

Research Article

Evaluation of Matrix Square Root Operations for UKF within a UAV GPS/INS Sensor Fusion Application

Matthew Rhudy, Yu Gu, Jason Gross, and Marcello R. Napolitano

Department of Mechanical and Aerospace Engineering, West Virginia University, P.O. Box 6106, Morgantown, WV 26506, USA

Correspondence should be addressed to Matthew Rhudy, mbr5002@gmail.com

Received 22 July 2011; Revised 14 December 2011; Accepted 21 December 2011

Academic Editor: Jinling Wang

Copyright © 2011 Matthew Rhudy 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.

Using an Unscented Kalman Filter (UKF) as the nonlinear estimator within a Global Positioning System/Inertial Navigation System (GPS/INS) sensor fusion algorithm for attitude estimation, various methods of calculating the matrix square root were discussed and compared. Specifically, the diagonalization method, Schur method, Cholesky method, and five different iterative methods were compared. Additionally, a different method of handling the matrix square root requirement, the square-root UKF (SR-UKF), was evaluated. The different matrix square root calculations were compared based on computational requirements and the sensor fusion attitude estimation performance, which was evaluated using flight data from an Unmanned Aerial Vehicle (UAV). The roll and pitch angle estimates were compared with independently measured values from a high quality mechanical vertical gyroscope. This manuscript represents the first comprehensive analysis of the matrix square root calculations in the context of UKF. From this analysis, it was determined that the best overall matrix square root calculation for UKF applications in terms of performance and execution time is the Cholesky method.

1. Introduction

The improvement of microprocessors and sensors has increased civilian use of Unmanned Aerial Vehicles (UAVs) for various applications, many of which requiring an accurate estimate of the aircraft attitude [1, 2]. For example, attitude estimation is an important requirement for UAV remote sensing applications such as 3D mapping with direct georeferencing [3] or constructing large mosaics [4]. Due to cost and weight restrictions [5, 6], high-quality military grade inertial navigation systems may not be practical [6, 7]. Therefore, attitude estimation algorithms that rely only on low-cost sensors have become essential for civilian applications. A popular approach to the attitude estimation problem involves fusing together information from a low-cost Inertial Navigation System (INS) with information from a Global Positioning System (GPS) receiver [8, 9]. Various formulations and analyses of GPS/INS sensor fusion exist in the literature [10–12], including a detailed comparison by the authors with respect to attitude estimation performance and computational cost [13].

The Unscented Kalman Filter (UKF) [14] is a well-known nonlinear estimator for use within GPS/INS sensor fusion [10, 11, 15]. Within this effort, the UKF is used to obtain an accurate estimate of the aircraft attitude, in particular the roll and pitch angles. For this type of problem, an alternative nonlinear estimator is the Extended Kalman Filter (EKF) [16], whose differences are discussed in detail in various sources [11, 12, 15, 17], including a sensitivity analysis by the authors with respect to different design parameters [18]. One primary difference between the EKF and UKF is that the UKF requires the calculation of a matrix square root, which is a computationally demanding operation [19]. In particular, computation is an important consideration for small UAV systems, due to resource limitations onboard the aircraft, for example, power, weight, and cost [5, 6]. The matrix square root computation is also an important consideration when designing a UKF algorithm because there are many different ways to compute the square root of a matrix, potentially with different accuracy and computational requirements [20].

The selection of matrix square root operation in the UKF differs among authors. Julier and Uhlmann state that “the

sigma points capture the same mean and covariance irrespective of the choice of matrix square root which is used. Numerically efficient and stable methods such as the Cholesky decomposition can be used [14]. Crassidis [10] and Wendel et al. [21] also recommended the use of Cholesky decomposition. These claims, however, are provided without any theoretical or empirical justification. Stastny et al. found that using the Cholesky decomposition method caused divergence; therefore, they used the Schur method instead [22]. Some authors using the UKF do not explicitly state which matrix square root operation was used [23–26]. A different means of handling the square root operation was developed by van der Merwe and Wan called the “square-root UKF (SR-UKF)” method [19]. This method provides a prediction and update of the square root of the covariance matrix directly at each time step, which reduces computational requirements of the algorithm [19]. A simulated example was used to show that the SR-UKF and UKF performances were the same [19]. Since there are inconsistencies in the selected matrix square root method for UKF applications, a detailed comparison of available approaches is necessary. Although some matrix square root comparison studies exist, for example, by Higham [20] and Meini [27], these studies are mostly theoretical, with a few examples using known matrices, such as the Moler, Chebyshev-Vandermonde, and Frobenius matrices. These comparison studies, therefore, do not consider the potential error propagation effects that are introduced by application in a recursive filter such as the UKF.

This paper aims to expand upon the existing matrix square root comparison studies through an example application of the matrix square root within a UKF-based GPS/INS sensor fusion algorithm for attitude estimation that relies on experimentally collected flight data. By analyzing different matrix square root methods in the context of the UKF, a matrix square root is required at each discrete time step in the algorithm, allowing for a more general comparison since many different matrices are considered. In addition this recursive filtering application introduces the effects of the propagation of uncertainties in the matrix square root computation. Furthermore, the flight data used for this study were selected from a large library of data in order to obtain diversity with respect to different flight conditions, thus providing an additional level of generalization. The use of experimental data for this study provides a necessary empirical approach to a problem that has previously only been considered in a theoretical context. To evaluate the performance of the attitude estimation algorithm, the aircraft pitch and roll estimates are compared with measurements acquired independently from the IMU and GPS with a high-quality mechanical vertical gyroscope. This sensor fusion study builds upon previous work conducted by various West Virginia University (WVU) researchers [13, 18, 28, 29].

The work presented herein represents the first comprehensive analysis of matrix square root calculations used for UKF applications. This study utilizes a diverse set of experimental flight data to analyze the performance speed and accuracy of different matrix square root calculations in the context of a UKF-based GPS/INS sensor fusion algorithm. The rest of the paper is organized as follows. Section 2

describes the theoretical background required for this study. Section 3 discusses the UAV attitude estimation problem, including information about the UAV research platform as well as the problem formulation. Section 4 presents the results, and finally Section 5 provides a summary and discussion of the results.

2. Theoretical Background

2.1. Unscented Kalman Filter (UKF). Formulations of the UKF for GPS/INS sensor fusion are well known and can be found in a number of publications [10, 11, 14]. For this application of the UKF, the state vector is augmented to include navigation states, x , and process noise states, v , leading to the augmented state vector, $x^a = [x^T v^T]^T$, where the superscript a denotes augmentation, and the process noise states are assumed to be zero mean Gaussian and mutually independent. The total dimension of the augmented state vector is denoted by l . The source of process noise modeled in this formulation is additive upon an input vector comprised of IMU measurements: $x_{k|k-1} = f(x_{k-1|k-1}, u_k + v_k)$, where the state prediction equations corresponding to the process noise states are simply $v_k = v_{k-1} + w_k$, where w is a vector of mutually independent zero mean Gaussian noise. These states are initialized to zero and remain constant with time. The effect of the process noise, however, is implemented implicitly by the use of the unscented transformation. Due to the state augmentation, the state covariance matrix is augmented accordingly to include the noise characteristics of the process noise states.

The UKF uses an unscented transformation [14] to obtain the *a posteriori* estimates of mean and covariance. The first step of the UKF implementation is to calculate $2l + 1$ sigma points based on the square-root of the augmented state covariance matrix:

$$\begin{aligned} \chi_{k-1|k-1}^a &= \begin{bmatrix} \chi_{k-1|k-1}^x \\ \chi_{k-1|k-1}^v \end{bmatrix} \\ &= \left[x_{k-1|k-1}^a x_{k-1|k-1}^a + \eta \sqrt{P_{k-1|k-1}^a} x_{k-1|k-1}^a \right. \\ &\quad \left. - \eta \sqrt{P_{k-1|k-1}^a} \right], \end{aligned} \quad (1)$$

where χ is a matrix of state sigma points, and η is the sigma point spread parameter, given by [11]

$$\eta = \sqrt{l + \lambda}, \quad (2)$$

$$\lambda = l(\alpha^2 - 1), \quad (3)$$

where λ is the compound sigma point parameter, and α is the primary sigma point scaling parameter, which is suggested to vary between 0.001 and 1 [11].

The prediction step for the UKF consists of passing each state sigma point through the state prediction equations:

$${}^i\chi_{k|k-1}^x = f\left({}^i\chi_{k-1|k-1}^x, u_k + {}^i\chi_{k-1|k-1}^v\right), \quad i = 0, 1, \dots, 2l, \quad (4)$$

where the superscript i denotes the $(i + 1)$ th column of the matrix. Then, the state mean and covariance are predicted

using a weighted average of the transformed sigma points using

$$x_{k|k-1} = \sum_{i=0}^{2l} w_m^i \chi_{k|k-1}^x, \quad (5)$$

$$P_{k|k-1} = \sum_{i=0}^{2l} w_c^i \left(\chi_{k|k-1}^x - x_{k|k-1} \right) \left(\chi_{k|k-1}^x - x_{k|k-1} \right)^T, \quad (6)$$

where w_m and w_c are weight vectors [11]. For the measurement update, this paper considers a problem that has linear output equations; therefore a linear Kalman filter update can be used [30].

2.2. Matrix Square Root Algorithms. An important requirement of the UKF algorithm is the calculation of the square root of the state covariance matrix, P . A covariance matrix by definition is both symmetric and positive semidefinite. To calculate the square root of a positive semidefinite $n \times n$ matrix, A , various methods can be used. The matrix principal square root, $A^{1/2}$, exists only for positive semidefinite matrices and is defined by [20]

$$A = A^{1/2} \cdot A^{1/2}. \quad (7)$$

If A is symmetric, it can be diagonalized using a similarity transformation [31], and the principal square root can be calculated using the diagonalization method:

$$A^{1/2} = X \Lambda^{1/2} X^{-1}, \quad (8)$$

where $\Lambda^{1/2}$ is a diagonal matrix with the square roots of the eigenvalues of A along the main diagonal, and X is a matrix containing a corresponding set of eigenvectors of A .

Another common matrix square root method is the Schur method, which uses the Schur decomposition:

$$A = UTU^*, \quad (9)$$

where T is an upper triangular matrix and U is a unitary matrix whose columns form the Schur basis of A [32], and the $(*)$ denotes the complex conjugate transpose of a matrix. Once in this form, the matrix square root can be calculated from

$$A^{1/2} = UT^{1/2}U^*, \quad (10)$$

where $T^{1/2}$ can be calculated algebraically since T , and therefore also $T^{1/2}$, are upper triangular matrices. Let $S = T^{1/2}$. The diagonal elements of S are calculated directly from the diagonal elements of T such that [20]

$$s_{ii} = \sqrt{t_{ii}}, \quad i = 1, \dots, n. \quad (11)$$

The strictly upper triangular elements are then calculated using [20]

$$s_{ij} = \frac{t_{ij} - \sum_{k=i+1}^{j-1} s_{ik}s_{kj}}{u_{ii} + u_{jj}}, \quad j = 2, \dots, n, \quad i < j. \quad (12)$$

In addition to analytical methods for calculating a matrix square root, various iterative methods have been derived. One of the most common iterative methods is Newton's

method, which can be written as

$$X_{k+1} = \frac{1}{2} \left(X_k + X_k^{-1} A \right), \quad X_0 = A, \quad (13)$$

where the matrix X converges quadratically to $A^{1/2}$ under certain conditions [33]. One variant of Newton's method is known as Denman Beavers (DB) iteration, given by [20]

$$\begin{aligned} X_{k+1} &= \frac{1}{2} \left(X_k + Y_k^{-1} \right), \quad X_0 = A, \\ Y_{k+1} &= \frac{1}{2} \left(Y_k + X_k^{-1} \right), \quad Y_0 = I, \end{aligned} \quad (14)$$

where the matrix X converges to $A^{1/2}$, and the matrix Y converges to $A^{-1/2}$. A product form of the Denman-Beavers iteration was identified by Cheng et al., which is a more efficient version of the Denman-Beavers iteration [34]:

$$\begin{aligned} M_{k+1} &= \frac{1}{2} \left(I + \frac{M_k + M_k^{-1}}{2} \right), \quad M_0 = A, \\ X_{k+1} &= \frac{1}{2} X_k \left(I + M_k^{-1} \right), \quad X_0 = A, \\ Y_{k+1} &= \frac{1}{2} Y_k \left(I + M_k^{-1} \right), \quad Y_0 = I, \end{aligned} \quad (15)$$

where M converges to the identity matrix, I , X converges to $A^{1/2}$, while Y converges to $A^{-1/2}$. Another iterative method is the cyclic reduction (CR) iteration [27]:

$$\begin{aligned} Y_{k+1} &= -Y_k Z_k^{-1} Y_k, \quad Y_0 = I - A, \\ Z_{k+1} &= Z_k + 2Y_{k+1}, \quad Z_0 = 2(I + A), \end{aligned} \quad (16)$$

where Y converges to 0 and Z converges to $4A^{1/2}$. A variant of the cyclic reduction iteration is the IN iteration given by [20]

$$\begin{aligned} X_{k+1} &= X_k + E_k, \quad X_0 = A, \\ E_{k+1} &= -\frac{1}{2} E_k X_{k+1}^{-1} E_k, \quad E_0 = \frac{1}{2} (I - A), \end{aligned} \quad (17)$$

where X converges to $A^{1/2}$, and E converges to 0.

All of the matrix square root algorithms discussed up to this point are methods of calculating the principle square root of a matrix. Another form of the matrix square root is found using the Cholesky decomposition [35]:

$$A = LL^T, \quad (18)$$

where L is a lower triangular matrix which can be considered as a form of the matrix square root. Although most numerical methods of calculating the Cholesky decomposition of a matrix require positive definiteness, there are ways to calculate this decomposition for positive semidefinite matrices [36, 37], although in general this result is not unique. For this application, the state covariance matrix, P , was positive definite at each time step. In general, however, it is possible in a given application for the state covariance matrix to be positive semidefinite; in particular this can occur if some linear combinations of the states are known perfectly, that is, zero uncertainty in that combination of states.

TABLE 1: Matrix square root algorithm computational requirement summary.

Algorithm	FLOPs
Diagonalization	$49n^3/3 - n^2/2 + 43n/6$
Schur	$85n^3/3$
Newton's iteration	$8n^3/3$ (per iteration)
Denman-Beavers (DB) iteration	$4n^3$ (per iteration)
Product DB iteration	$4n^3$ (per iteration)
Cyclic Reduction (CR) iteration	$14n^3/3$ (per iteration)
IN iteration	$14n^3/3$ (per iteration)
Cholesky	$n^3/3$

The computational complexity of the matrix square root operation is important for use in the UKF because it is a significantly expensive part of the UKF algorithm [19]. To analyze the computational complexity of the different matrix square root algorithms, the number of floating point operations (FLOPs) as a function of the matrix dimension, n , was determined from various sources [20, 35, 38]. These results were derived theoretically based on the fundamental requirements of the algorithm and are summarized in Table 1. Note that the number of FLOPs listed for the iterative methods is the number of FLOPs required for one iteration of the algorithm.

It is shown in Table 1 that each of the matrix square root algorithms requires computations on the order of n^3 , but scaled by different factors. A graphical comparison of these requirements is presented later in the paper in Figure 3(b).

$$\begin{bmatrix} \dot{V}_x \\ \dot{V}_y \\ \dot{V}_z - g \end{bmatrix} = \begin{bmatrix} \cos \theta \cos \psi & -\cos \phi \sin \psi + \sin \phi \sin \theta \cos \psi & \sin \phi \sin \psi + \cos \phi \sin \theta \cos \psi \\ \cos \theta \sin \psi & \cos \phi \cos \psi + \sin \phi \sin \theta \sin \psi & -\sin \phi \cos \psi + \cos \phi \sin \theta \sin \psi \\ -\sin \theta & \sin \phi \cos \theta & \cos \phi \cos \theta \end{bmatrix} \begin{bmatrix} a_{\tilde{x}} \\ a_{\tilde{y}} \\ a_{\tilde{z}} \end{bmatrix}. \quad (20)$$

The acceleration due to gravity, g , must be removed from the IMU accelerometer measurements, since those measurements are relative to free fall. Since the direction of g varies in the aircraft body frame, it is not easy to remove this effect directly from the IMU accelerometer measurements. However, since g acts in the negative local Cartesian z -direction, g is instead subtracted from the local Cartesian z -acceleration in order to compensate for this effect, as shown in +(20).

The IMU measurements of roll, pitch, and yaw rates, p , q , and r , are related to the corresponding Euler angles, ϕ , θ , and ψ using the attitude state equations, also known as the Inverted Kinematic equations [39]:

$$\begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} 1 & \sin \phi \tan \theta & \cos \phi \tan \theta \\ 0 & \cos \phi & -\sin \phi \\ 0 & \sin \phi \sec \theta & \cos \phi \sec \theta \end{bmatrix} \begin{bmatrix} p \\ q \\ r \end{bmatrix}. \quad (21)$$

The state equations (19)–(21) are initially provided in the continuous time domain. In order to implement the discrete

3. UAV Attitude Estimation Problem

3.1. GPS/INS Sensor Fusion Formulation. The sensor fusion formulation considered for this study has states given by the local Cartesian position r_x, r_y , and r_z , local Cartesian velocity, V_x, V_y , and V_z , and the attitude of the aircraft represented by the Euler angles, ϕ, θ , and ψ , corresponding to the roll, pitch, and yaw angles, respectively. Therefore, the state vector is given by $x = [r_x \ r_y \ r_z \ V_x \ V_y \ V_z \ \phi \ \theta \ \psi]^T$. Since the position and velocity states are given in the same coordinate frame, their relationship defines the position state equations:

$$\begin{bmatrix} \dot{r}_x \\ \dot{r}_y \\ \dot{r}_z \end{bmatrix} = \begin{bmatrix} V_x \\ V_y \\ V_z \end{bmatrix}. \quad (19)$$

An Inertial Measurement Unit (IMU) provides measurements of the linear accelerations as specific force, $a_{\tilde{x}}, a_{\tilde{y}}$, and $a_{\tilde{z}}$, and angular rates, p, q , and r . The IMU data are measured in the aircraft body frame, which is distinguished from the local Cartesian frame using (\sim) . The rotation between the aircraft body frame and the local Cartesian frame is determined from the aircraft attitude angles. When this rotation is applied to the acceleration, a coupled relationship between the velocity and attitude states is defined. This coupling is critical because correction of the velocity states allows for implicit regulation of the attitude states. This relationship defines the velocity state equations:

time data from the IMU, the state equations are discretized using a first-order approximation [40]:

$$x_k = f(x_{k-1}, u_k) = x_{k-1} + T_s f_c(x_{k-1}, u_k), \quad (22)$$

where T_s is the sampling time of the IMU, k is the discrete time index, f is the discrete state function, f_c is the continuous state function defined using (19)–(21), and u is the input vector given by the IMU measurements: $u = [a_{\tilde{x}} \ a_{\tilde{y}} \ a_{\tilde{z}} \ p \ q \ r]^T$. The source of process noise for this formulation is this input vector consisting of IMU measurements. Since information is available about the noise characteristics of the IMU, the process noise covariance matrix is approximated based on precalculated values of the variance of each component of the IMU measured in a static setting:

$$Q = \text{diag}[\sigma^2(a_{\tilde{x}}) \ \sigma^2(a_{\tilde{y}}) \ \sigma^2(a_{\tilde{z}}) \ \sigma^2(p) \ \sigma^2(q) \ \sigma^2(r)]. \quad (23)$$



FIGURE 1: WVU YF-22 research UAVs.

Measurements from GPS are used to calculate the position and velocity of the receiver in the Earth-Centered-Earth-Fixed (ECEF) coordinate frame. These calculations are done within the GPS receiver. For navigation purposes, the position and velocity are transformed into a local Coordinate frame with origin at the WVU flight testing facility. Therefore, the GPS is used to obtain calculations of local position, r_x , r_y , and r_z , and local velocity, V_x , V_y , and V_z . These values are represented as the vector: $z =$

$$R = \text{diag}[\sigma^2(r_x^{\text{GPS}}), \sigma^2(r_y^{\text{GPS}}), \sigma^2(r_z^{\text{GPS}}), \sigma^2(V_x^{\text{GPS}}), \sigma^2(V_y^{\text{GPS}}), \sigma^2(V_z^{\text{GPS}})]. \quad (25)$$

Note that this assumption does not model any direct correlation between the uncertainty in GPS position and velocity, which is reasonable since the measurements are obtained independently with the studied GPS receiver.

3.2. Software Implementation. The results of this study were derived using an off-line postprocessing analysis of previously recorded flight data. The attitude estimation algorithm using UKF was coded manually in-house using MATLAB. Different versions of the attitude estimation code were created for each matrix square root calculation, keeping all other portions of the code constant. All postprocessing was done on the same HP Pavilion Elite HPE Series desktop PC with Intel CORE i5 processor, which has processor speed of 2.67 GHz and 6 GB RAM.

3.3. Research Platform. This sensor fusion study was conducted using flight data collected on the three WVU YF-22 research UAVs (Green, Red, and Blue), shown in Figure 1. The WVU YF-22 platforms have been utilized for various projects, including the validation of GPS-based formation flight control laws [41–43] and fault-tolerant flight control laws [44].

Flight data were collected with two different avionics system architectures. Avionics system number 1 [41] consists of a Crossbow VG400 series IMU and a NovAtel OEM4 GPS receiver, which reports a 1.8-meter Circular Error Probable (CEP) for position measurements. One version of avionics system number 1 was used in each of the three UAVs.

$[r_x^{\text{GPS}} \ r_y^{\text{GPS}} \ r_z^{\text{GPS}} \ V_x^{\text{GPS}} \ V_y^{\text{GPS}} \ V_z^{\text{GPS}}]^T$. Since the GPS does not provide any information directly related to the aircraft attitude, the outputs of the system are defined as the position and velocity states of the system, leading to the following output equations:

$$y = h(x, u) = Hx = [r_x \ r_y \ r_z \ V_x \ V_y \ V_z]^T, \quad (24)$$

where h is the observation function, which for this formulation due to linearity can be written using the observation matrix, $H = [I_{6 \times 6} \ 0_{6 \times 3}]$, where I is an identity matrix, and 0 is a matrix of zeros with given dimensions. Due to the simplicity of the observation equations, the measurement noise comes directly from the GPS position and velocity calculations. The GPS noise is assumed to be zero mean Gaussian and mutually independent, which leads to a diagonal measurement noise covariance matrix. Similar to the process noise covariance matrix, this matrix is assumed constant and approximated based on precalculated values of the variance of each component of the GPS:

Avionics system number 2 [45] is a more recently developed system and was used in the retrofitted Blue YF-22. The GPS used in system 2 was also a NovAtel OEM4, while the IMU was an Analog Devices ADIS-16405. In total, data collected with four different IMUs, that is, ADIS-16405, Crossbow VG400CA-200, DMU(VG400)-100, and IMU400CC-200, were used for this effort. The specifications for the different IMUs are shown in Table 2.

Independent roll and pitch measurements for both systems were obtained with a Goodrich VG34 mechanical vertical gyroscope. The VG34 has a $\pm 90^\circ$ sensitivity on the roll axis and $\pm 60^\circ$ sensitivity on the pitch axis and is sampled with 16-bit resolution. The VG34 has a self-erection system and reported accuracy of within 0.25° of true vertical. The output of the mechanical vertical gyroscope is used as the “truth data” for the sensor fusion study.

3.4. Flight Data Selection. To evaluate the sensitivity of the UKF, 23 sets of flight data were selected from a large library of flight data collected at WVU. Data were selected from flights on the WVU YF-22 platform, which consists of three UAVs. Each data set consists of three-axis IMU measurements, GPS position, and velocity calculations in local geodetic coordinates, along with roll and pitch angle measurements from a mechanical vertical gyroscope. Flights were selected with the goal of obtaining a diverse set of data in order to fairly compare sensitivity effects. The distribution of the flight selection is summarized in Tables 3 and 4.

3.5. Performance Evaluation. The performance of the UKF sensor fusion algorithm is evaluated from the estimated roll

TABLE 2: Inertial measurement unit (IMU) specifications.

Inertial measurement unit (IMU)	Crossbow VG400CA-200/VG400CC-200	Crossbow DMU(VG400)-100	Analog devices ADIS-16405
Resolution	16-bit	16-bit	14-bit
Rate gyroscope sensitivity	$\pm 200^\circ/\text{s}$	$\pm 100^\circ/\text{s}$	$\pm 150^\circ/\text{s}$
Accelerometer sensitivity	$\pm 10 \text{ g}$	$\pm 8 \text{ g}$	$\pm 18 \text{ g}$
Rate gyroscope random walk	$1.7^\circ/\text{h}^{1/2}$	$0.85^\circ/\text{h}^{1/2}$	$2.0^\circ/\text{h}^{1/2}$
Accelerometer random walk	$0.5 \text{ m/s/h}^{1/2}$	$0.1 \text{ m/s/h}^{1/2}$	$0.2 \text{ m/s/h}^{1/2}$

TABLE 3: Summary of the distribution of flight systems.

System type	Distribution of flights		
UAV	(8) Green YF-22	(5) Red YF-22	(10) Blue YF-22
Avionics system	(21) Avionics system number 1	(2) Avionics system number 2	
Control system	(12) Pilot only	(11) Pilot with sections of autonomous flight	

TABLE 4: Summary of the distribution of flight conditions.

Approximate flight condition	Minimum	Maximum	Mean	Standard deviation
Temperature ($^\circ\text{F}$)	38	86	65.1	13.9
Wind speed (mph)	0	11	5.8	3.5
Flight duration (sec)	375	640	485	76.0

and pitch angles. These estimates are compared with independent measurements obtained from a high-quality mechanical gyroscope. The gyroscope data are considered to be a “truth” value, and therefore the difference between the sensor fusion estimate and the corresponding gyroscope measurement represents the error. To quantify this error, the mean of the absolute value of the error and the standard deviation of the error are considered for both roll and pitch. To simplify performance comparisons, a scalar composite performance cost, J , is defined as a weighted average of these values such that

$$J = 0.3[\sigma(\phi_{\text{error}}) + \sigma(\theta_{\text{error}})] + 0.2[\text{mean}(|\phi_{\text{error}}|) + \text{mean}(|\theta_{\text{error}}|)], \quad (26)$$

where ϕ_{error} and θ_{error} represent the error of the roll and pitch estimates with respect to the corresponding vertical gyroscope measurement. The weights for this performance metric were selected such that their sum is unity, equal importance is given to the roll and pitch errors, and less importance is placed on the mean errors because of potential alignment errors between the IMU and vertical gyroscope. Since the weights add up to one and the units for each value are in degrees, the performance cost, J , can be considered as a generalized error value in degrees.

4. Results

4.1. Sensitivity to UKF Matrix Square Root Calculation. To analyze the sensitivity of this formulation of GPS/INS sensor fusion to the matrix square root operation, the UKF algorithm was executed for each set of flight data using different methods of calculating the matrix square root. In parti-

cular, the diagonalization method, Schur method, Cholesky method, and five different iterative methods were implemented. For each of the iterative methods, the UKF was executed for each set of flight data using a set number of iterations throughout the entire flight. This process was repeated for the number of iterations ranging from 5 to 20 by unit increments. For each individual case, results were evaluated based on performance cost, J , total execution time of the UKF, and the accuracy of the matrix square root calculation. A mean was taken of each of these values over all of the 23 flights in order to establish a generalized result. All of the matrix square root operations executed without error, except for certain cases of Newton’s iteration which incurred matrix square root divergence errors on some flights when the number of iterations exceeded 16. In order to fairly compare the results, these cases of Newton’s iteration are omitted from the data set. The performance cost of the different algorithms is shown in Figure 2.

In Figure 2, the performance curves were combined for some of the algorithms for clarity since those methods yielded nearly identical performance results. The performance of the noniterative methods is also included in Figure 2 to compare with the iterative methods. The plot on the right in Figure 2 shows a zoomed-in section to show the convergence of the algorithms. In terms of performance, the Cyclic Reduction (CR) algorithm showed the fastest convergence with respect to number of iterations out of the five considered iterative methods. Another interesting observation from these performance curves is the number of iterations required for each of the iterative algorithms to converge to the same performance as the noniterative methods. For this application, the CR iteration requires 12 iterations and the other four iterative methods require 13 iterations to

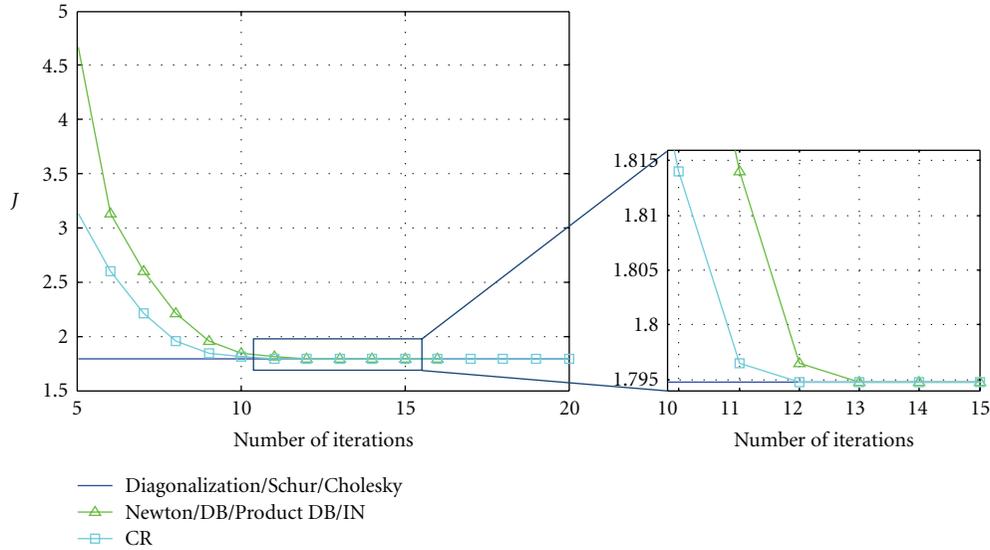


FIGURE 2: Performance cost of UKF for different matrix square root operations.

achieve similar performance as the noniterative methods to four significant figures. However, a smaller number of iterations can be used to reduce computations and still achieve reasonable performance. The CR method with 8 iterations could be used, for example, if performance cost of 2 degrees is acceptable for the application.

Since computational requirements are also important for a matrix square root algorithm, the actual execution time of the entire UKF algorithm was calculated for each of the different methods. These execution times are intended to provide an approximate empirical verification of the theoretical FLOP estimates in order to compare the computational requirements of the different algorithms. All execution time analyses were conducted using the same computer under approximately the same operating conditions. A mean of these execution times over each of the 23 flights is shown in Figure 3(a). Also in Figure 3(b) the corresponding estimated number of FLOPs using Table 1 is shown.

It is important to note that the number of FLOPs estimate is for the matrix square root operation only, while the execution times represent the run times of the entire sensor fusion algorithm. However, this is representative of the overall trend since the only difference between the curves is the matrix square root operation used. Because the diagonalization, Schur, and Cholesky methods do not require iterations, these algorithms are represented in Figure 3 by horizontal lines. Also, in Figure 3(b), the Denman-Beavers (DB) and Product DB are represented by a single line, since the estimated FLOPs for these algorithms are the same as listed in Table 1. The CR and IN iterations are similarly combined in Figure 3(b). Similar trends are observed between the estimated number of FLOPs and the UKF execution time with a few observations. First, the Denman-Beavers (DB) method demonstrates a longer execution time that grows at a steeper rate with the number of iterations than the product DB method, even though the FLOP estimations were the same. Another

difference between the FLOP estimates and the UKF execution time is the location of the noniterative methods with respect to the iterative methods. With respect to the empirical execution time results, the iterative methods are all found to be more efficient in execution time than the Schur method for cases up to 15 iterations, with the exception of the DB iteration. This is an important result because previously, in Figure 2, it was shown that the iterative methods all achieve performance accuracy to four significant figures by using at most 13 iterations. This indicates the potential value of using iterative methods over the Schur method. It is also shown in Figure 3 that the Cholesky method has the fastest execution time with respect to any of the tested cases.

In order to compare accuracy of the actual matrix square root calculation itself, the L_1 norm [46] of the matrix $|P^{1/2} \cdot P^{1/2} - P|_1$ was calculated as a measure of the accuracy of the matrix square root operation. This norm, which is equal to the maximum of the absolute column sums of the matrix, was calculated at each time step of the UKF algorithm. To analyze the overall accuracy of the matrix square root operation over an entire flight, only the maximum of this norm over all discrete time was considered for each flight. This maximum represents the worst matrix square root estimate that occurred over the entire flight. A mean was taken of each of these maximum norms over all 23 flights, and the results are shown in Figure 4.

Figure 4 shows the convergence of the iterative methods in terms of the matrix square root accuracy. For smaller numbers of iterations, all of the iterative methods except for CR are very close in accuracy. These curves start to separate only at higher numbers of iterations, as shown in the right side plot of Figure 4. All of these algorithms converge to very high matrix square root accuracy after a sufficient number of iterations, with the exception of Newton's iteration. Figure 4 demonstrates the divergence of Newton's iteration. After 13 iterations, the matrix square root accuracy starts to degrade

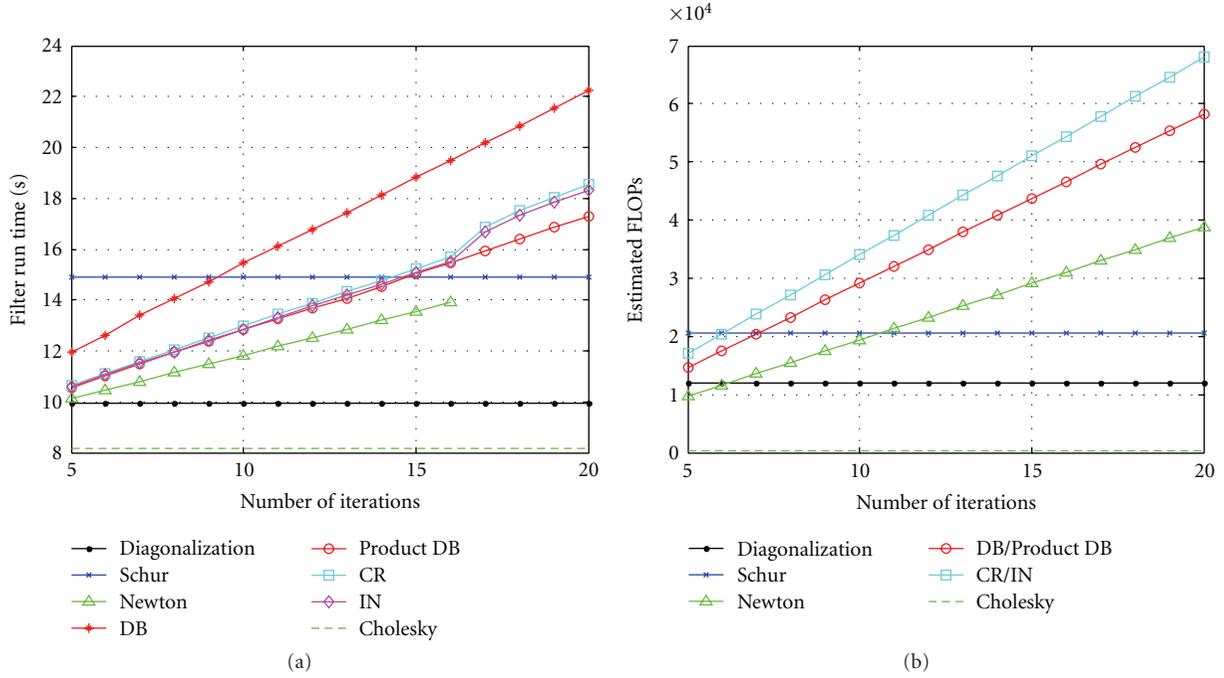


FIGURE 3: UKF computational requirements for different matrix square root operations.

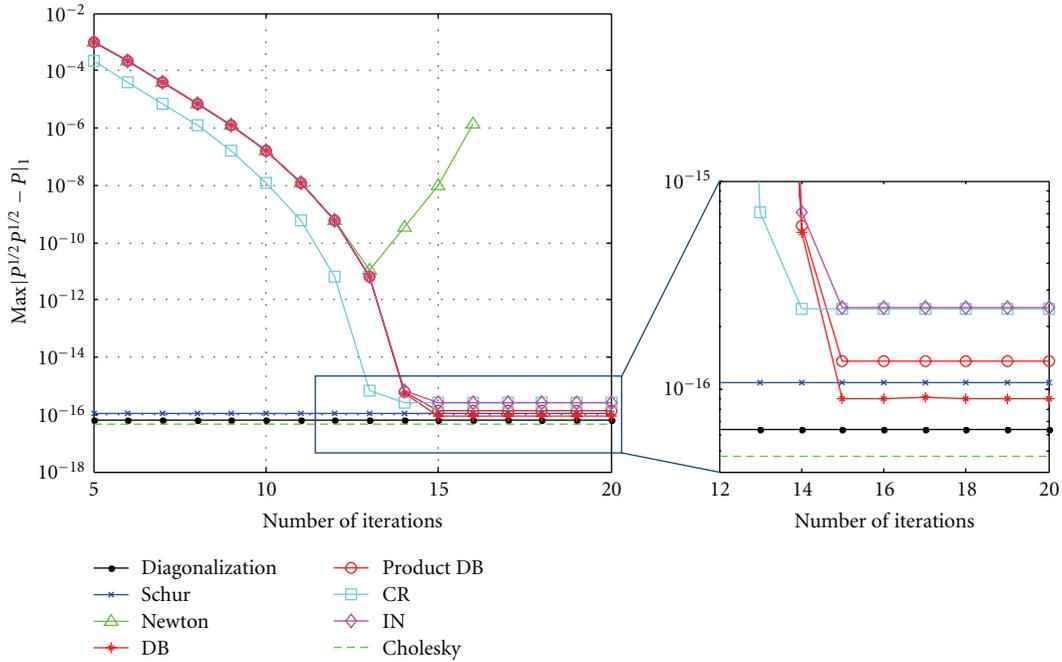


FIGURE 4: Matrix square root operation accuracy.

and eventually reaches a point where the matrix square root accuracy is too poor to use within the UKF algorithm. Because of the divergence issues associated with Newton’s iteration, it is not recommended for UKF applications, even though it is the most computationally efficient iterative matrix square root method with respect to both FLOP estimate and execution time. Because the accuracy of the matrix

square root operation has a direct effect on the accuracy of the prediction stage of the UKF, the relationship between the matrix square root accuracy and UKF performance accuracy for all considered matrix square root operations is shown in Figure 5.

As shown in Figure 5, there is a clear nonlinear relationship between the matrix square root accuracy and the UKF

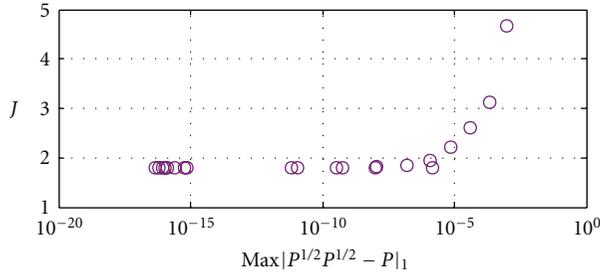


FIGURE 5: Relationship between matrix square root accuracy and UKF performance.

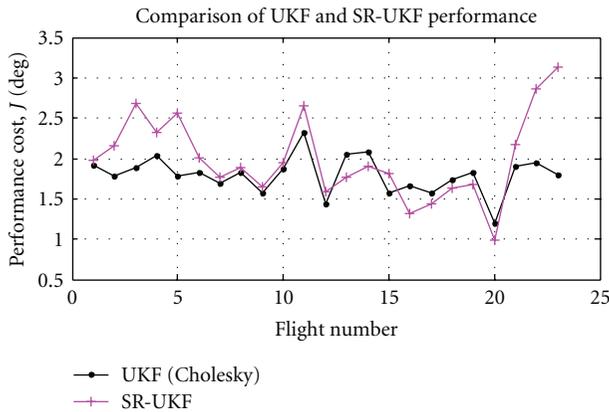


FIGURE 6: Comparison of UKF and SR-UKF performance.

performance for this application. This demonstrates the significant effect of the matrix square root accuracy on the performance of the UKF.

4.2. Comparison of Direct Matrix Square Root Methods to SR-UKF. A different method of handling the square root requirement of the UKF, named the “square-root UKF (SR-UKF)”, was suggested by van der Merwe and Wan [19]. In this method, the square root of the state covariance matrix is estimated directly. This eliminates the need to refactorize the state covariance matrix at each time step, and instead it is updated using Cholesky updates. A significant advantage of this method is a decrease in computational complexity, which leads to a faster run time of the UKF. For comparison purposes, the Cholesky method was selected as a representative case of calculating the matrix square root at each time step from the state covariance estimate. Performance results were calculated using each of these two methods for each of the 23 flights, and the performance cost is plotted in Figure 6.

The mean performance costs of the two different methods are shown in Table 5. To determine if there is a statistically significant performance advantage of the UKF over the SR-UKF, a one-tailed paired samples hypothesis test [46] was done using the t -statistic to determine the probability that the SR-UKF has better performance than the UKF. Using this null hypothesis, the probability was calculated to be 1.49%, which is less than the commonly considered 5% null

TABLE 5: Comparison of UKF and SR-UKF.

Matrix square root method	Mean performance cost J (deg)	Mean execution time (sec)
UKF (Cholesky)	1.795	8.16
SR-UKF	1.996	5.73

hypothesis rejection criterion. Therefore, the UKF achieves statistically significant better performance than the SR-UKF for this application, at the cost of additional computational complexity. To compare the computational complexity of the two different algorithms, the mean execution time of the algorithms was calculated and is also shown in Table 5.

It is also interesting to note from Figure 6 that there are some cases where the SR-UKF method has better performance than the Cholesky method, for example, flight number 16. If this single flight alone was used to analyze results, the opposite conclusion could be drawn about the accuracy of this method. This demonstrates the value of using multiple data sets.

5. Conclusions

This paper presented a comparison of different matrix square root calculations within the UKF. The GPS/INS sensor fusion attitude estimation problem for UAV applications was used as an example to evaluate the performance with respect to matrix square root accuracy, computational cost, and attitude estimation performance. In terms of attitude estimation performance, the Cholesky, diagonalization, and Schur methods yielded the highest accuracy; however this same performance can be reached using a sufficient number of iterations in any of the iterative methods. Newton’s iteration was found to diverge in certain instances and is therefore not recommended for UKF applications. The cyclic reduction (CR) iteration demonstrated the fastest performance convergence of the iterative methods. In terms of execution time, the SR-UKF is computationally efficient, but at the cost of performance. Overall, the Cholesky method was found to provide the best compromise in terms of both performance and execution time.

For real-time applications of the UKF, such as attitude estimation for small UAVs, computation is an important consideration factor. For most cases, the Cholesky method is the best suited matrix square root method due to its fast execution and high accuracy. If computational cost is even more important than the accuracy of the filter, the SR-UKF could be considered. The diagonalization and Schur methods are acceptable approaches for off-line applications, because the accuracy is similar to the Cholesky method, although they require more computation time. These methods also might be more desirable than the Cholesky method because they provide a more intuitive representation of the matrix square root, that is, the principle square root. Any of the iterative methods, except for Newton’s iteration, could also be used with a sufficient number of iterations, though these methods are a bit less intuitive.

Acknowledgments

This research was partially supported by NASA Grant no. NNX07AT53A and Grant no. NNX10AI14G.

References

- [1] L. Changchun, S. Li, W. Hai-bo, and L. Tianjie, "The research on unmanned aerial vehicle remote sensing and its applications," in *Proceedings of the IEEE International Conference on Advanced Computer Control (ICACC '10)*, pp. 644–647, Shenyang, China, March 2010.
- [2] A. M. Jensen, M. Baumann, and Y. Chen, "Low-cost multi-spectral aerial imaging using autonomous runway-free small flying wing vehicles," in *Proceedings of the IEEE International Geoscience and Remote Sensing Symposium (IGARSS '08)*, pp. 506–509, Boston, Mass, USA, July 2008.
- [3] M. Nagai, T. Chen, R. Shibasaki, H. Kumagai, and A. Ahmed, "UAV-borne 3-D mapping system by multisensor integration," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 47, no. 3, Article ID 4783021, pp. 701–708, 2009.
- [4] T. Suzuki, Y. Amano, and T. Hashizume, "Vision based localization of a small UAV for generating a large mosaic image," in *Proceedings of the SICE Annual Conference (SICE '10)*, pp. 2960–2964, Taipei, Taiwan, October 2010.
- [5] S. Dascalu, "Remote sensing using autonomous UAVs suitable for less developed countries," *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, Enschede, Netherlands*, vol. 34, no. 30, 2006.
- [6] G. Zhou and D. Zang, "Civil UAV system for earth observation," in *Proceedings of the IEEE International Geoscience and Remote Sensing Symposium (IGARSS '07)*, Barcelona, Spain, 2007.
- [7] J. Everaerts, "The use of unmanned aerial vehicles (UAVs) for remote sensing and mapping," *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, Beijing*, vol. 36, no. B1, pp. 1187–1192, 2008.
- [8] M. S. Grewal, L. R. Weill, and A. P. Andrew, *Global Positioning, Inertial Navigation & Integration*, John Wiley & Sons, New York, NY, USA, 2nd edition, 2007.
- [9] E. Kaplan and C. Heagarty, *Understanding GPS Principles and Applications*, Arttech House, Norwood, Mass, USA, 2nd edition, 2006.
- [10] J. L. Crassidis, "Sigma-point kalman filtering for integrated GPS and inertial navigation," in *Proceedings of the AIAA Guidance, Navigation and Control Conference and Exhibit*, San Francisco, Calif, USA, 2005.
- [11] R. D. Van Merwe, E. A. Wan, and S. I. Julier, "Sigma-point kalman filters for nonlinear estimation and sensor-fusion—applications to integrated navigation," in *Proceedings of the AIAA Guidance, Navigation, and Control Conference*, pp. 1735–1764, Providence, RI, USA, August 2004.
- [12] T. Fiorenzani et al., "Comparative Study of Unscented Kalman Filter and Extended Kalman Filter for Position/Attitude Estimation in Unmanned Aerial Vehicles," IASI-CNR, R. 08-08, 2008.
- [13] J. Gross, Y. Gu, M. Rhudy, S. Gururajan, and M. Napolitano, "Flight test evaluation of GPS/INS sensorfusion algorithms for attitude estimation," *IEEE Transactions on Aerospace Electronic Systems*. In press.
- [14] S. J. Julier and J. K. Uhlmann, "New extension of the Kalman filter to nonlinear systems," in *Signal Processing, Sensor Fusion, and Target Recognition VI*, vol. 3068 of *SPIE Proceedings Series*, pp. 182–193, July 1997.
- [15] J. Gross et al., "A comparison of extended kalman filter, sigma-point kalman filter, and particle filter in GPS/INS sensor fusion," in *Proceedings of the AIAA Guidance, Navigation, and Control Conference*, Toronto, Canada, 2010.
- [16] R. E. Kalman and R. S. Bucy, "New results in linear filtering and prediction theory," *Journal of Basic Engineering*, vol. 83, pp. 95–108, 1961.
- [17] N. El-Sheimy, E. H. Shin, and X. Niu, "Kalman filter face-off: extended vs. Unscented kalman filters for integrated GPS and MEMS inertial," in *Proceedings of the International Symposium on Global Navigation Satellite Systems (GNSS '06)*, pp. 48–54, 2006.
- [18] M. Rhudy, Y. Gu, J. Gross, and M. Napolitano, "Sensitivity analysis of EKF and UKF in GPS/INS sensor fusion," in *Proceedings of the AIAA Guidance, Navigation, and Control Conference*, Portland, Ore, USA, 2011.
- [19] R. van der Merwe and E. A. Wan, "The square-root unscented Kalman filter for state and parameter-estimation," in *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing*, pp. 3461–3464, Salt Lake City, Utah, USA, May 2001.
- [20] N. J. Higham, *Functions of Matrices: Theory and Computation*, Society for Industrial and Applied Mathematics (SIAM), Philadelphia, Pa, USA, 2008.
- [21] J. Wendel, J. Metzger, R. Moenikes, A. Maier, and G. F. Trommer, "A Performance comparison of tightly coupled GPS/INS navigation systems based on extended and sigma point kalman filters," *Journal of the Institute of Navigation*, vol. 53, no. 1, pp. 21–31, 2006.
- [22] N. B. Stastny, R. A. Bettinger, and F. R. Chavez, "Comparison of the extended and unscented Kalman filters for angles based relative navigation," in *Proceedings of the AIAA/AAS Astrodynamics Specialist Conference and Exhibit*, Honolulu, Hawaii, USA, August 2008.
- [23] Y. Wu, D. Hu, M. Wu, and X. Hu, "Unscented Kalman filtering for additive noise case: augmented vs. non-augmented," in *Proceedings of the American Control Conference (ACC '05)*, pp. 4051–4055, Portland, Ore, USA, June 2005.
- [24] F. Sun, G. Li, and J. Wang, "Unscented kalman filter using augmented state in the presence of additive noise," in *Proceedings of the IITA International Conference on Control, Automation and Systems Engineering (CASE '09)*, pp. 379–382, Zhangjiajie, China, July 2009.
- [25] M. C. VanDyke, J. L. Schwartz, and C. D. Hall, "Unscented kalman filtering for spacecraft attitude state and parameter estimation," in *Proceedings of the AIAA Space Flight Mechanics Meeting*, Maui, Hawaii, USA, 2004.
- [26] S. A. Banani and M. A. Masnadi-Shirazi, "A new version of unscented kalman filter," *World Academy of Science, Engineering, and Technology*, vol. 20, pp. 192–197, 2007.
- [27] B. Meini, "The matrix square root from a new functional perspective: theoretical results and computational issues," *SIAM Journal on Matrix Analysis and Applications*, vol. 26, no. 2, pp. 362–376, 2004.
- [28] J. Gross, Y. Gu, and M. R. Napolitano, "A systematic approach for extended kalman filter tuning and low-cost inertial sensor calibration within a GPS/INS application," in *Proceedings of the AIAA Guidance, Navigation, and Control Conference*, Toronto, Canada, 2010.
- [29] J. Jarrell, Y. Gu, B. Seanor, and M. Napolitano, "Aircraft attitude, position, and velocity determination using sensor fusion," in *Proceedings of the AIAA Guidance, Navigation and*

- Control Conference and Exhibit*, Honolulu, Hawaii, USA, August 2008.
- [30] D. Simon, *Optimal State Estimation*, Wiley, New York, NY, USA, 2006.
- [31] D. A. McQuarrie, *Mathematical Methods for Scientists and Engineers*, chapter 10, University Science Books, Sausalito, Calif, USA, 1st edition, 2003.
- [32] M. M. Konstantinov, *Perturbation Theory for Matrix Equations*, Elsevier Science B.V., Amsterdam, The Netherlands, 2003.
- [33] N. J. Higham, "Newton's method for the matrix square root," *Mathematics of Computation*, vol. 46, no. 174, pp. 537–549, 1986.
- [34] S. H. Cheng, N. J. Higham, C. S. Kenney, and A. J. Laub, "Approximating the logarithm of a matrix to specified accuracy," *SIAM Journal on Matrix Analysis and Applications*, vol. 22, no. 4, pp. 1112–1125, 2001.
- [35] L. N. Trefethen and D. Bau, *Numerical Linear Algebra*, SIAM, Philadelphia, Pa, USA, 1st edition, 1997.
- [36] N. J. Higham, "Analysis of the cholesky decomposition of a semi-definite matrix," in *Reliable Numerical Computation*, pp. 161–185, Oxford University Press, Oxford, UK, 1990.
- [37] N. J. Higham, *Accuracy and Stability of Numerical Algorithms*, Society for Industrial and Applied Mathematics (SIAM), Philadelphia, Pa, USA, 2nd edition, 2002.
- [38] D. W. Tufts and C. D. Melissinos, "Simple, effective computation of principal eigenvectors and their eigenvalues and application to high-resolution estimation of frequencies," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 34, no. 5, pp. 1046–1053, 1986.
- [39] J. Roskam, *Airplane Flight Dynamics and Automatic Flight Controls*, chapter 1, DARcorporation, Lawrence, Kan, USA, 2003.
- [40] F. L. Lewis and V. L. Syrmos, *Optimal Control*, chapter 6, Wiley & Sons, New York, NY, USA, 2nd edition, 1995.
- [41] Y. Gu, B. Seanor, G. Campa, M. R. Napolitano, S. Gururajan, and L. Rowe, "Autonomous formation flight: Hardware development," in *Proceedings of the 14th Mediterranean Conference on Control and Automation (MED '06)*, Ancona, Italy, June 2006.
- [42] B. Seanor, Y. Gu, M. R. Napolitano, G. Campa, S. Gururajan, and L. Rowe, "3-Aircraft formation flight experiment," in *Proceedings of the 14th Mediterranean Conference on Control and Automation (MED '06)*, Ancona, Italy, June 2006.
- [43] Y. Gu, B. Seanor, G. Campa et al., "Design and flight testing evaluation of formation control laws," *IEEE Transactions on Control Systems Technology*, vol. 14, no. 6, pp. 1105–1112, 2006.
- [44] M. G. Perhinschi, J. Burken, and G. Campa, "Comparison of different neural augmentations for the fault tolerant control laws of the WVU YF-22 model aircraft," in *Proceedings of the 14th Mediterranean Conference on Control and Automation (MED '06)*, Ancona, Italy, June 2006.
- [45] J. Gross, Y. Gu, B. Seanor, S. Gururajan, and M. R. Napolitano, "Advanced Research Integrated Avionic (ARIA) system for fault-tolerant flight research," in *Proceedings of the AIAA Guidance, Navigation, and Control Conference and Exhibit*, Chicago, Ill, USA, August 2009.
- [46] E. Kreyszig, *Advanced Engineering Mathematics*, chapter 20, Wiley & Sons, New York, NY, USA, 9th edition, 2006.



Hindawi

Submit your manuscripts at
<http://www.hindawi.com>

