

Article

Maximum Correntropy Unscented Kalman Filter for Ballistic Missile Navigation System based on SINS/CNS Deeply Integrated Mode

Bowen Hou ¹, Zhangming He ^{1,2}, Dong Li ³, Haiyin Zhou ¹ and Jiongqi Wang ^{1,2,*}

¹ Department of System Science, College of Liberal Arts and Science, National University of Defense Technology, Fuyuan Road No.1, Changsha 410072, China; houbowen12@nudt.edu.cn (B.H.); hzmnudt@sina.com (Z.H.); gfkd_zhy@sina.com (H.Z.)

² Beijing Institute of Control Engineering, China Academy of Space Technology, Beijing 100080, China

³ Unit 94, PLA 91550, Dalian 116023, China; nudtlidong@163.com

* Correspondence: wjq_gfkd@163.com; Tel.: +86-137-8719-8870

Received: 17 April 2018; Accepted: 23 May 2018; Published: 27 May 2018



Abstract: Strap-down inertial navigation system/celestial navigation system (SINS/CNS) integrated navigation is a high precision navigation technique for ballistic missiles. The traditional navigation method has a divergence in the position error. A deeply integrated mode for SINS/CNS navigation system is proposed to improve the navigation accuracy of ballistic missile. The deeply integrated navigation principle is described and the observability of the navigation system is analyzed. The nonlinearity, as well as the large outliers and the Gaussian mixture noises, often exists during the actual navigation process, leading to the divergence phenomenon of the navigation filter. The new nonlinear Kalman filter on the basis of the maximum correntropy theory and unscented transformation, named the maximum correntropy unscented Kalman filter, is deduced, and the computational complexity is analyzed. The unscented transformation is used for restricting the nonlinearity of the system equation, and the maximum correntropy theory is used to deal with the non-Gaussian noises. Finally, numerical simulation illustrates the superiority of the proposed filter compared with the traditional unscented Kalman filter. The comparison results show that the large outliers and the influence of non-Gaussian noises for SINS/CNS deeply integrated navigation is significantly reduced through the proposed filter.

Keywords: unscented Kalman filter; maximum correntropy; SINS/CNS deeply integrated navigation; non-Gaussian noises; ballistic missile

1. Introduction

With the development of missile technology, even higher accuracy of navigation is requested. Single navigation mode cannot satisfy the application requirement. Because of the uncertainty in the actual navigation system, a number of algorithms have been designed to improve the accuracy of state estimation [1]. To sum up, the navigation accuracy of ballistic missile has focused on two parts, navigation system construction and navigation filter algorithm designation.

For navigation system construction, a number of navigation modes have been designed to improve navigation accuracy. Among these modes, the strap-down inertial navigation system (SINS) has been playing an important role among navigation systems for several decades [2–4]. It can autonomously achieve a navigation mission completely relying on micro electro mechanical systems (MEMS) and has a strong anti-interference capability. Moreover, the attitude, the velocity, and the position of ballistic missiles can be easily obtained by its output information. However, the accumulated navigation errors caused by the inertial components cannot be avoided, especially in the long-term missile

navigation [5]. Therefore, in the practical situations, some other supporting facilities, including the Global Navigation Satellite System (GNSS), BeiDou Navigation Satellite System (BDS), the Doppler Velocity Log (DVL), and the celestial navigation system (CNS), are always integrated with SINS for the purpose of improving navigation accuracy [6,7]. However, GNSS is always susceptible to interference, which will have a strong impact on missile performance. DVL is always used underwater and is limited by the distance, which is not appropriate for long-distance missiles. The CNS is a navigation system that calculates the accurate attitude on the basis of measuring the azimuth of a celestial body through the celestial sensor. Because the source of information in the CNS is a celestial body, it has many merits, such as good concealment, strong autonomy, and high navigation accuracy, but the output frequency is much lower and discontinuous, and the output information can be easily influenced by surroundings [8–10]. Hence, the CNS is frequently used for the purpose of assisting the SINS in the aerospace field, utilizing the SINS/CNS integrated navigation system. The SINS/CNS integrated navigation system can correct the attitude error and the gyro drift using the starlight information and greatly improve navigation accuracy [11]. At present, it is an important developing direction for missile, airplane, and spacecraft navigation technology [12]. Nevertheless, the traditional system only makes use of the information of star sensors to reckon the gyro drift, but it cannot estimate the accelerometer bias, which leads to position divergence. Therefore, some improved SINS/CNS integrated navigation systems have been proposed [4,13,14]. They make use of more astronomical observation information to realize SINS/CNS deeply integrated navigation systems and effectively improve their performance.

For navigation filter design, many algorithms have been applied to the navigation system. The Kalman filter (KF) is the most popular optimal state estimation method used for the linear dynamic system [15]. However, for navigation systems, the navigation system equation is always nonlinear [16]. The frequently used nonlinear algorithms for solving nonlinear problems mainly include the extended Kalman filter (EKF) [17] and the unscented Kalman filter (UKF) [18]. EKF is a popular nonlinear extension of the KF, which uses Taylor series expansions to approximate the nonlinear system. However, the Taylor series expansions in many navigation systems are cumbersome and always lead to implementation difficulties [15]. The iterated extended Kalman filter (IEKF) overcomes the problem and improves the accuracy with the Gauss-Newton method [19]. Furthermore, the nonlinear iterated filter (NIF) improves the stability of the Jacobian matrix in the EKF and the IEKF through the Levenberg–Marquardt algorithm [20]. In the UKF, a set of selected sigma points are used to approximate the probability distribution function of the state and propagate through the nonlinear process and the measurement equation. Thus, the UKF saves the trouble of calculating the Taylor series is more accurate than the EKF. The posterior linearization filter (PLF), using statistical linear regression, is a newly proposed algorithm to improve the filter accuracy based on sigma-point approximations [21]. However, the filters above are always used to deal with Gaussian white noise. Nevertheless, owing to the operational error or the complex atmospheric environment, the non-Gaussian noises, such as large outliers and Gaussian mixture noises, often exist during the actual navigation process.

For non-Gaussian process, many nonlinear filters do not perform satisfactorily [22]. However, the filter robustness can be improved through the approaches as follows:

1. A classical method involves creating filters directly for the systems under the conditions of non-Gaussian noises to deal with noises with different heavy-tailed distributions [23,24]. However, it is difficult to take effect in multidimensional situations, which limits its applicability [25].
2. There are two efficient methods to handle non-Gaussian noises based on estimating the a posteriori probability density with a series of samples. The UKF approximates the mean and the covariance of the state estimation using unscented transformation (UT) through sigma points [26]. The ensemble Kalman filter (EnKF) is another algorithm to capture state estimation using samples set for the purpose of handling non-Gaussian noises [27].

3. Another popular method is the Huber-based Kalman filter. It is a robust state estimator to handle the non-Gaussian noises based on the robust estimation theory [28]. Many navigation and target problems were often handled through it under the conditions of non-Gaussian noises in the past few years [29–32].
4. Chang [33] proposed a new robust Kalman filter in recent years. He defined a judging index calculating the difference between the observation and the prediction based on Mahalanobis distance to solve the outliers, which can also successfully solve the observation noises that obey the thick-tailed distribution during the actual observation process.

Maximum correntropy criterion (MCC) is an information theoretic method proposed in recent years. It has been applied to the Bayesian estimation and the error analysis [34,35]. The maximum correntropy Kalman filter (MCKF), which is an improved Kalman filter algorithm using the MCC theory, is one of the latest designed estimators for the purpose of solving non-Gaussian noises. In addition, the maximum correntropy extended Kalman filter (MCEKF) and the maximum correntropy unscented Kalman filter (MCUKF) are proposed for the nonlinear system [36], and the MCUKF has been proven to show a better performance than MCEKF [37]. However, its robustness is not analyzed and it has not been applied to SINS/CNS deeply integrated navigation.

The key contributions of this paper are expressed as follows. First of all, the observability of the new deeply integrated navigation system is analyzed. The MCUKF is then deduced on the basis of maximum correntropy theory and unscented transformation to deal with the nonlinearity and non-Gaussian noises in the navigation system. The computational complexity of it is also analyzed. Furthermore, the robustness of the MCUKF is discussed under the condition of the non-Gaussian noises. Finally, the conditions of non-Gaussian noise, mainly including the large outliers and the Gaussian mixture noises, existing in SINS/CNS deeply integrated navigation are effectively handled by the MCUKF.

The organization of this paper proceeds as follows. SINS/CNS deeply integrated navigation is briefly reviewed and the observability is analyzed in Section 1. In Section 2, the algorithm steps are designed for the MCUKF and the robustness performance is analyzed compared with a Huber-based filter. In Section 3, simulations are demonstrated to compare the performances using several filter methods under the conditions of different kinds of noises. Finally, we draw conclusions in Section 4.

2. SINS/CNS Deeply Integrated Navigation System

2.1. Integrated Navigation Principle

The traditional SINS/CNS integrated navigation system outputs the attitude, the velocity, and the position information of a missile through the IMU (including a gyro and an accelerometer). The star sensor of the CNS outputs the attitude information of the missile after star capturing and star identification. The attitude information calculated by the star sensor and the gyro is then integrated, and the integrated information is served as the measurement equation. The optimal estimation methods are then used to estimate the gyro drift to correct the gyro. Figure 1 shows the traditional SINS/CNS integrated navigation principle.

Although SINS/CNS integrated navigation system can correct the gyro drift to some extent, the error of the state variables still has a divergence [4,13,14,16]. SINS/CNS deeply integrated navigation can solve the problem. Based on the traditional integrated navigation system, the height, the pitching angle, and the azimuth angle are added as the measurement information according to the altimeter and the astronomical information. In fact, these three new measurements are the function of the position of the navigation target. Therefore, the estimation of the gyro drift and the accelerometer bias can be gained to correct the acceleration information of SINS. Figure 2 shows the navigation principle of the SINS/CNS deeply integrated mode.

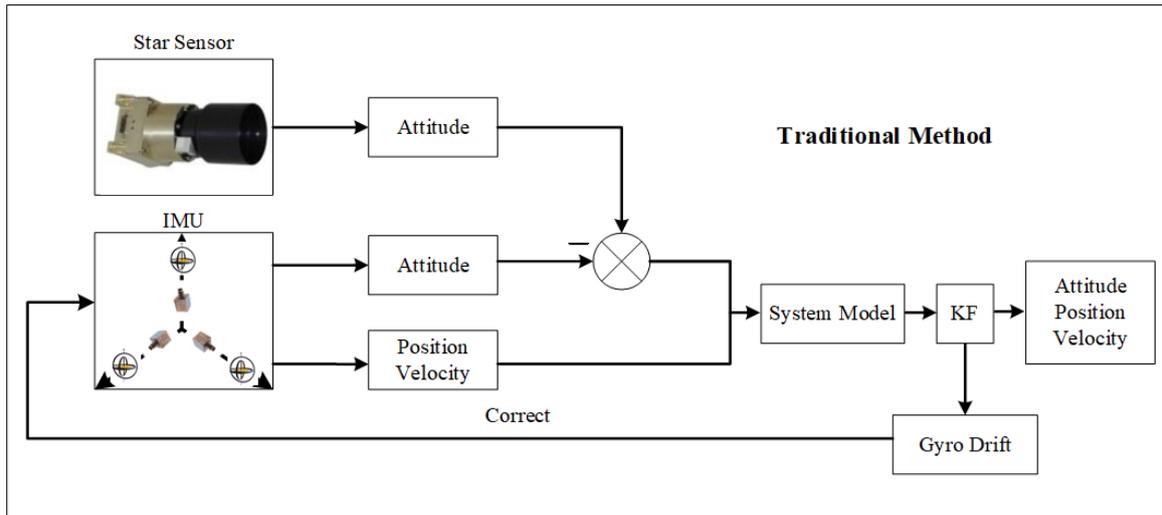


Figure 1. Traditional strap-down inertial navigation system/celestial navigation system (SINS/CNS) integrated navigation principle.

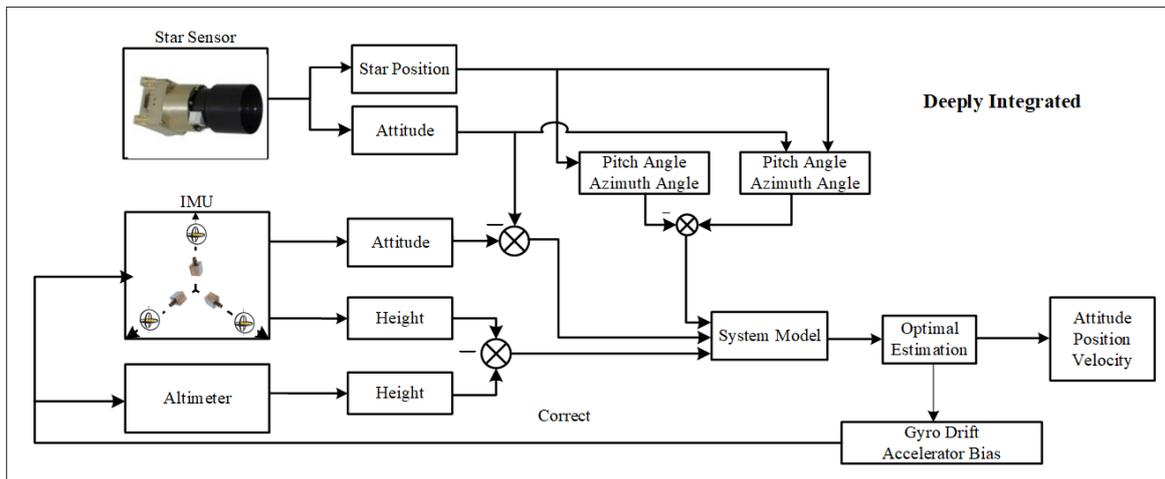


Figure 2. SINS/CNS deeply integrated navigation principle.

The deeply integrated SINS/CNS integrated navigation system has not yet been realized in real scenarios. When it is applied on the missile, the star sensor measures not only the attitude information but also the partial position information of the missile. The attitude can be calculated through star identification. The position can be calculated by measuring the height angle and the azimuth angle of the star. An altimeter needs to be installed on the system to measure the height information of the missile. The altimeter can be a radar altimeter [38], a pressure altimeter [39], or a laser altimeter [40], decided according to the specific case. Combined with the position information from the star sensor, the position of the missile can be obtained. In this way, the deeply SINS/CNS integrated navigation system can be realized and applied to practical situations. The detailed principle of the system is introduced in the next section.

2.2. System Model

2.2.1. State Equation

Suppose that the navigation frame is the launch point frame (l-frame) and the SINS is strapdown-installed along the three axes of the missile. Then the state equation is identical to the traditional algorithm:

$$\dot{\mathbf{X}}(t) = \mathbf{F}(t) \mathbf{X}(t) + \mathbf{G}(t) \mathbf{W}(t), \quad (1)$$

where $\mathbf{X}(t) = \left[\boldsymbol{\Psi}(t) \ \delta \mathbf{v}(t) \ \delta \mathbf{r}_l(t) \ \boldsymbol{\varepsilon}(t) \ \nabla(t) \right]^T$ denotes the system state, $\boldsymbol{\Psi}(t) = \left[\phi_x(t) \ \phi_y(t) \ \phi_z(t) \right]$ denotes the platform angles error (the navigation frame misalignment angle) in the l-frame; $\delta \mathbf{v}(t) = \left[\delta v_{xl}(t) \ \delta v_{yl}(t) \ \delta v_{zl}(t) \right]$ denotes the velocity error in the l-frame; $\delta \mathbf{r}_l(t) = \left[\delta x_l(t) \ \delta y_l(t) \ \delta z_l(t) \right]$ denotes the position error in the l-frame; $\boldsymbol{\varepsilon}(t) = \left[\varepsilon_x(t) \ \varepsilon_y(t) \ \varepsilon_z(t) \right]$ denotes the gyro constant drift; $\nabla(t) = \left[\nabla_x(t) \ \nabla_y(t) \ \nabla_z(t) \right]$ denotes the accelerometer constant bias; $\mathbf{W}(t) = \left[\varepsilon_{sx}(t) \ \varepsilon_{sy}(t) \ \varepsilon_{sz}(t) \ \nabla_{sx}(t) \ \nabla_{sy}(t) \ \nabla_{sz}(t) \right]^T$ denotes the process noise vector, including the random noises of the gyro and the accelerometer; $\mathbf{G}(t)$ denotes the process noise input matrix; $\mathbf{F}(t)$ denotes the process input matrix. The detailed information of $\mathbf{F}(t)$ and $\mathbf{G}(t)$ can be found in [41]—Equations (6-5) and (6-6), respectively. The solution of the differential Equation (1) for the discrete mode can be expressed as:

$$\mathbf{X}(k) = f(\mathbf{X}(k-1), \mathbf{W}(k-1)), \quad (2)$$

Equation (2) can be obtained through the Laplace transformation which is introduced in Chapter 2 of Reference [41]. In simulation, the equation is solved through Runge-Kutta algorithm.

And the linear version for Equation (2), which is used for MCEKF [36] can be expressed as:

$$\mathbf{X}(k) = \boldsymbol{\Phi}(k-1) \mathbf{X}(k-1) + \boldsymbol{\Gamma}(k-1) \mathbf{W}(k-1), \quad (3)$$

where k denotes the epoch, $\boldsymbol{\Phi}(k-1)$ and $\boldsymbol{\Gamma}(k-1)$ are the discrete versions of $f(\mathbf{X}(k-1))$ and $\mathbf{G}(k-1)$, respectively, which can be expressed as:

$$\begin{aligned} \boldsymbol{\Phi}(k-1) &= \mathbf{I} + \mathbf{F}(k-1)T + \frac{1}{2!} \mathbf{F}^2(k-1)T^2 \\ \boldsymbol{\Gamma}(k-1) &= T \left(\mathbf{I} + \frac{1}{2!} \mathbf{F}(k-1)T + \frac{1}{3!} \mathbf{F}^2(k-1)T^2 \right) \mathbf{G}(k). \end{aligned} \quad (4)$$

2.2.2. Measurement Equation

(1) Measurement of Attitude Error

Suppose that the attitude output of the gyro is $\boldsymbol{\theta}_g(k) = \left[\theta_g(k) \ \varphi_g(k) \ \psi_g(k) \right]^T$, and the attitude output of the star sensor is $\boldsymbol{\theta}_s(k) = \left[\theta_s(k) \ \varphi_s(k) \ \psi_s(k) \right]^T$. The three angles are equivalent to the three Euler angles, denoting roll angle, pitch angle, and yaw angle, respectively. The difference between the two sets of attitude angles can be denoted as $\delta \boldsymbol{\theta}(k) = \boldsymbol{\theta}_g(k) - \boldsymbol{\theta}_s(k)$, which can be transformed into the error-angles of the platform and can be expressed as the function of the state $\boldsymbol{\Psi}(k)$. Denote $\mathbf{Z}_1(k) = \delta \boldsymbol{\theta}(k)$ as the attitude error measurement, which can be expressed as:

$$\mathbf{Z}_1(k) = \mathbf{M}(k) \mathbf{X}(k) + \mathbf{V}_1(k), \quad (5)$$

where $M(k) = \begin{bmatrix} M_0(k) & \mathbf{O}_{3 \times 12} \end{bmatrix}$, and $M_0(k)$ denotes the attitude error transition matrix, which can be expressed as:

$$M_0(k) = \begin{bmatrix} 0 & \cos \varphi(k) & -\cos \psi(k) \sin \varphi(k) \\ 0 & \sin \varphi(k) & \cos \psi(k) \cos \varphi(k) \\ 1 & 0 & \sin \psi(k) \end{bmatrix}, \quad (6)$$

where $\varphi(k)$ denotes the pitch angle, $\psi(k)$ denotes the yaw angle, and $V_1(k)$ represents the measurement noises of the star sensor. The detailed derivation process of $M_0(k)$ can be obtained in [42].

(2) Measurement of Position Error

The measurement information relies on the azimuth angle, the pitch angle of the guide star, and the outputs of the altimeter. The measurement equation of the position error is derived in [4]. The measurement equation is expressed as:

$$\begin{bmatrix} \delta P(k) \\ \delta A(k) \\ \delta h(k) \end{bmatrix} = M_1(k) \left(C_i^l C_e^i(k) M_2(k) \right)^{-1} \begin{bmatrix} \delta x_l(k) \\ \delta y_l(k) \\ \delta z_l(k) \end{bmatrix} + \begin{bmatrix} v_P(k) \\ v_A(k) \\ v_h(k) \end{bmatrix}, \quad (7)$$

where $\delta P(k)$ is the pitch angle difference between the measurement and the computation value; $\delta A(k)$ is the azimuth angle difference between the measurement and the computation value; $\delta h(k)$ is the difference between the height measurement of the altimeter and the computation value; C_i^l stands for the rotation matrix from the l-frame (the launching point coordinate frame which is also the navigation frame) to the i-frame (the geocentric inertial coordinate frame), which is a constant matrix and can be expressed as:

$$C_i^l = \begin{bmatrix} -\cos \zeta \sin \gamma_0 \cos \phi_0 - \sin \zeta \sin \phi_0 & \cos \gamma_0 \cos \phi_0 & \sin \zeta \sin \gamma_0 \cos \phi_0 - \cos \zeta \sin \phi_0 \\ -\cos \zeta \sin \gamma_0 \sin \phi_0 + \sin \zeta \cos \phi_0 & \cos \gamma_0 \sin \phi_0 & \sin \zeta \sin \gamma_0 \sin \phi_0 + \cos \zeta \cos \phi_0 \\ \cos \zeta \cos \gamma_0 & \sin \gamma_0 & -\sin \zeta \cos \gamma_0 \end{bmatrix}, \quad (8)$$

where ζ denotes the launching angle of missile, ϕ_0 and γ_0 denote the longitude and the latitude of the launching point, respectively, $C_e^i(k)$ denotes the rotation matrix from the i-frame to the e-frame (the earth-based frame), which can be expressed as:

$$C_e^i(k) = \begin{bmatrix} \cos \omega_e t(k) & \sin \omega_e t(k) & 0 \\ -\sin \omega_e t(k) & \cos \omega_e t(k) & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad (9)$$

where ω_e is the earth's rotation angular velocity, $t(k)$ denotes the time in the k-th epoch. $\delta r_l(t) = \begin{bmatrix} \delta x_l(k) & \delta y_l(k) & \delta z_l(k) \end{bmatrix}$ denotes the position error in three directions; $v_P(k)$ and $v_A(k)$ are the observation noises of the pitch angle and the azimuth angle, respectively; $v_h(k)$ is the measurement noise of the altimeter; $M_1(k)$ and $M_2(k)$ are as follows:

$$M_1(k) = \begin{bmatrix} -\cos A(k) & -\sin A(k) \cos \varphi_b(k) & 0 \\ -\tan P(k) \sin A(k) & \tan P(k) \cos A(k) \cos \varphi_b(k) - \sin \varphi_b(k) & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad (10)$$

$$M_2(k) = \begin{bmatrix} -(h(k) + R_e) \sin \varphi_b(k) \cos \lambda_b(k) & -(h(k) + R_e) \cos \varphi_b(k) \sin \lambda_b(k) & \cos \varphi_b(k) \cos \lambda_b(k) \\ -(h(k) + R_e) \sin \varphi_b(k) \sin \lambda_b(k) & (h(k) + R_e) \cos \varphi_b(k) \cos \lambda_b(k) & \cos \varphi_b(k) \sin \lambda_b(k) \\ (h(k) + R_e) \cos \varphi_b(k) & 0 & \sin \varphi_b(k) \end{bmatrix}, \quad (11)$$

where $P(k)$ is the pitch angle of the guide star, $A(k)$ is the azimuth angle of the guide star, $h(k)$ is the height of the missile, R_e is the earth radius, $\varphi_b(k)$ is the latitude of the missile, and $\lambda_b(k)$ is the longitude of the missile. Denote $Z_2(k) = \begin{bmatrix} \delta P(k) & \delta A(k) & \delta h(k) \end{bmatrix}^T$, that is,

$$Z_2(k) = T(k) \delta r_l(k) + V_2(k), \quad (12)$$

where $T(k) = M_1(k)(C_i^1 C_e^i(k) M_2(k))^{-1}$, and $V_2(k) = [v_p(k) \ v_A(k) \ v_h(k)]^T$.

Combined with Equations (5) and (12), the total measurement equation can be expressed:

$$Z(k) = \begin{bmatrix} Z_1(k) \\ Z_2(k) \end{bmatrix} = H(k)X(k) + V(k), \quad (13)$$

where the measurement matrix $H(k)$ is:

$$H(k) = \begin{bmatrix} M_0(k) & O_{3 \times 3} & O_{3 \times 3} & O_{3 \times 6} \\ O_{3 \times 3} & O_{3 \times 3} & T(k) & O_{3 \times 6} \end{bmatrix}, \quad (14)$$

and O is the zero matrix. The measurement noises $V(k)$ are expressed as:

$$V(k) = [V_1(k) \ V_2(k)]^T, \quad (15)$$

and it is independent of the state noises $W(k)$ in Equation (3). Compared with the traditional mode, the deeply integrated mode adds the dimension of the measurements. In this way, the computational complexity will increase, which is analyzed in Section 3.3.

In many cases, $V(k)$ denotes the Gaussian white noises [4,13,14,43–45]. However, in practice, due to the sensor faults or the complex environment, the measurement noises may be contaminated noises with different distributions or include large outliers. It can be shown as follows:

$$V(k) = V_k^{(1)} + V_k^{(2)} \quad (16)$$

or

$$V(k) = V_k^{(1)} + \Delta V_k \quad (17)$$

where $V_k^{(1)}$ denotes the Gaussian white noises, $V_k^{(2)}$ is distributed differently from $V_k^{(1)}$, and ΔV_k is the outlier.

The non-Gaussian noises can influence the performance of the navigation system; therefore, it is necessary to use the robust filter to solve the problem. The maximum correntropy Kalman filter is newly designed to effectively tackle the non-Gaussian noise.

2.2.3. Observability Analysis

The observability analysis is needed to determine the efficiency of the Kalman filter designed for the purpose of estimating the system state variable [46]. If the system observability is poor, the estimation of the state will be inaccurate. Therefore, it is necessary to analyze the observability of the new integrated navigation system.

Considering the state Equation (3) and the measurement Equation (13), according to [47], the observability matrix U_1 of the deeply integrated navigation system is as follows:

$$U_1 = \begin{bmatrix} H \\ H\Phi \\ \vdots \\ H\Phi^{n_x-1} \end{bmatrix} \quad (18)$$

where H is the measurement matrix of the deeply integrated method, Φ denotes the discrete version of the state transition matrix, n_x is the dimension of state variable, which is 15 in the deeply integrated navigation system, as described in Equation (1), so $\text{rank}(U_1) = n_x = 15$. Hence, the SINS/CNS deeply

integrated navigation mode is observable. For comparison, the observability matrix of the traditional navigation mode \mathbf{U}_2 is analyzed as follows:

$$\mathbf{U}_2 = \begin{bmatrix} \mathbf{M} \\ \mathbf{M}\Phi \\ \vdots \\ \mathbf{M}\Phi^{n_x-1} \end{bmatrix}, \quad (19)$$

where \mathbf{M} is the measurement matrix of the traditional method. Thus, $\text{rank}(\mathbf{U}_2) = 6 < n_x$. Therefore, the traditional integrated navigation system is unobservable. In other words, the new navigation system, compared with the traditional one, will have a more precise estimation.

3. The Maximum Correntropy Unscented Kalman Filter

3.1. Maximum Correntropy Criterion

Correntropy [48] denotes the measure of an information entropy field, which defines the novel metric in the sample space, using the information available in the higher-order statistics of the signals. Given two random variables with the joint probability density function (PDF) $p(a, b)$, the definition of correntropy is expressed as follows.

Definition 1. Given two random variables \mathbf{A} and \mathbf{B} , the correntropy is defined as:

$$C_\lambda(\mathbf{A}, \mathbf{B}) = \mathbb{E}[\kappa(\mathbf{A}, \mathbf{B})] = \iint \kappa_\lambda(a, b) p(a, b) da db, \quad (20)$$

where \mathbb{E} denotes the expectation, $\kappa_\lambda(\cdot, \cdot)$ denotes a real-valued continuous, symmetric, and nonnegative definite kernel function $\lambda > 0$ denotes the kernel bandwidth.

The Gaussian kernel, which is expressed as follows, is used as the kernel function in Definition 1:

$$\kappa_\lambda(a, b) = G_\lambda(e) = \exp\left(-\frac{e^2}{2\lambda^2}\right), \quad (21)$$

where $e = a - b$.

In many practical cases, there are a finite amount of samples to estimate the unknown joint PDF $p(a, b)$. Hence, the sample mean estimator can calculate the correntropy as follows:

$$\hat{C}_\lambda(\mathbf{A}, \mathbf{B}) = \frac{1}{N} \sum_{i=1}^N G_\lambda(e_i), \quad (22)$$

where

$$e_i = a_i - b_i, \quad (23)$$

$\{a_i, b_i\}_{i=1}^N$ denotes N samples, which are drawn from $p(a, b)$.

Assuming $\lambda = 0.5$, $a_i \in [0, 5]$, $b_i \in [0, 5]$, and $p(a, b) = \frac{1}{2}e^{-(a^2 + b^2)}$. The current correntropy value is depicted in Figure 3.

According to Figure 3, correntropy is considered a correlation scale in the joint space, which reaches maximum under the conditions that $a(i) = b(i)$. Based on this, the cost function can be defined to represent the maximum correntropy criterion (MCC) as follows:

$$J_{MCC} = \max \sum_{i=1}^N G_\lambda(e_i). \quad (24)$$

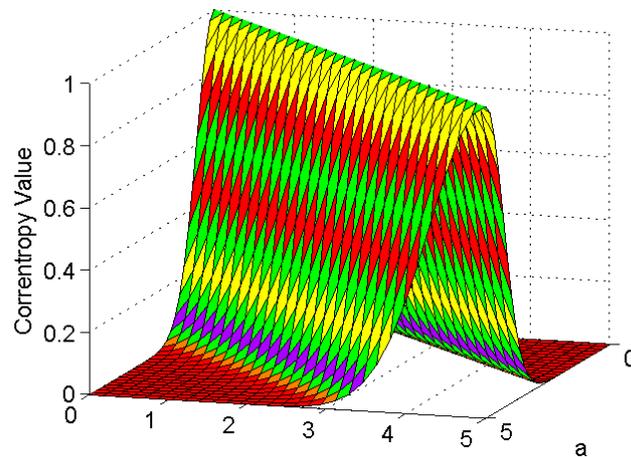


Figure 3. Correntropy when $\lambda = 1$.

From Figure 3, it is manifest that MCC represents a local similarity criterion that reflects the maximum error probability density from the view of probabilistic meaning [48]. Non-Gaussian noises, such as large outliers and Gaussian mixture noises, can be handled effectively on the basis of the property [49]. That is to say, MCC reaches the maximum value on the joint space’s bisector because the value of Gaussian kernel maximizes on the line $A = B$, which is different from the mean square error (MSE) estimation [50]. Based on this, MCC has been used for parameter estimation in recent years [34,35,51].

3.2. The Maximum Correntropy Unscented Kalman Filter

The MCKF is a novel algorithm combined with a UKF framework based on the MCC, and can perform better for nonlinear systems in non-Gaussian noise environments [37].

Consider the state equation and the measurement equation described in Equations (2) and (13) in Section 2:

$$\begin{cases} \mathbf{X}(k) = f(\mathbf{X}(k-1), \mathbf{W}(k-1)) \\ \mathbf{Z}(k) = \mathbf{H}(k)\mathbf{X}(k) + \mathbf{V}(k), \end{cases} \quad (25)$$

$\mathbf{W}(k-1)$ and $\mathbf{V}(k)$ are mutually uncorrelated process noises and measurement noises, respectively. The mean of both noises are zero and the covariances of them are expressed as:

$$\mathbb{E}[\mathbf{W}(k-1)\mathbf{W}^T(k-1)] = \mathbf{Q}(k-1), \quad \mathbb{E}[\mathbf{V}(k)\mathbf{V}^T(k)] = \mathbf{R}(k). \quad (26)$$

Similar to the Kalman filter, the MCKF includes the time update and the measurement update.

3.2.1. Time Update

The sigma points set are generated from the estimated state $\hat{\mathbf{X}}(k-1|k-1)$ and the covariance matrix $\mathbf{P}(k-1|k-1)$ at the previous step $(k-1)$.

$$\begin{aligned} \chi^{(0)}(k-1|k-1) &= \hat{\mathbf{X}}(k-1|k-1) \\ \chi^{(i)}(k-1|k-1) &= \hat{\mathbf{X}}(k-1|k-1) + \left(\sqrt{(n_x + \lambda)\mathbf{P}(k-1|k-1)} \right)_i, \quad i = 1, 2, \dots, n_x \\ \chi^{(i)}(k-1|k-1) &= \hat{\mathbf{X}}(k-1|k-1) - \left(\sqrt{(n_x + \lambda)\mathbf{P}(k-1|k-1)} \right)_{i-n}, \quad i = n_x + 1, n_x + 2, \dots, 2n_x, \end{aligned} \quad (27)$$

where n_x denotes the dimension number of the state, $\left(\sqrt{(n_x + \lambda) \mathbf{P}(k-1|k-1)}\right)_i$ denotes the i -th column of $\sqrt{(n_x + \lambda) \mathbf{P}(k-1|k-1)}$, and the composite scaling factor λ is expressed as:

$$\lambda = \alpha^2 (n_x + \kappa) - n_x, \quad (28)$$

where κ is a parameter that is set to $(3 - n_x)$ [37], and α controls the distribution conditions of the sigma points.

The unscented transformed points are given as:

$$\chi^{(i)}(k|k-1) = f\left(\chi^{(i)}(k-1|k-1), \mathbf{W}(k-1)\right), i = 0, 1, \dots, 2n_x. \quad (29)$$

The predicted state and the covariance matrix are estimated as:

$$\hat{\mathbf{X}}(k|k-1) = \sum_{i=0}^{2n_x} \omega_c^{(i)} \chi^{(i)}(k|k-1) \quad (30)$$

$$\mathbf{P}(k|k-1) = \sum_{i=0}^{2n_x} \omega_c^{(i)} \left[\hat{\mathbf{X}}(k|k-1) - \chi^{(i)}(k|k-1) \right] \left[\hat{\mathbf{X}}(k|k-1) - \chi^{(i)}(k|k-1) \right]^T + \mathbf{Q}(k-1) \quad (31)$$

where the weights of the sigma points are as follows:

$$\begin{cases} \omega_m^{(0)} = \frac{\lambda}{n + \lambda} \\ \omega_c^{(0)} = \frac{\lambda}{n + \lambda} + (1 - \alpha^2 + \beta) \\ \omega_m^{(i)} = \omega_c^{(i)} = \frac{\lambda}{2(n + \lambda)}, i = 1, 2, \dots, 2n_x, \end{cases} \quad (32)$$

where α is expressed in Equation (28) and β is a weighted parameter that is non-negative. When the measurement data are not available, $\hat{\mathbf{X}}(k|k-1)$ can replace the current state estimation.

The sigma points of the prior state mean and the predicted covariance are set as:

$$\begin{aligned} \chi^{(0)}(k|k-1) &= \hat{\mathbf{X}}(k|k-1) \\ \chi^{(i)}(k|k-1) &= \hat{\mathbf{X}}(k|k-1) + \left(\sqrt{(n_x + \lambda) \mathbf{P}(k|k-1)}\right)_i, i = 1, 2, \dots, n_x \\ \chi^{(i)}(k|k-1) &= \hat{\mathbf{X}}(k|k-1) - \left(\sqrt{(n_x + \lambda) \mathbf{P}(k|k-1)}\right)_{i-n}, i = n_x + 1, n_x + 2, \dots, 2n_x. \end{aligned} \quad (33)$$

The sigma points are then transferred using the measurement equation as:

$$\mathbf{Z}^{(i)}(k|k-1) = \mathbf{H}(k) \chi^{(i)}(k|k-1), i = 0, 1, \dots, 2n_x. \quad (34)$$

The predicted measurement mean can be estimated as:

$$\hat{\mathbf{Z}}(k|k-1) = \sum_{i=1}^{2n_x} \omega_m^i \mathbf{Z}^{(i)}(k|k-1). \quad (35)$$

3.2.2. Measurement Update

Next, the measurement update is calculated based on the MCC. For the model described in the previous section, we have

$$\begin{bmatrix} \hat{\mathbf{X}}(k|k-1) \\ \mathbf{Z}(k) \end{bmatrix} = \begin{bmatrix} \mathbf{I} \\ \mathbf{H}(k) \end{bmatrix} \mathbf{X}(k) + \mathbf{u}(k), \quad (36)$$

where \mathbf{I} denotes the unit matrix, and $\mathbf{u}(k) = \begin{bmatrix} \hat{\mathbf{X}}(k|k-1) - \mathbf{X}(k) \\ \mathbf{V}(k) \end{bmatrix}$, with

$$\begin{aligned} & \mathbb{E} [\mathbf{u}(k) \mathbf{u}^T(k)] \\ &= \begin{bmatrix} \mathbf{P}(k|k-1) & \mathbf{O} \\ \mathbf{O} & \mathbf{R}(k) \end{bmatrix} \\ &= \begin{bmatrix} \mathbf{B}_p(k|k-1) \mathbf{B}_p^T(k|k-1) & \mathbf{O} \\ \mathbf{O} & \mathbf{B}_r(k) \mathbf{B}_r^T(k) \end{bmatrix} \\ &= \mathbf{B}(k) \mathbf{B}^T(k), \end{aligned} \quad (37)$$

where $\mathbf{B}(k)$, $\mathbf{B}_p(k|k-1)$, and $\mathbf{B}_r(k)$ is calculated on the basis of Cholesky decomposition of $\mathbb{E} [\mathbf{u}(k) \mathbf{u}^T(k)]$, the predicted covariance $\mathbf{P}(k|k-1)$, and the covariance of the measurement noises $\mathbf{R}(k)$, respectively. In fact, \mathbf{u} denotes a random variable matrix, which consists of the prior estimation error of the state and the measurements prediction error. Equation (36) is multiplied by $\mathbf{B}^{-1}(k)$ on the left of both sides, and we get

$$\mathbf{D}(k) = \mathbf{w}(k) \mathbf{X}(k) + \mathbf{e}(k), \quad (38)$$

where $\mathbf{D}(k) = \mathbf{B}^{-1}(k) \begin{bmatrix} \hat{\mathbf{X}}(k|k-1) \\ \mathbf{Z}(k) \end{bmatrix}$, $\mathbf{w}(k) = \mathbf{B}^{-1}(k) \begin{bmatrix} \mathbf{I} \\ \mathbf{H}(k) \end{bmatrix}$, and $\mathbf{e}(k) = \mathbf{B}^{-1}(k) \mathbf{u}(k)$. In this way, $\mathbf{e}(k)$ denotes a new random variable matrix, which is just multiplied by a matrix \mathbf{B}^{-1} .

Now the MCC for Equation (24) based on the cost function can be rewritten as follows:

$$J_{MCC}(\mathbf{X}(k)) = \frac{1}{N} \sum_{i=1}^N G(\mathbf{e}_i(k)) = \frac{1}{N} \sum_{i=1}^N G(d_i(k) - \mathbf{w}_i(k) \mathbf{X}(k)), \quad (39)$$

where $\mathbf{e}_i(k)$ denotes the i -th unit of $\mathbf{e}(k)$, $d_i(k)$ denotes the i -th unit of $\mathbf{D}(k)$, $\mathbf{w}_i(k)$ denotes the i -th row of $\mathbf{w}(k)$, and $N = n_x + n_z$ denotes the entire dimension of $\mathbf{D}(k)$, which represents the sum of the dimension of state variable $\mathbf{X}(k)$ and the measurement $\mathbf{Z}(k)$. In fact, $d_i(k)$ is equivalent to a_i in Equation (23), and $\mathbf{w}_i(k) \mathbf{X}(k)$ is equivalent to b_i in Equation (23). That is, $\mathbf{e}_i(k)$ is equivalent to e_i in Equations (22) and (23), and $J_{MCC}(\mathbf{X}(k))$ is equivalent to $\hat{C}_\lambda(\mathbf{A}, \mathbf{B})$ in Equation (22).

In particular, according to Figure 3, when the error $\mathbf{e}_i(k)$ reaches the minimum, the correntropy reaches its maximum value. In other words, the optimal estimation of the state can minimize the error and maximize the correntropy as well. In this way, the optimal estimate of $\mathbf{X}(k)$ is considered with respect to MCC.

Then, the optimal estimate under the MCC criterion of $\mathbf{X}(k)$ is

$$\hat{\mathbf{X}}(k) = \arg \max_{\mathbf{X}} J_{MCC}(\mathbf{X}(k)) = \arg \max_{\mathbf{X}(k)} \sum_{i=1}^N G_\lambda(\mathbf{e}_i(k)). \quad (40)$$

The optimal solution can thus be obtained by solving

$$\frac{\partial J_{MCC}(\mathbf{X}(k))}{\partial \mathbf{X}(k)} = 0. \quad (41)$$

It follows that

$$\mathbf{X}(k) = \left(\sum_{i=1}^N [\mathbf{G}_\lambda(\mathbf{e}_i(k)) \mathbf{w}_i^T(k) \mathbf{w}_i(k)] \right)^{-1} \times \left(\sum_{i=1}^N [\mathbf{G}_\lambda(\mathbf{e}_i(k)) \mathbf{w}_i^T(k) \mathbf{d}_i(k)] \right). \quad (42)$$

The optimal solution expressed by Equation (42) is actually a fixed-point equation of $\mathbf{X}(k)$ and can be rewritten as:

$$\mathbf{X}(k) = g(\mathbf{X}(k)), \quad (43)$$

with

$$g(\mathbf{X}(k)) = \left(\sum_{i=1}^N [\mathbf{G}_\lambda(\mathbf{d}_i(k) - \mathbf{w}_i(k) \mathbf{X}(k)) \mathbf{w}_i^T(k) \mathbf{w}_i(k)] \right)^{-1} \times \left(\sum_{i=1}^N [\mathbf{G}_\lambda(\mathbf{d}_i(k) - \mathbf{w}_i(k) \mathbf{X}(k)) \mathbf{w}_i^T(k) \mathbf{d}_i(k)] \right).$$

A fixed-point iterative algorithm can be readily obtained as [51]:

$$\hat{\mathbf{X}}(k)_{t^*+1} = g(\hat{\mathbf{X}}(k)_{t^*}) \quad (44)$$

where $\hat{\mathbf{X}}(k)_{t^*}$ denotes the solution at the fixed-point iteration t^* .

The fixed-point Equation (42) can also be expressed in matrix form as:

$$\mathbf{X}(k) = \left(\mathbf{w}^T(k) \mathbf{C}(k) \mathbf{w}(k) \right)^{-1} \mathbf{w}^T(k) \mathbf{C}(k) \mathbf{D}(k), \quad (45)$$

where $\mathbf{C}(k) = \begin{bmatrix} \mathbf{C}_X(k) & \mathbf{O} \\ \mathbf{O} & \mathbf{C}_Z(k) \end{bmatrix}$, and

$$\mathbf{C}_X(k) = \text{diag}(\mathbf{G}_\lambda(\mathbf{e}_1(k)), \dots, \mathbf{G}_\lambda(\mathbf{e}_{n_x}(k)))$$

$$\mathbf{C}_Z(k) = \text{diag}(\mathbf{G}_\lambda(\mathbf{e}_{n_x+1}(k)), \dots, \mathbf{G}_\lambda(\mathbf{e}_{n_x+n_z}(k))).$$

Equation (45) can be further expressed as follows (the detailed derivation is in [52]):

$$\mathbf{X}(k) = \hat{\mathbf{X}}(k|k-1) + \bar{\mathbf{K}}(k) (\mathbf{Z}(k) - \hat{\mathbf{Z}}(k|k-1)) \quad (46)$$

$$\begin{cases} \bar{\mathbf{K}}(k) = \bar{\mathbf{P}}(k|k-1) \mathbf{H}^T(k) (\mathbf{H}(k) \bar{\mathbf{P}}(k|k-1) \mathbf{H}^T(k) + \bar{\mathbf{R}}(k))^{-1} \\ \bar{\mathbf{P}}(k|k-1) = \mathbf{B}_p(k|k-1) \mathbf{C}_X^{-1}(k) \mathbf{B}_p^T(k|k-1) \\ \bar{\mathbf{R}}(k) = \mathbf{B}_r(k) \mathbf{C}_Z^{-1}(k) \mathbf{B}_r^T(k) \end{cases}. \quad (47)$$

With the above derivations, we summarize the proposed MCKF algorithm as follows:

Step 1: Choose a proper kernel bandwidth λ and a small positive number ε ; the estimate $\hat{\mathbf{X}}(0|0)$ and the covariance matrix $\mathbf{P}(0|0)$ are initialized; let $k = 1$.

Step 2: Use Equations (27)–(32) to obtain $\hat{\mathbf{X}}(k|k-1)$ and $\mathbf{P}(k|k-1)$, and make use of Cholesky decomposition to calculate $\mathbf{B}_p(k|k-1)$.

Step 3: According to Equations (32)–(35), compute the predicted measurement $\hat{\mathbf{Z}}(k|k-1)$ and use Equations (25) and (30) to construct the model (36).

Step 4: Transform Equation (36) into Equation (38). Let $t^* = 1$ and $\hat{\mathbf{X}}(k|k)_0 = \hat{\mathbf{X}}(k|k-1)$, where $\hat{\mathbf{X}}(k|k)_{t^*}$ denotes the estimated state at the fixed-point iteration t^* .

Step 5: Use Equations (48)–(54) to compute $\hat{\mathbf{X}}(k|k)_{t^*}$:

$$\hat{\mathbf{X}}(k|k)_{t^*} = \hat{\mathbf{X}}(k|k-1) + \tilde{\mathbf{K}}(k) \hat{\mathbf{Z}}(k|k-1) \quad (48)$$

and

$$\tilde{\mathbf{K}}(k) = \tilde{\mathbf{P}}(k|k-1) \mathbf{H}^T(k) \left(\mathbf{H}(k) \tilde{\mathbf{P}}(k|k-1) \mathbf{H}^T(k) + \tilde{\mathbf{R}}(k) \right)^{-1} \quad (49)$$

$$\tilde{\mathbf{P}}(k|k-1) = \mathbf{B}_p(k|k-1) \tilde{\mathbf{C}}_X^{-1}(k) \mathbf{B}_p^T(k|k-1) \quad (50)$$

$$\tilde{\mathbf{R}}(k) = \mathbf{B}_r(k) \tilde{\mathbf{C}}_Z^{-1}(k) \mathbf{B}_r^T(k) \quad (51)$$

$$\tilde{\mathbf{C}}_X(k) = \text{diag}(G_\lambda(\tilde{e}_1(k)), \dots, G_\lambda(\tilde{e}_{n_x}(k))) \quad (52)$$

$$\tilde{\mathbf{C}}_Z(k) = \text{diag}(G_\lambda(\tilde{e}_{n_x+1}(k)), \dots, G_\lambda(\tilde{e}_{n_x+n_z}(k))) \quad (53)$$

$$\tilde{e}_i(k) = \mathbf{d}_i(k) - \mathbf{w}_i(k) \hat{\mathbf{X}}(k|k)_{t^*-1}. \quad (54)$$

Step 6: Compare the estimation of the current step and the estimation of the last step. If Equation (55) holds, set $\hat{\mathbf{X}}(k|k) = \hat{\mathbf{X}}(k|k)_{t^*}$ and continue to Step 6. Otherwise, go back to Step 4.

$$\frac{\|\hat{\mathbf{X}}(k|k)_{t^*} - \hat{\mathbf{X}}(k|k)_{t^*-1}\|}{\|\hat{\mathbf{X}}(k|k)_{t^*-1}\|} \leq \varepsilon. \quad (55)$$

Step 7: Update the estimation covariance matrix, make $k+1 \rightarrow k$, and go back to Step 2.

$$\mathbf{P}(k|k) = (\mathbf{I} - \tilde{\mathbf{K}}(k) \mathbf{H}(k)) \mathbf{P}(k|k-1) (\mathbf{I} - \tilde{\mathbf{K}}(k) \mathbf{H}(k))^T + \tilde{\mathbf{K}}(k) \mathbf{R}(k) \tilde{\mathbf{K}}^T(k). \quad (56)$$

For obtaining the optimal solution in Equation (42), the MCKF updates the state estimation through a fixed-point algorithm, which is shown as Equation (55). Therefore, the small positive number ε provides a threshold for the fixed-point iteration. When the result of the iteration is in the threshold range, we call it convergence and the solution is optimal. The convergence of the solution will be fast because of the predicted state $\hat{\mathbf{X}}(k|k-1)$ [52].

3.3. Computational Complexity

For the future implementation on real hardware, it is necessary to analyze the computational complexity of the proposed approach in the article. In terms of the floating points operations, the computation complexity of the basic equations of the MCKF is analyzed in Table 1.

Table 1. Computational complexities of some equations, where n denotes the dimension of the state variable, m denotes the dimension of the measurements.

Equation	Addition/Subtraction and Multiplication	Equation	Addition/Subtraction and Multiplication
(25)	$\frac{1}{3}n^3 + 3n^2$	(47)	$4n^2m + 4nm^2 - 3nm$
(27)	$4n^3 - n$	(48)	$2n^3$
(28)	$2n^2 + 2n$	(49)	$2m^3$
(29)	$4n^3 + 5n^2 + 2n$	(50)	$2n^2$
(31)	$\frac{1}{3}n^3 + 3n^2$	(51)	$2nm$
(32)	$4n^2m - m$	(52)	$2n$
(33)	$2nm + 2m$	(54)	$4n^3 + 6n^2m - 2n^2 + 2nm^2 - nm$
(46)	$4mn$		

The traditional unscented Kalman filter is analyzed in [53]. The computation complexity of it is

$$S_{\text{UKF}} = \frac{26}{3}n^3 + 15n^2 + 10n^2m + 5n + 8nm^2 + 6mn + m^3 + 3m^2 + 3m. \quad (57)$$

In fact, Equations (27)–(35) in the MCKF are the same as that in the UKF. Equations (48)–(56) in the MCKF mainly involves the different equations expressed by Equations (48)–(56). Equation (55) gives a fixed-point algorithm to update the posterior estimate of the state that can provide a stop condition in several steps. Assuming that the fixed-point iteration number is T , then the computation complexity of MCKF is

$$S_{\text{MCKF}} = \left(\frac{38}{3} + 2\right) Tn^3 + (11 + 2T)n^2 + (10 + 4T)n^2m + (2T + 3)n + (4T + 2)nm^2 + (1 + 3T)mn + 2Tm^3 + m. \quad (58)$$

According to Equations (57) and (58), the computational complexity of the MCKF is much higher than that of the UKF because of the fixed-point algorithm and more matrix inverse problems.

According to the complexity analysis above, the increasing dimension of the measurements of the deeply integrated mode has a higher computational complexity than the traditional mode.

3.4. Robustness Analysis

The robustness is a very important feature for the navigation filter algorithm, especially in the complex space environment. One of the most significant advantages of MCC is robustness that can tackle non-Gaussian noises. In fact, it uses a method similar to the Huber-based filter (HF) to improve the robustness [54]. Both of them use the weighted function to tackle the non-Gaussian noises. HF is a classical method to tackle the non-Gaussian noises, and it has been proved in detail [55]. Additionally, some articles have demonstrated the ability of MCC in accommodating non-Gaussian noises, especially outliers and Gaussian mixture noises [56,57]. Next, the algorithm robustness is compared with HF through the weighting function.

HF is created based on the concept of the robustness through the maximum likelihood method. Assume that the maximum likelihood estimator (MLE) stems from the data observation $\mathbf{Z} = \{z_1, \dots, z_N\}$ and the state variable $\mathbf{X} = \{x_1, \dots, x_N\}$. Then $\mathbf{e} = \mathbf{Z} - \mathbf{H}\mathbf{X}$ is drawn from the distribution $p(\mathbf{e}; \mathbf{X})$, which is known apart from the parameters \mathbf{X} . In this way, the MLE of \mathbf{X} can be defined as:

$$\hat{\mathbf{X}}_{MLE} = \arg \max_{\mathbf{X}} p(e_1, \dots, e_N; \mathbf{X}). \quad (59)$$

Assuming that e_1, \dots, e_N are i.i.d. observations, then it can be rewritten as:

$$\begin{aligned} \hat{\mathbf{X}}_{MLE} &= \arg \max_{\mathbf{X}} \prod_{i=1}^N p(e_i; \mathbf{X}) = \arg \max_{\mathbf{X}} \sum_{i=1}^N \ln p(e_i; \mathbf{X}) \\ &= \arg \min_{\mathbf{X}} \sum_{i=1}^N (-\ln p(e_i; \mathbf{X})) \end{aligned} \quad (60)$$

Let $\rho(e_i; \mathbf{X}) = a(-\ln p(e_i; \mathbf{X})) + b$, where $a > 0$ and b is uncorrelated with \mathbf{X} . Thus, $\hat{\mathbf{X}}_{MLE}$ obeys

$$\sum_{i=1}^N \rho(e_i; \mathbf{X}) = \sum_{i=1}^N \rho(z_i - \mathbf{H}x_i) = \min_{\mathbf{X}} \sum_{i=1}^N \rho(e_i; \mathbf{X}). \quad (61)$$

In fact, the errors in the target track model are always considered as pure Gaussian white noises with zero mean and constant variance σ^2 in some parametric models. In these cases, the maximum likelihood estimation can be solved. However, in practice, the external environment influences the

measurement equipment and causes the filter algorithm to break down. Hence, the non-Gaussian noises in the measurements cannot be avoided. A typical non-Gaussian noises are Gaussian mixture noises, which obey the Gaussian distributions with different mean values and variances with most errors. For instance, assume that the distribution is expressed as:

$$F(e_i; \mathbf{X}) = (1 - v) F_0(e_i; \mathbf{X}) + v F_c(e_i; \mathbf{X}), \quad (62)$$

where $F(\cdot)$, $F_0(\cdot)$, and $F_c(\cdot)$ denote the Gaussian mixture distribution, the known normal Gaussian distribution, and an unknown contaminated distribution, respectively, and v is the contamination ratio. It is much smaller than 1. For Equation (62), it is of necessity to solve the maximum likelihood estimation through Equation (61).

For the purpose of solving the problem, Huber proposed that ρ can change freely within limits, leading to the birth of M -estimation, which is an effective way of solving Equation (61).

M -estimator is defined as a generalization of MLE as:

$$\min_{\mathbf{X}} \sum_{i=1}^N \rho(e_i; \mathbf{X}) \quad \text{or} \quad \sum_{i=1}^N \varphi(e_i; \mathbf{X}) = 0 \quad \text{with} \quad \varphi(e_i; \mathbf{X}) = \frac{d\rho(e_i; \mathbf{X})}{de_i}, \quad (63)$$

where $\rho(e)$ denotes the cost or penalty function.

Let

$$\rho_{MCC}(e_i) = \left(1 - \exp\left(-e_i^2/2\lambda^2\right)\right) / \sqrt{2\pi}\lambda \quad (64)$$

where λ denotes the kernel bandwidth.

Then

$$\begin{aligned} & \min \sum_{i=1}^N \rho_{MCC}(e_i) \\ &= \min \sum_{i=1}^N \left(1 - \exp\left(-e_i^2/2\lambda^2\right)\right) / \sqrt{2\pi}\lambda \\ &\Leftrightarrow \max \sum_{i=1}^N \exp\left(-e_i^2/2\lambda^2\right) / \sqrt{2\pi}\lambda \\ &= \max \sum_{i=1}^N G_\lambda(e_i) \end{aligned} \quad (65)$$

Therefore, MCC is equivalent to Equation (64):

$$\min_{\mathbf{X}} \sum_{i=1}^N \rho_{MCC}(e_i) \quad \text{is equivalent to} \quad MCC = \max_{\mathbf{X}} \sum_{i=1}^N G_\lambda(e_i).$$

According to the M -estimation problem, $\psi_{MCC}(e_i)$, as well as the weighting function of MCC, can be rewritten as:

$$\psi_{MCC}(e_i) = \frac{\rho_{MCC}'(e_i)}{e_i}, \quad (66)$$

where $\rho_{MCC}'(e_i) = \frac{d\rho_{MCC}(e_i)}{de_i}$. Therefore,

$$\psi_{MCC}(e_i) = \exp\left(-e_i^2/2\lambda^2\right) / \sqrt{2\pi}\lambda^3. \quad (67)$$

On the basis of analysis of the weighting function above, we will compare it with the Huber filter to prove the superiority of MCC.

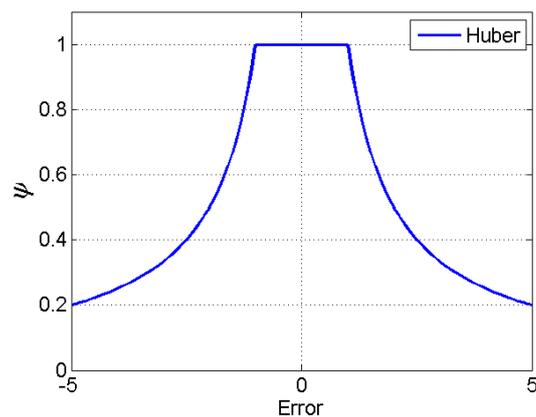
The weighted functions of them are shown in Table 2.

Table 2. The weighted functions of two methods.

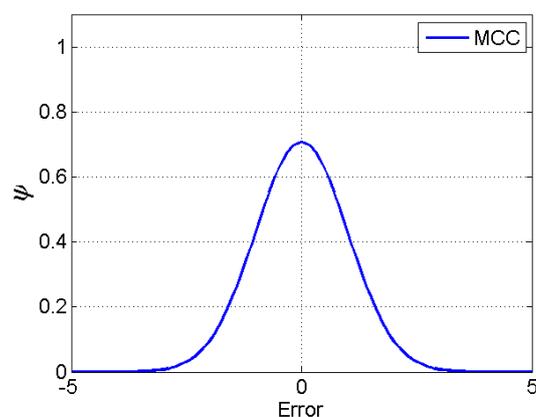
Item	Penalty Function	Weighted Function
Huber	$\rho_{\text{Huber}}(e_i) = \begin{cases} e_i^2/2 & e_i \leq \delta \\ \delta(e_i - \delta) & e_i > \delta \end{cases}$	$\psi_{\text{Huber}}(e_i) = \begin{cases} 1 & e_i \leq \delta \\ \delta/ e_i & e_i > \delta \end{cases}$
MCC	$\rho_{\text{MCC}}(e_i) = (1 - \exp(-e_i^2/2\lambda^2))/\sqrt{2\pi}\lambda$	$\psi_{\text{MCC}}(e_i) = \exp(-e_i^2/2\lambda^2)/\sqrt{2\pi}\lambda^3$

Some researchers have performed robustness analysis of MCC in theory. The robustness of two methods are shown through Figure 4 to intuitively analyze the robustness of MCC.

In Figure 4, the weight of HF decreases rapidly as the estimate residual grows. However, the weight of MCC approaches zero faster compared with that of the Huber method. In addition, with the growth of the error, the weight of the Huber method is not zero. In other words, Huber-based methods can select more measurements containing large errors than MCC-based methods, which may cause large errors to the filter algorithm, although with small weights. Therefore, the filter based on the MCC is more robust than the Huber-based filter.



(a) The weighted function of the Huber method



(b) The weighted function of the maximum correntropy criterion (MCC)

Figure 4. The weight comparison with the error growth of two methods where $\lambda = \delta = 1$.

4. Simulation Results

In this section, the superiority of SINS/CNS deeply integrated navigation is proved firstly. The performances of the MCKF are then illustrated under the conditions of the Gaussian noises, the large outliers, and the Gaussian mixture noises, respectively. The estimation performance are compared on the basis of the following indices:

$$\text{Residual}(k) = \mathbf{X}(k) - \hat{\mathbf{X}}(k|k), \quad k = 1, \dots, m, \quad (68)$$

$$\text{RMSE} = \sum_{j=1}^L \sqrt{\frac{1}{m} \sum_{i=1}^m \left(\mathbf{X}(k) - \hat{\mathbf{X}}(k|k) \right)^2} / L, \quad j = 1, \dots, M, \quad (69)$$

where m is the simulation time steps and L represents the total simulation numbers. In addition, the time cost of every method is given to compare the computational complexity. All methods are implemented in the same precision (64-point floating point) in MATLAB running on a PC with processor Intel(R) Core(TM) i7-7500U CPU 2.70GHz and with 8 GB of installed memory (RAM).

The simulation parameters are as Table 3.

Table 3. Simulation conditions.

Parameter	Values	Parameter	Values
Initial latitude	39.98°	Random noise of altimeter	50 m
Initial longitude	116.34°	Random noise of star sensor	3"
Initial velocity	$v_x = 355.49 \text{ m/s}$ $v_y = v_z = 0$	Time of the vertical rise	10 s
Initial pitch angle (Launching angle)	90°	Filter start time	40 s
Gravity acceleration	$g_0 = 9.78 \text{ m/s}^2$	Ending time of the powered phase turn	60 s
Constant drifts of the gyro	$\varepsilon_x = \varepsilon_y =$ $\varepsilon_z = 1^\circ/h$	Time of the engine shutting off	160 s
Random noise of the gyro	$\varepsilon_{sx} = \varepsilon_{sy} =$ $\varepsilon_{sz} = 0.5^\circ/h$	Total flying time	1110 s
Constant biases of the accelerometers	$\nabla_x = \nabla_y =$ $\nabla_z = 100 \text{ ug}$	Filter period	1 s
Random noise of the accelerometers	$\nabla_{sx} = \nabla_{sy} =$ $\nabla_{sz} = 50 \text{ ug}$	Gyro Sample period	0.01 s
Monte Carlo times	$L = 10$	Star Sensor Sampling period	1 s

The missile trajectory is shown in Figure 5.

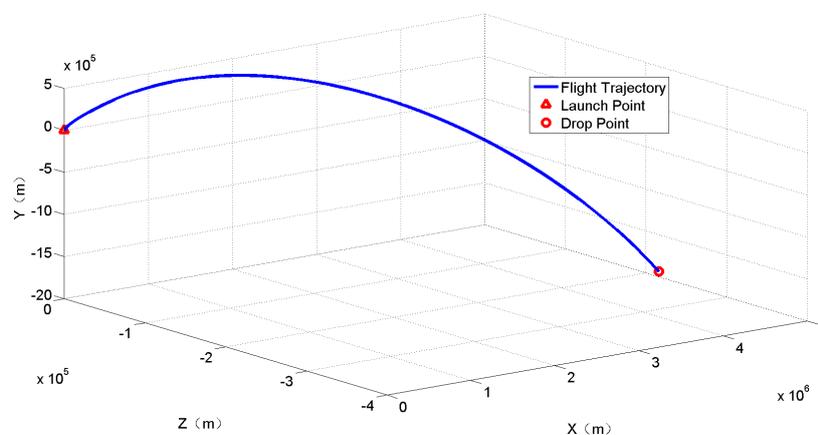


Figure 5. Missile flight trajectory.

To prove the high performance of the new integrated navigation system, a simulation is done only under the condition of the Gaussian white noises. Compared with the traditional integrated mode, the position and the attitude angle residual are shown in Figures 6 and 7.

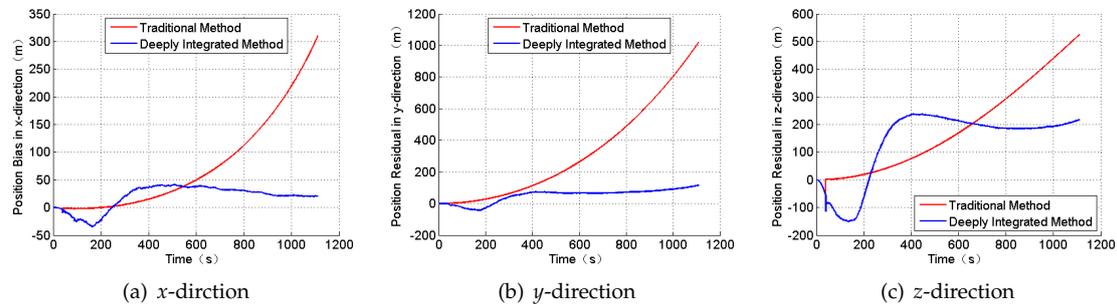


Figure 6. The position residual.

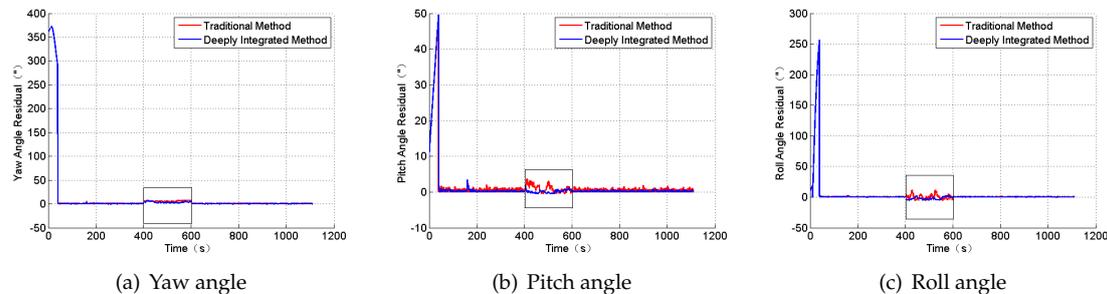


Figure 7. The attitude angle residual.

The RMSE of the position and attitude angle and the time cost of every filter are shown in Table 4.

Table 4. RMSE of the position, attitude, and time cost of the two navigation methods.

	Position (m)			Attitude (")			Time (min)
	<i>x</i>	<i>y</i>	<i>z</i>	Yaw	Pitch	Roll	
Traditional Method	114.4777	434.0437	245.1516	27.4501	65.9759	6.1114	0.738
Deeply Integrated Mode	30.9459	67.5309	184.4883	27.4490	65.9718	6.1009	0.934

From the Figures and the table above, the navigation position error has evidently improved when constructing the measurement Z_2 in Equation (12). In addition, the SINS/CNS deeply integrated navigation mode for the ballistic missile can tackle the divergence of the position error effectively, which obeys the observability analysis results. However, the newly proposed navigation system needs more time to gain the results.

Next, we will compare the MCKF performance with other algorithms under the conditions of the Gaussian noises, the large outliers, and the Gaussian mixture noises for SINS/CNS deeply integrated navigation.

4.1. Case 1: Gaussian Noises

In this section, we will apply the MCKF on SINS/CNS deeply integrated navigation under the conditions of the Gaussian noises to compare performances with other filter algorithms including the

HKF [55], the IEKF, the NIF, the PLF, the UKF, and the MCEKF [36]. The simulation parameters are set as Table 1. The simulation results are shown in Figures 8 and 9, and Table 5.

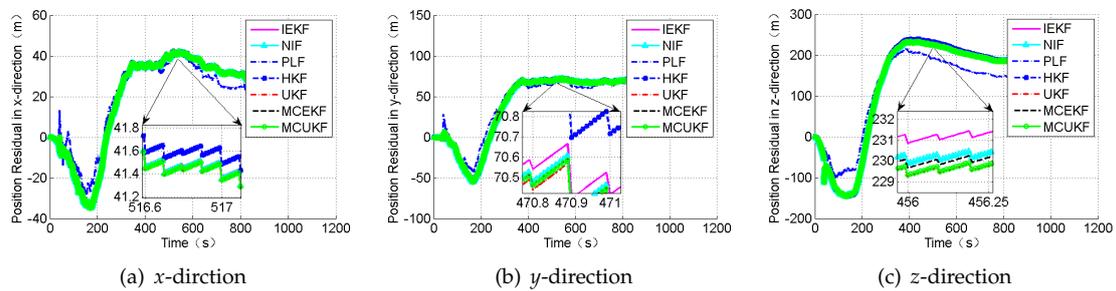


Figure 8. The position residual under the condition of Gaussian noise.

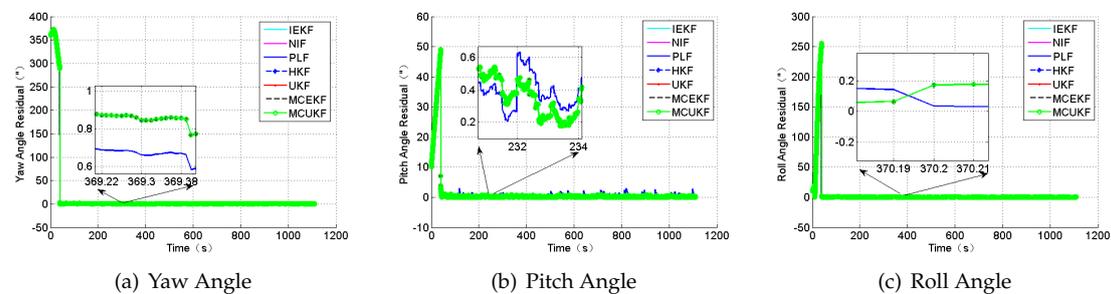


Figure 9. The attitude angle residual under the condition of Gaussian noise.

Table 5. RMSE of position, attitude, and time cost of different methods in the presence of Gaussian noise for SINS/CNS deeply integrated navigation.

	Position (m)			Attitude (")			Time (min)
	x	y	z	Yaw	Pitch	Roll	
IEKF	29.5693	66.0529	184.5749	27.4483	66.0447	6.0555	0.9748
NIF	29.5635	66.0312	184.2973	27.4483	66.0447	6.0555	1.3277
PLF	25.5944	66.6029	172.7334	27.4425	66.0406	6.0086	1.4536
HKF	29.6274	66.2252	186.4335	27.4483	66.0447	6.0555	0.5878
UKF	29.5598	66.0247	184.2777	27.4483	66.0447	6.0555	0.1124
MCEKF	29.5841	66.0968	185.0428	27.4483	66.0447	6.0555	0.2280
MCUKF	29.5659	66.0427	184.4676	27.4483	66.0447	6.0555	0.3230

According to Table 5, PLF demonstrates the highest accuracy among the seven filters in this condition because of the a posteriori estimation and the iteration. It can be observed that robust filters cannot always perform as well as the PLF, the NIF, or the UKF under the condition of the Gaussian noises.

4.2. Case 2: Large Outliers

In this section, the case in which the measurement value including large outliers is considered. The attitude outliers are added artificially to the attitude angle data at the 5000th and 9000th epochs, respectively. The position outliers are also added artificially to the position data at the 5000th epochs. The constant outliers are added to all epochs between the 4001th to the 4005th to the attitude angle

data. The constant outliers are added to all epochs between 4001th and the 4005th to the position data. The obtained position errors and the attitude angle errors are displayed in Figures 10 and 11, and Table 6.

According to the presented results, it can be concluded that, in the presence of outliers, the filter results mostly depend on the outliers. In addition, Figures 10 and 11 show that the HKF, the MCEKF, and the MCKUF can resist the influence of the outliers to some extent. Furthermore, the two figures show that the MCKUF has better performance than the other algorithms, so the influence of the single and the constant outliers is reduced using the MCKUF algorithm. The RMSE values and time cost of all used filters are presented in Table 6.

Compared with other filters, the filtering algorithm of the MCEKF and the MCKUF is improved to a certain degree, especially for the outliers. The MCKUF has better performance than other filters and lower time cost than the iterated filters.

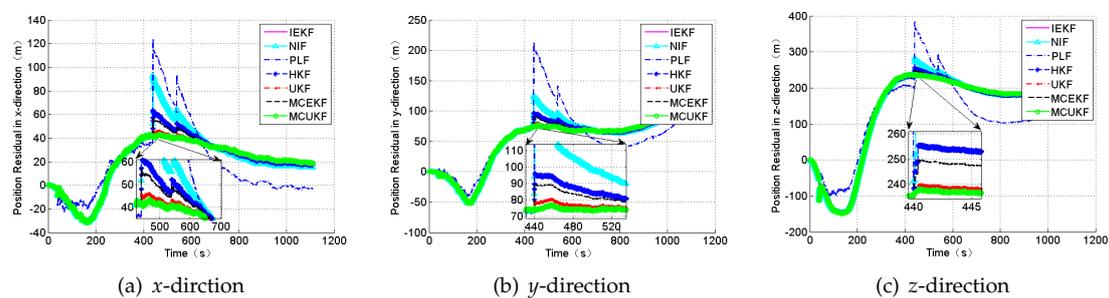


Figure 10. The position residual under the condition of the large outliers.

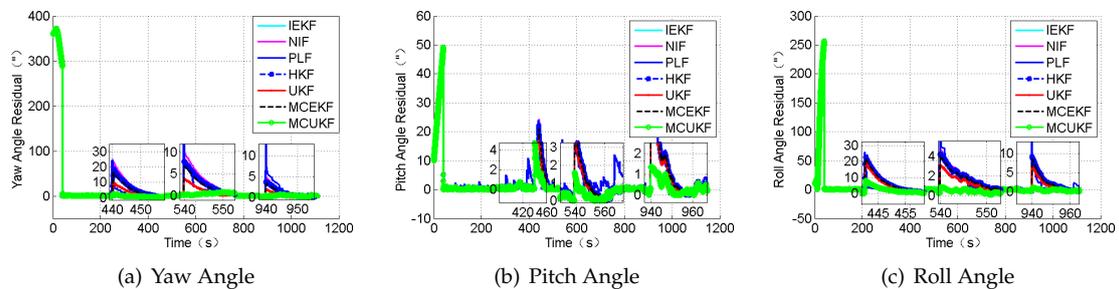


Figure 11. The attitude angle residual under the condition of the large outliers.

Table 6. RMSE of position, attitude, and time cost of different methods in the presence of large outliers for SINS/CNS deeply integrated navigation.

	Position (m)			Attitude (")			Time (min)
	x	y	z	Yaw	Pitch	Roll	
	x	y	z	p	ph	g	time
IEKF	34.9042	70.5450	186.9381	27.5917	66.1098	6.2019	0.4629
NIF	34.8999	70.5382	186.9179	27.5918	66.1098	6.2021	0.8702
PLF	36.1497	73.6876	176.1497	27.5687	66.1091	6.1272	0.9432
HKF	31.5010	68.6713	185.1142	27.5874	66.1095	6.1775	0.2228
UKF	29.6882	67.6938	184.3485	27.5715	66.1088	6.1275	0.1332
MCEKF	30.8100	68.2925	184.7478	27.5852	66.1093	6.1657	0.2257
MCKUF	29.3169	67.5903	184.3414	27.5579	66.1081	6.0574	0.2665

4.3. Case 3: Gaussian Mixture Noises

In this section, the case in which the measurement noises are the Gaussian mixture noises, whose distributions are as follows:

$$\text{Star Sensor Noise: } v_s \sim 0.9N(0, (3'')^2) + 0.1N(0, (30'')^2).$$

$$\text{Altimeter Noise: } v_h \sim 0.9N(0, (50m)^2) + 0.1N(0, (500m)^2).$$

The obtained position errors and the attitude angle errors are displayed in Figures 12 and 13, and Table 7.

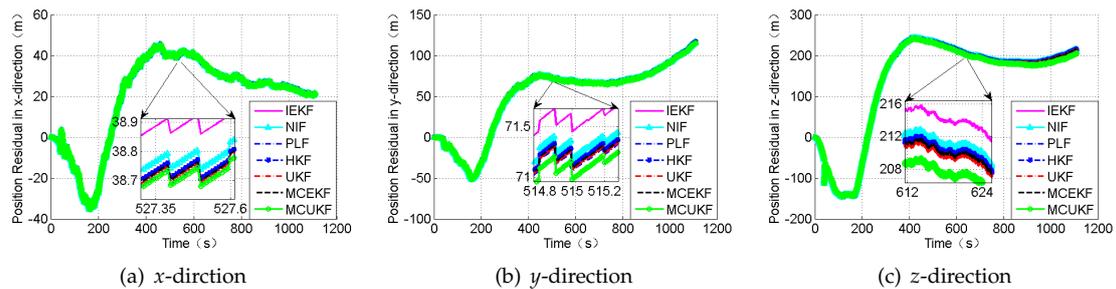


Figure 12. The position residual under the condition of the Gaussian mixture noises.

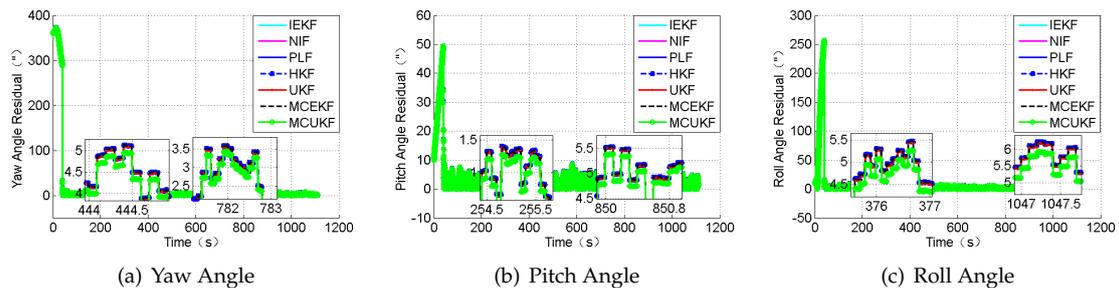


Figure 13. The attitude angle residual under the condition of the Gaussian mixture noises.

Table 7. RMSE of position, attitude, and time cost of different message under the condition of the Gaussian mixture noises for SINS/CNS deeply integrated navigation.

	Position (m)			Attitude (")			Time (min)
	x	y	z	Yaw	Pitch	Roll	
	x	y	z	p	ph	g	time
IEKF	30.1388	67.7275	185.7022	26.6707	66.0896	6.6902	0.8293
NIF	30.0166	67.4352	183.6012	27.8260	67.2466	6.6826	1.2689
PLF	30.6610	67.5520	183.6714	27.6757	66.0895	6.6811	2.9075
HKF	30.0526	67.5224	184.2160	27.6750	66.0893	6.6787	1.3422
UKF	30.0090	67.4125	183.4182	27.6757	66.0869	6.6811	0.1277
MCEKF	30.0233	67.4515	183.7159	27.6750	66.0893	6.6787	0.3502
MCKUF	30.0049	67.4062	183.3980	27.6679	66.0696	6.6521	0.3889

Figures 12 and 13 reveal the residual using different filters under the conditions of the Gaussian mixture noises. In addition, Table 7 illustrates the RMSE of the position, the attitude angle, and the time cost of every method. It can be observed that the MCKUF achieves better performance than other filters.

5. Conclusions

SINS/CNS deeply integrated navigation, which consists in modified SINS/CNS navigation, is introduced, and the observability of the navigation system was analyzed. However, the simulation results demonstrate that the new system needs more time. Measurement noises are often the research focus with regard to navigation systems. The non-Gaussian noises, including the large outliers and the Gaussian mixture noises, always exist during the measurement process. On the basis of the maximum correntropy theory, the maximum correntropy unscented Kalman filter (MCUKF) is a newly designed robust unscented Kalman filter (UKF). The robustness for state estimation, which can resist the effects of non-Gaussian noises effectively, is also analyzed and compared with a Huber-based filter. Simulated experimental results demonstrate that the MCC-based filter provides much better performance when tackling the non-Gaussian noises of SINS/CNS deeply integrated navigation, but with a higher computational complexity than the traditional UKF.

In practical applications, the added star sensors and the altimeter of the deeply integrated system will increase the load of the missile. Future work will focus on the installation method of the measurement facilities and applications in real scenarios. The coupled problem of multiple sensors may also need to be solved.

Author Contributions: B.H. proposed the main idea, finished the draft manuscript; Z.H. and D.L. conceived of the experiments and drew the figures and tables; H.Z. analyzed the data; J.W. conducted the simulations.

Funding: This work was supported in part by the National Defense Technology Foundation of China under Grant No. 3101065, and the National Natural Science Foundation of China (NSFC) under Grant No. 61773021, 61703408, and 61573367.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

SINS	Strap-Down Inertial Navigation System
CNS	Celestial Navigation System
MCC	Maximum Correntropy Criterion
MCUKF	Maximum Correntropy Unscented Kalman Filter
MCEKF	Maximum Correntropy Extended Kalman Filter
UKF	Unscented Kalman Filter
HKF	Huber Kalman Filter
IEKF	Iterated Extended Kalman Filter
NIF	Nonlinear Iterated Filter
PLF	Posterior Linearization Filter

References

1. Pan, J.; Xiong, Z.; Zhao, H.; Yu, F.; Wang, L. SINS/GPS/CNS multi-integrated navigation system algorithm in launch inertial coordinate system and realization. *Chin. Space Sci. Technol.* **2015**, *35*, 9–16. [[CrossRef](#)]
2. Grewal, M.S.; Weill, L.R.; Andrews, A.P. *Global Positioning Systems, Inertial Navigation, and Integration*; John Wiley and Sons: New York, NY, USA, 2007; pp. 23–39.
3. Titterton, D.; Weston, J. Strapdown inertial navigation technology-2nd edition-[book review]. *IEEE Aerosp. Electron. Syst. Mag.* **2004**, *20*, 33–34. [[CrossRef](#)]
4. Yang, L.; Li, B.; Ge, L. A novel SINS/CNS integrated navigation algorithm used in a ballistic missile. *Int. J. Secur. Appl.* **2015**, *9*, 65–76. [[CrossRef](#)]
5. Yu, F.; Lv, C.; Dong, Q. A novel robust H-infinity filter based on Krein space theory in the SINS/CNS attitude reference system. *Sensors* **2016**, *16*, 396. [[CrossRef](#)] [[PubMed](#)]
6. Zhang, H.; Zheng, W.; Tang, G. Stellar/inertial integrated guidance for responsive launch vehicles. *Aerosp. Sci. Technol.* **2012**, *18*, 35–41. [[CrossRef](#)]

7. Wang, Q.; Li, Y.; Diao, M.; Gao, W.; Yu, F. Coarse alignment of a shipborne strapdown inertial navigation system using star sensor. *IET Sci. Meas. Technol.* **2015**, *9*, 852–860. [[CrossRef](#)]
8. Gao, W.; Zhang, Y.; Wang, J. A strapdown inertial navigation system/beidou/doppler velocity log integrated navigation algorithm based on a cubature kalman filter. *Sensors* **2014**, *14*, 1511–1527. [[CrossRef](#)] [[PubMed](#)]
9. Nobahari, H.; Ghanbarpour Asl, H.; Abtahi, S.F. A back-propagation approach to compensate velocity and position errors in an integrated inertial/celestial navigation system using unscented Kalman filter. *Proc. Inst. Mech. Eng. Part G J. Aerosp. Eng.* **2014**, *228*, 1702–1712. [[CrossRef](#)]
10. Rad, A.M.; Nobari, J.H.; Nikkhah, A.A. Optimal attitude and position determination by integration of INS, star tracker, and horizon sensor. *IEEE Aerosp. Electron. Syst. Mag.* **2014**, *29*, 20–33. [[CrossRef](#)]
11. Yang, Y.; Zhang, C.; Lu, J. Local observability analysis of star sensor installation errors in a SINS/CNS integration system for near-earth flight vehicles. *Sensors* **2017**, *17*, 167. [[CrossRef](#)] [[PubMed](#)]
12. Yang, S.; Yang, G.; Zhu, Z.; Li, J. Stellar refraction-based SINS/CNS integrated navigation system for aerospace vehicles. *J. Aerosp. Eng.* **2016**, *29*, 04015051. [[CrossRef](#)]
13. Lai, J.Z.; Yu, Y.J.; Xiong, Z.; Liu, J.Y. SINS/CNS tightly integrated navigation positioning algorithm with nonlinear filter. *Control Decis.* **2012**, *27*, 11.
14. Qian, H.; Lang, X.; Qian, L.; Peng, Y.; Wang, H. Ballistic missile SINS/CNS integrated navigation method. *J. Beijing Univ. Aeronaut. Astronaut.* **2017**, *43*, 857–864. [[CrossRef](#)]
15. Liu, X.; Qu, H.; Zhao, J.; Yue, P.; Wang, M. Maximum correntropy unscented Kalman filter for spacecraft relative state estimation. *Sensors* **2016**, *16*, 1530. [[CrossRef](#)] [[PubMed](#)]
16. Wang, M.; Wang, S.; Mu, J. SINS/CNS integrated navigation system. *J. Proj. Rockets Missiles Guid.* **2009**, *29*, 87–90. [[CrossRef](#)]
17. Anderson, B.D.O.; Moore, J.B.; Eslami, M. Optimal filtering. *IEEE Trans. Syst. Man Cybern.* **2007**, *12*, 235–236. [[CrossRef](#)]
18. Juler, S.J.; Uhlmann, J.K.; Durrant-Whyte, H.F. A new method for nonlinear transformation of means and covariances in filters and estimator. *IEEE Trans. Autom. Control* **2000**, *45*, 477–482. [[CrossRef](#)]
19. Bell, B.M.; Cathey, F.W. The iterated Kalman filter update as a Gauss-Newton method. *IEEE Trans. Autom. Control* **1993**, *38*, 294–297. [[CrossRef](#)]
20. Bellaire, R.L.; Kamen, E.W.; Zabin, S.M. New nonlinear iterated filter with applications to target tracking. In Proceedings of SPIE's 1995 International Symposium on Optical Science, Engineering, and Instrumentation, San Diego, CA, USA, 9–14 July 1995; pp. 240–252.
21. García-Fernández, Á.F.; Svensson, L.; Morelande, M.R.; Särkkä, S. Posterior linearization filter: Principles and implementation using sigma points. *IEEE Trans. Signal Process.* **2015**, *63*, 5561–5573. [[CrossRef](#)]
22. Plataniotis, K.N.; Androustos, D.; Venetsanopoulos, A.N. Nonlinear filtering of non-Gaussian noise. *J. Intell. Rob. Syst.* **1997**, *19*, 207–231. [[CrossRef](#)]
23. Sorenson, H.W.; Alspach, D.L. Recursive Bayesian estimation using Gaussian sums. *Automatica* **1971**, *7*, 465–479. [[CrossRef](#)]
24. Harvey, A.; Luati, A. Filtering with heavy tails. *J. Am. Stat. Assoc.* **2014**, *109*, 1112–1122. [[CrossRef](#)]
25. Izanloo, R.; Fakoorian, S.A.; Yazdi, H.S.; Simon, D. Kalman filtering based on the maximum correntropy criterion in the presence of non-Gaussian noise. In Proceedings of the 2016 Annual Conference on Information Science and Systems (CISS), Princeton, NJ, USA, 16–18 March 2016; pp. 500–505. [[CrossRef](#)]
26. Wan, E.A.; Van Der Merwe, R. The unscented Kalman filter for nonlinear estimation. In Proceedings of the Adaptive Systems for Signal Processing, Communications, and Control Symposium, Lake Louise, AB, Canada, 4 October 2000; pp. 153–158. [[CrossRef](#)]
27. Curn, J.; Marinescu, D.; Lacey, G.; Cahill, V. Estimation with non-white Gaussian observation noise using a generalised ensemble Kalman filter. In Proceedings of the 2012 IEEE International Symposium on Robotic and Sensors Environments Proceedings, Magdeburg, Germany, 16–18 November 2012; pp. 85–90. [[CrossRef](#)]
28. Rousseeuw, P.J.; Leroy, A.M. Robust regression and outlier detection. *Technometrics* **2005**, *31*, 260–261.
29. Wang, R.; Xiong, Z.; Liu, J.Y.; Li, R.; Peng, H. SINS/GPS/CNS information fusion system based on improved Huber filter with classified adaptive factors for high-speed UAVs. In Proceedings of the 2012 IEEE/ION Position, Location and Navigation Symposium, Myrtle Beach, SC, USA, 23–26 April 2012; pp. 441–446. [[CrossRef](#)]

30. Huang, J.; Yan, B.; Hu, S. Centralized fusion of unscented kalman filter based on huber robust method for nonlinear moving target tracking. *Math. Prob. Eng.* **2015**, *2015*. [[CrossRef](#)]
31. Chang, L.; Li, K.; Hu, B. Huber's M-estimation-based process uncertainty robust filter for integrated INS/GPS. *IEEE Sens. J.* **2015**, *15*, 3367–3374. [[CrossRef](#)]
32. Tseng, C.H.; Lin, S.F.; Jwo, D.J. Robust huber-based cubature kalman filter for gps navigation processing. *J. Navig.* **2016**, *70*, 527–546. [[CrossRef](#)]
33. Chang, G. Robust Kalman filtering based on Mahalanobis distance as outlier judging criterion. *J. Geod.* **2014**, *88*, 391–401. [[CrossRef](#)]
34. Chen, B.; Xing, L.; Liang, J.; Zheng, N.; Principe, J.C. Steady-state mean-square error analysis for adaptive filtering under the maximum correntropy criterion. *IEEE Signal Process. Lett.* **2014**, *21*, 880–884. [[CrossRef](#)]
35. Chen, B.; Principe, J.C. Maximum correntropy estimation is a smoothed MAP estimation. *IEEE Signal Process. Lett.* **2012**, *19*, 491–494. [[CrossRef](#)]
36. Liu, X.; Qu, H.; Zhao, J.; Chen, B. Extended Kalman filter under maximum correntropy criterion. In Proceedings of the 2016 International Joint Conference on Neural Networks (IJCNN), Vancouver, BC, Canada, 24–29 July 2016; pp. 1733–1737. [[CrossRef](#)]
37. Liu, X.; Chen, B.; Xu, B.; Wu, Z.; Honeine, P. Maximum correntropy unscented filter. *Int. J. Syst. Sci.* **2017**, *48*, 1607–1615. [[CrossRef](#)]
38. Vacanti, D.C. Radar Altimeter. U.S. Patent 7,239,266, 3 July 2007.
39. Pressure Altimeter. In *Dictionary Geotechnical Engineering/Wörterbuch GeoTechnik*; Springer: Berlin/Heidelberg, Germany, 2014. [[CrossRef](#)]
40. Smith, D.E.; Zuber, M.T.; Frey, H.V.; Garvin, J.B.; Head, J.W.; Muhleman, D.O.; Pettengill, G.H.; Phillips, R.J.; Solomon, S.C.; Banerdt, W.B. Mars orbiter laser altimeter: Experiment summary after the first year of global mapping of mars. *J. Geophys. Res. Planets* **2001**, *106*, 23689–23722. [[CrossRef](#)]
41. Quan, W.; Liu, B.Q.; Gong, X.L.; Fang, J.C. *INS/CNS/GNSS Integrated Navigation Technology*; Springer: Berlin/Heidelberg, Germany, 2015.
42. Deng, H.; Liu, G.B.; Chen, H.M.; Liu, Z.G. Deduction and simulation of angular error relationship in "SINS/CNS" integrated navigation system. *J. Astronaut.* **2011**, *32*, 781–786.
43. Xiong, Z.; Liu, J.Y.; Yu, F.; Chen, H.M. Research of airborne INS/CNS integrated filtering algorithm based on celestial angle observation. *J. Astronaut.* **2010**, *31*, 397–403.
44. Wu, X.; Wang, X. A SINS/CNS deep integrated navigation method based on mathematical horizon reference. *Aircr. Eng. Aerosp. Technol.* **2011**, *83*, 26–34. [[CrossRef](#)]
45. Qian, H.; Sun, L.; Cai, J.; Peng, Y. A novel navigation method used in a ballistic missile. *Meas. Sci. Technol.* **2013**, *24*, 105011. [[CrossRef](#)]
46. Goshen-Meskin, D.; Bar-Itzhack, I.Y. Observability analysis of piece-wise constant systems with application to inertial navigation. In Proceedings of the 29th IEEE Conference on Decision and Control, Honolulu, HI, USA, 5–7 December 1990; pp. 821–826. [[CrossRef](#)]
47. Chen, Z.; Jiang, K.; Hung, J.C. Local observability matrix and its application to observability analyses. In Proceedings of the 16th Annual Conference of IEEE Industrial Electronics Society (IECON), Pacific Grove, CA, USA, 27–30 November 1990; pp. 100–103.
48. Principe, J.C. *Information Theoretic Learning: Renyi's Entropy and Kernel Perspectives*; Springer: New York, NY, USA, 2010.
49. Liu, W.; Pokharel, P.P.; Principe, J.C. Correntropy: Properties and applications in non-Gaussian signal processing. *IEEE Trans. Signal Process.* **2007**, *55*, 5286–5298. [[CrossRef](#)]
50. He, R.; Hu, B.; Yuan, X.; Wang, L. *Robust Recognition via Information Theoretic Learning*; Springer: Cham, Switzerland, 2014.
51. Chen, B.; Wang, J.; Zhao, H.; Zheng, N.; Principe, J.C. Convergence of a fixed-point algorithm under maximum correntropy criterion. *IEEE Signal Process. Lett.* **2015**, *22*, 1723–1727. [[CrossRef](#)]
52. Chen, B.; Liu, X.; Zhao, H.; Principe, J.C. Maximum correntropy kalman filter. *Automatica* **2015**, *76*, 70–77. [[CrossRef](#)]
53. Zhang, Z.; Hao, Y.; Wu, X. Complexity analysis of three deterministic sampling nonlinear filtering algorithms. *J. Harbin Inst. Technol.* **2013**, *45*, 111–115.
54. Huber, P.J. *Robust Estimation of a Location Parameter*; Springer: New York, NY, USA, 1992.

55. Petrus, P. Robust huber adaptive filter. *IEEE Trans. Signal Process.* **1999**, *47*, 1129–1133. [[CrossRef](#)]
56. Chen, B.; Xing, L.; Zhao, H.; Xu, B.; Principe, J.C. Robustness of maximum correntropy estimation against large outliers. *arXiv* **2017**, arXiv:1703.08065.
57. Su, Z.; Yang, L.; Zhu, S.; Si, N.; Lv, X. Gaussian mixture image restoration based on maximum correntropy criterion. *Electron. Lett.* **2017**, *53*, 715–716. [[CrossRef](#)]



© 2018 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).