Sensors and Materials, Vol. 34, No. 6 (2022) 2427–2445
MYU Tokyo

S & M 2981

Inertial Sensor Error Compensation for Global Positioning System Signal Blocking —Extended Kalman Filter vs Long- and Short-term Memory—

Guo-Shing Huang,[1*] Yu-Fan Wu,[1] and Ming-Cheng Kao[2]

[1]Institute of Electronic Engineering, National Chin-Yi University of Technology, Taichung City 411, Taiwan, R.O.C.
[2]Department of Electronic Engineering, Hsiuping University of Science and Technology, Taichung City 412, Taiwan, R.O.C.

(Received December 28, 2021; accepted May 31, 2022)

Keywords: GPS, RTK, navigation, positioning, inertia, error compensation, LSTM, longitude, latitude conversion

At present, various applications have a high demand for navigation systems. With the example of self-driving cars, the navigation system has to provide pinpoint accuracy for positioning. Inertial navigation system (INS) and global positioning system (GPS) are some of the common ways to navigate. However, these two systems have the disadvantages of continuity, cumulative error, divergence over time, and reliability. A solution based on the extended Kalman filter (EKF) and long- and short-term memory (LSTM) is proposed in this study to correct the divergence due to cumulative errors in INS. It has been proven effective by a number of studies to combine a Kalman filter with GPS and INS data. However, there are still issues in the integration of the Kalman filter with INS/GPS, such as random error model, noise resistance, and observability of inertial sensors. The proposed system is designed to incorporate deep learning to comb through long-, medium-, and short-term memories as well as predict INS and GPS errors using recurrent neural network (RNN), as LSTM is used to learn INS errors while the GPS is working well and to predict GPS errors when GPS signals are lost. Unlike the traditional way of learning, LSTM contains time variants. To verify the accuracy of the proposed design, the EKF is introduced as a means to compare with LSTM. EKF is very suitable for more flexible coordination between INS and GPS, so EKF is used for deep learning comparison with LSTM for prediction and control in a nonlinear environment. Then, the LSTM deep learning is used to correct the predictions. This computation reduces the errors in position and speed. Finally, an emulation model developed in MATLAB is used to simulate the INS–GPS integrated system error compensation model. The experiment results indicate that the errors in parameters are the smallest with the integration of LSTM in INS and GPS, thus providing the effects of error correction and compensation.

*Corresponding author: e-mail: hgs@ncut.edu.tw
https://doi.org/10.18494/SAM3790

ISSN 0914-4935
https://myukk.org/

1. Introduction

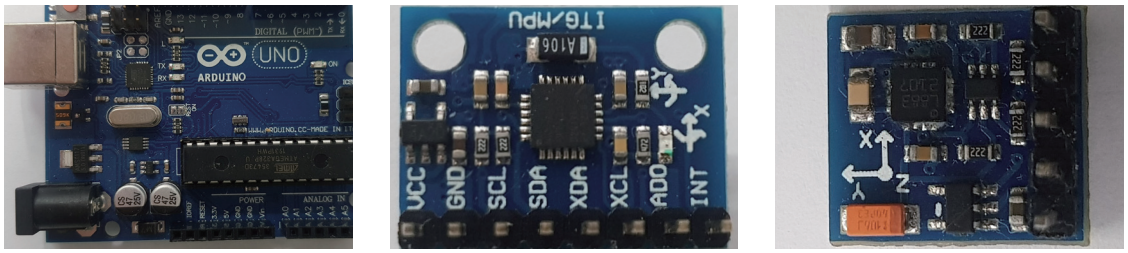
Global positioning system (GPS) is a widely used way of positioning and navigation in the world. It has become an essential tool in daily life as it is seen in positioning for logistics and travel. GPS is a part of the satellite navigation system, which was developed under the NAVSTAR project of the US Department of Defense (DoD). GPS satellites run 24 h a day and orbit the Earth on six circular orbits in general. On each of these orbits, there are four or more satellites orbiting the Earth at the same time; i.e., there are 24 or 28 GPS satellites orbiting 24/7. In theory, any place on Earth's surface with a clear view can receive signals from three or more GPS satellites, while there are four or more GPS satellites constantly monitoring users' positions at any place on Earth every hour and every day. GPS is a very accurate positioning system, as its accuracy does not deteriorate over time. However, it does have its downsides, as the signal reception is poor in covered areas such as mountains and tunnels, since it relies on radio waves for navigation parameter transmission. To maintain uninterrupted navigation, inertial sensors are added for compensation most of the time. The inertial navigation system (INS) itself is an independent system used extensively for dead reckoning, as it incorporates an accelerometer, a gyroscope, and a magnetic compass.[1] To provide continuous speed, position, and attitude estimations and convert them into navigational coordinates for the transmission of position, speed, and attitude components, the accelerometer measures the acceleration in the direction of the measurement carrier, while the gyroscope monitors the angular speed in the INS. These estimations are accurate for a short time. However, INS does not work as an independent navigation system, since the estimations by the accelerometer and gyroscope are subject to displacement, resulting in residual deviations in the sensors. The divergence rate depends on the quality of sensors, and the positioning accuracy of the navigation system decreases with time. GPS and INS complement one another. By combining inertial navigation and satellite positioning, the complementing nature enables a system that allows positioning and navigation even with interrupted satellite signals. The INS/GPS system is often made possible with a Kalman filter (KF),[2] which is proven to be one of the best available integrated solutions. However, Wen-jiang *et al.* mentioned in their study that the downside of KF is that it depends heavily on a predefined dynamic model; without a reasonable mathematical model, the desired compensation is impossible.[3] A dynamic model is usually based on INS position errors, speed errors, and attitude errors. Therefore, many switch to the extended Kalman filter (EKF), such as Deepika and Arun,[4] or introduce a sigma-point Kalman filter (SPKF) for error correction.[5] In addition, the sensor errors, such as accelerometer deviation and gyroscope shift, add to those errors. In fact, every inertial sensor has its own random errors. It is difficult to define an appropriate random model for every inertial sensor to work effectively in every environment and reflect sensor errors, as this will increase the difficulty of INS error modeling and compromise the accuracy of INS/GPS navigation. For this, Malleswaran *et al.* used RBF-NN for INS error correction. They simulated the prediction and correction of 5 points with a software program and obtained the best predicted values at epoch 25.[6] Li *et al.* proposed a new learning method, the Extreme Learning Machine (ELM).[7] The interactive multi-model extended Kalman filter (IMM-EKF) was introduced to improve the KF's accuracy, and ELM was used to predict and correct INS positions. This design achieved 40% or more of correction, but the error was still

close to 10 m. Yang *et al.* proposed a model-predictive-filter-based neural network (MPFNN) for INS error compensation.[8] Unlike any traditional algorithm, the model predictive filter (MPF) uses network weights for system status variants. Two field tests were performed in this study. From 720 training samples, the RMSEs of east and north positions were determined to be 8.58 and 9.23 m, respectively. Xiong *et al.* incorporated Doppler velocity log (DVL) in GPS/INS for a hybrid positioning method and proposed a robust adaptive federated strong tracking Kalman filter (RAFSTKF) algorithm, where the least square adaptive signals are used to obtain the optimized estimations for better overall reliability.[9] Lee *et al.* used visual images and determined the position-compensated GPS/INS by identifying the characteristics of the same points in two consecutive images.[10] Apart from the KF, Malleswaran *et al.* incorporated the Input Delayed Dynamic Neural Network (IDNN) for KF without GPS signals.[11] In this study, insignificant and inexpensive inertial sensors are used as a viable navigational backup to compensate the blocked GPS signals, and then long- and short-term memory (LSTM) is used to predict the error divergence. Long- and short-term memories are used in this study because this has been extensively used for predictions in a number of fields, such as predictions of wind speed,[12] stock market, and depression,[13,14] to name a few. The difference between LSTM and recurrent neural network (RNN) is that a processor is added to the algorithm to determine whether a message is useful; the message is memorized if it is useful or discarded if not. LSTM is as effective in solving the dependence on long-time sequence as it is adaptive for application in many fields. The LSTM prediction keeps the error growing within a certain period of time for the accuracy of inertial navigation without resulting in error drift generated over time.

2. System Design

2.1 System hardware design

The proposed system hardware consists of an Arduino development board, an MPU6050 triaxial accelerometer, and a QMC5883L triaxial magnetic field sensor, as shown in Fig. 1. The MPU6050 is in turn composed of a triaxial gyroscope and a triaxial accelerometer module. It measures the accelerations and angular speeds in A_x and A_y directions of the onboard unit, while the QMC5883L triaxial magnetic field sensor is a low-cost electronic compass ideal for a



(a) (b) (c)

Fig. 1. (Color Online) (a) Arduino development board, (b) MPU6050 triaxial gyroscope/accelerometer sensor, and (c) QMC5883L triaxial magnetic field sensor.

handheld device. The assembly of Arduino, QMC5883L, and MPU6050 is presented in Fig. 2. The overall structure of inertial sensors, RTK GPS receiver, and PC is shown in Fig. 3, and the integrated wiring diagram is provided in Fig. 4. The PC comes with an i7 for CPU and RTX2070

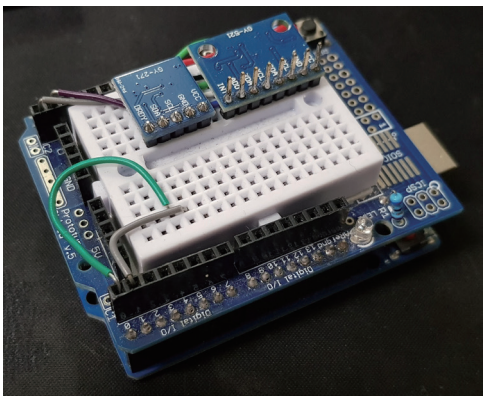


Fig. 2. (Color Online) Assembly of Arduino, QMC5883L, and MPU6050INS.

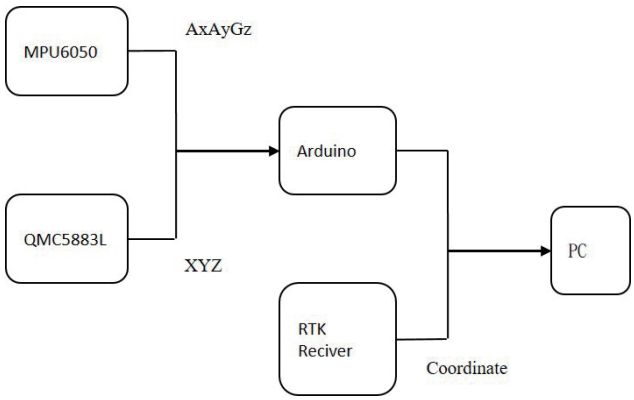


Fig. 3. (Color Online) Structure of inertial sensors, RTK GPS receiver, and PC.

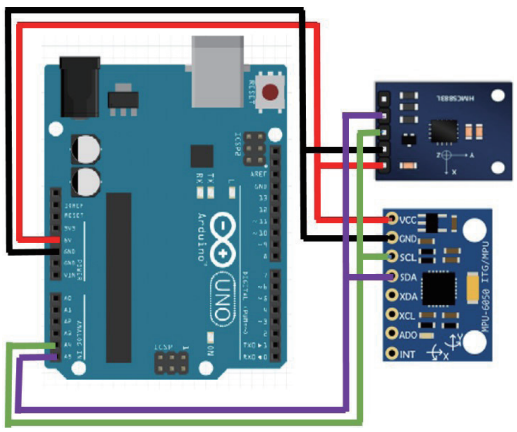


Fig. 4. (Color Online) Wiring layout of hardware system modules.

for GPU. The VCC voltage input is 3–5 V for Arduino, MPU6050, and QMC5883L, and the PC connection baud rate is 9600 Hz.

2.2 System software design

MATLAB is the core of the system design program for this study, as the coordinate errors are corrected using MATLAB. Figure 5 is the flowchart for coordinate error correction with MATLAB. MATLAB is a useful program that is ideal for algorithm development, data visualization, data analysis, and numeric calculations. It is capable of plotting functions, data images as well as building user interfaces. However, MATLAB is used in this experiment only to determine the input values from inertial sensors and to plot the EFK-converted longitudes and latitudes and points measured by RTK, as shown in Fig. 6.

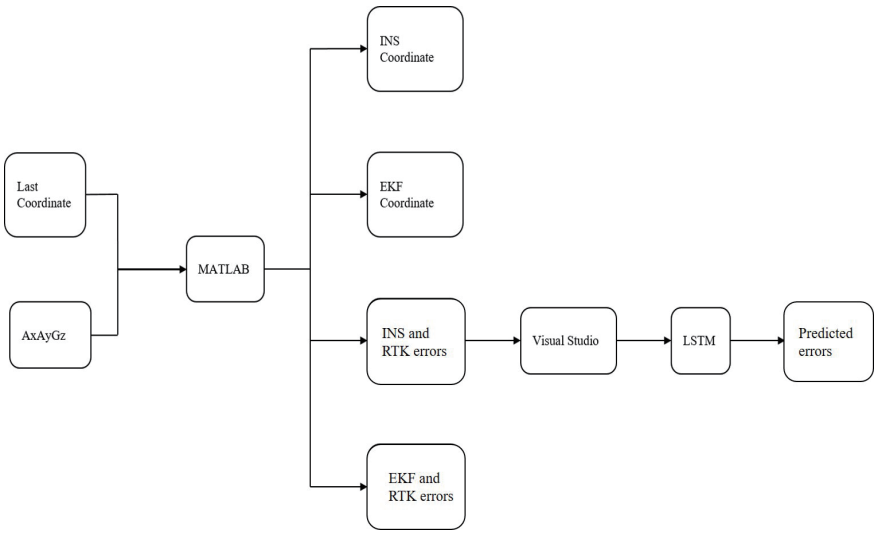


Fig. 5. Coordinate error correction flow.

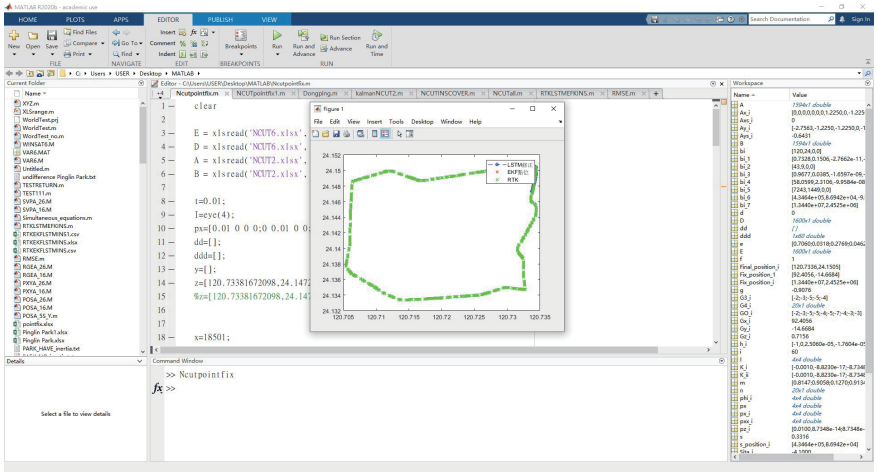


Fig. 6 (Color Online) MATLAB interface.

Another program used is Visual Studio, as shown in Fig. 7. It provides virtually all the tools needed for programming, including UML, program code control tool, and integrated development environment (IDE). The LSTM used in this study is programmed in python through Visual Studio.

2.3 RTK satellite receiver

GPS consists of user, space, and control. User refers to the receiver that receives and decodes satellite signals for positioning, space refers to the satellites in space that synchronize and transmit navigation signals for user positioning, and control is a series of monitoring stations that monitor the serial numbers and quality of GPS satellite transmission and correct the satellite signals in a timely manner. However, when a user is using a GPS receiver, a number of factors result in certain errors.(15)

The idea of RTK is the use of a base station that is highly accurate in engineering measurement and positioning for phase measurement of GNSS carrier waves, correction of parameter values, and transmission to a mobile station where the correct position is determined on the basis of real-time GPS signal measurements when the corrected parameters are received. A network RTK is a GNSS network consisting of multiple base stations for the evaluation of positioning errors in the area covered by the base stations. By introducing the observation data from neighboring physical base stations, a virtual base station (VBS) is generated for an RTK base station, as shown in Fig. 8.

The RTK used in this study, which is accurate up to 10 cm, is a product of Century Instruments Co., Ltd., as shown in Fig. 9. Century Instruments has established 21 (GPS+GLONASS) reference stations in Taiwan, and each of them transmits back observation data every second. Every second, the computation core performs a system model computation to determine corrections for satellite signals, satellite coordinates, and time. These correction values allow the correction of all errors at the current satellite status, from receiving a satellite signal at a reference station to transmitting differential data to a user.

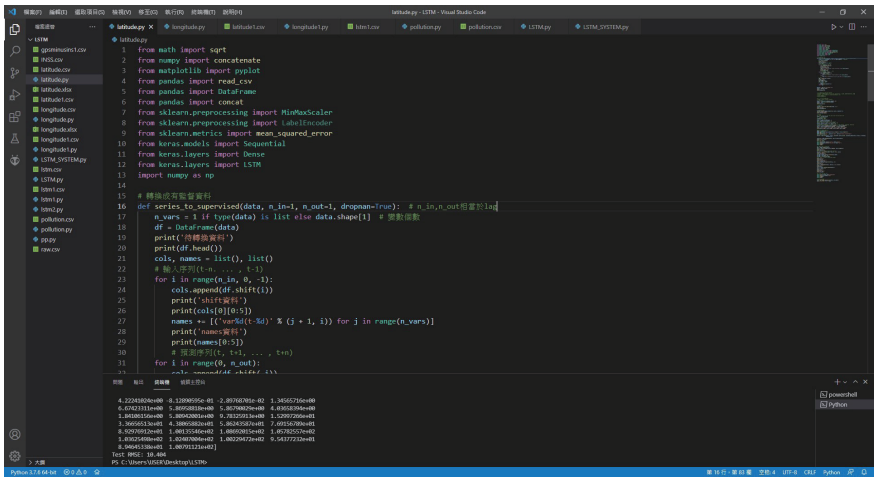


Fig. 7. (Color Online) Visual Studio interface.

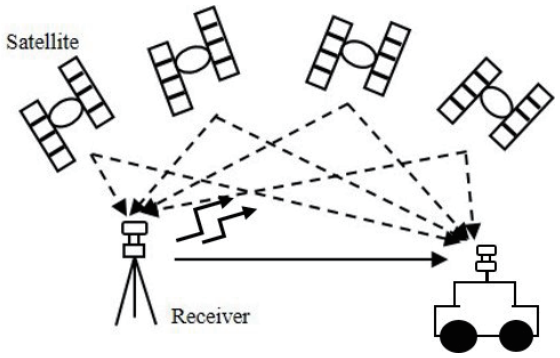


Fig. 8. RTK GPS positioning scheme.

Fig. 9. (Color Online) RTK GPS receiver.

3. Experiment Design

First of all, the Arduino, gyroscope, and accelerometer are connected to the inertial sensors of the test system before being connected to the PC where the captured values are exported to EXCEL as a text file. The RTK receiver captures the longitudes and latitudes, and these coordinates are copied to EXCEL. The connection of inertial sensors, PC, and RTK GPS receiver is shown in Fig. 10. Then, MATLAB reads the values in the EXCEL file for computation and output of corrected coordinates. The position coordinates are plotted along with the coordinate errors plotted by RTK and inertial sensors. The plots are substituted into LSTM for learning and prediction. Figure 11 is a photo showing how the assembly of inertial sensors, RTK GPS receiver, and a laptop are mounted on a car.

4. Principles and Methodology

Figure 12 shows the experiment flowchart. The experiment consists of two parts as it starts: interrupted and uninterrupted GPS signals. While the GPS signals are not interrupted, the coordinates captured by RTK, accelerations (A_x and A_y), and change in angular speed (Gz) measured by INS are fed to MATLAB for geometric calculation to determine the coordinates from vectors and RT errors, and then the errors are fed to LSTM for learning, as shown in

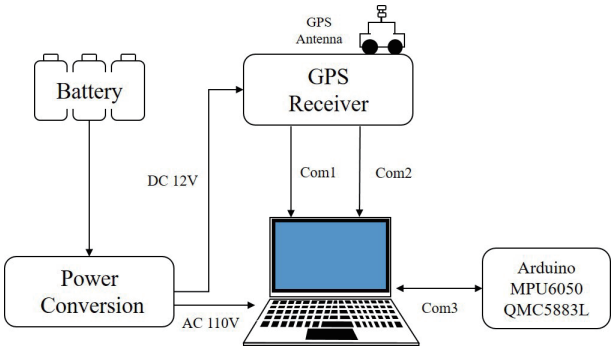


Fig. 10. (Color Online) Configuration of inertial sensors, RTK GPS receiver, and laptop.

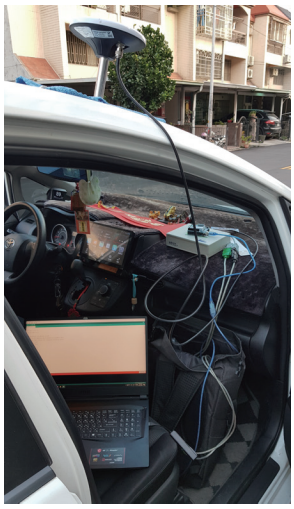


Fig. 11. (Color Online) Inertial sensors, RTK GPS receiver, and laptop mounted on a car.

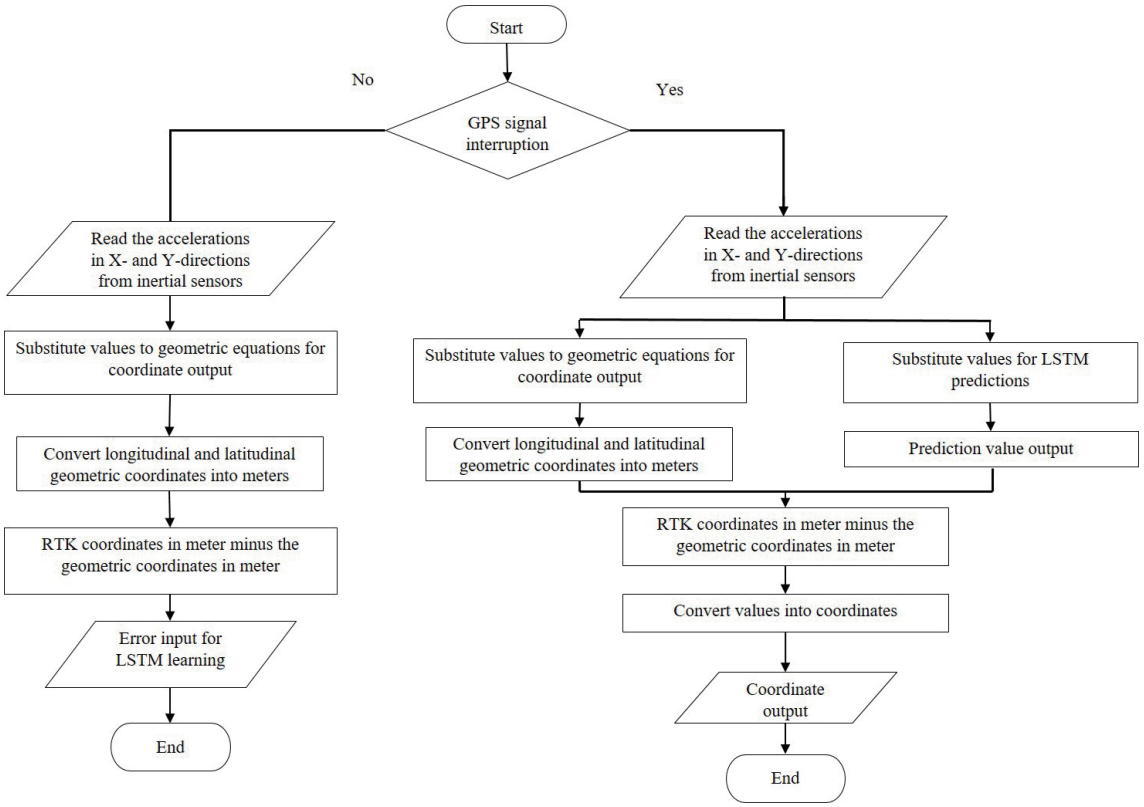


Fig. 12. Compensation process.

Fig. 13. When the GPS signals are interrupted, the INS and RTK errors are predicted and fed to MATLAB for correction. The corrected errors of both are compared with the coordinates generated by EKF.

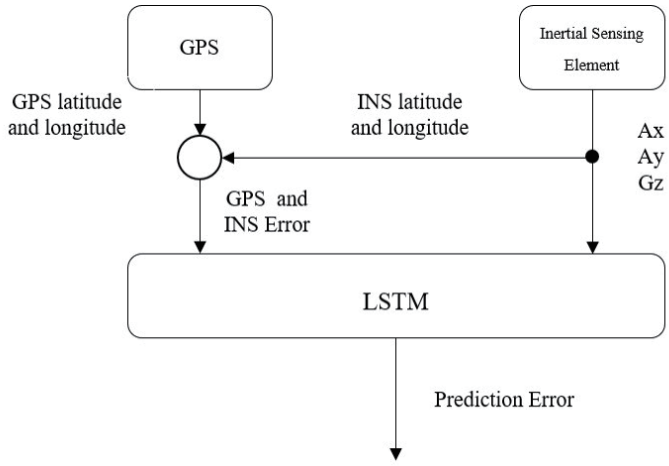


Fig. 13. LSTM learning and prediction with uninterrupted GPS.

4.1 Geometric computation of inertial sensors

The distance that the vehicle travels is determined from the displacement Eq. (1) as

$$s = \frac{1}{2}at^2. \tag{1}$$

It is clear in Fig. 14 that the velocity of the vehicle in the *X*-direction is defined as V_x and that in the *Y*-direction is V_y, and the angle measured by the gyroscope is G_a. Figure 15 shows that the movement of the vehicle in the *E*-axis is determined by adding the two component vectors in the same direction, V_{y1} and V_{x1}, together. Therefore, the displacement in the *X*-axis is expressed as Eq. (2). The movement in the *N*-direction is determined by subtracting the two component vectors in opposite directions, V_{y2} and V_{x2}. Therefore, the displacement in the *Y*-axis is expressed as Eq. (3).

$$D_x = V_{x1} + V_{y1} \tag{2}$$

$$D_y = V_{x2} - V_{y2} \tag{3}$$

As shown in Fig. 15, V_{x2} and V_{y1} are determined as the *sin* components of V_x and V_y, whereas V_{x1} and V_{y2} are the cos components of V_x and V_y, respectively, which are expressed as Eqs. (4) and (5). Finally, the direction in which the vehicle turns (G_a) is measured with a gyroscope and combined with Eqs. (2) and (3) to obtain the forward positions for Eqs. (4) and (5).

$$D_x = V_x \cos(G_a) - V_y \sin(G_a) \tag{4}$$

$$D_y = V_x \sin(G_a) + V_y \cos(G_a) \tag{5}$$

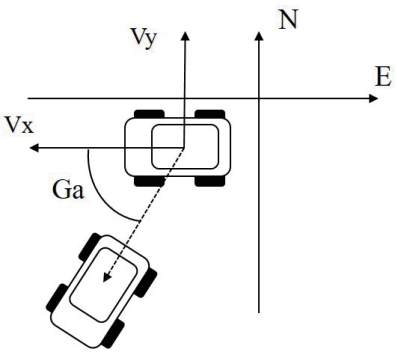


Fig. 14. Vectors detected by inertial sensors.

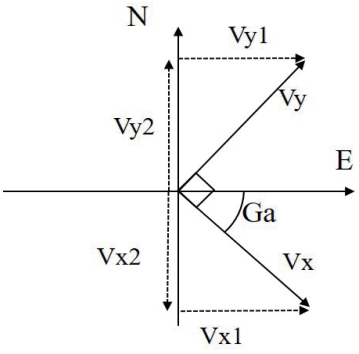


Fig. 15. Vehicle vectors.

4.2 EKF algorithm

4.2.1 Introduction to EKF

A KF is often used to correct errors in linear equation for ideal data. The algorithm is based on the estimation of the optimized linear mean square using the least square, thus optimizing a linear equation. It uses a series of prediction and measurement steps for the optimal estimation of status vectors that include the minimum variance. There are two sets of equations; one predicts the system status based on the current status and assumed system model, thus obtaining the advanced estimation for the next time step; the other is co-variants, which are the measurement of estimated uncertainty for the system status prediction.

The EKF is developed by improving the weaknesses of the KF.[11,12] The first-order Taylor expansion for the estimation of current surroundings is used, and the equation is updated with time and measurements as in KF. This linearizes nonlinear data for substitution into the KF algorithm for the optimized result of the nonlinear data. If the use of first-order Taylor expansion still does not lead to optimization, higher-order items may be added to the expansion.

When a measurement contains unpredictable or random error or uncertain change, it uses a set of equations and continuous data input for quick estimations of the real value, position, and velocity of the measured object.

The KF has the advantage over other filters in navigation algorithm, as it uses only the current and previous statuses to predict the next status, thus accelerating the calculation.

4.2.2 EKF calculation steps

First, assume that the initial estimated status $\hat{x}(0|0)$ is m_x and that the error status covariance matrix $P_x(0|0)$ is equal to 0.

$$\hat{x}(0|0) = m_x(0) \text{ and } P_x(0|0) = 0 \tag{6}$$

The current predicted status is expressed as

$$\hat{x}\left(k+1|k\right)=f\left(\hat{x}\left(k|k\right),k\right). \tag{7}$$

Linearize the status:

$$f\left(x\left(k\right),k\right)=f\left(\hat{x}\left(k|k\right),k\right)+\phi\left(k+1,k\right)\left[x\left(k\right)-\hat{x}\left(k|k\right)\right]+\ldots \tag{8}$$

Take the differential of $f\left(x\left(k\right),k\right)$:

$$\phi\left(k+1\,|\,k\right)=\left.\frac{\partial f\left(x\left(k\right),k\right)}{\partial x\left(k\right)}\right|_{\hat{x}\left(k|k\right)} \tag{9}$$

Then, the predicted covariance matrix becomes

$$P_x\left(k+1|k\right)=\phi\left(k+1|k\right)P_x\left(k|k\right)\phi^T\left(k+1|k\right)+\Gamma\left(k+1,k\right)Q\left(k\right)\Gamma^T\left(k+1|k\right). \tag{10}$$

$\hat{x}\left(k+1|k\right)$ is obtained from linearization:

$$h\left(x\left(k+1\right),\left(k+1\right)\right)=h\left(\hat{x}\left(k+1|k\right),k+1\right)+H\left(k+1\right)\left[x\left(k-1\right)-\hat{x}\left(k+1|k\right)\right]+\ldots \tag{11}$$

Take the differential of $x\left(k+1\right)$:

$$H\left(k+1\right)=\left.\frac{\partial h\left(x\left(k+1\right),k+1\right)}{\partial x\left(k+1\right)}\right|_{\hat{x}\left(k+1|k\right)} \tag{12}$$

Then, determine the gain K for $k+1$:

$$K\left(k+1\right)=P_x\left(k+1|k\right)H^T\left(k+1\right)^*\left[H\left(k+1\right)P_x\left(k+1|k\right)H^T\left(k+1\right)+R\left(k+1\right)\right]^{-1} \tag{13}$$

Correct and update with the gain in Eq. (13), $K\left(k+1\right)$:

$$\hat{x}\left(k+1|k+1\right)=\hat{x}\left(k+1|k\right)+K\left(k+1\right)\left[z\left(k+1\right)-\hat{z}\left(k+1\right)|\,k\right] \tag{14}$$

Update the covariance matrix $P_x\left(k+1|k\right)$. I in Eq. (15) is equal to an n*n matrix of the same size as the gain $K\left(k+1\right)H\left(k+1\right)$

$$P_x\left(k+1|k+1\right)=\left[I-K\left(k+1\right)H\left(k+1\right)\right]P_x\left(k+1|k\right). \tag{15}$$

The EKF estimation process is presented in Fig. 16.

4.3 LSTM algorithm

The LSTM algorithm is a type of RNN. It is widely used for time sequence prediction and proven to perform well in long- and short-term predictions. As shown in Fig. 17, LSTM consists of three levels, namely, input level, hidden level, and output level.[9]

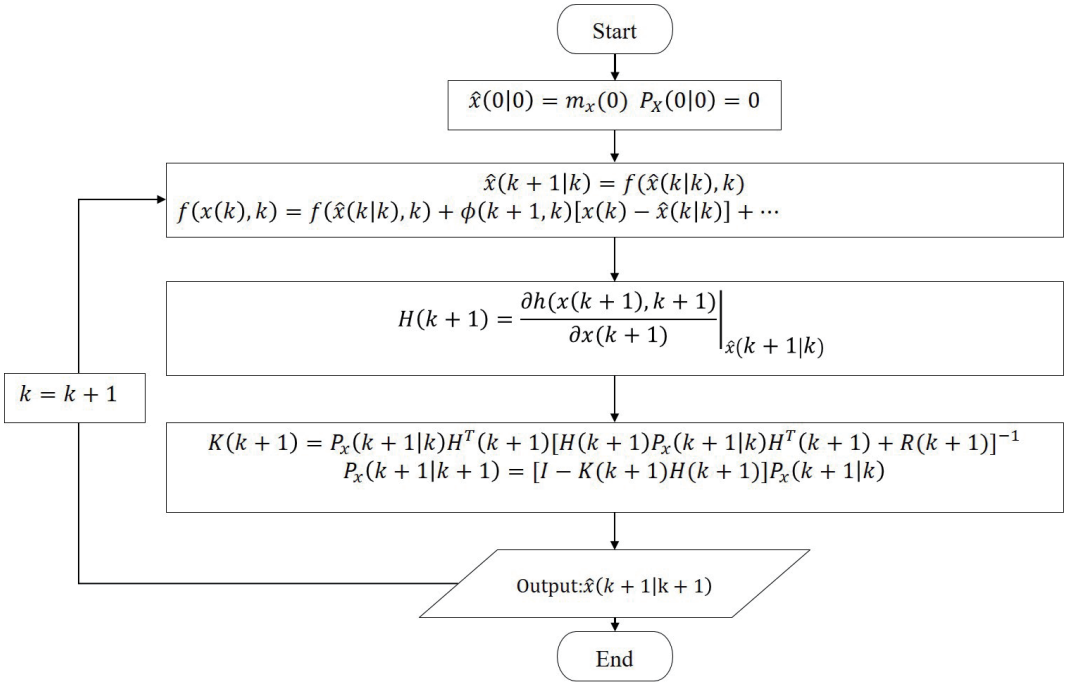


Fig. 16. EKF calculation process.

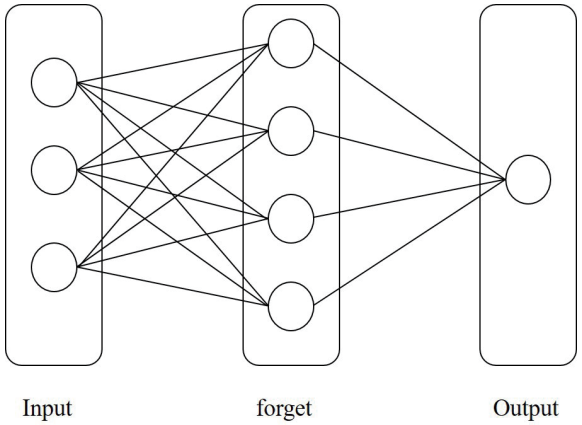


Fig. 17. LSTM structure.

It takes at least three levels to build the LSTM, namely, the input level (i_t), forget level (f_t), and output level (O_t). The data to be stored for the next status are selected at the input level. The values not to be stored in the status data are selected at the hidden level. If a value is 0 as calculated, it will be kept at the hidden level. On the other hand, it will be forgotten if it is calculated as 1. In a simple term, the hidden level is a level where data are determined to be useful or not. Finally, the data are transmitted to the output level where a confirmed status message is generated.

x_t is input, f_t is the forget level, i_t is the input level, c_t is the current status, O_t is the output level, and h_t is the output. The forget level controls whether to transmit the message in the memory cell of last time, c_{t-1}, to the current time, and the output time controls the input of the current time, x_t, to the memory cell of the current time through the candidate memory cell, c_t If the forget level stays close to 1 and the input level to 0, the past memory cell will keep transmitting to the current time through time storage. This design is the response to the gradient decay issue in RNN, as shown in Fig. 18.

5. System Verification

There were three interruptions in this experiment. It started on Zhongshan Rd at coordinates 24.15028692183025 and 120.7318637891659. The GPS signals were lost at the first turn on Dongying Rd at coordinates 24.14853077029017 and 120.7063391068262. As the vehicle traveled to Leye Rd at coordinates 24.1376073742325 and 120.70535724891008, the GPS signals were blocked at the second turn. Next, the vehicle made the third turn to Section 1, Zhongshan Rd at coordinates 24.13473977507443 and 120.73280792666246, and the GPS signals were lost at coordinates 24.147464780480675 and 120.7337514383481. Figure 19 shows the start, turns, and signal interruptions on Google Map. For a better presentation of the travel route, the RTK points were plotted into a closed route of the entire travel for comparison in greater detail, as shown in Fig. 20.

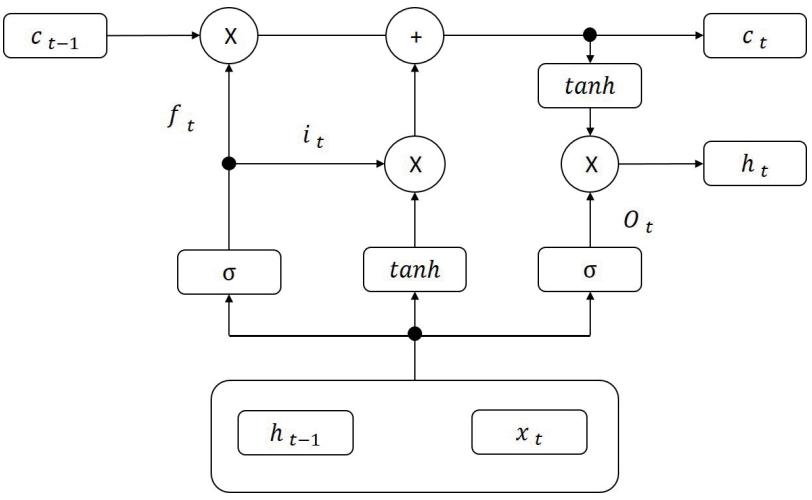


Fig. 18. LSTM algorithm composition.

Fig. 19. (Color Online) Actual road map.

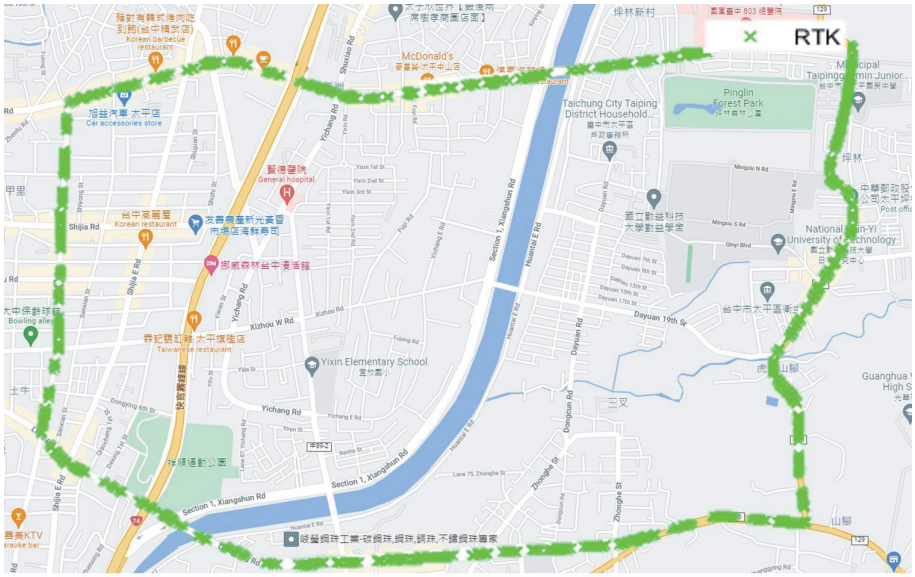


Fig. 20. (Color Online) Road map and RTK.

It can be seen from Fig. 21(a) that the RTK GPS output positioning map within 9 km around the campus can be drawn using MATLAB. When the vehicle turns left in the upper right corner, the RTK receiver signal is blocked to cause the RT signal to be interrupted to verify the correction method outside the system design, as shown in Fig. 21(b). The acceleration and angular speed measured by inertial sensors on RTK signal interruption were recorded and substituted into MATLAB to calculate and plot the route compensated by INS, as shown in Fig. 22(a). Apparently, the accuracy of simple compensation is very low. In Fig. 22(b), therefore,

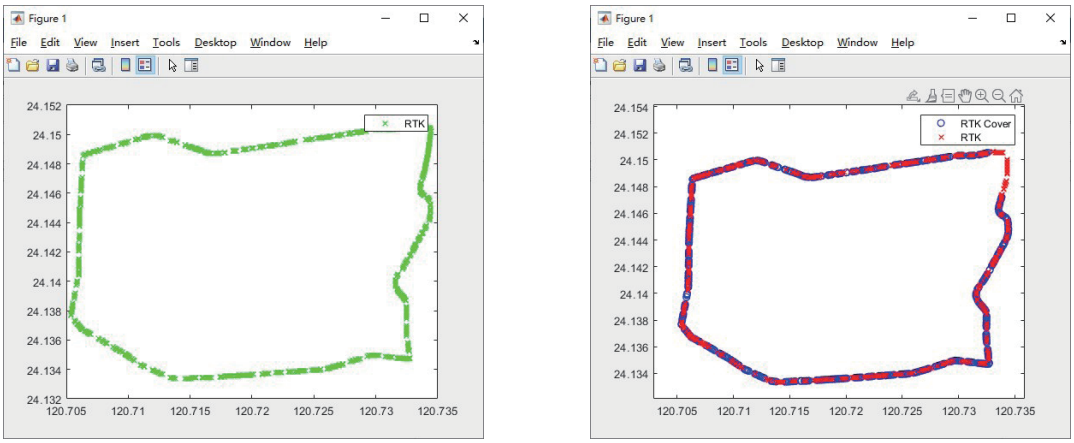


Fig. 21. (Color Online) (a) RTK GPS receiver output plots and (b) RTK GPS receiver signals blocked.

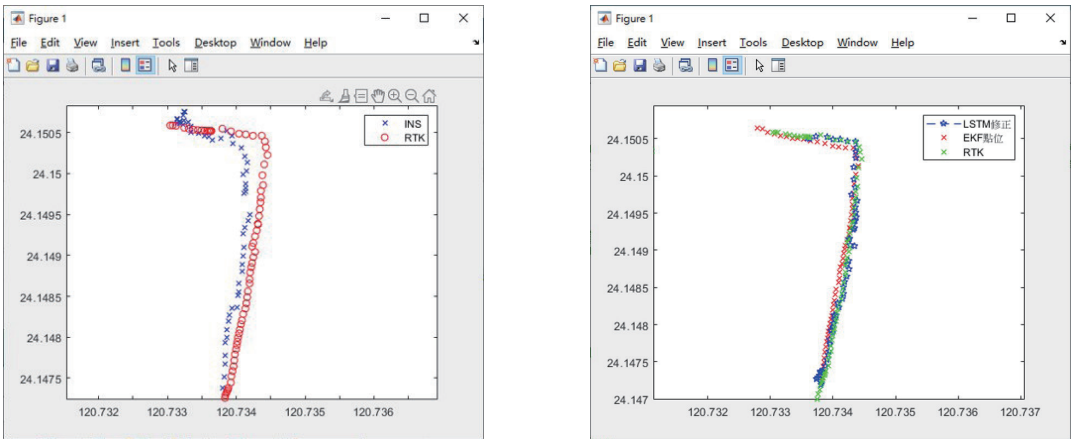
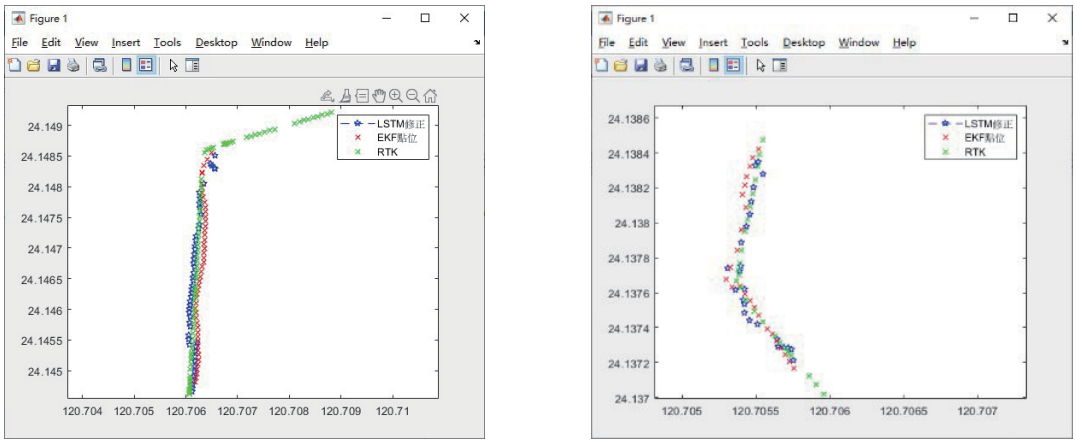


Fig. 22. (Color Online) (a) RTK and INS-compensated points and (b) LSTM and EKF-compensated points.

EKF and LSTM, which are both commonly used, were introduced in this study for the correction of INS-compensated points, so that the longitude is close to the RTK positioning point. With EKF and LSTM compensation, it is found that the points after compensation were very close to the RTK points. However, the values of LSTM-corrected points were much closer to RTK point values than EKF. It is also clear in the other two interruptions in Figs. 23(a) and 23(b) that the LSTM compensation resulted in positioning points closer to reality along every section of the travel. All points are plotted in Fig. 24 for a clear picture of the difference between before and after correction.

6. Experiment and Test Results and Discussion

A field test was performed to evaluate the performance of the learning algorithm proposed. A car was used as the carrier and mounted with simple inertial sensors and an RTK receiver. The



(a) (b)

Fig. 23. (Color Online) (a) Upper left LSTM and EKF-compensated points and (b) lower left LSTM and EKF-compensated points.

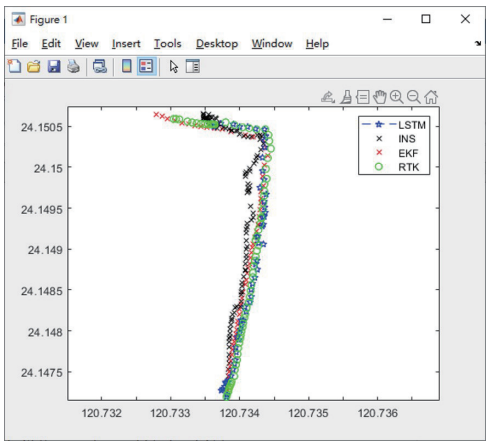


Fig. 24. (Color Online) LSTM, INS, and EKF-compensated points in relation to RTK receiving points.

car traveled in streets and alleys lined with buildings at a fluctuating speed, since the car did not travel on open roads. The average speed of the travel was 50 km/h. Table 1 shows that, at the interruption at the upper right corner, the minimum longitude and latitude errors were 0.893 and 1.083 m, respectively, with the simple compensation of inertial sensors. The minimum longitude and latitude errors with EKF correction, on the other hand, were 0.303 and 0.28 m, respectively. Compared with the minimum errors with EKF correction, the minimum longitude and latitude errors with LSTM correction were merely 0.049 and 0.036 m, respectively. In Table 2, the EKF-corrected minimum longitude and latitude errors at the upper left corner were 0.168 and 0.257 m, and they were 0.174 and 0.386 m, respectively, with LSTM correction. In Table 3, the EKF-corrected minimum longitude and latitude errors at the lower left corner turn were 0.462 and 0.51 m, and they were 0.328 and 0.264 m, respectively, with LSTM correction. The result suggests that LSTM is more effective than EKF in terms of reducing longitude and latitude errors. The experiment results in Table 4 indicate that the LSTM-corrected RMSE values are

Table 1
Minimum coordinate errors with INS, EKF, and LSTM corrections.

Min. error	Correction		
	INS (m)	EKF (m)	LSTM (m)
Longitude	0.893	0.303	0.049
Latitude	1.083	0.28	0.36

Table 2
Min. coordinate errors at upper left turn with EKF and LSTM corrections.

Min. error	Correction	
	EKF (m)	LSTM (m)
Longitude	0.168	0.174
Latitude	0.257	0.386

Table 3
Min. coordinate errors at lower left turn with EKF and LSTM corrections.

Min. error	Correction	
	EKF (m)	LSTM (m)
Longitude	0.462	0.328
Latitude	0.51	0.264

Table 4
RMSE values with INS, EKF, and LSTM corrections.

RMSE	Correction		
	INS (m)	EKF (m)	LSTM (m)
Longitude	9.877	9.435	4.529
Latitude	22.908	10.076	3.433

Table 5
RMSE values at upper left turn with EKF and LSTM corrections.

RMSE	Correction	
	EKF (m)	LSTM (m)
Longitude	7.961	2.747
Latitude	8.696	4.163

Table 6
RMSE values at lower left turn with EKF and LSTM corrections.

RMSE	Correction	
	EKF (m)	LSTM (m)
Longitude	13.492	1.609
Latitude	19.106	4.083

4.529 and 3.433 m, greater than the EKF-corrected RMSE longitude and latitude, which are 9.435 and 10.076 m, respectively. The results in Table 5 suggest that the TM-corrected RMSE values are 2.747 and 4.163 m at the upper left turn. The results in Table 6 indicate that the LSTM-corrected RMSE values are 1.609 and 4.083 m at the lower left turn. In both cases, they are more

accurate than EKF-corrected RMSE longitudes and latitudes, which are 7.961 and 8.696 m, and 13.492 and 19.106 m, respectively. The compensations and corrections at the three turns verify that the proposed method is more accurate than the common correction with EKF and produces results closer to points captured by RTK.

7. Conclusions

The GPS system often becomes unstable or even interrupted due to climate and surroundings, among other factors. It is common to couple it with inertial sensors to compensate roads where signals are blocked and to keep the onboard positioning system running. However, the accuracy of inertial sensors will deteriorate over time owing to the divergence of accumulated errors. Therefore, a KF is added to correct the data generated by the inertial sensor and improve the positioning accuracy of the system. However, the use of relatively inexpensive inertial sensors leads to significant positioning errors in the navigation system. Even the correction with EKF does not necessarily provide the accuracy within 10 cm for the errors captured by the RTK GPS receiver. Therefore, long- and short-term memory with high accuracy is introduced to correct the system errors. The results show that LSTM compensation and correction are more accurate than EKF correction at all three turns and closer to the points captured by RTK. The LSTM-compensated points are closest to the RTK points even with cheap and simple sensors. Also, the maximum longitude and latitude errors with EKF correction are 13 and 15 m, whereas those with LSTM correction are merely 4.529 and 3.433 m, respectively. Clearly, LSTM correction is more accurate than EKF correction.

However, this study uses offline calculation and prediction, and compensates and corrects the system positioning coordinates. An idea for future development is to streamline the entire process and develop the software with C or Python for real-time calculation, compensation, and correction, allowing highly accurate positioning and navigation with a simple inertial sensor when GPS signals are blocked.

Acknowledgments

The authors are very grateful to the Ministry of Science and Technology, Taiwan for the financial support (MOST 108-2221-E-167-027-MY2).

References

1 A. Kaur, P. Balsundar, V. Kumar, A. Mantri, and A. N. J. Raj: 2016 5th Int. Conf. Wireless Networks and Embedded Systems (WECON, 2016) 1.
2 N. C. Yadav, A. Shanmukha, B. M. Amruth, and Basavaraj: 2017 Int. Conf. Algorithms, Methodology, Models and Applications in Emerging Technologies (ICAMMAET, 2017) 1.
3 W.-J. Feng, S.-Z. Yang, and F. Zhao: 2001 Int. Conf. Info-Tech and Info-Net. Proceedings (Cat. No. 01EX479, 2001) 352.
4 D. M. G. and A. Arun: 2018 Int. Conf. Design Innovations for 3Cs Compute Communicate Control (ICDI3C, 2018) 18.
5 Z. Guo, Y. Hao, F. Sun, and W. Gao: 2008 IEEE Int. Symp. Knowledge Acquisition and Modeling Workshop (ISKAMW, 2008) 844.

6 M. Malleswaran, S. A. Deborah, S. Manjula, and V. Vaidehi: 2010 11th Int. Conf. Control Automation Robotics & Vision (CARV, 2010) 2427.
7 D. Li, X. Jia, and J. Zhao: IEEE Access **8** (2020) 53984. http://doi.org/10.1109/ACCESS.2020.2981015.
8 Y. Yang, Y. Zhong, and Y. Gao: 2017 7th IEEE Int. Conf. Electronics Information and Emergency Communication (ICEIEC, 2017) 469–472.
9 H. Xiong, Z. Mai, J. Tang, and F. He: IEEE Access **7** (2019) 26168. https://doi.org/10.1109/ACCESS.2019.2897222.
10 Y. D. Lee, W. J. Yoo, L. W. Kim, B. S. Choi, and H. K. Lee: 2019 European Navigation Conf. (ENC, 2019) 1–4.
11 M. Malleswaran, V. Vaidehi, and M. Mohankumar: 2011 Third Int. Conf. Advanced Computing (ICAC, 2011) 378–383.
12 M. A. Istiake Sunny, M. M. S. Maswood, and A. G. Alharbi: 2020 2nd Novel Intelligent and Leading Emerging Sciences Conf. (NILES, 2020) 87–92.
13 S. D. Kumar and D. Subha: 2019 3rd Int. Conf. Trends in Electronics and Informatics (ICOEI, 2019) 1248–1253.
14 J. C. Juang: Satellite Navigation (Chuan Hwa Publishing Ltd., Taiwan, 2012).
15 S. Hochreiter and J. Schmidhuber: Neural Computation **9** (1997) 1735. https://doi.org/10.1162/neco.1997.9.8.1735.

About the Authors

Guo-Shing Huang (Senior Member, IEEE) received his B.S. degree in electrical engineering in 1980 and M.S. degree in automatic control engineering in 1983 from Feng Chia University, and his Ph.D. degree in 1998 from the Department of Electrical Engineering, National Cheng Kung University. He is currently a professor with the Institute of Electronic Engineering, National Chin-Yi University of Technology and the Chair of the Taiwan Industrial Robot Association. His research interests include control application, integrated GPS/INS electronic navigation, fuzzy/neural network control, intelligent robotic location navigation and control, dual robotic arms motion control, and wearable lower limb exoskeleton robots.
(hgs@ncut.edu.tw)

Yu-Fan Wu is studying for his M.S. degree in electrical engineering at National Chin-Yi University. His research interest is integrated GPS/INS electronic navigation. (4a913010@gm.student.ncut.edu.tw)

Ming-Cheng Kao received his M.S. and Ph.D. degrees in electrical engineering from National Sun Yat-Sen University in 2000 and 2004, respectively. Since 2006, he has been at Hsiuping University of Science and Technology. His current research interests are in the areas of navigation, thin-film technology, and electronic devices. (kmc@hust.edu.tw)