

# Quaternion-based Kalman Filtering on INS/GPS

Yuhong Yang, Junchuan Zhou and Otmar Loffeld  
Center for Sensorsystems (ZESS), University of Siegen  
Siegen, Germany

**Abstract**— In this paper, two quaternion-based nonlinear filtering methods are applied on the processing of measurements from the low-cost Micro-electromechanical Systems (MEMS) based Inertial Navigation system (INS) and Global Positioning System (GPS). One approach employs an Extended Kalman filter (EKF) propagating the quaternion vector using conventional vector addition operation. However, due to the fact that the unit sphere defined by the quaternion vector is not an Euclidean vector space, the vector addition and scaling should principally not be directly applied. Therefore, in the second approach, an Unscented Kalman filter (UKF) is used which propagates the quaternion vector based on the quaternion product chain rule, having a natural way of maintaining the normalization constraint. A field experiment based on the train ride is made for the comparison. The objective is to verify whether different handlings of nonlinearity in system models and different ways of propagating quaternion vector over time will practically yield differences in the estimation of attitude and sensor bias errors.

**Keywords** - *Quaternion-based filtering; low-cost; INS/GPS integration, nonlinear filtering*

## I. INTRODUCTION

The cheap MEMS-based inertial sensors do not exhibit highly accurate navigation performance. However, they can meet the requirements of many land-based navigation applications when aided with GNSS devices. Nowadays, INS and GPS integrated solutions are the back-bone of many modern navigation systems, which are employed in industrial and military applications. Substantial research effort has been devoted to extensive algorithmic developments and performance analysis. The objective is mainly the promotion of system estimation accuracy with low-cost sensor systems, putting the focus on powerful sensor fusion algorithms.

In this paper, the tightly-coupled integration of INS and GPS is addressed, which is one of the promising approaches to fuse measurements of both sensors. However, when modeling the underlying problem, the system propagation and observation models are nonlinear. The most common application of the Kalman filter (KF) on nonlinear systems is the EKF [1-3], which is based on a first-order linearization of the nonlinear stochastic system models. Although the EKF maintains the elegant and computationally efficient update form of the KF, it suffers from a number of drawbacks. For instance, using the first order approximations could result in filter instabilities in the presence of higher order effects [4, 5], and the derivation of the Jacobian matrices itself can be a complicated mathematical task. The UKF algorithm is reported that it can accurately capture the mean and covariance estimates up to higher order terms of a Taylor series expansion

for any nonlinearity [6-8], leading to a better convergence performance from inaccurate initial conditions in estimation problems [5]. In this contribution, the EKF and UKF algorithms are applied on the INS/GPS integration using quaternion as the representation of attitude. Different ways of propagating quaternion vector over time are employed. In the EKF, the quaternion vector is treated as a usual vector on which the vector addition operation is directly applied. However, due to the fact that the unit sphere defined by the quaternion vector is not an Euclidean vector space [9, 10], the vector addition and scaling should principally not be applied on quaternion vectors for representing rotations. In addition, the quaternion unity magnitude constraint allows only three degrees of freedom instead of four. Thus, in the UKF algorithm, the quaternion vector is transformed to the rotational space to preserve the nonlinear nature of the unit vector [11-13]. In this way, the quaternion vector is updated using the quaternion product chain rule, having a natural way of maintaining the normalization constraint. That means, the successive rotation can be accomplished using quaternion multiplication in the same order as the direction cosine matrix multiplication [14]. In this paper, we use data collected from land-based field experiment to verify whether or not these two approaches will yield differences in the estimation of attitude and inertial sensor errors (i.e., gyro bias errors).

In the remainder of the paper, in Section 2, the quaternion-based EKF and UKF algorithms are introduced. In Section 3, field experiment is conducted based on a train ride. The data is processed with correct initial attitude information and with 30 degrees initial attitude errors in each axis for verifying the robustness of the proposed methods. Numerical results are compared and analyzed.

## II. QUATERNION BASED LOW-COST INS/GPS INTEGRATION ALGORITHMS

The navigation performance of a low-cost MEMS-based inertial measurement unit (IMU) deteriorates over time due to the accumulation of combined sensor errors, such as noise, biases, etc. The accuracy achievable by inertial navigation depends on the quality of sensors in use and the modeling and compensation of sensor errors during run-time. Among other errors, the inaccuracies in attitude determination are the main contributors to position and velocity errors in inertial navigation systems. Up to today, this is the most researched topic related to inertial navigation [5]. For the integration of measurements coming from a single GPS receiver antenna and an INS without redundant attitude information (e.g., from magnetometers, or multi-antenna GPS systems), the INS

attitude and inertial sensor error observability is weak and depends on the dynamics of the platform during application [15-17]. It is the nonlinear system propagation model that relates the attitude and inertial sensor errors to the position and velocity errors. Therefore, the nonlinearity in the system state space models should be treated with care. As a conventional nonlinear filtering approach, in the EKF algorithm, the inertial error propagation equations are often used as the system propagation model, where the navigation errors and inertial sensor errors are chosen as states (i.e., error states) [2, 3, 9]. In an UKF algorithm, the navigation solutions (i.e., position, velocity and attitude) and inertial sensor errors are chosen as states (i.e., total states). The nonlinear system propagation and observation models are directly employed in the filter. In this way, the derivation of the complicated Jacobian matrices is prevented. In the following sections, the quaternion-based EKF and UKF algorithms are given. They differ not only in handling nonlinearities, but also in the way of propagating the quaternion vector over time.

#### A. Quaternion based EKF Algorithm

Among many definitions of a quaternion vector, in this paper, the quaternion vector is denoted as  $\mathbf{q} = [q_1 \bar{\mathbf{q}}^T]^T$ , where  $\bar{\mathbf{q}} = [q_2 \ q_3 \ q_4]^T$ . It is used to represent the rotation from the navigation frame to body frame.

The attitude differential equation in terms of the quaternion vector  $\mathbf{q}$  is given in (1). For a detailed derivation, the reader is referred to [9, 18].

$$\dot{\mathbf{q}} = \frac{1}{2} \mathbf{Q}_{\omega_{bn}^b} \mathbf{q}, \text{ with } \mathbf{Q}_{\omega_{bn}^b} = \begin{bmatrix} 0 & | & -[\omega_{bn}^b]^T \\ \hline \omega_{bn}^b & | & [\omega_{bn}^b \times] \end{bmatrix} \quad (1)$$

In (1), the angular rate measurement vector  $\omega_{bn}^b$  from the navigation frame to the body frame, expressed in the body frame, is equal to:

$$\omega_{bn}^b = \omega_{in}^b - \omega_{ib}^b \quad (2)$$

where  $\omega_{ib}^b$  is the rotational rate vector of the body frame relative to the inertial frame, expressed in the body frame (i.e., IMU gyroscope raw measurements);  $\omega_{in}^b$  represents the sum of the rotation of the earth with respect to the inertial frame plus the turn rate of the navigation frame with respect to the earth, expressed in the body frame, i.e.,  $\omega_{in}^b = \mathbf{R}_n^b (\omega_{ie}^n + \omega_{en}^n)$ .

Using a low-cost MEMS-based IMU, the earth rotation is buried in sensor errors, and cannot be detected by the sensor. Thus, the Coriolis and centrifugal terms are not considered in the following. Moreover, for short distance applications, the transport rate is negligible. Considering these effects, we have  $\omega_{in}^b = \mathbf{0}$ . Therefore, (2) turns to be:

$$\omega_{bn}^b = \mathbf{0} - \omega_{ib}^b \Rightarrow \omega_{bn}^b = -\omega_{ib}^b \quad (3)$$

where  $\omega_{ib}^b = [\omega_{ib,x}^b \ \omega_{ib,y}^b \ \omega_{ib,z}^b]^T$  is the gyroscope raw data, resolved in the body frame.

In this way, (1) can be approximated as:

$$\dot{\mathbf{q}} = \frac{1}{2} \mathbf{Q}_{\omega_{ib}^b} \mathbf{q}, \text{ with} \quad (4)$$

$$\mathbf{Q}_{\omega_{ib}^b} = \begin{bmatrix} 0 & | & -[\omega_{ib}^b]^T \\ \hline \omega_{ib}^b & | & [\omega_{ib}^b \times] \end{bmatrix} \approx \begin{bmatrix} 0 & | & [\omega_{ib}^b]^T \\ \hline -\omega_{ib}^b & | & [-\omega_{ib}^b \times] \end{bmatrix}$$

The simplified mechanization model for the IMU can be expressed in the navigation frame, as shown in (5). More sophisticated models can be found in [3, 19, 20].

$$\begin{aligned} \dot{\mathbf{p}}_n &= \mathbf{v}_n \\ \dot{\mathbf{v}}_n &= \mathbf{R}_n^b(\mathbf{q}) \mathbf{f}_{ib}^b + \mathbf{g}_n \\ \dot{\mathbf{q}} &= \frac{1}{2} \mathbf{Q}_{\omega_{ib}^b} \mathbf{q} \end{aligned} \quad (5)$$

Here  $\mathbf{g}_n$  represents gravity indicated in the navigation frame, which is assumed to be constant. The rotational transformation matrix  $\mathbf{R}_n^b(\mathbf{q})$  from the body frame to the navigation frame is expressed using quaternion as:

$$\mathbf{R}_n^b(\mathbf{q}) = \begin{bmatrix} q_1^2 + q_2^2 - q_3^2 - q_4^2 & 2(q_2q_3 - q_1q_4) & 2(q_1q_3 + q_2q_4) \\ 2(q_2q_3 + q_1q_4) & q_1^2 - q_2^2 + q_3^2 - q_4^2 & 2(q_3q_4 - q_1q_2) \\ 2(q_2q_4 - q_1q_3) & 2(q_1q_2 + q_3q_4) & q_1^2 - q_2^2 - q_3^2 + q_4^2 \end{bmatrix}^T \quad (6)$$

Equation (5) represents the mechanization model for the strapdown IMU, and it can be used as the INS/GPS propagation model for updating position, velocity and attitude over time. In this case, the IMU specific force and angular rate measurement errors (e.g., sensor biases) should be considered. They are modelled as constants superseded by random walk as:

$$\dot{\mathbf{f}}_b^{bias} = \mathbf{w}_f, \quad \dot{\omega}_b^{bias} = \mathbf{w}_\omega \quad (7)$$

where  $\mathbf{w}_f$  and  $\mathbf{w}_\omega$  are assumed to be zero mean, Gaussian distributed white noises.

It is worth mentioning that when using (5) as INS/GPS system propagation model, the IMU incoming measurements should be compensated by the current estimate of the sensor biases before further processing, i.e.,  $\mathbf{f}_{ib}^b = \tilde{\mathbf{f}}_{ib}^b - \mathbf{f}_b^{bias} - \mathbf{n}_f$ ,  $\omega_{ib}^b = \tilde{\omega}_{ib}^b - \omega_b^{bias} - \mathbf{n}_\omega$  ( $\mathbf{n}_f$  and  $\mathbf{n}_\omega$  are assumed to be the Gaussian white noises).

In the discrete-time domain with a sufficiently small time interval (e.g., 100Hz IMU update rate), and for low dynamic applications, we have:

$$\begin{aligned}
\mathbf{p}_{n,k+1} &= \mathbf{p}_{n,k} + \mathbf{v}_{n,k} \cdot T \\
\mathbf{v}_{n,k+1} &= \mathbf{v}_{n,k} + [\mathbf{R}_b^n(\mathbf{q}_k) \mathbf{f}_{ib,k}^b + \mathbf{g}_n] \cdot T \\
\mathbf{q}_{k+1} &= \mathbf{q}_k + \left( \frac{1}{2} \mathbf{Q}_{\omega_{bn,k}} \cdot T \right) \mathbf{q}_k \\
\mathbf{f}_{b,k+1}^{bias} &= \mathbf{f}_{b,k}^{bias} + \mathbf{w}_{f,k} \\
\boldsymbol{\omega}_{b,k+1}^{bias} &= \boldsymbol{\omega}_{b,k}^{bias} + \mathbf{w}_{\omega,k}
\end{aligned} \quad (8)$$

where “T” is system propagation time interval.

As shown in (8), a part of the system propagation model is nonlinear, e.g.,  $\mathbf{R}_b^n(\mathbf{q})$  contains quadratic terms of quaternion elements. Therefore, the linearization process is conducted.

In the scope of a tightly-coupled integration approach, we also need to model the receiver clock errors. The range-rate equivalent of the clock drift error is modelled as a constant plus a random walk process, while the range equivalent of the receiver clock bias error is the integral of the clock drift error. Thus, the linearized system propagation model used for INS/GPS tightly-coupled integration is formulated in (9), where error states are employed.

$$\begin{bmatrix} \delta \mathbf{p}_{n,k+1} \\ \delta \mathbf{v}_{n,k+1} \\ \delta \mathbf{q}_{k+1} \\ \delta \mathbf{f}_{b,k+1}^{bias} \\ \delta \boldsymbol{\omega}_{b,k+1}^{bias} \\ c\delta \tilde{t}_{k+1} \end{bmatrix} = \begin{bmatrix} \mathbf{I}_{3 \times 3} & \mathbf{I}_{3 \times 3} \cdot T & \mathbf{O}_{3 \times 3} & \mathbf{O}_{3 \times 3} & \mathbf{O}_{3 \times 3} & \mathbf{O}_{3 \times 3} & \mathbf{O}_{3 \times 3} \\ \mathbf{O}_{3 \times 3} & \mathbf{I}_{3 \times 3} & \mathbf{F}_{23,k} \cdot T & -\mathbf{R}_b^n(\mathbf{q}_k) \cdot T & \mathbf{O}_{3 \times 3} & \mathbf{O}_{3 \times 3} & \mathbf{O}_{3 \times 3} \\ \mathbf{O}_{4 \times 3} & \mathbf{O}_{4 \times 3} & \mathbf{I}_{4 \times 4} + \frac{1}{2} \mathbf{Q}_{\omega_{bn,k}} \cdot T & \mathbf{O}_{4 \times 3} & \frac{1}{2} \mathbf{Q}_{\omega_{bn,k}} \cdot T & \mathbf{O}_{4 \times 3} & \mathbf{O}_{4 \times 3} \\ \mathbf{O}_{3 \times 3} & \mathbf{O}_{3 \times 3} & \mathbf{O}_{3 \times 3} & \mathbf{I}_{3 \times 3} & \mathbf{O}_{3 \times 3} & \mathbf{O}_{3 \times 3} & \mathbf{O}_{3 \times 3} \\ \mathbf{O}_{3 \times 3} & \mathbf{O}_{3 \times 3} & \mathbf{O}_{3 \times 3} & \mathbf{O}_{3 \times 3} & \mathbf{I}_{3 \times 3} & \mathbf{O}_{3 \times 3} & \mathbf{O}_{3 \times 3} \\ \mathbf{O}_{3 \times 3} & \mathbf{O}_{3 \times 3} & \mathbf{O}_{3 \times 3} & \mathbf{O}_{3 \times 3} & \mathbf{O}_{3 \times 3} & \mathbf{I}_{3 \times 3} & \mathbf{O}_{3 \times 3} \\ \mathbf{O}_{3 \times 3} & \mathbf{O}_{3 \times 3} & \mathbf{O}_{3 \times 3} & \mathbf{O}_{3 \times 3} & \mathbf{O}_{3 \times 3} & \mathbf{O}_{3 \times 3} & \mathbf{I}_{3 \times 3} \end{bmatrix} \begin{bmatrix} \delta \mathbf{p}_{n,k} \\ \delta \mathbf{v}_{n,k} \\ \delta \mathbf{q}_k \\ \delta \mathbf{f}_{b,k}^{bias} \\ \delta \boldsymbol{\omega}_{b,k}^{bias} \\ c\delta \tilde{t}_k \end{bmatrix} + \mathbf{w}_k \quad (9)$$

where

$$\boldsymbol{\Omega}_{q_k} = \begin{bmatrix} -\vec{q}^T \\ q_1 \mathbf{I}_{3 \times 3} - [\vec{q} \times] \end{bmatrix}_k = \begin{bmatrix} -q_2 & -q_3 & -q_4 \\ q_1 & q_4 & -q_3 \\ -q_4 & q_1 & q_2 \\ q_3 & -q_2 & q_1 \end{bmatrix}_k \quad (10)$$

and  $\mathbf{F}_{23,k}$  is computed as:

$$\mathbf{F}_{23,k} = 2 \cdot \begin{bmatrix} q_1 \hat{f}_x + q_4 \hat{f}_y - q_3 \hat{f}_z & q_2 \hat{f}_x + q_3 \hat{f}_y + q_4 \hat{f}_z & -q_3 \hat{f}_x + q_2 \hat{f}_y - q_1 \hat{f}_z & q_1 \hat{f}_y - q_4 \hat{f}_x + q_2 \hat{f}_z \\ q_1 \hat{f}_y - q_4 \hat{f}_x + q_2 \hat{f}_z & q_3 \hat{f}_x - q_2 \hat{f}_y + q_1 \hat{f}_z & q_2 \hat{f}_x + q_3 \hat{f}_y + q_4 \hat{f}_z & -q_1 \hat{f}_z - q_4 \hat{f}_y + q_3 \hat{f}_z \\ q_3 \hat{f}_x - q_2 \hat{f}_y + q_1 \hat{f}_z & -q_1 \hat{f}_y + q_4 \hat{f}_x - q_2 \hat{f}_z & q_1 \hat{f}_x + q_4 \hat{f}_y - q_3 \hat{f}_z & q_2 \hat{f}_x + q_3 \hat{f}_y + q_4 \hat{f}_z \end{bmatrix}_k \quad (11)$$

where  $\hat{f}_x = (\tilde{f}_{ib,x}^b - \hat{f}_{b,x}^{bias})_k$ ,  $\hat{f}_y = (\tilde{f}_{ib,y}^b - \hat{f}_{b,y}^{bias})_k$ ,  $\hat{f}_z = (\tilde{f}_{ib,z}^b - \hat{f}_{b,z}^{bias})_k$  are the IMU raw data compensated by the current estimate of the sensor biases expressed in the body frame.

For the observation model of the tightly-coupled integration, (12) is used. And ‘j’ denotes the number of satellites in view; is the estimated line-of-sight unit vector

pointing from the initial estimate of the user position to the j-th satellite.

$$\mathbf{y}_k = \begin{bmatrix} \hat{\rho}_{1,k} - \tilde{\rho}_{1,k} \\ \vdots \\ \hat{\rho}_{j,k} - \tilde{\rho}_{j,k} \\ \vdots \\ \hat{\rho}_{1,k} - \tilde{\rho}_{1,k} \\ \vdots \\ \hat{\rho}_{j,k} - \tilde{\rho}_{j,k} \end{bmatrix} = \begin{bmatrix} (-\mathbf{I}_{1,k}^T)_{1 \times 3} & \mathbf{0}_{1 \times 3}^T & \mathbf{0}_{1 \times 3}^T & \mathbf{0}_{1 \times 3}^T & \mathbf{0}_{1 \times 3}^T & 1 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ (-\mathbf{I}_{j,k}^T)_{1 \times 3} & \mathbf{0}_{1 \times 3}^T & \mathbf{0}_{1 \times 3}^T & \mathbf{0}_{1 \times 3}^T & \mathbf{0}_{1 \times 3}^T & 1 & 0 \\ \mathbf{0}_{1 \times 3}^T & (-\mathbf{I}_{1,k}^T)_{1 \times 3} & \mathbf{0}_{1 \times 3}^T & \mathbf{0}_{1 \times 3}^T & \mathbf{0}_{1 \times 3}^T & 0 & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \mathbf{0}_{1 \times 3}^T & (-\mathbf{I}_{j,k}^T)_{1 \times 3} & \mathbf{0}_{1 \times 3}^T & \mathbf{0}_{1 \times 3}^T & \mathbf{0}_{1 \times 3}^T & 0 & 1 \end{bmatrix}_{2j \times 18} \begin{bmatrix} \delta \mathbf{p}_{n,k} \\ \delta \mathbf{v}_{n,k} \\ \delta \mathbf{q}_k \\ \delta \mathbf{f}_{b,k}^{bias} \\ \delta \boldsymbol{\omega}_{b,k}^{bias} \\ c\delta \tilde{t}_k \end{bmatrix} + \boldsymbol{\varepsilon}_k \quad (12)$$

As opposed to Euler angle based algorithms, (9) does not involve trigonometric operations which are not only error prone when implementing, but also exhibit the potential of singularities. In addition, this derivation does not contain the small angle assumptions from the perturbation analysis, which improves the system robustness.

In (8), the quaternion vector is updated using vector addition. This may cause additional errors as the usual definitions of vector addition and scaling can normally not be applied directly due to the fact that the unit sphere defined by a quaternion is not an Euclidean vector space [9]. In the UKF algorithm the quaternion product chain rule is applied to update the quaternion vector.

### B. Quaternion-based UKF Algorithm

In this approach, the quaternion vector is transformed to the rotational space to preserve the nonlinear nature of the unit quaternion. This transformation will be carried out throughout the derivation given below. The quaternion vector is updated using the quaternion product chain rule, leading to a natural way of maintaining the normalization constraint. That is, the successive rotations can be accomplished using quaternion multiplication in the same order as the direction cosine matrix multiplication [14]. The system position, velocity and attitude updates in the discrete time domain are formulated in (13).

$$\begin{aligned}
\mathbf{p}_{n,k+1} &= \mathbf{p}_{n,k} + \mathbf{v}_{n,k} \cdot T \\
\mathbf{v}_{n,k+1} &= \mathbf{v}_{n,k} + [\mathbf{R}_b^n(\mathbf{q}_k) \mathbf{f}_{ib,k}^b + \mathbf{g}_n] \cdot T \\
\mathbf{q}_{k+1} &= \Delta \mathbf{q}_k \otimes \mathbf{q}_k
\end{aligned} \quad (13)$$

where  $\otimes$  denotes the quaternion product and

$$\Delta \mathbf{q}_k = \begin{bmatrix} \cos(0.5 \|\boldsymbol{\theta}_k\|) \\ \frac{\sin(0.5 \|\boldsymbol{\theta}_k\|)}{0.5 \|\boldsymbol{\theta}_k\|} \mathbf{0.5} \boldsymbol{\theta}_k \end{bmatrix}; \quad \boldsymbol{\theta}_k \text{ represents the integral of the}$$

IMU body frame angular rate measurements over IMU measurement update interval (e.g., k-1 to k).

The sensor biases, e.g.,  $\mathbf{f}_{b,k}^{bias}$  and  $\boldsymbol{\omega}_{b,k}^{bias}$ , are modelled as constants superseded by random walk as:

$$\begin{aligned}
\mathbf{f}_{b,k+1}^{bias} &= \mathbf{f}_{b,k}^{bias} + \mathbf{w}_{f,k} \\
\boldsymbol{\omega}_{b,k+1}^{bias} &= \boldsymbol{\omega}_{b,k}^{bias} + \mathbf{w}_{\omega,k}
\end{aligned} \quad (14)$$

where  $\mathbf{w}_f$  and  $\mathbf{w}_\omega$  are assumed to be zero mean, Gaussian white noise processes.

The incoming specific force and angular rate raw data are compensated with the estimated sensor biases, and only hereupon used in (14). The range-rate-type clock drift error is modeled as a constant plus a random walk process, while the receiver clock bias is modeled as the integral of the clock drift error.

$$\begin{bmatrix} c\Delta t_{k+1} \\ c\Delta \hat{t}_{k+1} \end{bmatrix} = \begin{bmatrix} 1 & T \\ 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} c\Delta t_k \\ c\Delta \hat{t}_k \end{bmatrix} + \mathbf{w}_{clk} \quad (15)$$

Equation (13), (14) and (15) form the system propagation model. For the observation model, the nonlinear equations are directly employed. The predicted pseudorange measurement based on the current position estimate is formulated as:

$$\hat{\rho}_k^j = \sqrt{(x_{n,k}^j - \hat{x}_{n,k})^2 + (x_{e,k}^j - \hat{x}_{e,k})^2 + (x_{d,k}^j - \hat{x}_{d,k})^2} + c\Delta \hat{t}_k \quad (16)$$

where  $\hat{\rho}_k^j$  is the predicted pseudorange measurement from the j-th satellite;  $x_{n,k}^j, x_{e,k}^j, x_{d,k}^j$  are the j-th satellite position coordinates expressed in the North east down (NED) navigation frame;  $\hat{x}_{n,k}, \hat{x}_{e,k}, \hat{x}_{d,k}$  are the vehicle position estimates resolved in the NED navigation frame.

The predicted pseudorange-rate measurements are related to the velocity estimates as:

$$\hat{\rho}_k^j = a_{n,k}^j (\dot{x}_{n,k}^j - \dot{\hat{x}}_{n,k}) + a_{e,k}^j (\dot{x}_{e,k}^j - \dot{\hat{x}}_{e,k}) + a_{d,k}^j (\dot{x}_{d,k}^j - \dot{\hat{x}}_{d,k}) + c\Delta \hat{t}_k \quad (17)$$

$$\text{where } a_{n,k}^j = \frac{x_{n,k}^j - \hat{x}_{n,k}}{\hat{d}_k^j}, a_{e,k}^j = \frac{x_{e,k}^j - \hat{x}_{e,k}}{\hat{d}_k^j}, a_{d,k}^j = \frac{x_{d,k}^j - \hat{x}_{d,k}}{\hat{d}_k^j}$$

$$\hat{d}_k^j = \sqrt{(x_{n,k}^j - \hat{x}_{n,k})^2 + (x_{e,k}^j - \hat{x}_{e,k})^2 + (x_{d,k}^j - \hat{x}_{d,k})^2}$$

where  $\hat{\rho}_k^j$  is the predicted delta pseudorange measurement from the j-th satellite;  $\dot{x}_{n,k}^j, \dot{x}_{e,k}^j, \dot{x}_{d,k}^j$  are the j-th satellite velocity coordinates expressed in NED navigation frame;  $\dot{\hat{x}}_{n,k}, \dot{\hat{x}}_{e,k}, \dot{\hat{x}}_{d,k}$  are the vehicle velocity estimates in NED navigation frame.

Given the system propagation and observation models, the UKF algorithm can now be applied. Among several UKF algorithms, the approach employing 2n equally weighted sigma points is used, which can be found in [7, 21]. Due to the quaternion normalization constraint, the degree of freedom of a quaternion vector is three rather than four. Thus, if we use quaternion vector elements as states, the dimension of the state vector is 18×1, but the dimension of state error covariance matrix is 17×17. This dimensional mismatch can be solved by transforming the quaternion vector error into its corresponding rotation vector in the rotation space. The state vector and its associated errors are defined (i.e., estimate minus truth) as:

$$\mathbf{x}_k^+ = \begin{bmatrix} \hat{\mathbf{p}}_{n,k}^+ \\ \hat{\mathbf{v}}_{n,k}^+ \\ \hat{\mathbf{q}}_k^+ \\ \hat{\mathbf{f}}_{b,k}^{bias+} \\ \hat{\mathbf{\omega}}_{b,k}^{bias+} \\ c\Delta \hat{t}_k^+ \\ c\Delta \hat{t}_k^+ \end{bmatrix}_{18 \times 1}, \delta \mathbf{x}_k^+ = \begin{bmatrix} \hat{\mathbf{p}}_{n,k}^+ - \mathbf{p}_{n,k} \\ \hat{\mathbf{v}}_{n,k}^+ - \mathbf{v}_{n,k} \\ \hat{\mathbf{q}}_k^+ \\ \hat{\mathbf{f}}_{b,k}^{bias+} - \mathbf{f}_{b,k}^{bias} \\ \hat{\mathbf{\omega}}_{b,k}^{bias+} - \mathbf{\omega}_{b,k}^{bias} \\ c\Delta \hat{t}_k^+ - c\Delta t_k \\ c\Delta \hat{t}_k^+ - c\Delta \hat{t}_k \end{bmatrix}, \delta \mathbf{x}_k^+ = \begin{bmatrix} \delta \hat{\mathbf{p}}_{n,k}^+ \\ \delta \hat{\mathbf{v}}_{n,k}^+ \\ \delta \hat{\mathbf{q}}_k^+ \\ \delta \hat{\mathbf{f}}_{b,k}^{bias+} \\ \delta \hat{\mathbf{\omega}}_{b,k}^{bias+} \\ c\delta \hat{t}_k^+ \\ c\delta \hat{t}_k^+ \end{bmatrix}_{17 \times 1} \quad (18)$$

where  $\hat{\mathbf{q}}_k^+$  is the rotation vector corresponding to  $\hat{\mathbf{q}}_k^+ \otimes (\mathbf{q}_k)^{-1}$ .

The state error covariance matrix is formulated as:

$$\mathbf{P}_k^+ = \begin{bmatrix} (\sigma_{\delta \hat{\mathbf{p}}}^2)_{3 \times 3} & \mathbf{O}_{3 \times 3} & \mathbf{O}_{3 \times 3} & \mathbf{O}_{3 \times 3} & \mathbf{O}_{3 \times 3} & \mathbf{O}_{3 \times 1} & \mathbf{O}_{3 \times 1} \\ \mathbf{O}_{3 \times 3} & (\sigma_{\delta \hat{\mathbf{v}}}^2)_{3 \times 3} & \mathbf{O}_{3 \times 3} & \mathbf{O}_{3 \times 3} & \mathbf{O}_{3 \times 3} & \mathbf{O}_{3 \times 1} & \mathbf{O}_{3 \times 1} \\ \mathbf{O}_{3 \times 3} & \mathbf{O}_{3 \times 3} & (\sigma_{\hat{\mathbf{q}}}^2)_{3 \times 3} & \mathbf{O}_{3 \times 3} & \mathbf{O}_{3 \times 3} & \mathbf{O}_{3 \times 1} & \mathbf{O}_{3 \times 1} \\ \mathbf{O}_{3 \times 3} & \mathbf{O}_{3 \times 3} & \mathbf{O}_{3 \times 3} & (\sigma_{\delta \hat{\mathbf{f}}}^2)_{3 \times 3} & \mathbf{O}_{3 \times 3} & \mathbf{O}_{3 \times 1} & \mathbf{O}_{3 \times 1} \\ \mathbf{O}_{3 \times 3} & \mathbf{O}_{3 \times 3} & \mathbf{O}_{3 \times 3} & \mathbf{O}_{3 \times 3} & (\sigma_{\delta \hat{\mathbf{\omega}}}^2)_{3 \times 3} & \mathbf{O}_{3 \times 1} & \mathbf{O}_{3 \times 1} \\ \mathbf{0}_{3 \times 1}^T & \mathbf{0}_{3 \times 1}^T & \mathbf{0}_{3 \times 1}^T & \mathbf{0}_{3 \times 1}^T & \mathbf{0}_{3 \times 1}^T & (\sigma_{c\delta \hat{t}}^2)_{1 \times 1} & 0_{1 \times 1} \\ \mathbf{0}_{3 \times 1}^T & \mathbf{0}_{3 \times 1}^T & \mathbf{0}_{3 \times 1}^T & \mathbf{0}_{3 \times 1}^T & \mathbf{0}_{3 \times 1}^T & 0_{1 \times 1} & (\sigma_{c\delta \hat{t}}^2)_{1 \times 1} \end{bmatrix}_{17 \times 17} \quad (19)$$

We generate 2n (n=17) equally weighted sigma points as:

$$\left( \sqrt{n\mathbf{P}_{k-1}^+} \right)_i^T = \begin{bmatrix} \Delta \hat{\mathbf{p}}_{n,k-1,i}^+ \\ \Delta \hat{\mathbf{v}}_{n,k-1,i}^+ \\ \hat{\mathbf{q}}_{k-1,i}^+ \\ \Delta \hat{\mathbf{f}}_{b,k-1,i}^{bias+} \\ \Delta \hat{\mathbf{\omega}}_{b,k-1,i}^{bias+} \\ \Delta c\Delta \hat{t}_{k-1,i}^+ \\ \Delta c\Delta \hat{t}_{k-1,i}^+ \end{bmatrix}, \boldsymbol{\chi}_{k-1,i}^+ = \begin{bmatrix} \hat{\mathbf{p}}_{n,k-1}^+ + \Delta \hat{\mathbf{p}}_{n,k-1,i}^+ \\ \hat{\mathbf{v}}_{n,k-1}^+ + \Delta \hat{\mathbf{v}}_{n,k-1,i}^+ \\ \delta \mathbf{q}(\hat{\mathbf{q}}_{k-1,i}^+) \otimes \hat{\mathbf{q}}_{k-1}^+ \\ \hat{\mathbf{f}}_{b,k-1}^{bias+} + \Delta \hat{\mathbf{f}}_{b,k-1,i}^{bias+} \\ \hat{\mathbf{\omega}}_{b,k-1}^{bias+} + \Delta \hat{\mathbf{\omega}}_{b,k-1,i}^{bias+} \\ c\Delta \hat{t}_{k-1}^+ + \Delta c\Delta \hat{t}_{k-1,i}^+ \\ c\Delta \hat{t}_{k-1}^+ + \Delta c\Delta \hat{t}_{k-1,i}^+ \end{bmatrix}, \quad (20)$$

$$\boldsymbol{\chi}_{k-1,i+n}^+ = \begin{bmatrix} \hat{\mathbf{p}}_{n,k-1}^+ - \Delta \hat{\mathbf{p}}_{n,k-1,i}^+ \\ \hat{\mathbf{v}}_{n,k-1}^+ - \Delta \hat{\mathbf{v}}_{n,k-1,i}^+ \\ \delta \mathbf{q}(-\hat{\mathbf{q}}_{k-1,i}^+) \otimes \hat{\mathbf{q}}_{k-1}^+ \\ \hat{\mathbf{f}}_{b,k-1}^{bias+} - \Delta \hat{\mathbf{f}}_{b,k-1,i}^{bias+} \\ \hat{\mathbf{\omega}}_{b,k-1}^{bias+} - \Delta \hat{\mathbf{\omega}}_{b,k-1,i}^{bias+} \\ c\Delta \hat{t}_{k-1}^+ - \Delta c\Delta \hat{t}_{k-1,i}^+ \\ c\Delta \hat{t}_{k-1}^+ - \Delta c\Delta \hat{t}_{k-1,i}^+ \end{bmatrix}$$

where  $i = 1, \dots, n$ , and we denote:

$$\delta \mathbf{q}(\hat{\boldsymbol{\phi}}_{k-1,j}^+) = \begin{bmatrix} \cos(0.5 \|\hat{\boldsymbol{\phi}}_{k-1,j}^+\|) \\ \sin(0.5 \|\hat{\boldsymbol{\phi}}_{k-1,j}^+\|) \frac{\hat{\boldsymbol{\phi}}_{k-1,j}^+}{\|\hat{\boldsymbol{\phi}}_{k-1,j}^+\|} \end{bmatrix} \quad (21)$$

In the time update, we pass sigma points through the nonlinear system propagation model, and compute the mean as:

$$\boldsymbol{\chi}_{k,i}^- = f_{k,k-1}(\boldsymbol{\chi}_{k-1,i}^+) = \begin{bmatrix} \hat{\mathbf{p}}_{n,k,i}^- \\ \hat{\mathbf{v}}_{n,k,i}^- \\ \hat{\mathbf{q}}_{k,i}^- \\ \hat{\mathbf{f}}_{b,k,i}^{bias-} \\ \hat{\boldsymbol{\omega}}_{b,k,i}^{bias-} \\ c\Delta \hat{t}_{k,i}^- \\ c\Delta \hat{t}_{k,i}^- \end{bmatrix}, \quad \hat{\mathbf{x}}_k^- = \frac{1}{2n} \sum_{i=1}^{2n} \boldsymbol{\chi}_{k,i}^- = \begin{bmatrix} \hat{\mathbf{p}}_{n,k}^- \\ \hat{\mathbf{v}}_{n,k}^- \\ \hat{\mathbf{q}}_k^- \\ \hat{\mathbf{f}}_{b,k}^{bias-} \\ \hat{\boldsymbol{\omega}}_{b,k}^{bias-} \\ c\Delta \hat{t}_k^- \\ c\Delta \hat{t}_k^- \end{bmatrix} \quad (22)$$

Due to the fact that the unit quaternion is not mathematically closed for addition and scalar multiplications, the renormalization must be conducted, which is handled by

$$\hat{\mathbf{q}}_k^- = \frac{\hat{\mathbf{q}}_k^-}{\|\hat{\mathbf{q}}_k^-\|}.$$

For updating the state error covariance matrix, we calculate it as:

$$\mathbf{P}_k^- = \frac{1}{2n} \sum_{i=1}^{2n} [\Delta \hat{\mathbf{x}}_k^-] [\Delta \hat{\mathbf{x}}_k^-]^T + \mathbf{Q}_{k-1} \quad (23)$$

$$\text{where } \Delta \hat{\mathbf{x}}_k^- = \boldsymbol{\chi}_{k,i}^- - \hat{\mathbf{x}}_k^- = \begin{bmatrix} \hat{\mathbf{p}}_{n,k,i}^- - \hat{\mathbf{p}}_{n,k}^- \\ \hat{\mathbf{v}}_{n,k,i}^- - \hat{\mathbf{v}}_{n,k}^- \\ \hat{\mathbf{q}}_{k,i}^- \\ \hat{\mathbf{f}}_{b,k,i}^{bias-} - \hat{\mathbf{f}}_{b,k}^{bias-} \\ \hat{\boldsymbol{\omega}}_{b,k,i}^{bias-} - \hat{\boldsymbol{\omega}}_{b,k}^{bias-} \\ c\Delta \hat{t}_{k,i}^- - c\Delta \hat{t}_k^- \\ c\Delta \hat{t}_{k,i}^- - c\Delta \hat{t}_k^- \end{bmatrix}, \quad \text{and } \hat{\boldsymbol{\phi}}_{k,i}^- \text{ is the}$$

rotation vector corresponding to  $\hat{\mathbf{q}}_{k,i}^- \otimes (\hat{\mathbf{q}}_k^-)^{-1}$ .

Having the a priori mean and covariance matrix, we generate the sigma points for measurement updates as:

$$\left(\sqrt{n\mathbf{P}_k^-}\right)_i^T = \begin{bmatrix} \Delta \hat{\mathbf{p}}_{n,k,i}^- \\ \Delta \hat{\mathbf{v}}_{n,k,i}^- \\ \hat{\boldsymbol{\phi}}_{k,i}^- \\ \Delta \hat{\mathbf{f}}_{b,k,i}^{bias-} \\ \Delta \hat{\boldsymbol{\omega}}_{b,k,i}^{bias-} \\ \Delta c\Delta \hat{t}_{k,i}^- \\ \Delta c\Delta \hat{t}_{k,i}^- \end{bmatrix}, \quad \boldsymbol{\chi}_{k,i}^- = \begin{bmatrix} \hat{\mathbf{p}}_{n,k}^- + \Delta \hat{\mathbf{p}}_{n,k,i}^- \\ \hat{\mathbf{v}}_{n,k}^- + \Delta \hat{\mathbf{v}}_{n,k,i}^- \\ \delta \mathbf{q}(\hat{\boldsymbol{\phi}}_{k,i}^-) \otimes \hat{\mathbf{q}}_k^- \\ \hat{\mathbf{f}}_{b,k}^{bias-} + \Delta \hat{\mathbf{f}}_{b,k,i}^{bias-} \\ \hat{\boldsymbol{\omega}}_{b,k}^{bias-} + \Delta \hat{\boldsymbol{\omega}}_{b,k,i}^{bias-} \\ c\Delta \hat{t}_k^- + \Delta c\Delta \hat{t}_{k,i}^- \\ c\Delta \hat{t}_k^- + \Delta c\Delta \hat{t}_{k,i}^- \end{bmatrix},$$

$$\boldsymbol{\chi}_{k,i+n}^- = \begin{bmatrix} \hat{\mathbf{p}}_{n,k}^- - \Delta \hat{\mathbf{p}}_{n,k,i}^- \\ \hat{\mathbf{v}}_{n,k}^- - \Delta \hat{\mathbf{v}}_{n,k,i}^- \\ \delta \mathbf{q}(-\hat{\boldsymbol{\phi}}_{k,i}^-) \otimes \hat{\mathbf{q}}_k^- \\ \hat{\mathbf{f}}_{b,k}^{bias-} - \Delta \hat{\mathbf{f}}_{b,k,i}^{bias-} \\ \hat{\boldsymbol{\omega}}_{b,k}^{bias-} - \Delta \hat{\boldsymbol{\omega}}_{b,k,i}^{bias-} \\ c\Delta \hat{t}_k^- - \Delta c\Delta \hat{t}_{k,i}^- \\ c\Delta \hat{t}_k^- - \Delta c\Delta \hat{t}_{k,i}^- \end{bmatrix} \quad (24)$$

$$\text{where } i=1, \dots, n, \text{ and } \delta \mathbf{q}(\hat{\boldsymbol{\phi}}_{k,i}^-) = \begin{bmatrix} \cos(0.5 \|\hat{\boldsymbol{\phi}}_{k,i}^-\|) \\ \sin(0.5 \|\hat{\boldsymbol{\phi}}_{k,i}^-\|) \frac{\hat{\boldsymbol{\phi}}_{k,i}^-}{\|\hat{\boldsymbol{\phi}}_{k,i}^-\|} \end{bmatrix}.$$

We pass them through the nonlinear measurement models and compute the predicted measurement and covariance matrices as:

$$\begin{aligned} \hat{\mathbf{y}}_k &= \frac{1}{2n} \sum_{i=1}^{2n} h_k(\boldsymbol{\chi}_{k,i}^-) \\ \mathbf{P}_k^{yy} &= \frac{1}{2n} \sum_{i=1}^{2n} [h_k(\boldsymbol{\chi}_{k,i}^-) - \hat{\mathbf{y}}_k] [h_k(\boldsymbol{\chi}_{k,i}^-) - \hat{\mathbf{y}}_k]^T + \mathbf{R}_k \\ \mathbf{P}_k^{xy} &= \frac{1}{2n} \sum_{i=1}^{2n} [\boldsymbol{\chi}_{k,i}^- - \hat{\mathbf{x}}_k^-] [h_k(\boldsymbol{\chi}_{k,i}^-) - \hat{\mathbf{y}}_k]^T \end{aligned} \quad (25)$$

The Kalman gain and the state correction terms are computed as:

$$\mathbf{K}_k = \mathbf{P}_k^{xy} (\mathbf{P}_k^{yy})^{-1}, \quad \Delta \hat{\mathbf{x}}_k^+ = \mathbf{K}_k (\tilde{\mathbf{y}}_k - \hat{\mathbf{y}}_k) = \begin{bmatrix} \Delta \hat{\mathbf{p}}_{n,k}^+ \\ \Delta \hat{\mathbf{v}}_{n,k}^+ \\ \hat{\boldsymbol{\phi}}_k^+ \\ \Delta \hat{\mathbf{f}}_{b,k}^{bias+} \\ \Delta \hat{\boldsymbol{\omega}}_{b,k}^{bias+} \\ \Delta c\Delta \hat{t}_k^+ \\ \Delta c\Delta \hat{t}_k^+ \end{bmatrix} \quad (26)$$

The last step in the UKF algorithm is to apply the corrections.

$$\hat{\mathbf{x}}_k^+ = \begin{bmatrix} \hat{\mathbf{p}}_{n,k}^- + \Delta \hat{\mathbf{p}}_{n,k}^+ \\ \hat{\mathbf{v}}_{n,k}^- + \Delta \hat{\mathbf{v}}_{n,k}^+ \\ \mathbf{q}(\hat{\boldsymbol{\phi}}_k^+) \otimes \hat{\mathbf{q}}_k^- \\ \hat{\mathbf{f}}_{b,k}^{bias-} + \Delta \hat{\mathbf{f}}_{b,k}^{bias+} \\ \hat{\boldsymbol{\omega}}_{b,k}^{bias-} + \Delta \hat{\boldsymbol{\omega}}_{b,k}^{bias+} \\ c\Delta \hat{t}_k^- + \Delta c\Delta \hat{t}_k^+ \\ c\Delta \hat{t}_k^- + \Delta c\Delta \hat{t}_k^+ \end{bmatrix}, \quad \mathbf{P}_k^+ = \mathbf{P}_k^- - \mathbf{K}_k \mathbf{P}_k^{yy} \mathbf{K}_k^T \quad (27)$$

### III. FIELD EXPERIMENT

In this section, the quaternion-based approaches are analyzed based on a train ride field experiment from the city of Haiger to the city of Siegen in the north-west of Germany. The train ride takes about 23 min (1400 s) with long tunnels (GPS outages), as shown in Figure 1. One Landmark<sup>TM</sup>20 eXT MEMS-based IMU (100 Hz) and one u-blox Antaris 4 receiver (1 Hz) are used. The main sensor errors of the Landmark<sup>TM</sup>20 eXT IMU are given in Table I.

TABLE I LANDMARK<sup>TM</sup>20 IMU PERFORMANCE SPECIFICATION

Gyroscope (Angular rates)	Bias in-run stability ( $1\sigma$ )	Noise (ARW) ( $1\sigma$ )	Scale Factor Error [ppm]
	20 [ $^{\circ}$ /h]	0.035 [ $^{\circ}$ /s/ $\sqrt{\text{Hz}}$ ]	$\leq 1000$
Accelerometer (Specific forces)	Bias in-run stability ( $1\sigma$ )	Noise (VRW) ( $1\sigma$ )	Scale Factor Error [ppm]
	20 [ $\mu\text{g}$ ]	40 [ $\mu\text{g}/\sqrt{\text{Hz}}$ ]	$\leq 1000$

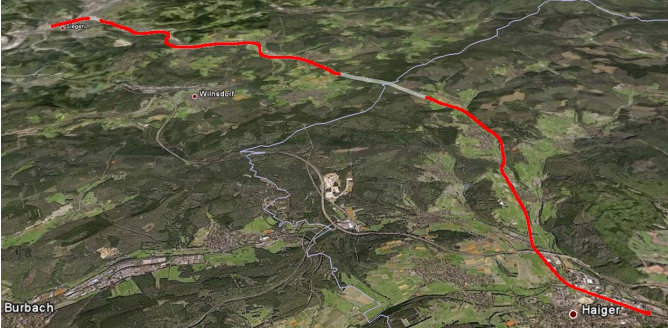


Figure 1. Train trajectory with long GPS outage periods (plotted in Google earth).

The number of tracked satellites is depicted in Fig. 2, where the first GPS outage event stretches over 93 s (about 3 km), and the second one stretches over 43 s.

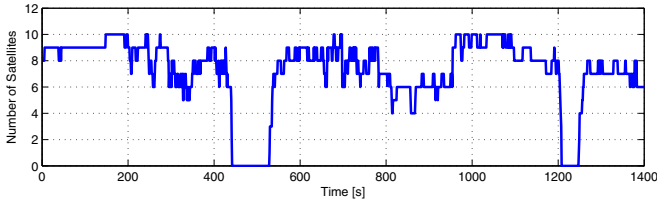


Figure 2. Number of satellites in view.

In the first test, the filter is initialized with roughly accurate attitude information (readings from the onboard magnetometer). And in the second test, the initial attitude error is set to be 30 degrees in each axis. In both tests, the IMU sensor errors have not been compensated before application started. They will be estimated during run-time.

#### A. With Correct Initial Heading Information

The EKF and UKF employ different quaternion update equations in the propagation phase of the KF (i.e., (8) and (13)). The long GPS outage periods provide the opportunity to

explore the behaviour of both quaternion propagation approaches. With the correct initial attitude information, the velocity estimates from the EKF and UKF methods are plotted in Fig. 3. The reason to plot velocity estimates is that, in this paper, the heading is derived from horizontal velocity when the norm of velocity is larger than 5 m/s. The quaternion estimates are plotted in Fig. 4.

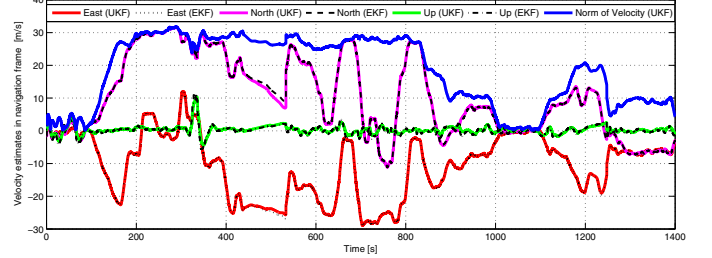


Figure 3. Velocity estimation results comparison.

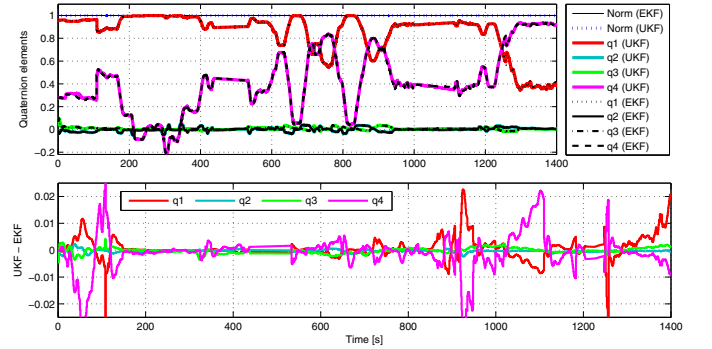


Figure 4. Attitude estimation results comparison.

As shown in Fig. 4, the quaternion estimates derived from the EKF and UKF are comparable (i.e., the unit of y axis in the 2nd subplot of Figure 4 is very small). These small differences may partly come from the different quaternion propagation approaches employed. They may also come from the different handling of nonlinearities in the system. From Fig. 2, we know that there are no GPS satellites in view between 450 s to 540 s. In this period, the quaternion elements are only propagated over time without measurement update. Fig. 4 shows that, during this period, the differences in the quaternion propagation from both approaches are very small, which can be almost neglected. The main differences are observed in q1 and q4 estimates (where heading information is contained) when GPS measurements are available, proposing that it's the measurement update which is responsible for the deviations. For providing a more tangible interpretation of the results, Fig. 5 shows the Euler angle time-history corresponding to the quaternion plots given in Fig. 4.

As shown in Fig. 5, both approaches exhibit almost same results in terms of roll and pitch angles. However, when it comes to the yaw angle, the difference is slightly larger. As depicted on the right-hand side of the 4th subplot in Fig. 5, the  $1\sigma$  value of the yaw angle estimation difference over time is 1.1 degree, which should come from the different handling of nonlinearity in the system models. However, without reference, we cannot exactly tell which method provides better results.

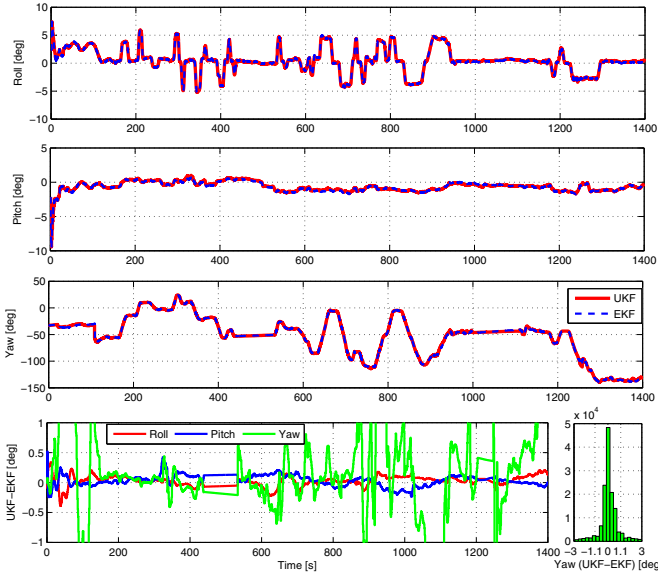


Figure 5. Attitude estimation results comparison (transformed to Euler angles).

The estimated gyro biases from the EKF and UKF algorithms and their differences are plotted in Fig. 6.

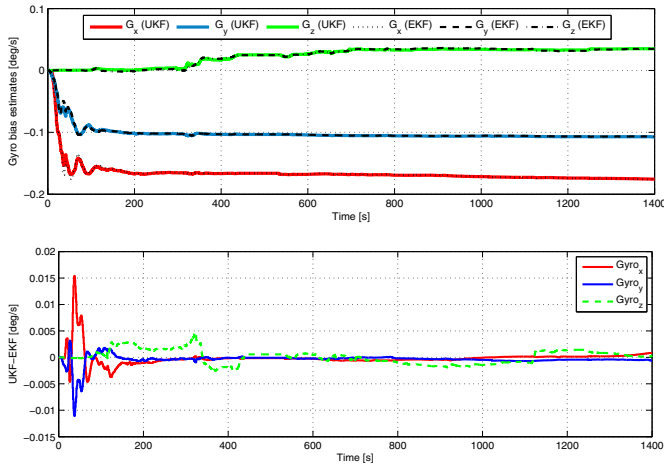


Figure 6. Gyro bias estimation differences between UKF and EKF.

Note that the inertial sensors have not been calibrated beforehand and that the initial gyro bias estimates are set to zero for the lack of better knowledge. As shown in Fig. 6, both approaches yield quite similar gyro bias estimation results, differing mainly during convergence, where the innovations are expected to be large. However, when looking at the y axis unit of the second sub-plot of Fig. 6, these differences are very small.

### B. With Initial Attitude Errors of 30 Degree in Each Axis

To test the robustness of the EKF and UKF based algorithms, an initial attitude error of 30 degrees is given in each axis. This error is not unrealistic for an actual application on heading, but is very large on roll and pitch for land-based navigation. The attitude estimation results are shown in Fig. 7, where the convergence process is seen at first few seconds of

the roll and pitch estimates, but not at the yaw angle. This is because that, the yaw angle is the least observable state for particular vehicle motion (e.g., low dynamic). In the algorithms presented in this paper, when the norm of vehicle velocity is over 5 m/s, the heading angle will be derived from the horizontal velocity estimates. Therefore, at about the 100<sup>th</sup> second, when the dynamic of the vehicle arises, the yaw angle is updated from the velocity estimates.

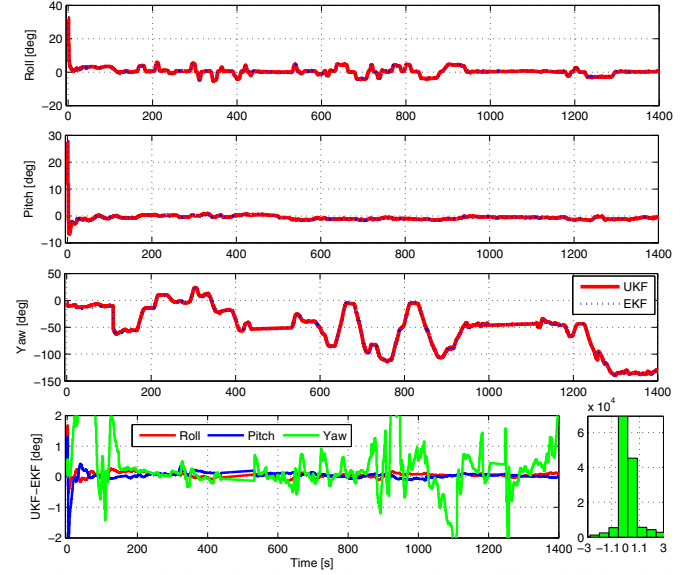


Figure 7. Attitude estimation results comparison with an initial attitude error of 30 degrees in each axis (transformed to Euler angles).

It is interesting to see that during the convergence process, both filters present quite similar performances. With the incorrect initial attitude information, the estimated gyro bias errors from the EKF and UKF algorithms and their differences are plotted in Fig. 8. Compared with the corresponding results shown in Fig. 6, we see slightly larger differences in the convergence processes at the first few seconds. However, when looking at the unit (y axis of the second sub-plot of Fig. 8), these differences are still very small.

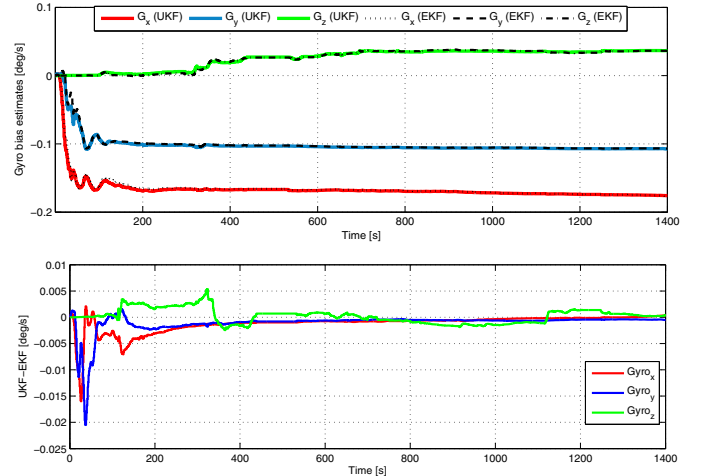


Figure 8. Gyro bias estimation differences between UKF and EKF with an initial attitude error of 30 degrees in each axis.



The reason why both approaches show similar convergence performances even when the initial attitude errors are set to be 30 degrees could be because that, using the quaternion as the representation of attitude, the direction cosine matrix does not feature trigonometric operations, which reduce nonlinearity problems compared to the Euler angle-based approach. Performances from the Euler-angle based approaches can be found in [22-24].

On the processing load, the EKF requires a significantly lower computational burden (about 40% of the UKF approach). This processing time is evaluated using Matlab programming, which cannot be considered as a direct measure of the exact computational burden. For the knowledge on the exact processing load, usually, the number of multiplications and additions should be counted separately, as the way done in [25-26]. It is also worth mentioning that for any KF-based integration approach, correct tuning of the filter covariance parameters is crucial. A variation of these parameters may significantly change the system estimation performance. In order to prevent this problem, the same set of covariance parameters is applied to both methods throughout the results presented in this article.

#### IV. CONCLUSION

In this contribution, the system performances of a tightly-coupled INS/GPS system using quaternion-based EKF and UKF algorithms are analyzed. A train ride field experiment is evaluated. Numerical results show that both approaches yield comparable attitude and bias estimation accuracy. The different ways of propagating the quaternion vector over time do not imply a difference. The UKF should principally handle the nonlinear nature of the problem, and quaternion propagation in a more accurate manner. However, we do not see much difference with respect to the EKF results, even when the initial attitude errors are set to be 30 degrees in each axis.

#### ACKNOWLEDGMENT

The first author gratefully acknowledges the support provided by the Multi Modal Sensor Systems for Environmental Exploration and Safety (MOSES) from Center for Sensorsystems (ZESS) in University of Siegen, Germany.

#### REFERENCES

- [1] O. Loffeld, *Estimationstheorie I + II*. München: Oldenbourg Verlag, 1990.
- [2] R. G. Brown and P. Y. C. Hwang, *Introduction to Random signals and Applied Kalman Filtering*, 3 ed. New York: John Wiley & Sons, 1997.
- [3] P. S. Maybeck, *Stochastic Models, Estimation and Control* vol. 1. New York: Academic Press Inc., 1979.
- [4] D. Lerro and Y. Bar-Shalom, "Tracking with debiased consistent converted measurements versus EKF," *IEEE Trans. on Aerospace and Electronic Systems*, vol. 29, pp. 1015 - 1022, 1993.
- [5] J. L. Crassidis, "Sigma-Point Kalman Filtering for Integrated GPS and Inertial Navigation," in *AIAA Guidance, Navigation, and Control Conference*, San Francisco, California, 2005.
- [6] S. J. Julier and J. K. Uhlmann, "Unscented filtering and nonlinear estimation," *IEEE Review*, vol. 92, pp. 401 - 422, 2004.
- [7] S. J. Julier, J. K. Uhlmann, and H. F. Durrant-Whyte, "A new method for the nonlinear transformation of means and covariances in filters and estimators," *IEEE Trans. on Automatic Control*, vol. 45, pp. 477 - 482, 2000.
- [8] S. J. Julier and J. K. Uhlmann, "A Consistent, Debiased Method for Converting Between Polar and Cartesian Coordinate Systems," in *AeroSense: Acquisition, Tracking and Pointing XI*, Orlando FL, 1997, pp. 110-121.
- [9] A. Farrell, *Aided Navigation: GPS with High Rate Sensors*. New York: McGraw-Hill Professional, 2008.
- [10] E.-H. Shin, X. Niu, and N. El-Sheimy, "Performance Comparison of the Extended and the Unscented Kalman Filter for Integrated GPS and MEMS-Based Inertial Systems," in *ION NTM 2005*, San Diego, CA, 2005, pp. 961-969.
- [11] E.-H. Shin, "A Quaternion-Based Unscented Kalman Filter for the Integration of GPS and MEMS INS," in *ION GNSS 2004*, Long Beach, CA, 2004, pp. 1060-1068.
- [12] E.-H. Shin and N. El-Sheimy, "Unscented Kalman Filter and Attitude Errors of Low-cost Inertial Navigation Systems," *NAVIGATION*, vol. 54, pp. 1-9, 2007.
- [13] W. Khoder, B. Fassnüt-Mombot, and M. Benjelloun, "Quaternion Unscented Kalman Filtering for integrated Inertial Navigation and GPS," in *11th International Conference on Information Fusion*, 2008 Cologne, Germany, 2008.
- [14] E. J. Lefferts, F. L. Markley, and M. D. Schuster, "Kalman Filtering for Spacecraft Attitude Estimation," *Journal of Guidance, Control, and Dynamics*, vol. 5, pp. 417-429, 1982.
- [15] S. Hong, M. H. Lee, H.-H. Chun, S.-H. Kwon, and J. L. Speyer, "Observability of error States in GPS/INS integration," *IEEE Trans. on Vehicular Technology*, vol. 54, pp. 731 - 743, 2005.
- [16] J. T. W. Ryley, P. M. G. Silson, A. Tsourdos, and S. Jordan, "Observability of a stationary GPS aided INS," in *AIAA Guidance, Navigation, and Control Conference*, Toronto, Ontario, 2010.
- [17] I. Rhee, M. F. Abdel-Hafez, and J. L. Speyer, "Observability of an integrated GPS/INS during maneuvers," *IEEE Trans. on Aerospace and Electronic Systems*, vol. 40, pp. 526 - 535, 2004.
- [18] W. R. Hamilton, *Elements of Quaternions*. London, England: Longmans, Green and Co., 1866.
- [19] S. Sukkariéh, E. M. Nebot, and H. F. Durrant-Whyte, "A High Integrity IMU/GPS Navigation Loop for Autonomous Land Vehicle Applications," *IEEE Trans. on Robotics and Automation*, vol. 15, pp. 572-578, 1999.
- [20] D. H. Titterton and J. L. Weston, *Strapdown Inertial Navigation Technology*, 2 ed.: The institution of Electrical Engineers, 2004.
- [21] D. Simon, *Optimal State Estimation: Kalman, H Infinity, and Nonlinear Approaches*, 1 ed. New Jersey: John Wiley & Sons, 2006.
- [22] J. Zhou, E. Edwan, S. Knedlik, and O. Loffeld, "Low-cost INS/GPS with nonlinear filtering methods," in *FUSION*, Edinburgh, 2010.
- [23] J. Zhou, Y. Yang, J. Zhang, E. Edwan, "Applying Quaternion-based Unscented Particle Filter on INS/GPS with Field Experiments", *Proceedings of the 24th International Technical Meeting of The Satellite Division of the Institute of Navigation (ION GNSS 2011)*, Portland, Oregon, September 20 - 23, 2011.
- [24] J. Zhou, S. Knedlik, O. Loffeld, "INS/GPS tightly-coupled integration using adaptive unscented particle filter", *The Journal of Navigation* (Royal Institute of Navigation), Vol. 63, no. 3, pp. 491-513, 2010.
- [25] J. Zhou, S. Knedlik, O. Loffeld, "Sequential Processing of Integrated Measurements in Tightly-coupled INS/GPS Integrated Navigation System", *AIAA Guidance, Navigation and Control*, Toronto, Ontario, Canada, 2-5 August 2010.
- [26] J. Zhou, S. Knedlik, and O. Loffeld, "INS/GPS for High-Dynamic UAV-Based Applications," *International Journal of Navigation and Observation*, vol. 2012, 2012.