

# An Online Smoothing Method Based on Reverse Navigation for ZUPT-Aided INSs

Qingzhong Cai, Gongliu Yang, Ningfang Song, Jianye Pan and  
Yiliang Liu

*(School of Instrument Science and Opto-electronics Engineering, Beihang University,  
Beijing, China)*

(E-mail: [qingzhong\\_cai@163.com](mailto:qingzhong_cai@163.com))

Zero velocity update (ZUPT) is widely discussed for error restriction in land vehicle Inertial Navigation Systems (INSs) and wearable pedestrian INSs to overcome the problems of Global Positioning System (GPS) unavailability in urban canyons or indoor scenarios. In this paper, an online smoothing method for ZUPT-aided INSs is presented. By introducing the Rauch–Tung–Striebel (RTS) smoothing method into the ZUPT-aided INS, position errors can be effectively restrained not only at stop points but during the whole trajectory. By integrating reverse navigation with a ZUPT smoother, the method realises forward and real-time processing. Compared with existing approaches, it can improve the position accuracy in real time without any other sensors, which is well suited for applications on high-accuracy navigation in GPS-challenging environments. Accuracy test results with different Inertial Measurement Units (IMUs) show that the developed method can significantly decrease position errors from hundreds or thousands of metres to below ten metres. During the whole trajectory, the online smoothing method ensures the maximum position errors at non-stop points can reach the same level of accuracy at stop points. A delay test result proves that the delay of the reverse online smoothing method proposed in this paper is much shorter than existing online smoothing methods.

## KEYWORDS

1. Inertial navigation.
2. Online smoothing.
3. Reverse navigation algorithm.
4. Zero velocity update.

Submitted: 18 March 2016. Accepted: 20 August 2016. First published online: 21 October 2016.

1. INTRODUCTION. Inertial Navigation System (INS) technology is widely applied for bridging GPS outages (Aloi and Korniyenko, 2007; Grejner-Brzezinska et al., 2001) or even replacing the Global Positioning System (GPS) in certain environments, such as urban canyons or indoor scenarios (Collin, 2015; Zhang et al., 2015; Correa et al., 2016) because of the signal blockages of GPS. Zero velocity update (ZUPT), as one of the high-accuracy position methods used in INS without any other assisting sensors, has been widely used in urban canyon land vehicle navigation

(Grejner-Brzezinska et al., 2001) and indoor pedestrian navigation (Godha and Lachapelle., 2008; Wang et al., 2015).

By periodically stopping and observing the velocity error using an estimation filter such as a Kalman filter, the ZUPT technology can correct the user's velocity, restrict the position and attitude errors, and further estimate the sensor bias errors. In land vehicle INS, previous work focuses on proposing various kinds of advanced ZUPT methods. An improved filter estimation method applied in ZUPT for land-based survey systems was proposed by Ben et al. (2009). A new hybrid extended particle filter combined with ZUPT is proposed for integrated navigation system (Yang et al., 2010). Li et al. (2012) proposed an adaptive ZUPT algorithm to identify the zero velocity condition and improve the position accuracy of the cart-mounted geolocation system for unexploded ordnance. Ramanandan et al. (2012) proposed a new frequency domain approach, using only IMU data, to detect the zero velocity condition for land vehicles. In indoor pedestrian navigation, ZUPT is a key method to improve indoor position accuracy without GPS data by mounting the Inertial Measurement Unit (IMU) on a foot/shoe (Godha and Lachapelle., 2008; Ojeda and Borenstein, 2007; Zhou et al., 2010). To improve the accuracy of the ZUPT, some researchers focus on zero velocity detection or walking motion (Wang et al., 2015; Bebek et al., 2010; Elhoushi et al., 2016). Also, some researchers focus on integration with other sensors, such as magnetic sensors (Yun et al., 2012; Fourati, 2015), visual sensors (Tian et al., 2014) and pressure sensors (Zihajehzadeh et al., 2015). However, research and improvement of the ZUPT method itself is still necessary to improve the position accuracy or the adaptability in flexible walking motion conditions, such as while running.

To improve the position accuracy of all points along the travel route, an optimal smoothing method is employed in the early inertial survey systems (Huddle, 1986). However, there is a drawback in that all traditional smoothing methods can only be applied for post-processing applications. Suh (2012) proposes a two-step smoother method consisting of an attitude smoother and a velocity smoother. In the two-step smoother, the post-processing works on the previous segment of the path, so that the delay time is the interval between the two stops. A step-wise smoothing algorithm of ZUPT-aided INSs was proposed by Colomar et al. (2012), and this is applied to the data in a step-wise way requiring a suggested varying-lag segmentation rule. Even though the step-wise smoothing algorithm can be used in near real-time applications, it cannot estimate the position error at the real-time point or the end point.

The existing smoothing methods cannot be used in real-time processing because all the smoothing techniques are backward processing methods that always start from the end of the forward filtering mission. However, real-time processing is necessary for most navigation applications. Thus, an online smoothing method is proposed in this paper for ZUPT-aided INS by combining the reverse navigation with the smoothing algorithm.

The paper is organised as follows. In Section 2, the traditional ZUPT method based on Kalman filter is briefly reviewed. In Section 3, the reverse navigation algorithm and its error model are introduced. In Section 4, an online smoothing method for ZUPT based on reverse navigation is elaborated. In Section 5, the test results as well as an accuracy verification are reported. The last section is conclusion and discussion.

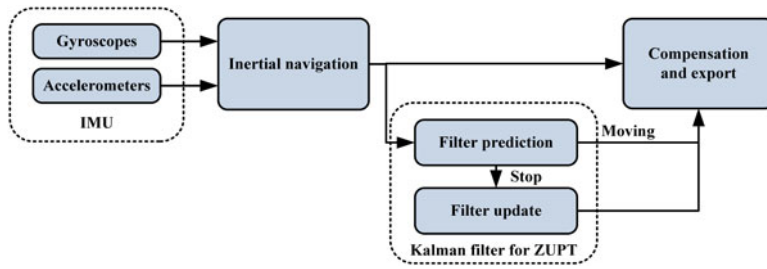


Figure 1. Principle of zero velocity update.

2. ZERO VELOCITY UPDATE.

2.1. *Principle of Zero Velocity Update.* The principle of ZUPT is shown in Figure 1. With the inertial navigation calculation, a Kalman filter is used to estimate the navigation errors by observing velocity error when the vehicle stops. The states of the Kalman filter including the navigation errors (position, velocity and attitude errors) and the sensor errors (gyro and accelerometer biases) are required. The state equation is the error function of the navigation algorithm. When the vehicle is moving, only the filter prediction step is carried out; when the vehicle stops, the filter updating step is carried out by using the observations of velocity error. The navigation results are exported after compensated by the errors estimated by the filter.

2.2. *Strapdown INS Algorithm and Error Model.* Most of the land vehicle and pedestrian INSs adopt micromechanical or optical gyroscope strapdown IMUs. In a strapdown INS, the IMU is mounted on the vehicle or the body without a static platform. Angular velocity and acceleration are measured by gyroscopes and accelerometers in the body frame (denoted by *b*). A mathematical platform is established in the computer to calculate the navigation results (attitude, velocity and position). The local geographical frame (E-N-U frame) is chosen as the navigation frame (denote by *n*), the strapdown INS algorithm can be written as:

$$\begin{aligned}
 \dot{\mathbf{C}}_b^n &= \mathbf{C}_b^n \boldsymbol{\Omega}_{nb}^b \\
 \dot{\mathbf{v}}^n &= \mathbf{C}_b^n \mathbf{f}^b - (2\boldsymbol{\omega}_{ie}^n + \boldsymbol{\omega}_{en}^n) \times \mathbf{v}^n + \mathbf{g}^n \\
 \dot{L} &= v_N^n / (R_N + h), \quad \dot{\lambda} = v_E^n \sec L / (R_E + h), \quad \dot{h} = v_U^n
 \end{aligned}
 \tag{1}$$

where  $\boldsymbol{\Omega}_{nb}^b = (\boldsymbol{\omega}_{nb}^b \times)$ ,  $\boldsymbol{\omega}_{nb}^b = \boldsymbol{\omega}_{ib}^b - (\mathbf{C}_b^n)^T (\boldsymbol{\omega}_{ie}^n + \boldsymbol{\omega}_{en}^n)$ ,  $\mathbf{g}^n = [0 \ 0 \ -g]^T$ ,  $\boldsymbol{\omega}_{ie}^n = [0 \ \omega_{ie} \cos L \ \omega_{ie} \sin L]^T$ ,  $\boldsymbol{\omega}_{en}^n = [-v_N^n / (R_N + h) \ v_E^n / (R_E + h) \ v_U^n \tan L / (R_E + h)]^T$ .

$\mathbf{C}_b^n$  is the direction cosine matrix to transform vectors from body frame to navigation frame;  $\mathbf{v}^n$  represents the vector defined by the velocity directions of east, north and up in the navigation frame,  $\mathbf{v}^n = [v_E^n \ v_N^n \ v_U^n]^T$ ;  $L$ ,  $\lambda$  and  $h$  are the latitude, longitude and height, respectively;  $\boldsymbol{\omega}_{ib}^b$  represents the angular velocity vector with respect to the inertial space measured by the gyroscope triad in the body frame;  $\mathbf{f}^b$  is the specific force vector measured by the accelerometer triad;  $\omega_{ie}$  and  $g$  are the self-rotation angular velocity and the gravity acceleration of the earth;  $R_N$  and  $R_E$  are the meridian radius and the transverse radius of curvature, respectively;  $\boldsymbol{\Omega}_{nb}^b$  is the skew symmetric

matrix constituted by the element of  $\omega_{nb}^b = [\omega_{nbx}^b \ \omega_{nby}^b \ \omega_{nbz}^b]^T$  (symbolised by

$$(\omega_{nb}^b \times) = \begin{bmatrix} 0 & -\omega_{nbz}^b & \omega_{nby}^b \\ \omega_{nbz}^b & 0 & -\omega_{nbx}^b \\ -\omega_{nby}^b & \omega_{nbx}^b & 0 \end{bmatrix} \text{ in this paper.}$$

According to the algorithm, the error model of the strapdown INS in Equation (1) can be written as (Titterton and Weston, 1997):

$$\begin{aligned} \dot{\psi} &= -\psi \times \omega_{in}^n + \delta\omega_{in}^n - C_b^n \delta\omega_{ib}^b \\ \delta\dot{v}^n &= f^n \times \psi - (2\omega_{ie}^n + \omega_{en}^n) \times \delta v^n - (2\omega_{ie}^n + \omega_{en}^n) \times v^n - \delta g^n + C_b^n \delta f^b \\ \delta\dot{L} &= \delta v_N^n / (R_N + h) - \delta h \cdot v_N^n / (R_N + h)^2 \\ \delta\dot{\lambda} &= \delta v_E^n \sec L / (R_E + h) + \delta L \cdot v_E^n \tan L \sec L / (R_E + h) - \delta h \cdot v_E^n \sec L / (R_E + h)^2 \\ \delta\dot{h} &= \delta v_U^n \end{aligned} \tag{2}$$

where

$$\begin{aligned} \omega_{in}^n &= \omega_{ie}^n + \omega_{en}^n, \delta\omega_{in}^n = \delta\omega_{ie}^n + \delta\omega_{en}^n, \delta\omega_{ie}^n = [0 \quad -\delta L \cdot \omega_{ie} \sin L \quad \delta L \cdot \omega_{ie} \cos L]^T, \\ \delta\omega_{en}^n &= \begin{bmatrix} -\delta v_N^n / (R_N + h) + \delta h \cdot v_N^n / (R_N + h)^2 \\ \delta v_E^n / (R_E + h) - \delta h \cdot v_E^n / (R_E + h)^2 \\ \delta v_E^n \tan L / (R_E + h) + \delta L \cdot v_E^n \sec^2 L / (R_E + h) - \delta h \cdot v_E^n \tan L / (R_E + h)^2 \end{bmatrix}. \end{aligned}$$

$\psi$  is the attitude error vector in the mathematical platform,  $\psi = [\delta\alpha \ \delta\beta \ \delta\gamma]^T$ .  $\delta\alpha$ ,  $\delta\beta$  and  $\delta\gamma$  are the three attitude error elements, which can be replaced by the roll, pitch and yaw Euler errors for small angle misalignments;  $\delta v$  is the velocity error vector,  $\delta v = [\delta v_E \ \delta v_N \ \delta v_U]^T$ . The elements  $\delta v_E$ ,  $\delta v_N$  and  $\delta v_U$  denote the velocity errors in the east, north and up direction respectively;  $\delta L$ ,  $\delta\lambda$  and  $\delta h$  stand for the latitude, longitude and height error respectively;  $\delta g^n$  is the gravity anomaly vector;  $\delta\omega_{ib}^b$  and  $\delta f^b$  are the error vectors of gyroscope and accelerometer triad respectively.

2.3. *Kalman Filter.* In this paper, the attitude errors, velocity errors, position errors, the gyroscope biases ( $\varepsilon_x$ ,  $\varepsilon_y$  and  $\varepsilon_z$ ) and the accelerometer biases ( $\nabla_x$ ,  $\nabla_y$  and  $\nabla_z$ ) are chosen as the state of the Kalman filter:

$$\begin{aligned} x &= [\delta\alpha \ \delta\beta \ \delta\gamma \ \delta v_e \ \delta v_n \ \delta v_u \ \delta L \ \delta\lambda \ \delta h \\ &\quad \varepsilon_x \ \varepsilon_y \ \varepsilon_z \ \nabla_x \ \nabla_y \ \nabla_z]^T \end{aligned} \tag{3}$$

Assuming that the gyroscope and accelerometer measurement errors are constant biases and ignoring the gravity anomaly, the error model can be expressed in a discrete form as:

$$x_k = \Phi_{k-1,k} x_{k-1} + w_k \tag{4}$$

where  $x_k$  is the state vector at time  $k$ ;  $\Phi_{k-1,k}$  is the state transition matrix from time  $k-1$  to  $k$ ;  $w_k$  is the system noise.

When the vehicle or the body stops, the velocity of the system should be zero in theory. Thus, the observation velocity is expressed as  $\mathbf{v}_{obv}^n = [0 \ 0 \ 0]^T$ . The measurement equation can be derived as:

$$\mathbf{z}_k = (\mathbf{v}^n - \mathbf{v}_{obv}^n) = \mathbf{H}_k \mathbf{x}_k + \mathbf{v}_k \tag{5}$$

where  $\mathbf{z}_k$  is the aiding measurement,  $\mathbf{H}_k = [0_{3 \times 3} \ \mathbf{I}_{3 \times 3} \ 0_{3 \times 9}]$  is the measurement matrix and  $\mathbf{v}_k$  is the measurement noise.

In the Kalman filter, the system and measurement noise are considered as a Gaussian distribution:

$$\mathbf{w}_k \sim N(0, \mathbf{Q}_k), \mathbf{v}_k \sim N(0, \mathbf{R}_k) \tag{6}$$

The Kalman filter calculation steps are:

Prediction:

$$\hat{\mathbf{x}}_{k/k-1} = \Phi_{k-1,k} \hat{\mathbf{x}}_{k-1} \tag{7}$$

$$\mathbf{P}_{k/k-1} = \Phi_{k-1,k} \mathbf{P}_{k-1} \Phi_{k-1,k}^T + \mathbf{Q}_k \tag{8}$$

Updating:

$$\mathbf{K}_k = \mathbf{P}_{k/k-1} \mathbf{H}_k^T [\mathbf{H}_k \mathbf{P}_{k/k-1} \mathbf{H}_k^T + \mathbf{R}_k]^{-1} \tag{9}$$

$$\hat{\mathbf{x}}_k = \hat{\mathbf{x}}_{k/k-1} + \mathbf{K}_k [\mathbf{z}_k - \mathbf{H}_k \hat{\mathbf{x}}_{k/k-1}] \tag{10}$$

$$\mathbf{P}_k = \mathbf{P}_{k/k-1} - \mathbf{K}_k \mathbf{H}_k \mathbf{P}_{k/k-1} \tag{11}$$

where  $\hat{\mathbf{x}}_{k/k-1}$  and  $\mathbf{P}_{k/k-1}$  are the predicted state and the covariance at time  $k$  with the given information at time  $k-1$ ;  $\hat{\mathbf{x}}_{k-1}$  and  $\mathbf{P}_{k-1}$  are the estimated state and covariance at time  $k-1$ ;  $\hat{\mathbf{x}}_k$  and  $\mathbf{P}_k$  are the estimated state and covariance at time  $k$ .

### 3. REVERSE NAVIGATION ALGORITHM

3.1. *Discrete Strapdown INS Algorithm.* In practice, in order to perform the recursion calculation in a computer, the strapdown INS algorithm in Equation (1) should be discretised:

$$\begin{aligned} \mathbf{C}_{bk}^n &= \mathbf{C}_{bk-1}^n (\mathbf{I} + T_s \boldsymbol{\Omega}_{nbk}^b) \\ \mathbf{v}_k^n &= \mathbf{v}_{k-1}^n + T_s [\mathbf{C}_{bk-1}^n \mathbf{f}_k^b - (2\boldsymbol{\omega}_{iek-1}^n + \boldsymbol{\omega}_{enk-1}^n) \times \mathbf{v}_{k-1}^n + \mathbf{g}^n] \\ L_k &= L_{k-1} + T_s v_{Nk-1}^n / (R_N + h_{k-1}) \\ \lambda_k &= \lambda_{k-1} + T_s v_{Ek-1}^n \sec L_{k-1} / (R_E + h_{k-1}) \\ h_k &= h_{k-1} + T_s v_{Uk-1}^n \end{aligned} \tag{12}$$

where

$$\begin{aligned} \boldsymbol{\Omega}_{nbk}^b &= (\boldsymbol{\omega}_{nbk}^b \times); \\ \boldsymbol{\omega}_{nbk}^b &= \boldsymbol{\omega}_{ibk}^b - (\mathbf{C}_{bk-1}^n)^T (\boldsymbol{\omega}_{iek}^n + \boldsymbol{\omega}_{en}^n); \quad \boldsymbol{\omega}_{enk}^n = \begin{bmatrix} -v_{Nk}^n / (R_N + h_k) \\ v_{Ek}^n / (R_E + h_k) \\ v_{Ek}^n \tan L_k / (R_E + h_k) \end{bmatrix}; \\ \boldsymbol{\omega}_{iek}^n &= [0 \ \omega_{ie} \cos L_k \ \omega_{ie} \sin L_k]^T; \end{aligned}$$

$k$  is each discretised time point of the navigation process ( $k = 1, 2, 3, \dots$ ).

3.2. *Reverse Strapdown INS Algorithm.* Assuming that the INS navigates from position  $A$  to position  $B$  from time  $t_0$  to  $t$ , Equation (12) can be transformed to Equation (13) to navigate from position  $B$  back to position  $A$  reversely (Yan et al., 2008).

$$\begin{aligned}
 \mathbf{C}_{bk-1}^n &= \mathbf{C}_{bk}^n (\mathbf{I} + T_s \mathbf{\Omega}_{nbk}^b)^{-1} \approx \mathbf{C}_{bk}^n (\mathbf{I} - T_s \mathbf{\Omega}_{nbk}^b) \approx \mathbf{C}_{bk}^n (\mathbf{I} + T_s \tilde{\mathbf{\Omega}}_{nbk-1}^b) \\
 \mathbf{v}_{k-1}^n &= \mathbf{v}_k^n - T_s [\mathbf{C}_{bk-1}^n \mathbf{f}_k^b - (2\boldsymbol{\omega}_{iek-1}^n + \boldsymbol{\omega}_{enk-1}^n) \times \mathbf{v}_{k-1}^n + \mathbf{g}^n] \\
 &\approx \mathbf{v}_k^n - T_s [\mathbf{C}_{bk-1}^n \mathbf{f}_{k-1}^b - (2\boldsymbol{\omega}_{iek}^n + \boldsymbol{\omega}_{enk}^n) \times \mathbf{v}_k^n + \mathbf{g}^n] \\
 L_{k-1} &= L_k - T_s v_{Nk-1}^n / (R_N + h_{k-1}) \approx L_k - T_s v_{Nk}^n / (R_N + h_k) \\
 \lambda_{k-1} &= \lambda_k - T_s v_{Ek-1}^n \sec L_{k-1} / (R_E + h_{k-1}) \approx \lambda_k - T_s v_{Ek}^n \sec L_k / (R_E + h_k) \\
 h_{k-1} &= h_k - T_s v_{Uk-1}^n \approx h_k - T_s v_{Uk}^n
 \end{aligned} \tag{13}$$

where  $\tilde{\mathbf{\Omega}}_{nbk-1}^b = (\tilde{\mathbf{\Omega}}_{nbk-1}^b \times)$ ;  $\tilde{\mathbf{\Omega}}_{nbk-1}^b = -[\boldsymbol{\omega}_{ibk-1}^b - (\mathbf{C}_{bk}^n)^T (\boldsymbol{\omega}_{iek}^n + \boldsymbol{\omega}_{enk}^n)]$ ;

If the reverse velocity is defined as  $\mathbf{v}_{rk}^n = -\mathbf{v}_k^n$ , the reverse strapdown INS algorithm can be written as:

$$\begin{aligned}
 \mathbf{C}_{bk-1}^n &= \mathbf{C}_{bk}^n (\mathbf{I} + T_s \tilde{\mathbf{\Omega}}_{nbrk-1}^b) \\
 \mathbf{v}_{rk-1}^n &= \mathbf{v}_{rk}^n + T_s [\mathbf{C}_{bk-1}^n \mathbf{f}_{k-1}^b - (-2\boldsymbol{\omega}_{iek}^n + \boldsymbol{\omega}_{enrk}^n) \times \mathbf{v}_{rk}^n + \mathbf{g}^n] \\
 L_{k-1} &= L_k + T_s v_{Nrk}^n / (R_N + h_k) \\
 \lambda_{k-1} &= \lambda_k + T_s v_{Erk}^n \sec L_k / (R_E + h_k) \\
 h_{k-1} &= h_k + T_s v_{Urk}^n
 \end{aligned} \tag{14}$$

where  $\tilde{\mathbf{\Omega}}_{nbrk-1}^b = (\tilde{\mathbf{\Omega}}_{nbrk-1}^b \times)$ ;  $\tilde{\mathbf{\Omega}}_{nbrk-1}^b = -\boldsymbol{\omega}_{ibk-1}^b - (\mathbf{C}_{bk}^n)^T (-\boldsymbol{\omega}_{iek}^n + \boldsymbol{\omega}_{enrk}^n)$ ;

$$\boldsymbol{\omega}_{enrk}^n = \begin{bmatrix} -v_{Nrk}^n / (R_N + h_k) \\ v_{Erk}^n / (R_E + h_k) \\ v_{Erk}^n \tan L_k / (R_E + h_k) \end{bmatrix}.$$

It should be pointed out that, the attitude and position results in the reverse algorithm are the same as those in the forward algorithm, but the velocity result in the reverse algorithm is opposite. Comparing Equation (12) with Equation (14), it can be seen that the reverse algorithm is much the same as the forward algorithm except the sign of gyroscope measurement and self-rotation angular velocity of the earth are opposite. To sum up, the error model of the reverse INS algorithm has a similar form to the model of the forward algorithm except the sign of the gyroscope measurement, the gyroscope bias and the earth rotation angular velocity are opposite.

#### 4. ONLINE SMOOTHING BASED ON REVERSE NAVIGATION

4.1. *Principle of Online Smoothing.* The principle of the online smoothing based on reverse navigation proposed in this paper is shown in Figure 2. In the forward navigation processing, the Kalman filter for ZUPT is used as usual and the IMU data are stored in the memory. At each time of the navigation process, the reverse navigation algorithm can be performed using the previous IMU data. The initial values of the reverse navigation algorithm are the forward navigation result at this moment. In the reverse navigation algorithm, a reverse Kalman filter for ZUPT is used, which is

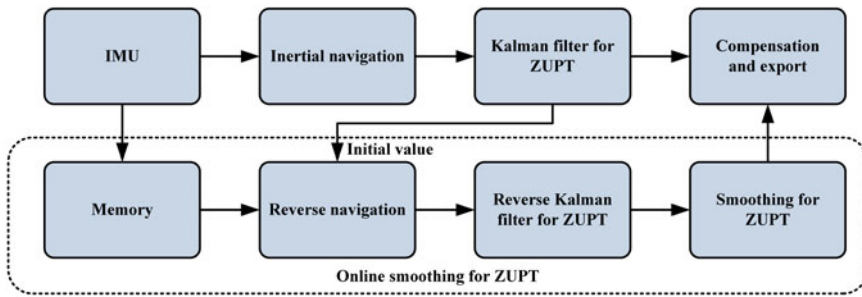


Figure 2. Integrated architecture of online smoothing for ZUPT based on reverse navigation.

much the same as the forward one except that the sign of the gyroscope measurement, the gyroscope bias and the earth rotation angular velocity in the state function are opposite. In the reverse navigation algorithm, the fixed-point smoothing method can be used to estimate the navigation errors at the real-time point, while the Rauch-Tung-Striebel (RTS) smoothing can be used to estimate the navigation error of all points in the trajectory including the real-time point. Because the reverse navigation and the smoothing are both reversed processing algorithm, the smoothing in the reverse navigation algorithm turns into forward processing, which can realise online processing in theory and thereby estimate the real-time point or the end point.

The process of the online smoothing proposed in this paper is shown in Figure 3. Each line expresses one smoothing calculation process. Taking land vehicle INS for example, the vehicle stops for 10–15 minutes as initial alignment time before navigation. Then, the vehicle also stops for 30–60 seconds for ZUPT after travelling every 8–10 minutes. In the forward navigation algorithm, the Kalman filter performs the updating equation of the Kalman filter (Equations (7) and (8)) by observing the velocity error when the vehicle stops and only carries out the prediction equation of the Kalman filter (Equations (9)–(11)) when the vehicle is travelling. At the same time, the online smoothing proposed in this paper can be performed during the navigation. At each point (normally at the stop point), the reverse navigation (Equation (14)) can be processed using the past IMU data. In the reverse navigation algorithm, when the vehicle is traveling, only the filter prediction step (Equations (7) and (8)) is carried out; when the vehicle stops, the filter updating (Equations (9)–(11)) is carried out by observing the velocity error; when the reverse navigation algorithm is calculated back to the start point, the filter updating is carried out by observing the velocity error and the position error. Then, reversed processing smoothing algorithms such as the RTS smoothing (Equations (18)–(21)) can be carried out in the forward process. Because the position information of the start point is used in the online smoothing algorithm, the position precision can be better than the traditional offline smoothing algorithm.

**4.2. Kalman Smoothing Algorithm.** In the online smoothing method proposed in this paper, various Kalman smoothing algorithms can be used. In practice, a suitable smoothing algorithm can be chosen according to the requirement.

**4.2.1. Fixed-point smoothing.** The fixed-point smoothing is used to estimate the states in a fixed time by all observable information after that time. The calculation steps of fixed-point smoothing are as below.

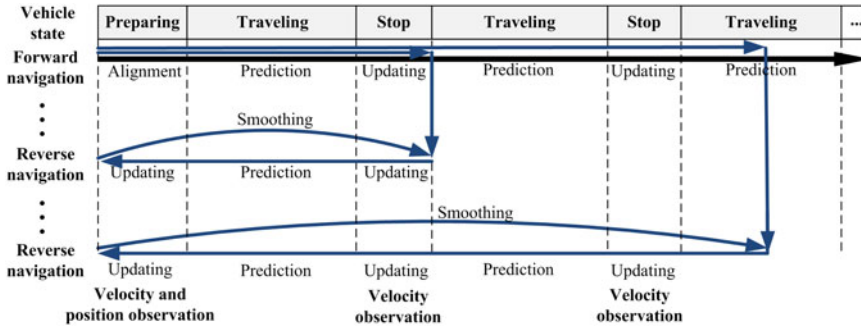


Figure 3. Process of online smoothing for ZUPT.

From the initial time, the updated states  $\hat{x}_k$  and covariance  $P_k$  are calculated by the Kalman filter. When the filtering time  $k$  is equal to or later than the smoothing time  $j$ , the state vector at the smoothing time can be estimated by the iterative equation after the Kalman filter:

$$\hat{x}_{j/k} = \hat{x}_{j/k-1} + K_k^a (z_k - H_k \hat{x}_{j/k-1}) \tag{15}$$

$$K_k^a = P_{k/k-1}^a H_k^T (H_k P_{k/k-1}^a H_k^T + R_k)^{-1} \tag{16}$$

The iterative equation of covariance matrix  $P_{k+1/k}^a$  in the fixed-point smoother can be presented as:

$$P_{k+1/k}^a = P_{k/k-1}^a (\Phi_{k+1,k} - \Phi_{k+1,k} K_k H_k^T)^T \tag{17}$$

Because the fixed-point smoothing does not need to store the filtering information, it has a low requirement of the computer’s speed and storage space, which applies to the online smoothing, but only one point can be estimated in each smoothing operation.

4.2.2. *RTS smoothing.* The RTS smoothing is also called fixed-interval smoothing, which can estimate the states in the whole trajectory using discontinuous observable information (Gong et al., 2013; Liu et al., 2010). The calculation steps of the RTS smoother are as below.

While the Kalman filter is working, the predicted states  $\hat{x}_{k/k-1}$  and covariance  $P_{k/k-1}$ , the updated states  $\hat{x}_k$  and covariance  $P_k$  are all stored in memory for smoothing later. We assume that there are  $M$  times in the whole trajectory, and each time can be denoted as  $j$  ( $0 < j < M$ ). After the calculation of the Kalman filter, the RTS smoother begins at time  $M$ . With  $j = M, M - 1, \dots, 1$ , the iterative equation of the state vector in the RTS smoother can be written as:

$$\hat{x}_{j-1/M} = \hat{x}_{j-1} + A_{j-1} (\hat{x}_{j/M} - \hat{x}_{j/j-1}) \tag{18}$$

$$A_{j-1} = P_{j-1} \Phi_{j,j-1}^T P_{j/j-1}^{-1} \tag{19}$$



The iterative equation of the covariance matrix in the RTS smoother can be presented as:

$$\mathbf{P}_{j-1/M} = \mathbf{P}_{j-1} + \mathbf{A}_{j-1}(\mathbf{P}_{j/M} - \mathbf{P}_{j/j-1})\mathbf{A}_{j-1}^T \quad (20)$$

where the updated states and covariance at time  $M$  of the Kalman filter is the initial value of the RTS smoother.

$$\hat{\mathbf{x}}_{M/M} = \hat{\mathbf{x}}_M \quad \mathbf{P}_{M/M} = \mathbf{P}_M \quad (21)$$

In this paper, the RTS smoothing is chosen to perform a high-accuracy estimation of the navigation errors. It should be pointed out that because the RTS smoothing has to store the filtering information during the navigation, it has a relatively high requirement of the computer's processing speed and storage space. To improve the software for making the online smoothing algorithm applicable to common hardware, the calculation frequency of the smoothing process can be reduced from 100 Hz to 1 Hz or even lower. In this way, only the averaged data for each second needs to be stored, and the storage space and the processing speed required of the computer can be reduced to 1% or lower.

## 5. FIELD TESTS

5.1. *Test Arrangement.* Field navigation tests were carried out in Changsha to verify the effectiveness of the online smoothing method for ZUPT-aided INS. Two IMUs with the same level of accuracy were used in the tests. Both IMUs consisted of three Ring Laser Gyros with an accuracy of  $0.003^\circ/\text{h}$  and three quartz accelerometers with an accuracy of  $10 \mu\text{g}$ . An indexing mechanism with an accuracy of  $5''$  was used for estimating the inertial sensor biases during the alignment. The sampling frequency was 100 Hz. A Differential GPS (DGPS) with an accuracy of 3 cm was used to provide the reference position information for comparison. The test car, the DGPS base station and the installation of the IMU on the indexing mechanism and on the vehicle are shown in Figure 4.

To verify the method used in different IMUs, paths and stop intervals, six navigation tests (listed in Table 1) were designed.

5.2. *Accuracy Test Results.* In the first two tests, IMU1 was used to navigate for 2 h and 1 h 50 min respectively. During the alignment, the azimuth axis of the IMU was turned  $90^\circ$  by the indexing mechanism to estimate and compensate the biases of the inertial sensors. During the navigation, the test car stopped for 30 s every 10 min for ZUPT. The navigation results of Test 1 and Test 2 are shown in Figures 5 and 6. In each figure, the navigation trajectory is shown in Figure (a), the errors of inertial navigation without ZUPT are shown in Figure (b), the errors with traditional ZUPT mentioned in Section 2 are shown in Figure (c) and the errors with online smoothing ZUPT proposed in this paper are shown in Figure (d), respectively. We can see from the figures that the max position error of inertial navigation without ZUPT is about 500 m over 2 h. The position error at stop points can be compensated to 5.5 m by ZUPT-aiding, but when the car was running, the position error accumulated to 15 m in 10 min. With the online smoothing ZUPT-aiding, the maximum position error during the whole trajectory was restrained to below 5 m in both tests. It has to be pointed out that the unconventional points such as the 20 m error in Figure 6(c)

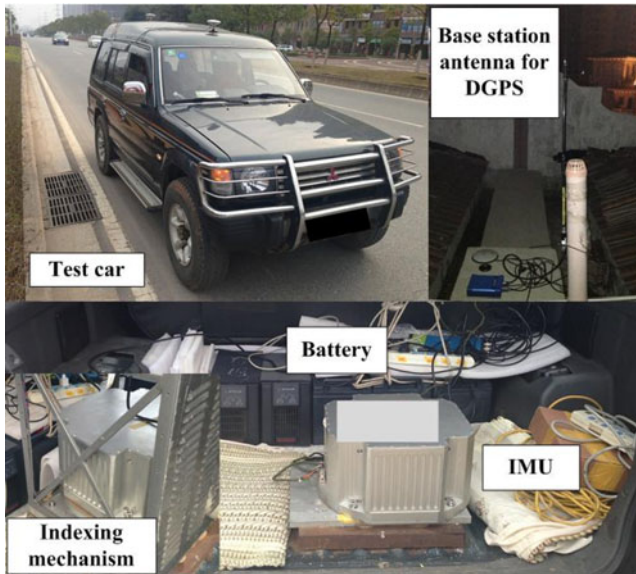


Figure 4. Field test facilities.

Table 1. Introduction of the navigation tests.

Test number	IMU	Navigation time	Trajectory	Alignment	Stop interval
Test1	IMU1	2 h	Random	Two-direction	10 min
Test2	IMU1	1h50 min	Random	Two-direction	10 min
Test3	IMU2	1h20 min	Random	Two-direction	10 min
Test4	IMU2	1h20 min	Straight and return	One-direction	10 min
Test5	IMU2	2 h	Random	One-direction	15 min
Test6	IMU2	2 h	Random	One-direction	15 min

and Figure 6(d) are caused by a break in the DGPS signal when the car traversed through bridges and other obstructions. That is also the reason why the position error at Stop point 5 in Test 2 cannot be obtained.

In Test 3 and Test 4 the method is verified using a different IMU (IMU2). Each test lasted for 1 h 20 min, and the test car also stopped for 30 s every 10 min for ZUPT. In Test 3, the biases of the inertial sensors were also estimated and compensated by the indexing mechanism, and the path was also random as in Test 1 and Test 2. The navigation result is similar to the first two tests as shown in Figure 7. The maximum position error at the stop points was decreased to 4.6 m by ZUPT-aiding, and during the whole trajectory was restrained to 4.3 m by the online smoothing ZUPT-aiding. It further illustrates the efficiency of the online smoothing method on high-accuracy IMU.

The position errors in Test 4 are larger than those in the previous three tests, especially at the third stop point, which is twice as large as the max error in the previous tests. This is because during the alignment and the navigation before the third stop

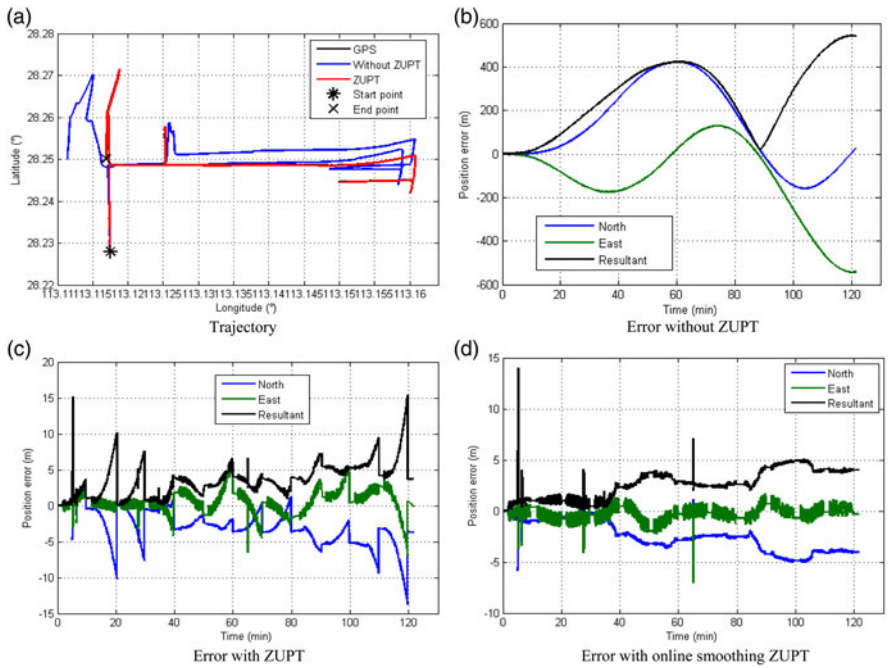


Figure 5. Results of Test 1.

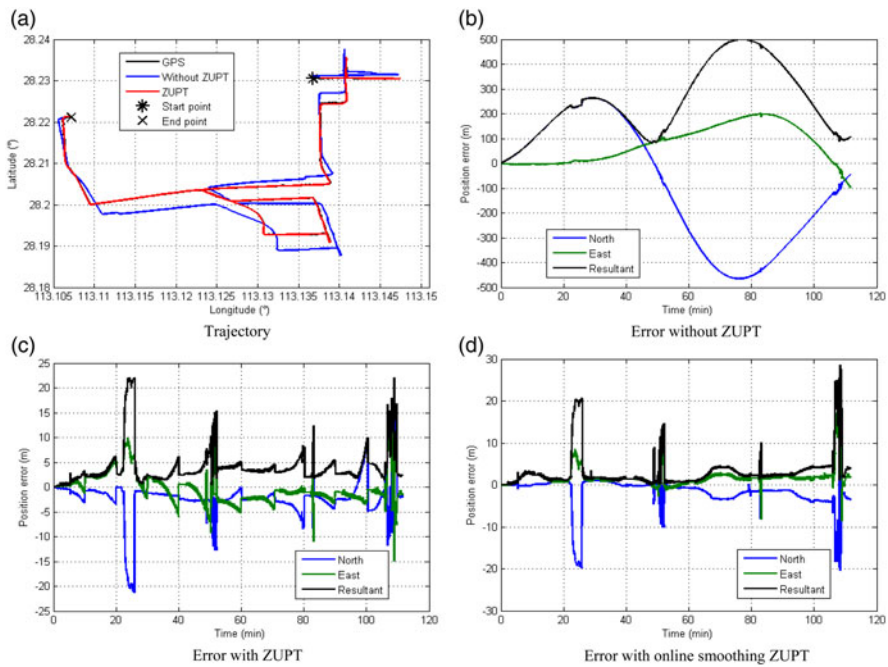


Figure 6. Results of Test 2.

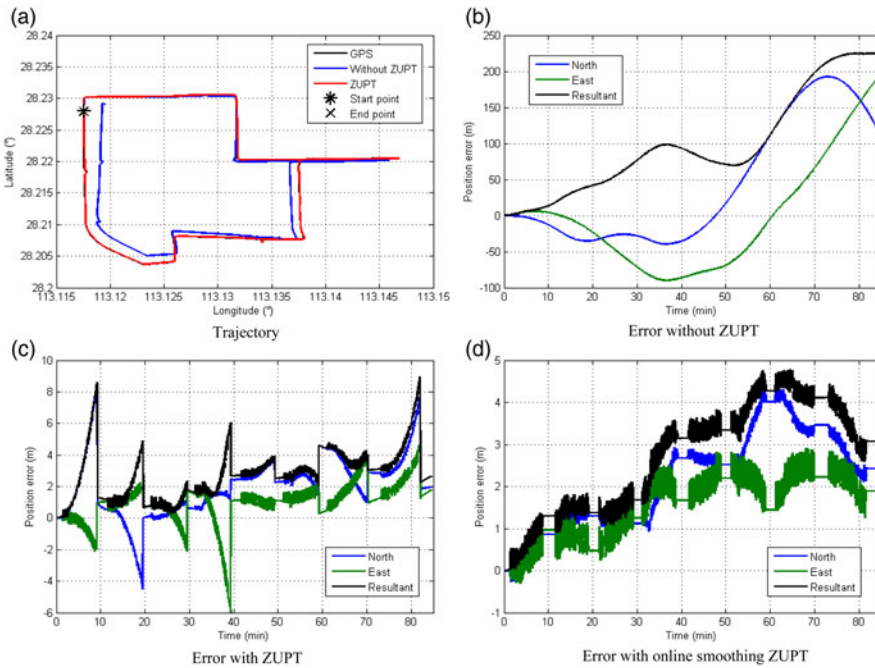


Figure 7. Results of Test 3.

point, the orientation of the IMU had not changed. Under this condition, the horizontal attitude angle errors are coupled with the accelerometer bias as well as the azimuth error and the gyro bias, and therefore some states of the Kalman filter cannot be convergent. After the vehicle turned 180°, the errors were decoupled and all states of the Kalman filter were convergent so that all the errors were estimated at the next stop point by observing the velocity error. Thus the position error was restrained after the fifth stop point. We can infer from the result that turning of the vehicle is necessary for a ZUPT-aiding INS to improve the observability and ensure the Kalman filter is convergent. According to the test result, the issue of low observability in straight trajectory can be solved by adding a single-axis indexing mechanism. The errors can be decoupled by only one 90° rotation around the azimuth axis during alignment with no need of rotation during navigation. In addition, it can be seen by comparing Figure 8(c) with Figure 8(d) that the online smoothing algorithm can clearly improve the accuracy of ZUPT under bad observability conditions as well.

The test conditions in Test 5 and Test 6 were the same as in Test 3, except the stop interval was extended to 15 min. Test results are shown in Figure 9 and Figure 10. The maximum position error at stop points was compensated to 8.7 m by ZUPT-aiding, and during the whole trajectory it was restrained from 70 m to 14.4 m by the online smoothing ZUPT-aiding in both tests. This proves that the online smoothing ZUPT method can maintain a high accuracy estimation even if the observation interval is extended to 15 min.

Statistical results of the six tests are shown in Table 2. When the stop interval is 10 min, the max position error at the stop points is 5.5 m by ZUPT-aiding, and it is 4.9 m

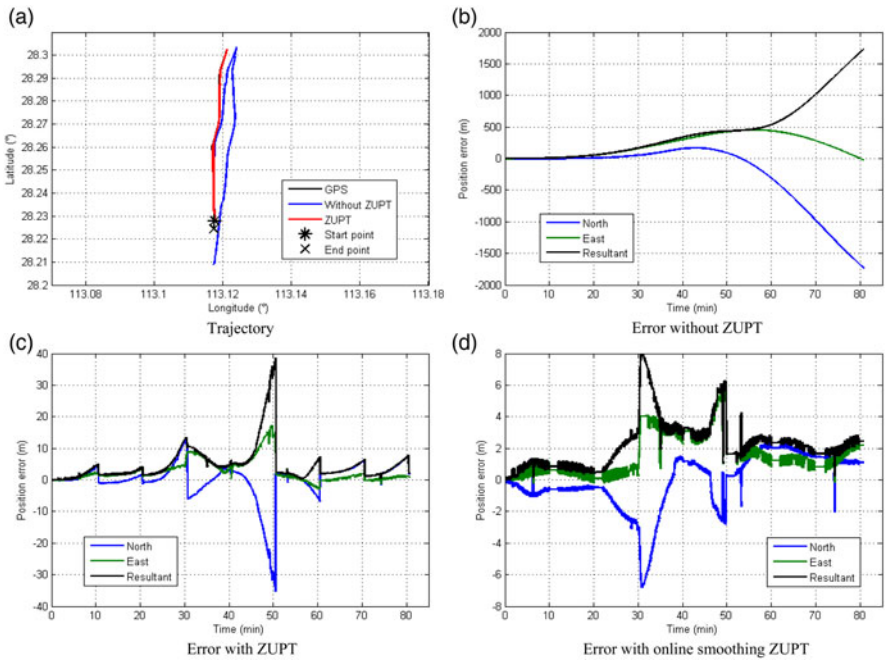


Figure 8. Results of Test 4.

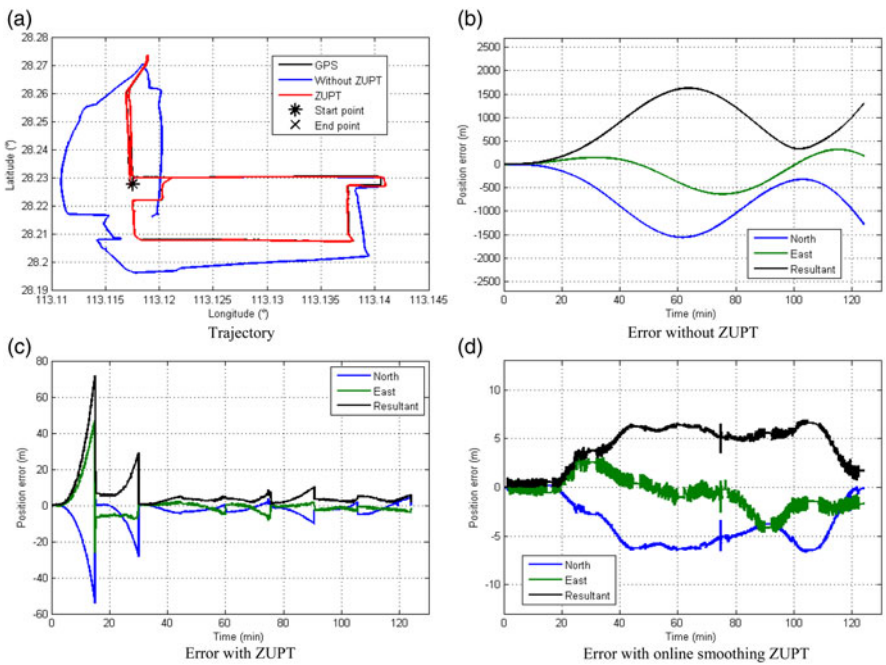


Figure 9. Results of Test 5.



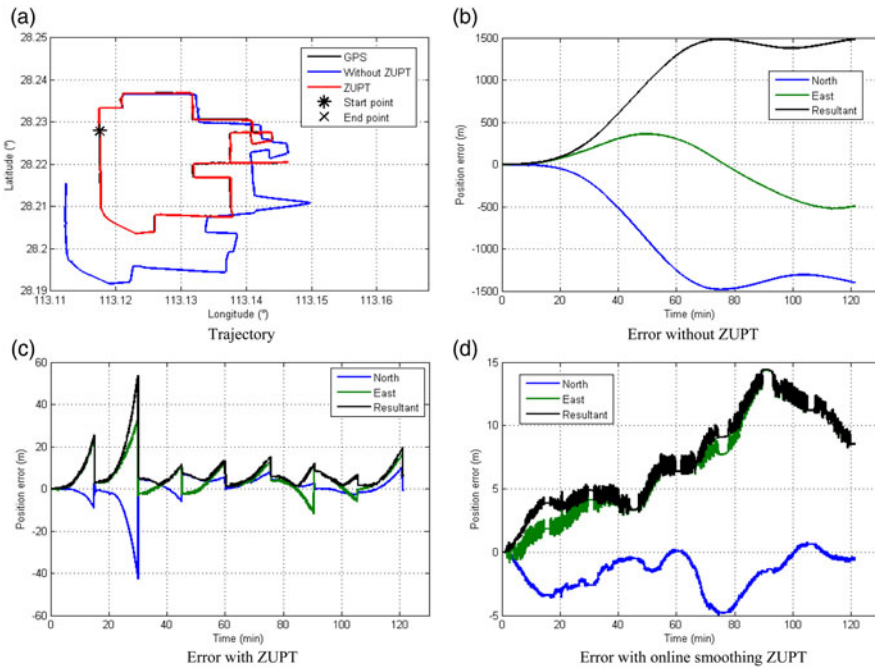


Figure 10. Results of Test 6.

Table 2. Statistical Results of the Six Tests.

Position error	Test1	Test2	Test3	Test4	Test5	Test6
Stop point 1	1.0 m	2.6 m	1.3 m	1.7 m	6.6 m	3.2 m
Stop point 2	0.2 m	2.5 m	0.7 m	1.6 m	0.7 m	4.8 m
Stop point 3	0.8 m	2.4 m	1.8 m	10.3 m	3.8 m	7.1 m
Stop point 4	3.8 m	2.3 m	2.7 m	5.0 m	3.9 m	1.2 m
Stop point 5	2.8 m	—	2.5 m	2.7 m	0.9 m	5.8 m
Stop point 6	4.0 m	3.4 m	4.6 m	2.0 m	3.5 m	8.7 m
Stop point 7	3.8 m	3.0 m	3.0 m	1.2 m	5.0 m	1.6 m
Stop point 8	4.0 m	2.0 m	2.2 m	2.1 m	2.0 m	6.0 m
Stop point 9	5.5 m	2.6 m	—	—	—	—
Stop point 10	5.2 m	5.0 m	—	—	—	—
Stop point 11	4.2 m	3.1 m	—	—	—	—
Stop point 12	3.7 m	—	—	—	—	—
Max error at stop points	5.5 m	5.0 m	4.6 m	10.3 m	6.6 m	8.7 m
Max error during the whole trajectory	4.9 m	4.2 m	4.3 m	7.9 m	6.6 m	14.4 m

during the whole trajectory by online smoothing ZUPT-aiding except the unusual points in Test 4. When the stop interval extends to 15 min, the max position error at the stop points was 8.7 m by ZUPT-aiding, and it was 14.4 m during the whole trajectory by the online smoothing ZUPT-aiding. This proves that by using the proposed online smoothing ZUPT method, the position error in the whole trajectory can be

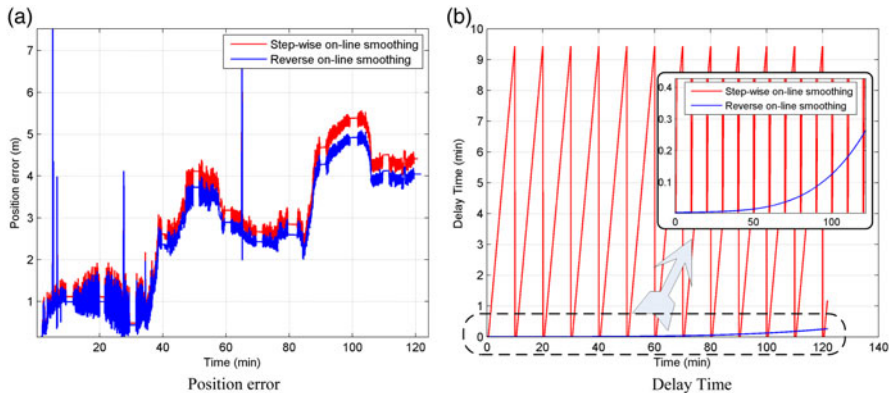


Figure 11. Results of delay test results.

restrained to the same level of those at the stop points. The test results reach quite a high level accuracy in a state-of-art technical condition, which verifies the advancement of online smoothing ZUPT method proposed in this paper.

**5.3. Delay Test Results.** A delay test with a calculation frequency of 1 Hz was designed for calculating the position accuracy and the delay time of the reverse online smoothing method proposed in this paper. A commonly used computer with a 4 GHz quad-core processor and a 120 GB solid-state drive is adopted in the test. The 100 Hz gyro and accelerometer data of Test 1 is read for each 10 ms to simulate the online condition and the delay time is recorded by computer. The test results of existing step-wise online smoothing method and the reverse online smoothing method proposed in this paper are shown in Figure 11. It can be seen from Figure 11(a), that the accuracy of the two smoothing methods are similar and the accuracy with 1 Hz smoothing is no worse than the accuracy of Test 1 in Figure 5. It can be seen from Figure 11(b), limited by the principle, that the step-wise online smoothing method can process a step of smoothing only when the observation is obtained. The largest delay time is about 10 min because the vehicle stops every 10 min. The theoretical delay of the proposed reverse online smoothing method is 1 s because the smoothing calculation frequency is 1 Hz. In practice, this is limited by the computer's calculation speed, and the delay time increases with navigation time because the data size accumulates. However, the delay is no more than 3 s after 1 h navigation and no more than 20 s after 2 h navigation using a common computer, which is much shorter than that with the traditional online smoothing method.

**6. CONCLUSION AND DISCUSSION.** Smoothing methods in ZUPT-aided navigation can effectively restrain position errors not only at stop points but also during the whole trajectory in independent navigation systems. To realise real-time high-accuracy estimation, an online smoothing method for ZUPT-aided INS based on reverse navigation is proposed in this paper. Six tests with two different IMUs are designed to verify the effectiveness of the proposed method. The online smoothing method was performed in different trajectories with different stop intervals. Test results prove that the online smoothing method with ZUPT can improve the position accuracy

during the whole trajectory, including at real time points and the end point, to the level nearly equivalent to the accuracy at the stop points, and the results can be even better when the stop interval is reduced to 10 min or shorter. In the delay test result, the delay is no more than 3 s and 20 s after 1 h and 2 h navigation using a common computer, respectively. This is much shorter than the traditional online smoothing method. The results reach a quite a high level of accuracy in a state-of-art technical condition, which verifies the advancement of the online smoothing ZUPT method proposed in this paper.

The proposed online smoothing method has shown a great potential in land vehicle navigation for the rapidly rising Vehicle Networking and self-driving technology, which can provide high accuracy position and attitude of vehicles in real time in an urban-canyon environment. The method researched in this paper is also helpful to improve the adaptability of ZUPT-aided pedestrian INSs in flexible walking motion conditions, such as while running. In the future, the proposed online smoothing method will be verified in a Micro-Electromechanical Systems (MEMS) pedestrian navigation system.

## REFERENCES

- Aloi, D. and Korniyenko, O. (2007). Comparative performance analysis of a Kalman filter and a modified double exponential filter for GPS-only position estimation of automotive platforms in an urban-canyon environment. *IEEE Transactions on Vehicular Technology*, **56**(5), 2880–2892.
- Bebek, O., Suster, M.A., Rajgopal, S., Fu, M.J., Huang, X., Cavusoglu, M.C., Young, D.J., Mehregany, M., van den Bogert, A.J. and Mastrangelo, C.H. (2010). Personal navigation via high-resolution gait-corrected inertial measurement units. *IEEE Transactions on Instrumentation and Measurement*, **59**(11), 3018–3027.
- Ben, Y.Y., Yin, G., Gao, W. and Sun, F. (2009). Improved filter estimation method applied in zero velocity update for SINS. *IEEE International Conference on Mechatronics and Automation*, Changchun, China, 3375–3380.
- Collin, J. (2015). MEMS IMU Carouseling for Ground Vehicles. *IEEE Transactions on Vehicular Technology*, **64**(6), 2242–2251.
- Colomar, D., Nilsson, J. and Händel, P. (2012). Smoothing for ZUPT-aided INSs. *International Conference on Indoor Positioning and Indoor Navigation*, Sydney, Australia, DOI: 10.1109/IPIN.2012.6418869.
- Correa, A., Barcelo, M., Morell, A. and Vicario, J. (2016). Indoor pedestrian tracking by on-body multiple receivers. *IEEE Sensors Journal*, DOI: 10.1109/JSEN.2016.2518872.
- Elhoushi, M., Georgy, J., Noureldin, A. and Korenberg, M. (2016). Motion mode recognition for indoor pedestrian navigation using portable devices. *IEEE Transactions on Instrumentation and Measurement*, **65**(1), 208–221.
- Fourati, H. (2015). Heterogeneous data fusion algorithm for pedestrian navigation via foot-mounted inertial measurement unit and complementary filter”. *IEEE Transactions on Instrumentation and Measurement*, **64**(1), 221–229.
- Godha, S. and Lachapelle, G. (2008). Foot mounted inertial system for pedestrian navigation. *Measurement Science. Technology*, **19**(7), 697–703.
- Gong, X., Zhang, R. and Fang, J. (2013). Application of unscented R-T-S smoothing on INS/GPS integration system post processing for airborne earth observation. *Measurement*, **46**(3), 1074–1083.
- Grejner-Brzezinska, D.A., Yi, Y., and Toth, C.K. (2001). Bridging GPS gaps in urban canyons: Benefits of ZUPT. *Navigation*, **48**(4), 217–225.
- Huddle, J.R. (1986). Historical perspective on estimation techniques for position and gravity survey with inertial systems. *Journal of Guidance Control and Dynamics*, **9**(3), 257–267.
- Li, L., Pan, Y., Lee, J., Ren, C., Liu, Y., Grejner-Brzezinska, D.A. and Toth, C.K. (2012). Cart-Mounted geolocation system for unexploded ordnance with adaptive ZUPT assistance. *IEEE Transactions on Instrumentation and Measurement*, **61**(4), 974–979.
- Liu, H., Nassar, S. and El-Sheimy, N. (2010). Two-filter smoothing for accurate INS/GPS land-vehicle navigation in urban centers. *IEEE Transactions on Vehicular Technology*, **59**(9), 4256–4267.



- Ojeda, L. and Borenstein, J. (2007). Non-GPS navigation for security personnel and first responders. *The Journal of Navigation*, **60**(3), 391–407.
- Ramanandan, A., Chen, A. and Farrell, J. (2012). Inertial navigation aiding by stationary updates. *IEEE Transaction on Intelligent Transportation Systems*, **13**(1), 235–248.
- Suh, Y.S. (2012). A smoother for attitude and position estimation using inertial sensors with zero velocity intervals. *IEEE Sensors Journal*, **12**(5), 1255–1262.
- Tian, Y., Hamel, W. and Tan, J. (2014). Accurate human navigation using wearable monocular visual and inertial sensors. *IEEE Transactions on Instrumentation and Measurement*, **63**(1), 203–213.
- Titterton, D.H. and Weston, J.L. (1997), *Strapdown Inertial Navigation Technology*. The Institution of Electrical Engineers, Stevenage, U.K.
- Wang, Z., Zhao, H., Qiu, S. and Gao, Q. (2015). Stance-phase detection for ZUPT-aided foot-mounted pedestrian navigation system. *IEEE Transactions on Instrumentation and Measurement*, **20**(6), 3070–3081.
- Yan, G., Yan, W. and Xu, D. (2008). On reverse navigation algorithm and its application to SINS gyro-compass in-movement Alignment. *27th Chinese Control Conference*, Kunming China, 724–729.
- Yang, C., Gao, Z. and Li, D. (2010). Hybrid filter combining with ZUPT for vehicle MINS. *2nd International Asia Conference on Informatics in Control, Automation and Robotics, IEEE*, Wuhan, China, 370–374.
- Yun, X., Calusdian, J., Bachmann, E. and McGhee, R. (2012). Estimation of human foot motion during normal walking using inertial and magnetic sensor measurements. *IEEE Transactions on Instrumentation and Measurement*, **61**(7), 2059–2072.
- Zhang, H., Yuan, W., Shen, Q., T. Li, and Chang, H. (2015). A handheld inertial pedestrian navigation system with accurate step modes and device poses recognition. *IEEE Sensors Journal*, **15**(3), 1421–1429.
- Zhou, C., Downey, J., Stancil, D. and Mukherjee, T. (2010). A low-power shoe-embedded radar for aiding pedestrian inertial navigation. *IEEE Transactions on Microwave Theory and Techniques*, **58**(10), 2521–2528.
- Zihajezadeh, S., Lee, T., Lee, J., Hoskinson, R. and Park, E. (2015). Integration of MEMS inertial and pressure sensors for vertical trajectory determination. *IEEE Transactions on Instrumentation and Measurement*, **64**(3), 804–814.