$ and $\hat{\boldsymbol{\omega}}_{b,k}^{\text{error}}$ are the estimated specific force and angular rate errors; $\mathbf{w}_f$ and $\mathbf{w}_\omega$ are the remaining noise errors, which are assumed to be zero-mean Gaussian white noises; $\mathbf{R}_{b2n}$ is the frame rotation matrix from body frame to NED navigation frame; $\mathbf{E}_{b2n}$ is the rotation rate transformation matrix from body to navigation frame,

$$\begin{aligned}
 \mathbf{R}_{b2n} &= \begin{pmatrix} CyC\beta & CyS\beta S\alpha - SyC\alpha & CyS\beta C\alpha + SyS\alpha \\ SyC\beta & SyS\beta S\alpha + CyC\alpha & SyS\beta C\alpha - CyS\alpha \\ -S\beta & C\beta S\alpha & C\beta C\alpha \end{pmatrix}, \\
 \mathbf{E}_{b2n} &= \begin{pmatrix} 1 & S\alpha T\beta & C\alpha T\beta \\ 0 & C\alpha & -S\alpha \\ 0 & S\alpha/C\beta & C\alpha/C\beta \end{pmatrix},
 \end{aligned} \tag{22}$$

where CX, SX , and TX represent the trigonometric operations of cosine, sine, and tangent of X , respectively. The α, β, γ denote the roll, pitch, and yaw, respectively.

Regarding the INS/GPS tightly coupled integration, the extended Kalman filter (EKF) is considered as state of the art for fusing the INS and GPS data, and the error states are often used [8]. In this paper, the error states are defined as

$\delta \mathbf{x} = \hat{\mathbf{x}}_{\text{estimated}} - \mathbf{x}_{\text{true}}$. The system transition model $\Phi_{k,k-1}$ (i.e., in (7)) for current error state vector is shown in

(23), where the first three rows come from the first order linearization of (21) with respect to the corresponding states:

$$\begin{bmatrix} \delta \mathbf{p}_{n,k+1} \\ \delta \mathbf{v}_{n,k+1} \\ \delta \psi_{k+1} \\ \delta \mathbf{f}_{b,k+1}^{\text{error}} \\ \delta \omega_{b,k+1}^{\text{error}} \\ c\delta t_{k+1} \\ c\delta \dot{t}_{k+1} \end{bmatrix} = \begin{bmatrix} \mathbf{I}_{3 \times 3} & \mathbf{I}_{3 \times 3} \Delta t & \mathbf{O}_{3 \times 3} & \mathbf{O}_{3 \times 3} & \mathbf{O}_{3 \times 3} & \mathbf{0}_{3 \times 1} & \mathbf{0}_{3 \times 1} \\ \mathbf{O}_{3 \times 3} & \mathbf{I}_{3 \times 3} & \mathbf{F}_{23} & -\mathbf{R}_{b2n} \Delta t & \mathbf{O}_{3 \times 3} & \mathbf{0}_{3 \times 1} & \mathbf{0}_{3 \times 1} \\ \mathbf{O}_{3 \times 3} & \mathbf{O}_{3 \times 3} & \mathbf{F}_{33} & \mathbf{O}_{3 \times 3} & -\mathbf{E}_{b2n} \Delta t & \mathbf{0}_{3 \times 1} & \mathbf{0}_{3 \times 1} \\ \mathbf{O}_{3 \times 3} & \mathbf{O}_{3 \times 3} & \mathbf{O}_{3 \times 3} & \mathbf{I}_{3 \times 3} & \mathbf{O}_{3 \times 3} & \mathbf{0}_{3 \times 1} & \mathbf{0}_{3 \times 1} \\ \mathbf{O}_{3 \times 3} & \mathbf{O}_{3 \times 3} & \mathbf{O}_{3 \times 3} & \mathbf{O}_{3 \times 3} & \mathbf{I}_{3 \times 3} & \mathbf{0}_{3 \times 1} & \mathbf{0}_{3 \times 1} \\ \mathbf{0}_{3 \times 1}^T & \mathbf{0}_{3 \times 1}^T & \mathbf{0}_{3 \times 1}^T & \mathbf{0}_{3 \times 1}^T & \mathbf{0}_{3 \times 1}^T & 1 & \Delta t \\ \mathbf{0}_{3 \times 1}^T & \mathbf{0}_{3 \times 1}^T & \mathbf{0}_{3 \times 1}^T & \mathbf{0}_{3 \times 1}^T & \mathbf{0}_{3 \times 1}^T & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} \delta \mathbf{p}_{n,k} \\ \delta \mathbf{v}_{n,k} \\ \delta \psi_k \\ \delta \mathbf{f}_{b,k}^{\text{error}} \\ \delta \omega_{b,k}^{\text{error}} \\ c\delta t_k \\ c\delta \dot{t}_k \end{bmatrix} + \mathbf{w}_k. \quad (23)$$

In (23), the gyroscope and accelerometer sensor errors $\delta \mathbf{f}_b^{\text{error}}$ and $\delta \omega_b^{\text{error}}$ are modeled as random-walk plus constant (e.g., [8]). The receiver clock drift error $c\delta \dot{t}$ is modeled as

constant plus random walk process, while the receiver clock bias $c\delta t$ is the integral of the clock drift error. The submatrices $\mathbf{F}_{33}$ and $\mathbf{F}_{23}$ are given as:

$$\mathbf{F}_{33} = \begin{bmatrix} 1 + \left(\frac{\hat{\omega}_y C \alpha T \beta}{\hat{\omega}_z T \beta S \alpha} \right) \Delta t & \left[\frac{\hat{\omega}_z C \alpha (T^2 \beta + 1)}{\hat{\omega}_y S \alpha (T^2 \beta + 1)} \right] \Delta t & 0 \\ - \left[\frac{\hat{\omega}_z C \alpha}{\hat{\omega}_y S \alpha} \right] \Delta t & 1 & 0 \\ \left[\frac{\hat{\omega}_y C \alpha - \hat{\omega}_z S \alpha}{C \beta} \right] \Delta t & \left[\frac{C \alpha S \beta \hat{\omega}_z + S \beta S \alpha \hat{\omega}_y}{C^2 \beta} \right] \Delta t & 1 \end{bmatrix}, \quad (24)$$

$$\mathbf{F}_{23} = \begin{bmatrix} \hat{f}_y (S \alpha S \gamma + C \alpha C \gamma S \beta) & \hat{f}_z C \beta C \alpha C \gamma & \hat{f}_z (C \gamma S \alpha + C \alpha S \beta S \gamma) \\ + \hat{f}_z (C \alpha S \gamma + C \gamma S \beta S \alpha) & - \hat{f}_x C \gamma S \beta & - \hat{f}_y (C \alpha C \gamma - S \beta S \alpha S \gamma) \\ - \hat{f}_z (S \beta S \alpha S \gamma + C \alpha C \gamma) & + \hat{f}_y C \beta C \gamma S \alpha & - \hat{f}_x C \beta S \gamma \\ - \hat{f}_y (C \gamma S \alpha - C \alpha S \beta S \gamma) & \hat{f}_z C \beta C \alpha S \gamma & \hat{f}_z (S \alpha S \gamma + C \alpha C \gamma S \beta) \\ - \hat{f}_x S \beta S \gamma & - \hat{f}_x S \beta S \gamma & - \hat{f}_y (C \alpha S \gamma - C \gamma S \beta S \alpha) \\ + \hat{f}_y C \beta S \alpha S \gamma & + \hat{f}_x C \beta C \gamma & \\ \hat{f}_y C \beta C \alpha - \hat{f}_z C \beta S \alpha & - \hat{f}_x C \beta & 0 \\ - \hat{f}_x C \alpha S \beta & & \\ - \hat{f}_y S \beta S \alpha & & \end{bmatrix} \cdot \Delta t,$$

where $C^2 X$, $T^2 X$ represent the square of cosine and tangent trigonometric operations of X , and $\hat{f}_x = \tilde{f}_x - \hat{f}_x^{\text{error}}$, $\hat{f}_y = \tilde{f}_y - \hat{f}_y^{\text{error}}$, $\hat{f}_z = \tilde{f}_z - \hat{f}_z^{\text{error}}$, $\hat{\omega}_y = \tilde{\omega}_y - \hat{\omega}_y^{\text{error}}$, $\hat{\omega}_z = \tilde{\omega}_z - \hat{\omega}_z^{\text{error}}$ are the IMU raw data compensated with the estimated sensor errors expressed in body frame. The linearization can, for example, be accomplished using the ‘‘Jacobian’’ function in MATLAB.

The transition matrix for the delay state vector is an identity matrix. Thus, according to (7), we have already specified the system propagation model.

For the system observation model (i.e., in (11)), $\mathbf{H}_{d,k}$ and $\mathbf{H}_k$ are the Jacobian matrices from the first order linearization of the pseudorange nonlinear observation equations around the INS-estimated position at the time instants of the start and end of the integration interval. The Jacobian matrix related with the current error state vector is shown as:

$$\mathbf{H}_k = \begin{bmatrix} l_{n,k}^{sv(i)} & l_{e,k}^{sv(i)} & l_{d,k}^{sv(i)} & 0 & \cdots & 0 & 1 & 0 \end{bmatrix}_{i \times 17}, \quad (25)$$

where $l_{n,k}^{sv(i)}$, $l_{e,k}^{sv(i)}$, $l_{d,k}^{sv(i)}$ ($i = 1$ to m) denote the unit direction vectors pointing from the i th satellite position to the INS

estimated user position in navigation frame at time instant k . The delay observation matrix $\mathbf{H}_{d,k}$ is identical with $\mathbf{H}_k$, except that all the parameters involved are related to the time instant $k-1$ instead of k . In the recursive KF algorithm, we do not need to specify $\mathbf{H}_{d,k}$. Instead, we can simply use previous epoch observation matrix $\mathbf{H}_{k-1}$ as $\mathbf{H}_{d,k}$ in the current recursion.

5. Simulation Experiments

In the following simulation tests, a field-collected UAV trajectory is used. The position and velocity dynamic profiles are depicted in Figures 3 and 4. On the purpose of verifying the correctness of the proposed algorithm in the processing of integrated measurements sequentially, we assume that the GPS receiver is operating at the DGPS mode, in which majority of GPS measurement errors are corrected or minimized to small values, that is, the ionospheric and tropospheric delays. Hereby, they are neglected in this simulation.

FIGURE 2: Computational burden analysis (pseudorange versus delta pseudorange) with varying number of measurements.

FIGURE 3: UAV trajectory.

For the GPS receiver, we assume that we use a NovAtel DL-4plus GPS receiver. The parameters shown in Table 3 are used to simulate the receiver-related measurement errors. The multipath errors are generated using SATNAV toolbox from GPSof LLC. The basic idea is to form the zero-elevation angle equivalent pseudorange multipath errors using a linear autoregressive model (i.e., “filter” function in MATLAB) and scale these errors by the cosine of their elevation angles before they are applied to the range measurements. Regarding the IMU, an automotive level LandMark 20 eXT MEMS-IMU from Gladiators Technologies Inc. is simulated. The main sensor errors are generated according to its specification as

FIGURE 4: Trajectory dynamics.

TABLE 3: Parameters used for generating the receiver-related measurement errors (1σ).

	Receiver thermal noise	Multipath errors
Pseudorange (DLL)	0.5 m	0.4 m
Doppler (FLL)	0.05 m/s	0.04 m/s
Carrier phase (PLL)	0.001 m	0.004 m

shown in Table 4. The IMU raw data (simulated errors plus the field collected high quality IMU raw data) are plotted in Figure 5. For the integrated KF, the system time-update happens at the IMU measurements update rate, which is 50 Hz, while the measurement-update happens at the GPS measurement update rate, which is 5 Hz. The simulation is conducted with 8 satellites in view.

With this simulation setup, two tests are made. In the first test, we verify the advantage of correctly handling the errors in the delta pseudorange measurements. And in the second one, we prove that the sequential measurement update of different types of measurements presents identical system estimation accuracy with respect to the batch processing, if certain conditions are met.

5.1. System Estimation Accuracy Using “Augmented” System Model. We initialize the filter to be 1 m away to the north, east and down directions from their true values. The errors analyzed here, represent the norm of position, velocity and attitude errors, respectively. For instance, the norm of position errors will be calculated as $\|\Delta \mathbf{x}\| = \sqrt{\Delta x_n^2 + \Delta x_e^2 + \Delta x_d^2}$. The comparisons are made between conditions listed in Table 5, and the results are depicted in Figures 6 and 7.

For the “augmented” system (red curves), both code-derived pseudorange and carrier-phase-derived delta pseudorange measurements are used in the position determination through the system observation model (i.e., (11)).

TABLE 4: Landmark 20 eXT MEMS-based IMU performance specification.

Gyroscope (angular rates)	Bias in-run stability 20 [$^{\circ}$ /h] (1σ)	Noise (ARW) 0.035 [$^{\circ}$ /s/ $\sqrt{\text{Hz}}$] (1σ)	Scale factor error ≤ 1000 [ppm]
Accelerometer (specific forces)	Bias in-run stability 20 [μg] (1σ)	Noise (VRW) 40 [μg / $\sqrt{\text{Hz}}$] (1σ)	Scale factor error ≤ 1000 [ppm]

FIGURE 5: Simulated LandMark 20 eXT MEMS IMU raw data.

TABLE 5: Conditions for comparison.

Red	“Augmented” system using pseudorange and delta pseudorange measurements
Green	“Augmented” system using delta pseudorange measurements alone
Blue	System using pseudorange for position and delta pseudorange for velocity
Black	System using pseudorange for position and Doppler shift for velocity

The pseudorange provides the absolute positioning information, while the delta pseudorange provides the relative displacement of the system position from one time instant

FIGURE 6: Comparison of position and velocity estimates (the 3rd subplot is the zoom-in of the 2nd subplot).

to the next. In such case, when the filter is tuned to give more confidence on the pseudorange measurements, the position estimation errors will converge faster, but the system performance will be noisier. On the other hand, when the filter gives more confidence on the delta pseudorange measurements, the estimation errors will converge slower, but system presents much smoothed estimation results. By giving the weights on the pseudorange and delta pseudorange measurements based on the errors introduced onto the GPS raw data, the red curves are obtained. In Figure 6, the third subplot is the “zoom-in” version of the second subplot, where the red curve shows the velocity estimation errors using the “augmented” system model. It demonstrates that with correctly handling the errors involved in the delta pseudorange measurement, the filter can correctly track the accelerations and jerks involved in the trajectory.

The green curves in Figure 6 show an extreme case. That is, if there are no pseudorange measurements (absolute positioning information) available in the filter, with only the delta pseudorange measurements (relative positioning information), the system presents the best velocity estimation

FIGURE 7: Comparison of attitude estimates (the 2nd subplot is the zoom-in of the 1st subplot).

accuracy. However, for position, the errors from the incorrect initialization remain through the whole trajectory, as shown in the first subplot of Figure 6.

The blue curves in Figure 6 represent the estimation results using the conventional approach. That is, to relate the pseudorange for position estimation, and the delta pseudorange for velocity estimation. In this test, the dynamics encountered in the trajectory is large. Therefore, large velocity estimation errors appear. The same problem can be observed from the attitude estimation results (i.e., Figure 7).

In this paper, we integrate the measurements obtained from a single-GPS receiver antenna with measurements from an INS without redundant attitude information (e.g., from magnetometers or a multi-antenna GPS system). Therefore, no attitude fixes are available. In such case, the attitude information is contained in the position and velocity information, and its errors are related to the position and velocity estimation errors through the off-diagonal parameters in the error covariance matrices. Hence, the greater the accuracy of the position and velocity estimates, the greater the dampening on the attitude errors.

As an alternate approach, the Doppler shift measurements are often used for velocity determination (black curves in Figures 6 and 7). However, they are the raw measurements from the receiver frequency lock loop (FLL), which is known to be noisier than the carrier-phase-derived measurements.

5.2. Comparison between Sequential and Batch Measurement Updates. In this test, the simulation setup remains unchanged as the aforementioned one. Figure 8 plots the position, velocity, and attitude estimation errors from 50 Monte Carlo runs (with 50 simulated GPS and IMU data sets based on parameters in Tables 3 and 4) using the “augmented” system with sequential and batch measurement updates.

FIGURE 8: System performance comparisons.

As depicted in the figure, the sequential measurement update presents equivalent performance as compared with batch measurement update. Nevertheless, two rules must be obeyed.

- (1) If we process a group of measurements, which involve both integrated and nonintegrated measurements, we must always process the integrated measurements (delta pseudorange) before the nonintegrated measurements (pseudorange) to guarantee that there is no measurement update before we process the integrated measurements.
- (2) The cross-covariance matrices (i.e., $\mathbf{P}_{cd,k}$ and $\mathbf{P}_{dc,k}$) need to be carefully treated. In sequential processing, among the measurement scalar updates, these cross covariance matrices, that is, $\mathbf{P}_{cd,k}$ and $\mathbf{P}_{dc,k}$, are updated from one measurement to the other with zero-width time interval, as shown in (15).

6. Conclusion

The errors involved in delta pseudorange measurements are related to the pseudorange errors at the start and end of the integration interval. The correct handling of these errors is essential in cases that the vehicle is maneuvering under high dynamics with low GPS update rate. In this contribution, the method with the “augmented” system state vector is considered, and is employed in the INS/GPS tightly coupled

TABLE 6: Computational cost of matrix inverse calculation.

Steps	Left side		Right side		Rows processed
	Multi.	Add.	Multi.	Add.	
1	n	$n-1$	n	n	$n-1$
2	$n-1$	$n-2$	n	n	$n-1$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
$n-1$	2	1	n	n	$n-1$
n	1	0	n	n	$n-1$
Division	0	0	n	0	n

integration. Simulation is conducted using a field-collected UAV trajectory. Numerical results show that the filter can correctly track the dynamics in the trajectory. Moreover, sequential processing is used for measurement update in the KF to prevent the matrix inverse calculation. Monte Carlo runs are conducted and verify that the sequential processing can update different types of measurements (i.e., integrated and nonintegrated) without degrading the system estimation accuracy, if certain rules are met.

Appendix

Calculation of the Matrix Inversion Using Gauss-Jordan Elimination Method

The Gauss-Jordan Elimination method is one of the methods which are widely used to calculate the inverse of a square matrix (e.g., [6]). It is done by augmenting the square matrix with an identity matrix of the same dimensions, and passing through the following matrix operations

$$\begin{aligned}
 &\text{multiply} \quad \left[\begin{array}{cccc|cccc} a_{11} & a_{12} & \cdots & a_{1n} & 1 & 0 & \cdots & 0 \\ a_{21} & a_{22} & \cdots & a_{2n} & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} & 0 & 0 & \cdots & 1 \end{array} \right], \\
 &\text{by } \mathbf{A}^{-1} \text{ to have} \quad \left[\begin{array}{cccc|cccc} 1 & 0 & \cdots & 0 & b_{11} & b_{12} & \cdots & b_{1n} \\ 0 & 1 & \cdots & 0 & b_{21} & b_{22} & \cdots & b_{2n} \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 1 & b_{n1} & b_{n2} & \cdots & b_{nn} \end{array} \right]. \quad (\text{A.1})
 \end{aligned}$$

After the Gauss-Jordan elimination, the right side of the augmented matrix is the inverse of the original square matrix. In this contribution, we firstly transform the left side of the augmented matrix (original square matrix) to be a diagonal matrix and divide the augmented matrix by its diagonal elements on each row to have an identity matrix on the left side and the inverse of original matrix on the right side. The computation is shown as follows.

In the first step, we use the elementary row operations to introduce zeros in the first column beginning from the

second row, which involves $2n$ multiplications and $2n-1$ additions:

$$\left[\begin{array}{cccc|cccc} a_{11} & a_{12} & \cdots & a_{1n} & b_{11} & b_{12} & \cdots & b_{1n} \\ 0 & a_{22}^{(1)} & \cdots & a_{2n}^{(1)} & b_{21}^{(1)} & b_{22}^{(1)} & \cdots & b_{2n}^{(1)} \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & a_{n2}^{(1)} & \cdots & a_{nn}^{(1)} & b_{n1}^{(1)} & b_{n2}^{(1)} & \cdots & b_{nn}^{(1)} \end{array} \right]_{n \times 2n}. \quad (\text{A.2})$$

After the first step, the elements in the rows below the first will change. We indicate this by denoting the elements with a superscript, which stands for the number of arithmetic operations involved.

In the second step, we introduce zeros in the second column except the second row, which involves $2n-1$ multiplications and $2n-2$ additions.

In the n -th step, we introduce zeros in the n -th column except the n -th row, which involves $n+1$ multiplications and n additions:

$$\left[\begin{array}{cccc|cccc} a_{11} & 0 & \cdots & 0 & b_{11}^{(n-1)} & b_{12}^{(n-1)} & \cdots & b_{1n}^{(n-1)} \\ 0 & a_{22}^{(1)} & \cdots & 0 & b_{21}^{(n-1)} & b_{22}^{(n-1)} & \cdots & b_{2n}^{(n-1)} \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & a_{nn}^{(n-1)} & b_{n1}^{(n-1)} & b_{n2}^{(n-1)} & \cdots & b_{nn}^{(n-1)} \end{array} \right]_{n \times 2n}. \quad (\text{A.3})$$

In the last step, we divide the augmented matrix by its diagonal elements on each row to have the matrix as shown in (A.1). The involved numerical operations are given in Table 6.

Using the summation formula

$$\sum_{i=1}^n i = \frac{n(n+1)}{2}. \quad (\text{A.4})$$

According to Table 6, the number of multiplications involved in Gauss-Jordan Elimination is

$$\begin{aligned}
 &(n-1) \sum_{i=1}^n i + n^2(n-1) + n^2 \\
 &= (n-1) \frac{n(n+1)}{2} + n^2(n-1) + n^2 = \frac{3}{2}n^3 - \frac{1}{2}n.
 \end{aligned} \quad (\text{A.5})$$

And the required number of additions is computed as

$$(n-1) \sum_{i=1}^{n-1} i + n^2(n-1) = \frac{3}{2}n^3 - 2n^2 + \frac{n}{2}. \quad (\text{A.6})$$

We can verify our calculation by inverting a scalar number as an extreme example for square matrix inversion (i.e., $n = 1$). The arithmetic operations involved contain obviously only one multiplication (division), which is the same as the results computed from (A.5) and (A.6).

Acknowledgment

Part of the work reported herein has been funded by the German Research Foundation (DFG), Grant no. KN 876/1-2, which is gratefully acknowledged.

References

- [1] R. G. Brown and P. Y. C. Hwang, *Introduction to Random signals and Applied Kalman Filtering*, John Wiley and Sons, New York, NY, USA, 1997.
- [2] P. Misra and P. Enge, *Global Positioning System: Signals, Measurements, and Performance*, Ganga-Jamuna Press, 2007.
- [3] D. Hartman and D. B. Tyler, "New Kalman filter formulation for GPS delta pseudorange processing," in *ION Proceedings of GPS*, pp. 1395–1400, Nashville, Tenn, USA, 1998.
- [4] P. S. Maybeck, *Stochastic Models, Estimation, and Control*, vol. 1, Academic Press, New York, NY, USA, 1982.
- [5] D. Simon, *Optimal State Estimation: Kalman, H Infinity, and Nonlinear Approaches*, Wiley-Interscience, Hoboken, NJ, USA, 2006.
- [6] G. Strang and K. Borre, *Linear Algebra, Geodesy, and GPS*, Wellesley-Cambridge Press, 1997.
- [7] Y. Li, C. Rizos, J. L. Wang, P. Mumford, and W. D. Ding, "Sigma-Point Kalman filtering for tightly coupled INS/GPS integration," *Journal of the Institute of Navigation*, vol. 55, no. 3, pp. 167–179, 2008.
- [8] J. Farrell, *Aided Navigation: GPS with High Rate Sensors*, McGraw-Hill Professional, New York, NY, USA, 2008.