Mock-Measurement Based Performance Evaluation of Inertial Navigation Without Knowing Ground Truth

Deqiang Han Center for Information Engineering Science Research Xi'an Jiaotong University Xi'an, Shaanxi, China 710049 Email: deqhan@gmail.com X. Rong Li and Yu Liu Department of Electrical Engineering University of New Orleans New Orleans, LA, U.S.A. 70148 Email: xli@uno.edu; lyu2@uno.edu

Abstract—As an important navigation technology, the inertial navigation system (INS) has been widely used in various applications and many INS algorithms have been proposed. Their performance evaluation is crucial for evaluating and improving the algorithms, where the ground truth is assumed known. However, knowledge of the ground truth is hard to acquire, and this presents a challenge to performance evaluation. In this paper, a truth-knowledge free approach to performance evaluation of INS algorithms is proposed. In this approach, we generate some mock measurements from the INS outputs and judging the performance by checking how close the mock measurements are to the real measurements. The INS algorithm whose mock measurements are closer to the real measurements is preferred. Simulations and related analysis are provided to illustrate and validate our proposed evaluation approach.

Index Terms—Mock measurements, performance evaluation, Strapdown INS, ground truth, navigation.

I. INTRODUCTION

The inertial navigation system (INS) [1], [2], which is a dead-reckoning navigation system in the 3-D space, integrates measurements of accelerometers and gyroscopes to provide estimates of position, velocity, and attitude of its carrier. INS has two different types: gimbaled and strapdown [3]. The major drawback of INS is the time-accumulating navigation errors. With more sophisticated inertial sensors and algorithms, INS performance has been improved significantly. INS is a critical component in many applications involving airplanes, ships, automobiles, and so on [2].

Performance evaluation (PE) of INS is important for navigation application [4], [5]. Existing work mainly focuses on two aspects. a) PE of inertial sensors (gyroscopes and accelerometers) [6]–[8]. For example, using Lab Testing [6] to provide parameters such as noise density, bias instability, scale factor instability, orthogonality, g-sensitivity, and temperature sensitivities. b) PE of the INS algorithms such as strapdown INS [3] alignment [9], error modeling, propagation and analysis [10], [11], and attitude/velocity/position (navigation) computation [12], [13].

In this work, we consider PE of INS algorithms for navigation computation. The most common way for PE of INS algorithms in practice is Field Testing; that is, to use information from sensors of higher accuracy (e.g., GPS signals or other navigation/tracking systems' output) as a "ground truth" to evaluate the performance [14], [15]. Another way for PE is simulation based [16]–[19], where the ground truth is predefined by the user and the artificially generated INS sensor measurements are processed by different INS algorithms to output different solutions for evaluation. People also use the results of a well-accepted INS algorithm, which provides good performance theoretically or practically, as the relative "ground truth" for PE.

As we can see, in the existing approaches to PE of INS algorithms, knowledge of the ground truth is always needed. In practice, exact knowledge of the ground truth is often difficult to obtain. For example, a navigation/tracking system with a higher accuracy such as GPS¹ may not be available for PE; the GPS might be temporarily unstable or blocked. Although artificial ground truth could be generated (e.g., using a "well-accepted" algorithm), such a "ground truth" is neither objective nor reliable for PE, since the optimal algorithm is usually either non-existent or infeasible to implement in practice.

In fact, PE without knowing the ground truth is badly called for in practice due to the difficulty to access the truth [20]. Then, how to implement PE without knowing the ground truth in practice? The real measurements acquired from INS sensors are always available and without artificial ingredients. Therefore, in this paper, the PE is implemented by comparing the real measurements and the mock measurements generated from each INS algorithm's navigation output. An

This work was supported by the Grant for State Key Program for Basic Research of China (973) (No. 2013CB329405), National Natural Science Foundation (No. 61104214, No. 61203222), Science and technology project of Shaanxi Province (No. 2013KJXX-46), Specialized Research Fund for the Doctoral Program of Higher Education (20120201120036), NASA/LEQSF(2013-15)-Phase3-06 through grant NNX13AD29A, and Fundamental Research Funds for the Central Universities (No. xjj2014122).

¹In realistic Field Testing, GPS/INS intergration is often used as the ground truth. However, our goal is to evaluate the performance of INS, and thus INS itself is inappropriate to serve as the ground truth.

INS algorithm is better if its mock measurements are closer to the real measurements. Simulation results and related analyses are provided to validate our proposed method for PE.

II. BASICS OF INS NAVIGATION COMPUTATION

The inertial measurement unit (IMU) in INS includes three accelerometers and three gyroscopes. A navigation processor processes the measurements acquired by the IMU to provide navigation information such as position, velocity and attitude. There are gimbaled INS and strapdown INS. In this paper, we focus on the strapdown INS (SINS) [3], where inertial sensors are aligned with the body of the carrier ². SINS must be initialized before the navigation computation. The attitude solution is used to transform the measurements under the body frame to the resolving (or navigation) coordinate frame used by the navigation processor. Integrating the acceleration yields the velocity, and integrating the velocity gives the position. There are many types of SINS algorithms [2], [13], [19]. A schematic diagram of basic SINS computation [1]–[3] is shown in Fig. 1.



Fig. 1. Schematic diagram of SINS.

In Fig. 1, Δt is the sampling interval, and $C_b^n(\cdot)$ denotes the direction cosine matrix (DCM) from the body frame (denoted by *b*) to the navigation frame (denoted by *n*).³ The elements of the DCM can be determined using attitude angles (θ, ψ, ϕ) [3].

The DCM update is very crucial in SINS computation. There are many methods based on rotation vectors, direction cosines, and quaternions, etc. A rotation can be represented by a quaternion, which is a 4D hyper complex number $q = (q_0, q_1, q_2, q_3)$. The DCM can be represented by a quaternion [1]–[3]. So the DCM update can be implemented by updating the quaternion q, which is briefly introduced below. For simplicity, here we use a gravity-free model. q is updated by

$$\dot{q} = \frac{1}{2}W \cdot q \tag{1}$$

where

$$W = \begin{bmatrix} 0 & -\omega_x & -\omega_y & -\omega_z \\ \omega_x & 0 & \omega_z & -\omega_y \\ \omega_y & -\omega_z & 0 & \omega_x \\ \omega_z & \omega_y & -\omega_x & 0 \end{bmatrix}$$
(2)

Here $[\omega_x, \omega_y, \omega_z]$ is the 3-D angular velocity from gyroscopes. Then the skew-symmetric matrix of the angle increment is

$$\Delta \Theta = \int_{t_{k-1}}^{t_k} W dt \approx \begin{bmatrix} 0 & -\Delta \theta_x & -\Delta \theta_y & -\Delta \theta_z \\ \Delta \theta_x & 0 & \Delta \theta_z & -\Delta \theta_y \\ \Delta \theta_y & -\Delta \theta_z & 0 & \Delta \theta_x \\ \Delta \theta_z & \Delta \theta_y & -\Delta \theta_x & 0 \end{bmatrix}_{(3)}$$

Rewrite Eq. (1) in discretized form as

$$q_k = e^{\frac{1}{2} \int_{t_{k-1}}^{t_k} W dt} \cdot q_{k-1}$$
 (4)

By Taylor series expansion,

$$q_{k} = \left[I + \frac{\frac{1}{2}\Delta\Theta}{1!} + \frac{\left(\frac{1}{2}\Delta\Theta\right)^{2}}{2!} + \cdots\right] \cdot q_{k-1}$$
$$= \underbrace{\left[I \cdot \cos\frac{\|\Delta\theta\|}{2} + \Delta\Theta\frac{\sin\frac{\|\Delta\theta\|}{2}}{\|\Delta\theta\|}\right]}_{A(\Delta\theta)} \cdot q_{k-1}$$
(5)

where I is the identity matrix. To save computation, approximations are often adopted in the calculation of $A(\Delta\theta)$ in Eq. (5) [3]. The 1st order approximation is

$$A(\Delta\theta) = I + \frac{\Delta\Theta}{2} \tag{6}$$

The 2nd order approximation is

$$A(\Delta\theta) = \left[I\left(1 - \frac{\|\Delta\theta\|^2}{8}\right) + \frac{\Delta\Theta}{2}\right]$$
(7)

The 4th order approximation is

$$A(\Delta\theta) = \left[I\left(1 - \frac{\|\Delta\theta\|^2}{8} + \frac{\|\Delta\theta\|^4}{384}\right) + \left(\frac{1}{2} - \frac{\|\Delta\theta\|^2}{48}\right)\Delta\Theta\right]$$
(8)

Different approximations lead to different navigation results. Intuitively and theoretically, a higher order approximation should result in a higher precision.

III. PERFORMANCE EVALUATION OF SINS ALGORITHMS KNOWING GROUND TRUTH

PE of SINS algorithms can be implemented based on the difference between the ground truth and the navigation results obtained. Suppose there are totally M SINS algorithms. $\hat{X}^i = [\hat{x}_1^i, ..., \hat{x}_N^i]$ is the navigation (estimation) result obtained using algorithm i, where N is the total time steps. Suppose that the

²Our approach for gimbaled INS is parallel.

³Here, the body frame used is "right-head-up" and the navigation frame used is "East-North-Up (ENU)".

ground truth is $X = [x_1, ..., x_N]$. Then the error vector of algorithm *i* is

$$\ddot{X}^i = X - \ddot{X}^i \tag{9}$$

1 /0

Then, root mean square error (RMSE) [5] for SINS algorithm i can be obtained as

$$\text{RMSE}^{i}(\hat{x}) = \left(\frac{1}{M} \sum_{j=1}^{M} ||\tilde{x}_{j}^{i}||^{2}\right)^{1/2}$$
(10)

RMSE is an important and widely used measure for PE of state estimation. However, RMSE is dominated by large errors. A better measure is average Euclidean error (AEE) [5]

$$AEE^{i}(\hat{x}) = \frac{1}{M} \sum_{j=1}^{M} ||\tilde{x}_{j}^{i}||$$
(11)

However, knowledge of the ground truth is seldom available in practice. This is a major difficulty for conventional PE methods.

IV. MOCK-MEASUREMENT BASED EVALUATION OF SINS PERFORMANCE WITHOUT KNOWING GROUND TRUTH

A. Basic Idea

What else is available if the ground truth is not available? We provide an answer in this section.

A SINS algorithm A can be considered as an estimator [21], whose input is the real measurements Z (in the measurement space Z) and output is the estimate \hat{X} (in the state space X). Here we do not know the ground truth X. Without knowing the ground truth, real measurements can be used for PE. If we can generate mock measurements $\hat{Z} = [\hat{z}_1, ..., \hat{z}_N]$ from \hat{X} , then we can implement the PE in Z by comparing Z and \hat{Z} . If \hat{Z} is close to Z, the algorithm is deemed to perform well. Such a PE in measurement space Z can be considered as an alternative to the one in the state space X. See Fig. 2 for an illustration.



Fig. 2. Diagram of mock measurements based PE.

B. Generating Mock Measurements

Mock-measurement generation is a mapping from the state space \mathcal{X} to the measurement space \mathcal{Z} . It includes the generation of mock 3-D acceleration measurements and mock 3-D angular velocity measurements.

1) Mock acceleration measurements: Suppose that in the navigation frame, the initial position of the moving body is $(p_x(0), p_y(0), p_z(0))$, the initial velocity is $(v_x(0), v_y(0), v_z(0))$, and the initial attitude is $(\theta(0), \psi(0), \phi(0))$. Suppose that the sample interval is Δt . At time k (k = 1, ..., N), the position estimate is

$$p(k) = [\hat{x}(k), \hat{y}(k), \hat{z}(k)]$$

and the 3-D attitude estimate is

$$Att(k) = [\hat{\theta}(k), \hat{\psi}(k), \hat{\phi}(k)]$$

Using Att(k), the DCM $C_b^n(k)$ at time k can be obtained.

The average velocity from time k - 1 to time k is $\frac{V^n(k-1)+V^n(k)}{2}$. Then

$$\frac{V^n(k-1)+V^n(k)}{2} \cdot \Delta t = p(k) - p(k-1)$$

$$\Rightarrow V^n(k) - V^n(k-1) = 2\left(\frac{p(k)-p(k-1)}{\Delta t} - V^n(k-1)\right)$$

Therefore, the acceleration in the navigation frame is

$$a^{n}(k-1) = \frac{V^{n}(k) - V^{n}(k-1)}{\Delta t} = \frac{2}{\Delta t} \left(\frac{p(k) - p(k-1)}{\Delta t} - V^{n}(k-1) \right)$$
(12)

Then, the acceleration in the body frame can be obtained as

$$a^{b}(k-1) = C_{n}^{b}(k) \cdot a^{n}(k-1) = [C_{b}^{n}(k)]^{T} \cdot a^{n}(k-1)$$
(13)

where $C_n^b = (C_b^n)^T$ denotes the DCM from the navigation frame to the body frame.

Mock acceleration measurements can be generated based on Eq. (12) and Eq. (13).

2) Mock angular velocity measurements: Based on the attitude at time k - 1 and k, DCM $C_n^b(k - 1)$ and $C_n^b(k)$ can be obtained. Using DCM $C_n^b(k - 1)$ and $C_n^b(k)$, we can obtain [3]

$$\Delta \Psi = C_n^b(k-1) \cdot \left(C_n^b(k)\right)^T - I \tag{14}$$

Then, the angular velocity measurements in the body frame can be obtained as

$$\begin{cases} \omega_x^b(k-1) = \Delta \hat{\psi}^b(k-1) / \Delta t = \Delta \Psi_{(2,1)} / \Delta t \\ \omega_y^b(k-1) = \Delta \hat{\theta}^b(k-1) / \Delta t = \Delta \Psi_{(1,3)} / \Delta t \\ \omega_z^b(k-1) = \Delta \hat{\phi}^b(k-1) / \Delta t = \Delta \Psi_{(3,2)} / \Delta t \end{cases}$$
(15)

where $\Delta \Psi_{(i,j)}$ denotes the (i,j)th element of $\Delta \Psi$.

Remark: The mock measurements \check{Z} generated based on the above method (denoted by $\hat{h}(\cdot)$ below) theoretically can be regarded as noiseless mock measurements:

$$Z = h(X)$$

It should be noted that each real measurement Z actually contains noise v (including random and non-random part) caused by the IMU sensors:

$$Z = h(X) + v \tag{16}$$

Therefore, to fairly and objectively quantify the difference between the real measurement and the mock measurement, noise with the same distribution f(v) as in the real measurement should be added to the "pure" mock measurement ⁴, ; that is, to quantify the difference between Z in Eq. (16) and \hat{Z} in Eq. (17)

$$\hat{Z} = \hat{h}(\hat{X}) + v = \check{Z} + v \tag{17}$$

Furthermore, adding only a single realization of the error v to the "pure" mock measurements is not sufficient due to the randomness as illustrated in Fig. 3.



Fig. 3. Illustration of the necessity for multi-generation.

Suppose there are six mock measurements generated by adding error as in Fig. 3. In Case I, a single realization of the mock measurement with noise is $\tilde{Z} + \tilde{v}_1$, which is close to the real measurements; however, in Case II, when the single realization of the mock measurement is $\tilde{Z} + \tilde{v}_5$, which is relatively far from the real measurements. This distance varies significantly due to the randomness of noise v. Therefore, we should generate the noise v multiple times (R times) to obtain { $\tilde{v}_1, ... \tilde{v}_R$ } and add them to the "noiseless" mock measurements to generate a group of mock measurements as

$$\{\check{Z}\} = \{\check{Z} + \check{v}_1, ..., \check{Z} + \check{v}_R\}$$
(18)

These mock measurements are generated as follows.

3) Multi-realization of mock measurements: The error of an accelerometer includes three parts [3]: (a) bias $(\beta_x^a, \beta_y^a, \beta_z^a)$ (in m/s^2), (b) zero-mean Gaussian white noise with standard deviations $(\sigma_x^a, \sigma_y^a, \sigma_z^a)$ (in m/s^2), (c) 3-D scale factor error (S_x^a, S_y^a, S_z^a) (in %). The error of acceleration measurements at time k is

$$\begin{cases} \varepsilon_x^a(k) = \Delta t \beta_x^a + \Delta t \mathcal{N} \left(0, \sigma_x^a \right) + \Delta t S_x^a a_x^b(k) / 100 \\ \varepsilon_y^a(k) = \Delta t \beta_y^a + \Delta t \mathcal{N} \left(0, \sigma_y^a \right) + \Delta t S_y^a a_y^b(k) / 100 \\ \varepsilon_z^a(k) = \Delta t \beta_z^a + \Delta t \mathcal{N} \left(0, \sigma_z^a \right) + \Delta t S_z^a a_z^b(k) / 100 \end{cases}$$
(19)

For each time k, we generate mock measurements $a^b(k)$ according to Eq. (12) and Eq. (13), and generate measurement

errors according to Eq. (19). Then, based on Eq. (18), a group of mock acceleration measurements $\{\hat{Z}^a\}$ is obtained.

The error of gyroscopes [3] also includes bias $(\beta_x^g, \beta_y^g, \beta_z^g)$ (in deg/hr), zero mean Guassian white noise with standard deviations of $(\sigma_x^g, \sigma_y^g, \sigma_z^g)$ (in deg/ \sqrt{hr}), and the scale factor error (S_x^g, S_y^g, S_z^g) (in %). Angular velocity measurements' error at time k is

$$\begin{aligned} \varepsilon_{x}^{g}(k) &= (\Delta t \beta_{x}^{g} + \sqrt{\Delta t} \mathcal{N} \left(0, \sigma_{x}^{g} \right)) \rho + \Delta t S_{x}^{g} \omega_{x}^{b}(k) / 100 \\ \varepsilon_{y}^{g}(k) &= (\Delta t \beta_{y}^{g} + \sqrt{\Delta t} \mathcal{N} \left(0, \sigma_{y}^{g} \right)) \rho + \Delta t S_{y}^{g} \omega_{y}^{b}(k) / 100 \\ \varepsilon_{z}^{g}(k) &= (\Delta t \beta_{z}^{g} + \sqrt{\Delta t} \mathcal{N} \left(0, \sigma_{z}^{g} \right)) \rho + \Delta t S_{z}^{g} \omega_{z}^{b}(k) / 100 \end{aligned}$$
(20)

where $\rho = \frac{\pi/180}{3600}$ is the coefficient for unit transformation from deg/hr to rad/s.

For each time k, we generate mock measurements $\omega^b(k)$ according to Eq. (14) and Eq. (15), and generate their error according to Eq. (20). Then, based on Eq. (18), a group of mock angular velocity measurements $\{\hat{Z}^g\}$ is obtained.

C. PE Based on Mock Measurements

Given real measurements $Z = \{Z^a, Z^g\}$ and M SINS algorithms, when M groups of mock measurements $\{\hat{Z}\} = \{\hat{Z}^a, \hat{Z}^g\}$ are generated, the mock measurements based PE is illustrated in Fig. 4.



Fig. 4. Implementation of mock measurements based PE.

Since acceleration and angular velocity have different units, the difference between Z^a and $\{\hat{Z}^a\}$ and the difference between Z^g and $\{\hat{Z}^g\}$ are calculated, respectively, to construct a pair $(\tilde{Z}^a, \tilde{Z}^g)$.

The difference between a measurement vector Z^a and a group of measurement vectors $\{\hat{Z}^a\}$ can be calculated as follows. The difference (e.g., Euclidean distance) between Z^a and each $\hat{Z}_j^a, j = 1, ..., R$ is computed, respectively, to obtain R different \tilde{Z}_j^a . The same is done for Z^g and $\{\hat{Z}^g\}$. Then we calculate the mean of the errors as

⁴At the same time, the mock measurement generation approach $\hat{h}(\cdot)$ should approximate the real measurement generation mechanism $h(\cdot)$ as closely as possible.

$$\begin{cases} \bar{\tilde{Z}}^a = \sum_{j=1}^R \tilde{Z}^a_j / R \\ \bar{\tilde{Z}}^g = \sum_{j=1}^R \tilde{Z}^g_j / R \end{cases}$$
(21)

We call this procedure "error-then-mean". Based on \tilde{Z}^a and \tilde{Z}^g , we can calculate RMSE and AEE in Eq. (10) and Eq. (11), respectively, to obtain the pairs (RMSE^{*a*}, RMSE^{*g*}) and (AEE^{*a*}, AEE^{*g*}).

We can also calculate the mean of the R generated measurements \hat{Z}_i^a (or \hat{Z}_i^g) first as

$$\begin{cases} \bar{\hat{Z}}^a = \sum_{j=1}^R \hat{Z}^a_j / R\\ \bar{\hat{Z}}^g = \sum_{j=1}^R \hat{Z}^g_j / R \end{cases}$$

and then calculate the difference between \hat{Z}^a (or \hat{Z}^g) and the real measurements Z^a (or Z^g). This is called "meanthen-error". Both "mean-then-error" and "error-then-mean" can calculate the closeness between a point and a point set. Which is better here? "Error-then-mean" is preferable. The reason is illustrated in Fig. 5.



Fig. 5. "Error-then-mean" vs. "mean-then-error"

In Fig. 5, for both cases, the real measurement is Z. In Case II, the four mock measurements (with noise) are distributed approximately symmetrically around the real measurement. If we take mean first and then calculate the difference between \tilde{Z} and Z, the difference is close to zero. In Case I, the four mock measurements (with noise) are not distributed symmetrically around the real measurement. If we take mean first and then calculate the difference is obviously larger than that in Case II. So, Case II is then deemed better. This is obviously bad, because every mock measurement in Case I is closer to the real measurement than in case II. When using "error-then-mean", there are no such problems.

V. SIMULATIONS STUDIES

If the real measurements of IMU are available, the above introduced mock measurements based PE for SINS algorithms can be directly applied. If they are not available, to apply our PE approach, we can generate simulated "real" measurements first. The "true" trajectory, attitude and other states can be preset. We can perform both the traditional PE knowing the ground truth and our proposed PE to validate the latter. Different noise levels can also be preset for a more detailed comparison. Furthermore, in simulation given the true state, the "real" measurement can be regenerated multiple times with different realizations of the noise. Therefore, PE can be implemented using the Monte Carlo method. The procedure of mock measurement based PE using Monte Carlo simulation is shown in Fig. 6.5



Fig. 6. Monte Carlo procedure for PE

A. Generation of Simulated Real Measurement

Here we use the INS simulation toolbox in Matlab [18]. The profiles of the motion can be preset according to the user's preference. The user can set the initial state, noise parameters, and sampling interval. Different types of motion models can be used, including constant velocity (CV), constant acceleration (CA), constant turn-rate (CT) with a constant height, and the transition between these types. The length of each motion stages can be set by the user. The simulated measurements for gyroscopes and accelerometers can be generated after the initialization. In the simulation, we use three noise levels given in Table 1. ⁶

B. Simulations of SINS and Mock Measurements Based PE

Three SINS algorithms are compared, including the 1st, 2nd, and 4th order approximations of the quaternion SINS method given by Eq. (6), Eq. (7) and Eq. (8), respectively. Intuitively, the performance of a higher order approximation is better.

⁵It should be noted that no Monte Carlo runs or generation of "real" measurement is needed if real measurements are available.

⁶For simplicity, the scale factor error is not considered in this paper.

 TABLE I

 Inertial sensor noise settings for simulation

Levels		Low	Medium	High	
Items	Unit	1011	mediam	mgn	
β_x^a		10	40	180	
β_y^a		15	50	200	
β_z^a		10	40	180	
σ^a_x	μg	2	8	30	
σ_y^a		3	10	30	
σ_y^a		2	8	30	
β_x^g		0.005	0.02	0.1	
β_y^g	deg/hr	0.008	0.03	0.1	
β_z^g	- /	0.005	0.02	0.1	
σ_x^g		0.002	0.004	0.03	
σ_y^g	deg / \sqrt{hr}	0.003	0.005	0.03	
σ_y^g		0.002	0.004	0.03	

For comparison of different SINS algorithms, in the motion settings, the moving object did multiple aggressive maneuvers (such as a turn with a large turn rate and speeding up with a high g). The working frequency of SINS was set as 10 Hz and the total length of simulation was 203.9 seconds. The true trajectory and the estimated trajectories using different SINS algorithms (at the high noise level) are shown in Fig. 7. A zoomed-in comparisons is given in Fig. 8. As shown in Figs. 7 and 8, the 1st order approximation departs from the truth farthest. The evaluated performance knowing the ground truth in terms of RMSE and AEE is given in Table II and Fig. 9 (for the high noise level).



Fig. 7. Ground truth and trajectories obtained using different SINS algorithms

We use the mock acceleration and angular velocity measurements and the simulated "real" counterparts to implement our PE method according to Eq. (21). The number of Monte-Carlo runs is 100. On each run, the simulated "real" measurements are regenerated with the error parameter setting given in Table I. Mock measurements at each time are realized 10 times with



Fig. 8. Zoomed in view of Fig. 7

TABLE II PE OF DIFFERENT SINS ALGORITHMS KNOWING GROUND TRUTH

Noise level	Metric	1st order	2nd order	4th order	Rank
High	RMSE	76.9617	0.2477	6.9706e-6	3-2-1
	AEE	53.6215.4884	0.1776	5.0125e-6	3-2-1
Medium	RMSE	76.9872	0.2473	6.9600e-6	3-2-1
	AEE	53.6381	0.1774	5.0055e-6	3-2-1
Low	RMSE	76.9939	0.2472	6.9575e-6	3-2-1
	AEE	53.6425	0.1773	5.0040e-6	3-2-1

the same noise parameter setting used for simulated "real" measurements.

The time average RMSE and AEE for different noise levels are listed in Tables III (for acceleration and angular velocity). The results at each time under the high level noise are illustrated in Figs. 10 (for angular velocity) and 11 (for acceleration). The large errors in Figs. 10 and 11 occur at those time steps with a large acceleration or angular velocity. For those steps with small measurement values, the difference



Fig. 9. Position error using PE knowing ground truth at the high noise level

between mock and real measurements should not be large due to their own small values.

Due to the periodicity of the angles, we should be cautious when calculating the difference between two angles. Their difference Δ must always be $-\pi < \Delta \leq \pi$.

TABLE III PE OF DIFFERENT SINS ALGORITHMS USING MOCK MEASUREMENTS AT DIFFERENT NOISE LEVELS

(a) using gyroscopes measurements

Noise level	Metric	1st order	2nd order	4th order	Rank
High	RMSE	4.9549e-4	0.6436e-4	0.6399e-4	3-2-1
	AEE	1.2822e-4	0.6258e-4	0.6228e-4	3-2-1
Medium	RMSE	4.9143e-4	0.1153e-4	0.0928e-4	3-2-1
	AEE	0.8070e-4	0.0976e-4	0.0902e-4	3-2-1
Low	RMSE	4.9138e-4	0.0851e-4	0.0505e-4	3-2-1
	AEE	0.7761e-4	0.0572e-4	0.0490e-4	3-2-1
(b) Using accelerometer measurements					

(1) 100					
Noise level	Metric	1st order	2nd order	3rd order	Rank
High	RMSE	609.06e-4	6.8087e-4	6.8080e-4	3-2-1
	AEE	424.46e-4	6.6260e-4	6.6250e-4	3-2-1
Medium	RMSE	609.2e-4	1.9754e-4	1.9742e-4	3-2-1
	AEE	423.0e-4	1.9208e-4	1.9196e-4	3-2-1
Low	RMSE	609.3e-4	0.5398e-4	0.5369e-4	3-2-1
	AEE	422.6e-4	0.5235e-4	0.5210e-4	3-2-1



Fig. 10. Mock angular velocity measurement based PE at the high noise level

Intuitively, the rank for the three SINS algorithms with the 1st, 2nd, or 4th order approximation should be "3-2-1"⁷. Such

a rank can also be obtained based on the average RMSE and AEE using our PE approach. To be more specific, the following Spearman rank distance [22]

$$d(\Lambda_S, \Lambda_i) = 1 - \frac{(\Lambda_S - \bar{\Lambda}_S)^T (\Lambda_i - \bar{\Lambda}_i)}{\sqrt{(\Lambda_S - \bar{\Lambda}_S)^T (\Lambda_S - \bar{\Lambda}_S)} \sqrt{(\Lambda_i - \bar{\Lambda}_i)^T (\Lambda_i - \bar{\Lambda}_i)}}$$
(22)

where $\bar{\Lambda}_S = \bar{\Lambda}_i = ((n+1)/2)e$, $(e = [1, 1, ..., 1]^T)$ and n is the length of a rank, can be used to check the accordance between the ranks obtained using the mock measurements based PE at each Monte-Carlo run (denoted by Λ_i , i = 1, ..., 100) and the standard rank "3-2-1" denoted by Λ_S . This distance ranges within the interval [0, 2]. d = 0 means two ranks are totally the same, and d = 2 means that they are exactly reversed.



Fig. 11. Mock acceleration measurements based PE at the high noise level

For example, for high level noise, the average rank distance at each time is shown in Fig. 12. As shown in Fig. 12, the rank of the three SINS algorithms does change at some Monte-Carlo runs. In average of the 100 Monte-Carlo runs, the ranks obtained using mock measurement PE have no contradictions with the standard rank as shown in Table III. Our proposed mock measurement based PE can work without knowing the ground truth and is always accordant with the traditional PE knowing the ground truth.

In Fig. 9, errors of the SINS navigation state space are accumulated (affected by history); however, differences in the measurement space are not significantly accumulated. This is because a state is accumulated (affected by history) while a measurement is not. In some cases, the 2nd order approximation algorithm can not be well distinguished from the 4th order algorithm. Although our PE approach is not as

⁷In a rank vector $[r_1, r_2, r_4]$, r_i denotes the rank of the *i*th order approximation, which could be anyone out of 1, 2, 3. 1 represents the best and 3 represents the worst. So 3-2-1 means that the 1st order approximation ranks third (the worst), the 2nd order approximation ranks second, and the 4th order approximation ranks first (the best).

good as the direct PE in the state space, when the ground truth is not known, it is the only way and it can at least accomplish the PE task.



Fig. 12. Rank distance at each Monte-Carlo run

VI. CONCLUSION

In this paper, a mock measurement based PE approach with an unknown ground truth for SINS algorithms is proposed. Simulation results show that our proposed PE approach is rational, which usually agrees with the PE with knowing the ground truth. Our proposed PE can be a vital alternative when knowledge of the ground truth is known. If the knowledge of the ground truth is not available, traditional PE can not be used while our proposed PE is the only choice, which can accomplish the PE task reasonably well.

Future work includes: first testing our PE approach by real data. Second, mock measurement generation has several approximations resulting in the loss of information. We will try to further analyze these approximations and their effects on PE and try to propose new methods to generate mock measurements with a high accuracy. Furthermore, our proposed PE is accomplished in the measurement space \mathcal{Z} , so more theoretical analyses of the credibility and sufficiency of our proposed PE in the measurement space are needed.

REFERENCES

- M. S. Grewal, L. R. Weill, and A. P. Andrews, *Global Positioning Systems, Inertial Navigation, and Integration.* Hoboken, New Jersey, USA: John Wiley & Sons, 2007.
- [2] P. D. Groves, Principles of GNSS, Inertial, and Multisensor Integrated Navigation Systems. London, UK: Artech House, 2013.
- [3] D. Titterton and J. L. Weston, *Strapdown Inertial Navigation Technology*, 2nd ed. Stevenage, UK: Institution of Engineering and Technology, 2004, vol. 17.

- [4] S. Hewitson, J. Wang, and A. Kearsley, "Performance evaluation of inertial navigation systems for surveying," in *Proceedings of the 6th International Symposium on Satellite Navigation Technology Including Mobile Positioning & Location Services*. Melbourne, Australia: Citeseer, July 22-25, 2003.
- [5] X. R. Li and Z. Zhao, "Evaluation of estimation algorithms part I: incomprehensive measures of performance," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 42, no. 4, pp. 1340–1358, 2006.
- [6] X. Niu, C. Goodall, S. Nassar, and N. El-Sheimy, "An efficient method for evaluating the performance of MEMS IMUs," in *Proceedings of the IEEE/ION Position, Location, And Navigation Symposium*, Coronado, CA, USA, April 25-27, 2006, pp. 766–771.
- [7] A. Kealy, G. Roberts, and G. Retscher, "Evaluating the performance of low cost MEMS inertial sensors for seamless indoor/outdoor navigation," in *Proceedings of the IEEE/ION Position Location and Navigation Symposium 2010 IEEE/ION*, Indian Wells, CA, USA, May 3-6, 2010, pp. 157–167.
- [8] A. Kealy, G. Retsche, D. Grejner-Brzezińska, V. Gikas, and G. Roberts, "Evaluating the performance of MEMS based inertial navigation sensors for land mobile applications," *Archiwum Fotogrametrii, Kartografii i Teledetekcji*, vol. 22, 2011.
- [9] Y. Wu, H. Zhang, M. Wu, X. Hu, and D. Hu, "Observability of strapdown ins alignment: a global perspective," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 48, no. 1, pp. 78–102, 2012.
 [10] Y. Wu, X. Hu, D. Hu, T. Li, and J. Lian, "Strapdown inertial navigation
- [10] Y. Wu, X. Hu, D. Hu, T. Li, and J. Lian, "Strapdown inertial navigation system algorithms based on dual quaternions," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 41, no. 1, pp. 110–132, 2005.
- [11] Y. Hao, J. Gong, W. Gao, and L. Li, "Research on the dynamic error of strapdown inertial navigation system," in *Proceedings of the IEEE International Conference on Mechatronics and Automation 2008 (ICMA* 2008), Takamatsu, Kagawa, Japan, August 5-8, 2008, pp. 814–819.
- [12] Y. Wu and X. Pan, "Velocity/position integration formula part I: application to in-flight coarse alignment," *IEEE Transactions on Aerospace* and Electronic Systems, vol. 49, no. 2, pp. 1006–1023, 2013.
- [13] —, "Velocity/position integration formula part II: application to strapdown inertial navigation computation," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 49, no. 2, pp. 1024–1034, 2013.
- [14] D. Li, R. Landry, and P. Lavoie, "Validation and performance evaluation of two different inertial navigation system design approaches," in *Proceedings of the International Global Navigation Satellite Systems Society Symposium*, Sydney, Australia, Dec 4-6, 2007, pp. 1–15.
- [15] W. Jiang, L. Wang, X. Niu, Q. Zhang, H. Zhang, M. Tang, and X. Hu, "High-precision image aided inertial navigation with known features: observability analysis and performance evaluation," *Sensors*, vol. 14, no. 10, pp. 19371–19401, 2014.
- [16] L. Wang, P. Ye, C. Zhai, and Y. Zhang, "High performance strapdown inertial navigation system algorithms for space flight," in *Proceedings of the 2nd International Symposium on Systems and Control in Aerospace and Astronautics 2008 (ISSCAA 2008)*, Shenzhen, China, Dec 10-12, 2008, pp. 1–5.
- [17] W. Zhang, M. Ghogho, and B. Yuan, "Mathematical model and Matlab simulation of strapdown inertial navigation system," *Modelling and Simulation in Engineering*, vol. 2012, pp. 1–25, 2012.
- [18] E. Shin and N. El-Sheimy. (2003) INS toolbox, a Matlab software for GPS. INS integration, Department of Geomatics Engineering, University of Calgary. [Online]. Available: http://www.geomatics.ucalgary.ca/ research/MMSensor/facilities/software/instoolboxformatLab.php
- [19] D. Wu and Z. Wang, "Strapdown navigation using geometric algebra: screw blade algorithm," *Positioning*, vol. 3, no. 02, p. 13, 2012.
- [20] X. R. Li, "Comprehensive evaluation of estimation performance without knowing ground truth," in *Proceedings of the ONR-GTRIWorkshop on Tracking and Sensor Fusion*, Monterey, CA, USA, May 2007.
- [21] Y. Bar-Shalom, X. R. Li, and K. Thiagalingam, *Estimation with Applications to Tracking and Navigation*. New York, USA: John Wiley & Sons, 2001.
- [22] J. L. Myers, A. Well, and R. F. Lorch, *Research Design and Statistical Analysis*. Hove, UK: Routledge, 2010.