Posture algorithm for mobile robot based on ultrawide band

Jing Sun¹, Bin Xie¹, Wei Xiong¹, Qiaolong Zhang¹ and Zhenyu Zhao²

Abstract
Inertial measurement unit or other navigation devices are used to obtain the posture information of most mobile robots based on ultrawide band. However, problems, such as zero drift, temperature drift, and error increasing with time, exist in inertial measurement unit and other navigation devices. To overcome these problems, a posture algorithm for mobile robot based on ultrawide band is proposed in this article. Real-time posture information can be obtained by the posture algorithm without using inertial measurement unit and other navigation devices. Simulated results show that the mean absolute error and root mean square error of posture angle in the global coordinate system are 0.951° and 1.169°, respectively. Both the simulated and experimental trajectories show good agreement with the ideal trajectory. The results indicate that the posture data can be obtained accurately by the posture algorithm and the accuracy can meet the requirements for most mobile robots. This article is expected to be of great guiding significance for the development of mobile robots.

Keywords
UWB, mobile robot, posture information, posture algorithm

Date received: 15 February 2022; accepted: 6 July 2022

Introduction
Positioning refers to the process of confirming the position of a robot in its operating environment. One of the most basic and important functions of a mobile robot is to perceive environmental information by using sensors to obtain reliable positioning.¹ Ultrawide band (UWB) technology is widely used in indoor navigation, personnel tracking, underground positioning, unmanned transportation, and other related fields because of high positioning accuracy, low power consumption, good penetration, and strong anti-interference ability.²,³ At present, the most common UWB positioning algorithms include the time of arrival (TOA) and the time difference of arrival (TDOA).⁴ The TOA positioning algorithm calculates the distance between the tag and the base station by recording the transmission time and reception time of the signal between the tag and the base station. Subsequently, the positioning information can be solved based on the knowledge of circle geometry. The TOA positioning algorithm is very simple, and its hardware requirements are relatively low.⁵,⁶ The TDOA positioning algorithm obtains the positioning information by recording the TDOA between the tag and different base stations.

¹ School of Electrical and Information Engineering, Hunan Institute of Engineering, Xiangtan, People’s Republic of China
² State Grid Xiangxiang Power Supply Company, Xiangtan, People’s Republic of China

Corresponding authors:
Jing Sun and Qiaolong Zhang, School of Electrical and Information Engineering, Hunan Institute of Engineering, Xiangtan 411104, People’s Republic of China.
Emails: jsun1206@163.com; 20809@hnie.edu.cn
Although the positioning accuracy of TDOA is better than that of TOA, the hardware requirements of the former are much higher than that of the latter, and the clock synchronization between the tag and the base station is needed.\textsuperscript{7,8} In fact, only the coordinate position of the tag can be obtained by executing the TOA positioning algorithm or the TDOA positioning algorithm. The prerequisite for the mobile robot to complete the task successfully is to obtain relatively reliable navigation data, such as the coordinate position, posture angle, and speed. However, most mobile robots that obtain posture information by using inertial measurement unit (IMU) need to be corrected continuously to reach the predetermined position.\textsuperscript{9–12} The reason is that IMU has inevitable defects, such as zero drift, low sensitivity, and temperature drift, due to the influence of manufacturing technology, and these defects lead to various errors in posture information.\textsuperscript{13–15} To overcome the above problems, Liu et al. proposed an approach to combine IMU, long-range Wi-Fi and short-range UWB for cooperative positioning in the infrastructure-less environment.\textsuperscript{16} The experimental results show that the combination of Wi-Fi and UWB measurements achieve a better positioning accuracy.\textsuperscript{17} Nguyen and Xie studied the four degree-offreedom relative transformation estimation problem based on the fusion of odometry and UWB ranging data. The simulated and experimental results show that the proposed methods outperform state-of-the-art methods, especially in geometrically poor or large measurement noise conditions.\textsuperscript{18} Cao et al. proposed an approach for relative localization without any infrastructure based on the fusion of multiple UWB ranging measurements from different nodes installed on the robots. The location accuracy is improved by utilizing the odometry constraints through a sliding window-based optimization.\textsuperscript{19} Zhao et al. combined the UWB and inertial navigation sensor (INS) to form a coupled UWB/INS localization framework, in which a minimum variance unbiased finite impulse response (MVU FIR) method is applied to obtain accurate position and velocity estimations from noisy measurements. The simulated and experimental results show that the MVU FIR filter exhibits better immunity to the errors about a priori knowledge of noise variances.\textsuperscript{20,21} In addition, lots of researcher studied the sensor fusion technology based on Kalman filtering in the robotics and vision.\textsuperscript{22} Although the accuracy evaluation is improved by the combination of UWB and IMU, the algorithms are complicated.

In this article, a posture algorithm is proposed for the mobile robot based on the UWB. Taking the TOA positioning algorithm as an example, two UWB tags and three UWB base stations are used in the posture algorithm. The coordinate data of the two tags are calculated by the TOA positioning algorithm. Subsequently, the posture information of the mobile robot can be solved by using the coordinate data of the two tags. In addition to the TOA algorithm, this posture algorithm can also be combined with most UWB-based positioning algorithms. This work is expected to have a guiding significance for the application of UWB and the development of mobile robots.

### Posture algorithm based on UWB

#### TOA algorithm

The TOA algorithm, in which a tag and at least three base stations $BS_i$ ($i = 1, 2, 3$) are required, achieves the positioning based on circumferential geometry knowledge. The distance $r_i$ between the tag and the base station can be obtained by the bilateral two-way ranging.\textsuperscript{23} Assuming that the tag and the base station are on the same plane, the three circles, which select the base station as the circle center and the distance $r_i$ between the tag and the base station as the radius, will intersect at one point ideally. The point is regarded as the position of the tag, and its coordinate is $(x_0, y_0)$, as shown by the red dot in Figure 1. The coordinates of the three base stations $BS_i$ in the plane coordinate system are $(x_1, y_1)$, $(x_2, y_2)$, and $(x_3, y_3)$, respectively.

![Schematic diagram of TOA positioning](image)

By using the distance formula between two points in the plane coordinate system, the equations can be obtained as follows

$$
\begin{align*}
(x_0 - x_1)^2 + (y_0 - y_1)^2 &= r_1^2 \\
(x_0 - x_2)^2 + (y_0 - y_2)^2 &= r_2^2 \\
(x_0 - x_3)^2 + (y_0 - y_3)^2 &= r_3^2
\end{align*}
$$

(1)

The coordinate data of the target point $(x_0, y_0)$ can be obtained by equation (1), given as follows

$$
\begin{pmatrix}
x_0 \\
y_0
\end{pmatrix} = \begin{pmatrix}
2 \times (x_1 - x_3) & 2 \times (y_1 - y_3) \\
2 \times (x_2 - x_3) & 2 \times (y_2 - y_3)
\end{pmatrix}^{-1}
\begin{pmatrix}
x_1^2 - x_3^2 + y_1^2 - y_3^2 - r_1^2 + r_3^2 \\
x_2^2 - x_3^2 + y_2^2 - y_3^2 - r_2^2 + r_3^2
\end{pmatrix}
$$

(2)
In fact, the data receiving time is longer than the transmission time because of the accuracy of the hardware, multipath effects, non-line of sight errors, and other factors. Thus the three circles will intersect in a region rather than at a point. In general, the algorithms, such as weighted least square algorithm, Fang algorithm, and Chan algorithm, can be used to estimate the results to obtain the optimal solution.

**Posture algorithm based on UWB**

Although the TOA algorithm can obtain the coordinate data of the target in the plane coordinate system, the coordinate data are not enough in the field of robotics. Therefore, knowing the posture data of the robot, which can be obtained by the posture algorithm based on UWB, is necessary for the application of the robot.

The coordinate axes in the plane global coordinate system are defined as $X_1$ and $Y_1$. The position of the robot in the global coordinate system is $(x_1, y_1)$, and the angle between the positive direction of the robot and the $X_1$ direction of the coordinate axis is $\theta_1$. Thus, the posture and the motion velocity of the robot in the global coordinate system can be described as $X_1=(x_1, y_1, \theta_1)^T$ and $\dot{X}_1=(\dot{x}_1, \dot{y}_1, \dot{\theta}_1)^T$, respectively. Taking the plane centroid of the robot as the local coordinate origin, the coordinate axes in the local coordinate system are $X_R$ and $Y_R$. Hence, the motion velocity of the robot in the local coordinate system can be described as $\dot{X}_R=(\dot{x}_R, \dot{y}_R, \dot{\theta}_R)^T$.

The relationship between $\dot{X}_1$ and $\dot{X}_R$ is

$$
\begin{pmatrix}
\dot{x}_R \\
\dot{y}_R \\
\dot{\theta}_R
\end{pmatrix} = R(\theta) \begin{pmatrix}
\dot{x}_1 \\
\dot{y}_1 \\
\dot{\theta}_1
\end{pmatrix}
$$

(3)

where $R(\theta)$ is the orthogonal rotation matrix, given as follows

$$
R(\theta) = \begin{pmatrix}
\cos \theta & \sin \theta & 0 \\
-\sin \theta & \cos \theta & 0 \\
0 & 0 & 1
\end{pmatrix}
$$

(4)

The positioning information of the robot can be obtained by using two UWB tags in the posture algorithm. The two tags are installed on both sides of the robot head. The left one is Tag1, and the right one is Tag2, as shown in Figure 2. Suppose that the coordinate data of the two tags obtained by the TOA positioning algorithm are $(x_1, y_1)$ and $(x_2, y_2)$, the relevant posture information of the robot can be calculated by using the coordinate data of the two tags.

**Posture of the robot in the global coordinate system.** There are four postures for the robot in the global coordinate system, as shown in Figure 3. The positioning information of the robot in the global coordinate system can be obtained by using the coordinate data $(x_1, y_1)$ and $(x_2, y_2)$ of the two tags. Take Figure 3(a) as an example, we can obtain the positioning information of the robot in the $X_1$ direction as $x_1 = \frac{x_1 + x_2}{2}$ and in the $Y_1$ direction as $y_1 = \frac{y_1 + y_2}{2}$. Therefore, the positioning information of the robot can be rewritten as

$$
\begin{pmatrix}
x_1 \\
y_1
\end{pmatrix} = \begin{pmatrix}
x_1 + x_2 \\
y_1 + y_2
\end{pmatrix}
$$

(5)

A straight line formed by connecting two tags intersects the $X_1$ axis of the global coordinate system in point A. The angle $\theta_A$ is formed by the straight line and the $X_1$ axis. Take Figure 3(a) as an example again, we can obtain the tangent of $\theta_A$

$$
\tan \theta_A = \frac{y_1 - y_2}{x_1 - x_2}
$$

(6)

Here $\theta_A$ will be specified in degrees not radians, so

$$
\theta_A = \frac{180 \times \arctan \frac{y_1 - y_2}{x_1 - x_2}}{\pi}
$$

(7)

The calculated $\theta_A \in [0^\circ, 90^\circ)$, and it can be treated as follows:

1) If $x_1 \leq x_2$ and $y_1 > y_2$, $\theta_1 = 90^\circ - \theta_A$, then the posture diagram is shown in Figure 3(a).

2) If $x_1 > x_2$ and $y_1 \leq y_2$, $\theta_1 = 90^\circ + \theta_A$, then the posture diagram is shown in Figure 3(b).

3) If $x_1 \leq x_2$ and $y_1 \leq y_2$, $\theta_1 = 90^\circ + \theta_A$, then the posture diagram is shown in Figure 3(c).

4) If $x_1 > x_2$ and $y_1 > y_2$, $\theta_1 = 90^\circ - \theta_A$, then the posture diagram is shown in Figure 3(d).

**Speed of the robot in the coordinate system.** In general, the speed of the robot is measured by the motor encoder. However, the use of the motor encoder results in errors caused by motor slippage. To overcome this problem, we use the
positioning information of the two UWB tags of the robot to calculate the speeds of the robot in the global and local coordinate systems.

From equation (5), the positioning information in the $X_1$ direction for the robot with two UWB tags is $\frac{x_1(t_1) + x_2(t_1)}{2}$ at time $t_1$ and $\frac{x_1(t_2) + x_2(t_2)}{2}$ at time $t_2$. Thus, the speed of the robot in the $X_1$ direction is

$$\dot{x}_1 = \frac{(x_1(t_2) + x_2(t_2)) - (x_1(t_1) + x_2(t_1))}{2 \times (t_2 - t_1)}$$ (8)

Similarly, the speed of the robot in the $Y_1$ direction can be obtained

$$\dot{y}_1 = \frac{(y_1(t_2) + y_2(t_2)) - (y_1(t_1) + y_2(t_1))}{2 \times (t_2 - t_1)}$$ (9)

The angle velocity $\dot{\theta}_1$ is the rate of the change of the angle over time, so it can be given as follows

$$\dot{\theta}_1 = \frac{\theta_I(t_2) - \theta_I(t_1)}{t_2 - t_1}$$ (10)

The speeds of the robot in the global coordinate systems can be rewritten as

$$\begin{bmatrix} \dot{x}_R \\ \dot{y}_R \\ \dot{\theta}_R \end{bmatrix} = \begin{bmatrix} \cos \theta & \sin \theta & 0 \\ -\sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} (x_1(t_2) + x_2(t_2)) - (x_1(t_1) + x_2(t_1)) \\ (y_1(t_2) + y_2(t_2)) - (y_1(t_1) + y_2(t_1)) \\ \theta_I(t_2) - \theta_I(t_1) \end{bmatrix} ÷ 2 \times (t_2 - t_1)$$ (11)

Here $x_1(t_1), x_1(t_2), x_2(t_1), x_2(t_2), y_1(t_1), y_1(t_2), y_2(t_1), y_2(t_2)$, $\theta_I(t_1)$, and $\theta_I(t_2)$ represent the global posture data of the robot at $t_1$ and $t_2$, and $t_2 - t_1$ represents a very short period. Consequently, the speed in the local coordinate system of the robot can be obtained by combination equation (3) and equation (11), given as

$$\begin{bmatrix} \dot{x}_R \\ \dot{y}_R \\ \dot{\theta}_R \end{bmatrix} = \begin{bmatrix} \cos \theta & \sin \theta & 0 \\ -\sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} (x_1(t_2) + x_2(t_2)) - (x_1(t_1) + x_2(t_1)) \\ (y_1(t_2) + y_2(t_2)) - (y_1(t_1) + y_2(t_1)) \\ \theta_I(t_2) - \theta_I(t_1) \end{bmatrix} ÷ 2 \times (t_2 - t_1)$$ (12)

Relative posture of the robot and the target. The relative posture of the robot and the target can be represented by $(d$ $\theta_T)^T$, where $d$ and $\theta_T$ are the distance and the relative angle between the robot and the target, respectively. The positioning information for the robot is $\left(\frac{x_1 + x_2}{2}, \frac{y_1 + y_2}{2}\right)$. According to the Pythagorean theorem, the $d$ can be obtained as

$$d = \sqrt{\left(\frac{x_3 - \frac{x_1 + x_2}{2}}{2}\right)^2 + \left(\frac{y_3 - \frac{y_1 + y_2}{2}}{2}\right)^2}$$ (13)
Here \( x_3 \) and \( y_3 \) are the coordinates of the target point. \( \theta_T \) needs to be handled differently according to various situations. If the target is in the positive direction of the robot shown in Figure 4, the angle \( \theta_T + \theta_1 \) is

\[
\theta_T + \theta_1 = \frac{180 \times \arctan \left( \frac{y_1 + y_2}{x_1 + x_2} - y_3 \right)}{\pi} \quad (14)
\]

So

\[
\theta_T = \frac{180 \times \arctan \left( \frac{y_1 + y_2}{x_1 + x_2} - y_3 \right)}{\pi} - \theta_1 \quad (15)
\]

Therefore, the relative posture of the robot and the target can be given as

\[
\begin{pmatrix}
\frac{x_3 - x_1 + x_2}{2} \\
\frac{y_1 + y_2}{2} - y_3
\end{pmatrix} = \left(\begin{array}{c}
\frac{x_3 - x_1 + x_2}{2} \\
\frac{y_1 + y_2}{2} - y_3
\end{array}\right) \\
\frac{180 \times \arctan \left( \frac{y_1 + y_2}{x_1 + x_2} - y_3 \right)}{\pi} - \theta_1
\]

\[
(\frac{d}{\theta_T}) = \left(\begin{array}{c}
\frac{x_3 - x_1 + x_2}{2} \\
\frac{y_1 + y_2}{2} - y_3
\end{array}\right) + \left(\begin{array}{c}
\frac{y_1 + y_2}{2} - y_3 \sin \theta_1 \\
\frac{180 \times \arctan \left( \frac{y_1 + y_2}{x_1 + x_2} - y_3 \right)}{\pi} - \theta_1 + \frac{y_1 + y_2}{2} - y_3 \cos \theta_1
\end{array}\right)
\quad (16)
\]

**Results and discussion**

The analysis shows that the accuracy of the algorithm is closely related to the UWB positioning accuracy that is affected by the measure errors and the position accuracy of the base stations\(^{27,28}\). At present, the positioning accuracy has been improved by the researchers, and the simulation error of 2 cm is chosen as the lower limit in this article\(^{29}\). We suppose the coordinates of the three base stations are (0,0), (0, 800), and (800, 0); the robot centroid coordinates (400, 400); target point position (800, 800); actual posture angle 45\(^\circ\); and the installation distance of the two tags 60 cm. The posture angle of the robot \( \theta_1 \) in the plane coordinate system is simulated with the positioning errors of 8, 6, 4, and 2 cm (Figure 6). The accuracy of the algorithm is closely related to the positioning accuracy of UWB when the tag installation distance is fixed. The angle errors increase with the increase of the positioning errors.
This phenomenon can be explained by equation (7), in which the angle solved by the algorithm is calculated by the positioning coordinates. The smaller the positioning error is, the more accurate the posture information obtained by the algorithm will be. Here the mean absolute error (MAE) and root mean square error (RMSE) are used as metrics for performance evaluation. As shown in Table 1, the MAE and RMSE of the posture angle in the plane coordinate system are 1.222° and 1.596° when the tag installation distance is 60 cm and the positioning error is within 2 cm. Such values can meet the accuracy requirements of most mobile robots.

We assume that the coordinates of the three base stations are (0,0), (0, 800), and (800, 0); the robot centroid coordinates (400, 400); target point position (800, 800); and the positioning error within 2 cm. The posture angle of the robot $\theta_1$ in the global plane coordinate system is simulated with the orientations of the robot changing from 0° to 180° when the installation distances are 50, 60, 70, and 80 cm between two tags, as shown in Figure 7. To show the differences of the four graphs in Figure 7 more obviously, the part of the curves is enlarged in the inset. The simulated curves become smoother with the increase of the installation distance when the positioning error is within a certain range. The accuracy of the simulated posture angle is almost unchanged when the orientations of the robot change from 0° to 180°. Equation (7) also shows that the greater the tag installation distance is, the smaller the influence of the positioning error on the posture angle will be. Table 2 shows that when the positioning error is within 2 cm and the tag installation distance is 80 cm, the MAE and the RMSE of the posture angle in the plane coordinate system reach 0.951° and 1.169°, respectively. Such finding indicates that the robot based on the posture algorithm can be applied in many occasions, such as catering and logistics.

To verify the validity of the posture algorithms, the control platform in which the TM32F103-ZET6 is selected as the control core is built to test the robot (Figure 8). The coordinates of the three base stations are (0,0), (0, 800), and

---

**Table 1. Data evaluation under different positioning errors.**

| Positioning error (cm) | 8    | 6    | 4    | 2    |
|------------------------|------|------|------|------|
| MAE (°)                | 4.527| 3.788| 2.510| 1.222|
| RMSE (°)               | 5.563| 4.771| 3.027| 1.596|

MAE: mean absolute error; RMSE: root mean square error.
The robot is placed at the point (400, 400), and the target point is set at (800, 800). The posture angle in the global coordinate system is adjusted to 45° and the installation distance is 24 cm between two tags. The posture data solved by the algorithm are transmitted through Bluetooth. Figure 9 shows the distribution of the experimental posture angles. The results show that the posture angle obtained by the posture algorithm fluctuates around 45°, and the MAE and RMSE are 2.3469° and 2.7093°, respectively. The simulated and experimental trajectories of the robot which moves from the point (400, 400) to (800, 800) are shown in Figure 10. Both the simulated and experimental trajectories

Table 2. Data evaluation under different tag installation distances.

| Installation distance (cm) | 50  | 60  | 70  | 80  |
|---------------------------|-----|-----|-----|-----|
| MAE (°)                   | 1.505 | 1.253 | 1.088 | 0.951 |
| RMSE (°)                  | 1.861 | 1.541 | 1.340 | 1.169 |

MAE: mean absolute error; RMSE: root mean square error.

Figure 7. Simulated results with different tag installation distances: (a) 50 cm, (b) 60 cm, (c) 70 cm, and (d) 80 cm. (Inset) The enlarged part of the curves.

Figure 8. Schematic diagram of the robot test.
show good agreement with the ideal trajectory, while from the point (700, 700) to (780, 780) the experimental trajectory cannot fit the ideal trajectory very well. The maximum offset is about 8.26 cm (see the inset). The experimental results are slightly worse than the simulated results because it is difficult to obtain a positioning error of 2 cm for UWB. The results show that the posture data can be obtained accurately by the posture algorithm and the accuracy can meet the requirements of most mobile robots.

Conclusions
A posture algorithm based on UWB is proposed in this article. The posture angle, speed, and other real-time posture data of the mobile robot in the established plane coordinate system can be calculated, and the problems by using IMU can be solved by the posture algorithm. The simulated results show that the posture accuracy increases with the decrease of the positioning errors when the installation distance of the two UWB tags is constant, and the error accuracy increases with the increase of the installation distance of the two UWB tags when the positioning error is within a certain range. Both the simulated and experimental trajectories show good agreement with the ideal trajectory. The results indicate that the posture data can be obtained by the posture algorithm and the accuracy can meet the requirements for most mobile robots. In fact, the accuracy of UWB is influenced by the non-line-of-sight blocking, which is not considered in this article. It will be studied in the following work.

Declaration of conflicting interests
The authors declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

Funding
The authors disclosed receipt of the following financial support for the research, authorship and/or publication of this article: This work was funded by the projects of the National Natural Science Foundation of China (No. 62071410 and No. 31502087), the Scientific Research Foundation of Hunan Provincial Department (No. 19A106, No. 20C0498 and No. 21C0581), the Science and Technology Innovation Talents Program of Hunan Province (No. 2021RC1011), and Hunan Graduate Scientific Research Innovation Project (No. CX20201173).

ORCID iD
Jing Sun https://orcid.org/0000-0002-7917-071X

References
1. Tao B, Wu H, Gong Z, et al. An RFID-based mobile robot localization method combining phase difference and readability. *IEEE Trans Auto Sci Eng* 2021; 18: 1406–1416.
2. Li S, Li G, Li W, et al. A three-dimensional robust ridge estimation positioning method for UWB in a complex environment. *Adv Space Res* 2017; 60: 2763–2775.
3. Mazhar F, Khan MG and Sällberg B. Precise indoor positioning using UWB: a review of methods, algorithms and implementations. *Wireless Personal Commun* 2017; 97: 4467–4491.
4. Pala S, Jayan S and Kurup DG. An accurate UWB based localization system using modified leading edge detection algorithm. *Ad Hoc Networks* 2020; 97: 102017.
5. Ridolfi M, Kaya A, Berkvens R, et al. Self-calibration and collaborative localization for UWB positioning systems: a survey and future research directions. *ACM Comput Surv* 2021; 54: 1–27.
6. Cxa C and Qlb C. An inertial neural network approach for robust time-of-arrival localization considering clock asynchronization. *Neural Networks* 2021; 146: 98–106.
7. Bottiglieri S, Milanesio D, Saccani M, et al. A low-cost indoor real-time locating system based on TDOA estimation of UWB pulse sequences. *IEEE Trans Instrum Meas* 2011; 70: 1–11.
8. Laaraiedh M, Lei Y, Avrillon S, et al. Comparison of hybrid localization schemes using RSSI, TOA, and TDOA. In: *17th
9. Feng D, Wang C, He C, et al. Kalman-filter-based integration of IMU and UWB for high-accuracy indoor positioning and navigation. *IEEE Int Things J* 2020; 7: 3133–3146.

10. Yang X, Wang J, Song D, et al. A novel NLOS error compensation method based IMU for UWB indoor positioning system. *IEEE Sens J* 2021; 21: 11203–11212.

11. Zhong Y, Liu T, Li B, et al. Integration of UWB and IMU for precise and continuous indoor positioning. In *2018 Ubiquitous Positioning, Indoor Navigation and Location-Based Services (UPLBLS)*, Wuhan, China, 22–23 March 2018.

12. Yao L, Wu Y, Yao L, et al. An integrated IMU and UWB sensor based indoor positioning system. In *2017 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, Sapporo, Japan, 18–21 September 2017.

13. Lu G and Zhang F. IMU-based attitude estimation in the presence of narrow-band noise. *IEEE Trans Mechatronics* 2019; 24: 841–852.

14. Zhang Q, Niu X and Shi C. Impact assessment of various IMU error sources on the relative accuracy of the GNSS/INS systems. *IEEE Sens J* 2020; 20: 5026–5038.

15. Wu Y, Niu X and Kuang J. A comparison of three measurement models for the wheel-mounted MEMS IMU-based dead reckoning system. 2020; arXiv: 2012.10589.

16. Liu R, Yuen C, Do TN, et al. Cooperative positioning for emergency responders using self IMU and peer-to-peer radios measurements. *Inform Fusion* 2020; 56: 93–102.

17. Nguyen TH and Xie L. Relative transformation estimation based on fusion of odometry and UWB ranging data. 2022; arXiv:2202.00279v1.

18. Cao Z, Liu R, Yuen C, et al. Relative localization of mobile robots with multiple ultra-wideband ranging measurements. 2021; arXiv:2107.08842v2.

19. Zhao S, Huang B and Liu F. Localization of indoor mobile robot using minimum variance unbiased FIR filter. *IEEE Trans Auto Sci Eng* 2016; 99: 1–10.

20. Zhao S, Shmaliy YS and Liu F. Fast Kalman-like optimal unbiased FIR filtering with applications. *IEEE Trans Signal Process* 2016; 64: 2284–2297.

21. Auger F, Hilairet M, Guerrero J, et al. Industrial applications of the Kalman filter: a review. *IEEE Trans Industr Electr* 2013; 60: 5458–5471.

22. Domuta I and Palade PT. Two-way ranging algorithms for clock error compensation. *IEEE Trans Veh Technol* 2021; 70: 8237–8250.

23. Sidorenko J, Schatz V, Scherer-Negenborn N, et al. Decawave ultra-wideband warm-up error correction. *IEEE Trans Aerosp Electr Syst* 2021; 57: 751–760.

24. Jiang W, Cao Z, Cai B, et al. Indoor and outdoor seamless positioning method using UWB enhanced multi-sensor tightly-coupled integration. *IEEE Trans Veh Technol* 2021; 70: 10633–10645.

25. Yan J, Tiberius C, Teunissen P, et al. A framework for low complexity least-squares localization with high accuracy. *IEEE Trans Signal Process* 2010; 58: 4836–4847.

26. Ho KC. Bias reduction for an explicit solution of source localization using TDOA. *IEEE Trans Signal Process* 2012; 60: 2101–2114.

27. Dardari D, Conti A, Lien J, et al. The effect of cooperation on localization systems using UWB experimental data. *EURASIP J Adv Signal Process* 2008; 124: 513873(1–22).

28. Han Y, Zhang X, Lai Z, et al. TOF-based fast self-positioning algorithm for UWB mobile base stations. *Sensors* 2021; 21: 6359(1–14).

29. Zhao Y, Zou J, Guo J, et al. A novel ultra-wideband double difference indoor positioning method with additional baseline constraint. *Int J Geo-Inform* 2021; 10: 634.

30. Shenoy M, Karuppiah A and Manjarekar N. A lightweight ANN based robust localization technique for rapid deployment of autonomous systems. *J Ambient Intell Humananiz Comput* 2020; 11: 2715–2730.

31. Cui Z, Liu T, Tian S, et al. Non-line-of-sight identification for UWB positioning using capsule networks. *IEEE Commun Lett* 2020; 24: 2187–2190.