

Research Article

A LSTM-RNN-Assisted Vector Tracking Loop for Signal Outage Bridging

Di Liu ¹, Qingyuan Xia ², Changhui Jiang ², Chaochen Wang,³ and Yuming Bo³

¹School of Automation, Nanjing Institute of Technology, Nanjing, China

²School of Computer Science and Engineering, Nanjing University of Science and Technology, Nanjing, China

³School of Automation, Nanjing University of Science and Technology, Nanjing, China

Correspondence should be addressed to Qingyuan Xia; xqy@njut.edu.cn and Changhui Jiang; chiang_changhui@outlook.com

Received 6 November 2019; Revised 24 June 2020; Accepted 1 July 2020; Published 12 August 2020

Academic Editor: Angelo Cervone

Copyright © 2020 Di Liu et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Global Navigation Satellite System (GNSS) has been the most popular tool for providing positioning, navigation, and timing (PNT) information. Some methods have been developed for enhancing the GNSS performance in signal challenging environments (urban canyon, dense foliage, signal blockage, multipath, and none-line-of-sight signals). Vector Tracking Loop (VTL) was recognized as the most promising and prospective one among these technologies, since VTL realized mutual aiding between channels. However, momentary signal blockage from part of the tracking channels affected the VTL operation and the navigation solution estimation. Moreover, insufficient available satellites employed would lead to the navigation solution errors diverging quickly over time. Short-time or temporary signal blockage was common in urban areas. Aiming to improve the VTL performance during the signal outage, in this paper, the deep learning method was employed for assisting the VTL navigation solution estimation; more specifically, a Long Short-Term Memory-Recurrent Neural Network (LSTM-RNN) was employed to aid the VTL navigation filter (navigation filter was usually a Kalman filter). LSTM-RNN obtained excellent performance in time-series data processing; therefore, in this paper, the LSTM-RNN was employed to predict the navigation filter innovative sequence values during the signal outage, and then, the predicted innovative values were employed to aid the navigation filter for navigation solution estimation. The LSTM-RNN was well trained while the signal was normal, and the past innovative sequence was employed as the input of the LSTM-RNN. A simulation was designed and conducted based on an open-source Matlab GNSS software receiver; a dynamic trajectory with several temporary signal outages was designed for testing the proposed method. Compared with the conventional VTL, the LSTM-RNN-assisted VTL could keep the horizontal positioning errors within 50 meters during a signal outage. Also, conventional Support Vector Machine (SVM) and radial basis function neural network (RBF-NN) were compared with the LSTM-RNN method; LSTM-RNN-assisted VTL could maintain the positioning errors less than 20 meters during the outages, which demonstrated LSTM-RNN was superior to the SVM and RBF-NN in these applications.

1. Introduction

Currently, users are capable of obtaining accurate positioning, navigation, and timing (PNT) information from a handheld or chip-scale Global Navigation Satellite System (GNSS) receiver [1–3]. Smartphones, smartwatches, or shared bicycles are all equipped with a chip-scale GNSS receiver for acquiring continuous PNT information. By its nature, the GNSS is a satellite-based navigation system, and the receivers receive the transmitted signals from the

navigation satellites in space orbit covering the earth. A GNSS receiver usually consists of three major functional blocks (shown in Figure 1), which are signal acquisition, signal tracking, and navigation solution estimation [3–5]. The working flow of a typical GNSS receiver can be illustrated as follows: firstly, Radio Front (RF) receives the signal and then samples and downconverts the high-frequency navigation satellite signals for further processing in the later blocks; secondly, signal acquisition accomplishes coarse code and frequency estimation, and the results are converted to

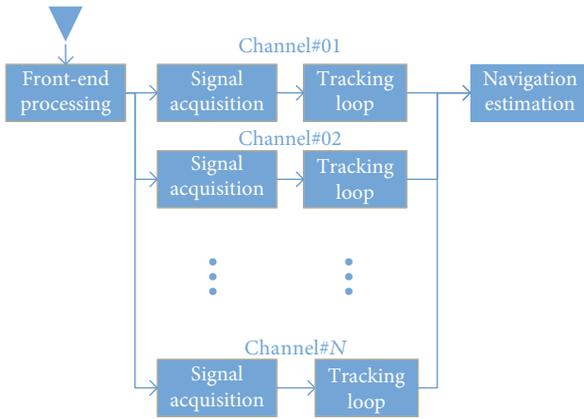


FIGURE 1: Scheme of the Scalar Tracking Loops.

tracking loop component for initializing the signal tracking operation; thirdly, signal tracking is responsible for locking the signal and generating code and frequency measurements for navigation solution (position and navigation) determination in the final block. Due to the existence of the clock bias and drift, at least four satellites are necessary for the navigation solution estimation.

In a receiver, signal tracking refers to the code tracking and carrier signal tracking. Code tracking generates the pseudorange measurements for position estimation. Meanwhile, the carrier tracking generates the pseudorange rate measurements for velocity estimation. Commonly, a Least Square Method (LSM) or Kalman filter is employed for accomplishing the navigation solution estimation [3–5]. As aforementioned, in the above architecture, each channel of the receiver carries out the signal tracking loop individually, and the navigation solutions are not fed back to aid the signal tracking. This architecture is termed as Scalar Tracking Loop (STL), and this scheme has the following characteristics: each channel accomplishes tracking its corresponding satellite signal individually, no information exchanging happens among different channels, no information is fed back to the signal tracking or signal acquisition block, and the three consecutive blocks are comparatively independent. Basically, the STL ignores the inherent relationship between navigation solutions and each channel's signal tracking parameters (position information is determined with code tracking results, and the velocity is determined with the carrier tracking results). By its nature, all channels share common navigation solutions; this characteristic might be helpful for signal acquisition and tracking [1–5].

Different from the STL, Vector Tracking Loop (VTL) utilizes a center navigation filter (usually a Kalman filter) to process all the channels tracking together and generate navigation solutions. The architecture of the VTL is presented in Figure 2, navigation solutions are fed back to signal tracking for generating the local signal. Then, the local signal replica is mixed with the received signal for generating measurements for navigation solution updating. VTL is commonly composed of a Vector Delay Lock Loop (VDLL) and a Vector Frequency Lock Loop (VFLL). A VDLL was firstly proposed by Spilker, and it attracts serious con-

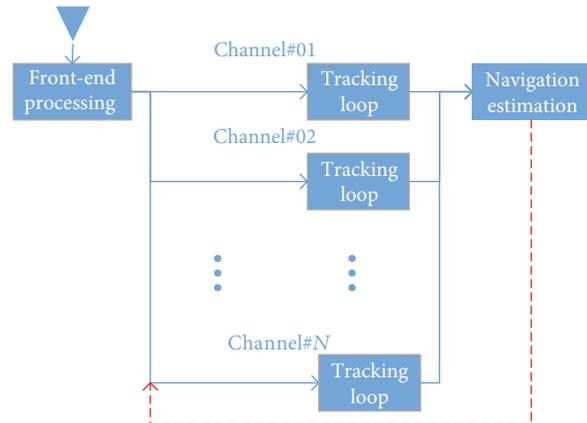


FIGURE 2: Vector Tracking Loops.

cern from researchers since it has superior performance than STL in tracking weak signals [6]. Lately, Lashley et al. implemented a VFLL/VDLL and compared its performance with STL; the results showed that VTL performed 6.2 dB weak signal tracking threshold improvement with eleven satellites in view and 2.4 dB improvement with five satellites available [7–10]. In addition, Pany and Eissfeller found that VTL could track 10 dB-Hz weaker signals than STL.

Based on the VTL performance analysis, researchers were devoted to enhance the performance of the VTL in a signal challenging environment [11]. According to the past researches, the VTL encountered three major problems influencing the VTL performance. The first problem was the error propagation between channels, which was brought by the centralized processing of all the channels. The second problem was nonlinearity of the measurement equations; usually, the measurement equation was linearized and a Kalman filter was employed for processing it. The third problem was that signal outage or blockage negatively affected the VTL performance. For the first problem, Won and Eissfeller defined a geometry-related range accuracy coefficient, which could be employed for analyzing the interactions between different channels and how these channels shared the common information [12, 13]. Channel interaction analysis could be a theoretical basis for designing an adaptive VTL for dealing with the condition that part of the signals was of low quality [12, 13]. For the nonlinear problem, Chen designed an adaptive iterative extended Kalman (AIEKF) filter for VTL [12, 13]; Xia designed an adaptive robust Cubature Kalman filter (AR-CKF) for enhancing the VTL navigation accuracy [14]. Apart from these methods, Grace proposed multivector tracking receiver architecture, and the positioning accuracy decreased with the collaborative positioning estimation of the multiple receivers [15]. Jiang employed a Chip Scale Atomic Clock (CSAC) to support VTL positioning, and the results showed that the CSAC could improve the VTL positioning accuracy [16]. The Dilution of Precision (DOP) values decreased with the CSAC as the local time reference [17].

Despite these efforts, a momentary signal blockage will inhibit VTL which generated continuously accurate navigation solutions. Limited work has been published on solving

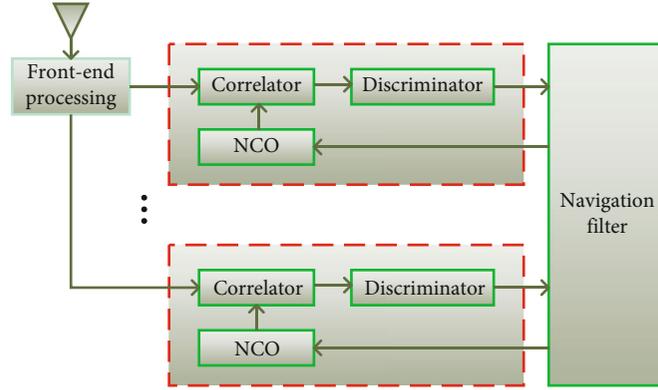


FIGURE 3: Basic architecture of the VTL.

this problem. Zhao proposed a neural network-assisted GPS/BDS VTL; in this method, NN was employed to predict the Doppler frequency and code phase during a signal outage [18]. Jwo et al. employed the radial basis function neural network (RBFNN) and the adaptive network-based fuzzy inference system (ANFIS) to predict the input of the adequate numerical control oscillator (NCO) inputs [19]. Different from conventional Recurrent Neural Networks (RNN), Long Short-Term Memory- (LSTM-) RNN was designed with a unique “gate” structure which could make full use of the history information to build the fitting model [20–22]. LSTM-RNN performed better performance in the time-series data processing [20–22]. In this work, LSTM-RNN was employed to aid the navigation filter and bridge the signal outage. In the training stage, the output information from the discriminators and navigation filter would be adopted as the inputs of the LSTM-RNN. During signal outages, the LSTM-RNN was employed to predict innovative sequence in the navigation filter for aiding navigation solution generation. Performance evaluation for the LSTM-RNN-assisted VTL was carried out by comparing it with the conventional VTL without any assistance.

The rest of this paper is organized as follows: Section 2 illustrates the basic architecture and equations of the VTL; the LSTM-RNN is introduced in Section 3, and the LSTM-RNN-assisted VTL is discussed; the LSTM-RNN-assisted VTL simulation and analysis are presented, and the results are discussed in Section 4. Conclusions are given in Section 5.

2. Vector Tracking Loop

Figure 3 presents the illustrative architecture of employed VTL; it has two essential components: signal tracking and navigation filter. In the VTL, firstly, the discriminators generate measurements for navigation filter, and then, the navigation filter employs the measurements to estimate the navigation solutions; secondly, the navigation solutions are fed back to generate local signal replica, which are mixed with the received signal in the correlators, and then generate the measurements again. The discriminators are placed after the correlators, and the information from the correlators is employed in the discriminators for measurement generation. Subsection 2.1 and Subsection 2.2 give the relationship

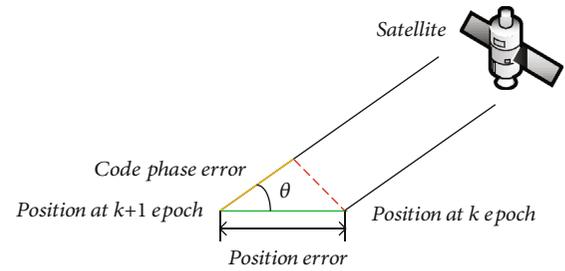


FIGURE 4: Relationship between the code phase and the position increment.

between discriminators and the navigation filter and the related equations.

2.1. Modeling Discriminator Output and Navigation Solutions. The relationship between position error and the code phase error is described in Figure 4. In this figure, the position error and the code phase error can be modelled using Equation (1):

$$\varphi_{\text{code},k+1} = \Delta \mathbf{X}_{k+1} \cdot \cos(\theta), \quad (1)$$

$$\Delta \mathbf{X}_{k+1} = \mathbf{X}_{k+1} - \mathbf{X}_k, \quad (2)$$

where the $\varphi_{\text{code},k}$ is the code phase error, $\Delta \mathbf{X}_{k+1}$ is the position error vector, the position error refers to the motion between k_{th} position \mathbf{X}_k and $(k+1)_{th}$ position \mathbf{X}_{k+1} , and its projection on the line-of-sight vector is corresponding to the code phase increment. In fact, the code phase error is also affected by the local clock bias, and the detailed relationship between the user's position and code phase error is listed as

$$\varphi_{\text{code},k+1} = \Delta \mathbf{X}_k^T \cdot \cos(\theta) + c \cdot t_{b,k+1} + \omega_{k+1}, \quad (3)$$

where the c is the light speed, $t_{b,k+1}$ is the clock bias at $(k+1)_{th}$ epoch, and ω_{k+1} is the code phase noise. Specifically, the $\cos(\theta)$ is determined by the relative position of the receiver and the user.

Apart from the relationship between position error and code phase error, the velocity and carrier frequency error have an analogous connection. Equations (4) and (5) describe the relationship in detail.

$$\Delta f_{k+1} = \Delta \mathbf{V}_{k+1}^T \cdot \cos(\theta) + c \cdot t_{d,k+1} + \eta_{k+1}, \quad (4)$$

$$\Delta \mathbf{V}_{k+1} = \mathbf{V}_{k+1} - \mathbf{V}_k, \quad (5)$$

where the Δf_{k+1} refers to the carrier frequency error at $(k+1)_{th}$ epoch, $t_{d,k+1}$ is the local time drift, and the η_{k+1} is the noise. \mathbf{V}_{k+1} is the velocity vector at $(k+1)_{th}$ epoch; \mathbf{V}_k is the estimated velocity vector at k_{th} epoch.

2.2. Navigation Filter Model. As aforementioned, subsection 2.1 illustrates how to utilize the code phase and frequency error estimate of the position and velocity increment. Based on the above analysis, this subsection will give the model of the navigation filter. The navigation filter is usually a Kalman filter, which consists of a state equation and measurement equation.

In the state equation, the position errors, velocity errors, clock bias, and clock drift are the state variables, and the equation is given as

$$\Delta \mathbf{all_X}_{k+1} \begin{bmatrix} \Delta \mathbf{X}_{k+1} \\ \Delta \mathbf{V}_{k+1} \\ c \cdot t_{b,k+1} \\ c \cdot t_{d,k+1} \end{bmatrix} = \mathbf{F}_{k,k+1} \cdot \begin{bmatrix} \Delta \mathbf{X}_k \\ \Delta \mathbf{V}_k \\ c \cdot t_{b,k} \\ c \cdot t_{d,k} \end{bmatrix} + \mathbf{W}, \quad (6)$$

where the $\mathbf{F}_{k,k+1}$ is the state transfer matrix and the \mathbf{W} is the state noise. More details of the state equation could be found in our previously published paper [3].

Based on the analysis of the relationship between the state variables and the measurements, thus, the measurement equation could be formed as

$$\mathbf{Z}_{k+1} = \mathbf{H}_k \cdot \begin{bmatrix} \Delta \mathbf{X}_k \\ \Delta \mathbf{V}_k \\ c \cdot t_{b,k} \\ c \cdot t_{d,k} \end{bmatrix} + \mathbf{\Lambda}_k, \quad (7)$$

where the \mathbf{Z} is the measurements which is composed of the outputs from discriminators; specifically, the code discriminators produce code increments and the carrier frequency discriminators offer frequency errors between epochs. \mathbf{H}_k is the matrix describing the relationship between the state variables and the measurement information; $\mathbf{\Lambda}_k$ is the measurement noise matrix. In addition, more information about how to calculate the above matrix or measurements is revealed in our previous papers [3].

2.3. Kalman Filter Implementation in VTL. As the navigation filter model illustrated in the above subsection, the following equations will give the updating and the details of the Kalman filter in VTL. Commonly, a typical Kalman filter contains a prediction procedure and state updating procedure [3]. The state prediction is as

$$\Delta \mathbf{all_X}_{k+1}^- = \mathbf{F}_{k+1|k} \Delta \mathbf{all_X}_k, \quad (8)$$

$$\mathbf{P}_{k+1}^- = \mathbf{F}_{k+1|k} \mathbf{P}_k \mathbf{F}_{k+1|k}^T + \mathbf{Q}. \quad (9)$$

The updating is as follows:

$$\mathbf{K}_{k+1} = \mathbf{P}_{k+1}^- \mathbf{H}_{k+1}^T (\mathbf{H}_{k+1} \mathbf{P}_{k+1}^- \mathbf{H}_{k+1}^T + \mathbf{R})^{-1}, \quad (10)$$

$$\mathbf{all_X}_{k+1} = \hat{\mathbf{X}}_{k+1}^- + \mathbf{K}_{k+1} (\mathbf{Z}_{k+1} - \mathbf{H}_{k+1} \mathbf{all_X}_{k+1}^-), \quad (11)$$

$$\mathbf{P}_{k+1} = (\mathbf{I} - \mathbf{K}_{k+1} \mathbf{H}_{k+1}) \mathbf{P}_{k+1}^-, \quad (12)$$

where the $\Delta \mathbf{all_X}_{k+1}^-$ is the predicted state vector using the previous state and state transfer matrix; \mathbf{P}_{k+1}^- is the variance matrix for the predicted state vector; \mathbf{K}_{k+1} is the gain matrix, which defines the updating weight between the predictions and the new measurements. In addition, the matrix \mathbf{Q} and \mathbf{R} represent the state matrix and observation noise covariance matrix individually, and the specifications are given as Equations (13)–(15).

$$E\{\mathbf{W}\} = E\{\mathbf{\Lambda}\} = 0, \quad (13)$$

$$E\{W_k W_i^T\} = \begin{cases} Q_k, & i = k, \\ 0, & i \neq k, \end{cases} \quad (14)$$

$$E\{\Lambda_k \Lambda_i^T\} = \begin{cases} R_k, & i = k, \\ 0, & i \neq k, \end{cases} \quad (15)$$

where the function $E\{\cdot\}$ denotes the expectation calculation.

3. LSTM-RNN-Assisted VTL

LSTM-RNN is a popular variant of basic RNN; firstly, we introduce the components and structure of it [16]. How the LSTM-RNN works will be illustrated and explained in this section. Secondly, the implementation of LSTM-RNN in VTL will be introduced, and the details will be given.

3.1. LSTM-RNN. A classic LSTM-RNN is composed of three “gates,” which are “input gate,” “output gate,” and “forget gate” [16]. These gates have different functions, and they work together for accomplishing the task. Figure 5 gives the structure of the LSTM-RNN unit. As presented, the first gate is the “input gate.” The input gate regulates the input data and the update of the state vector. The following two formulas are given to describe the procedure.

$$\mathbf{i}_t = \sigma(\mathbf{W}\mathbf{e}_i \cdot [\mathbf{h}_{t-1}, \mathbf{x}_t] + \mathbf{b}_i), \quad (16)$$

$$\tilde{\mathbf{C}}_t = \tanh(\mathbf{W}\mathbf{e}_C \cdot [\mathbf{h}_{t-1}, \mathbf{x}_t] + \mathbf{b}_C), \quad (17)$$

where the $\mathbf{W}\mathbf{e}_i$ and $\mathbf{W}\mathbf{e}_C$ are the weights, and the \mathbf{b}_i and \mathbf{b}_C are the bias; these parameters will be determined in the training procedure. As listed in Figure 5, the new candidate cell state $\tilde{\mathbf{C}}_t$ is constructed by the tanh function, then, $\tilde{\mathbf{C}}_t$ is multiplied with the vector \mathbf{i}_t , and the multiplication results are added to the cell state for updating.

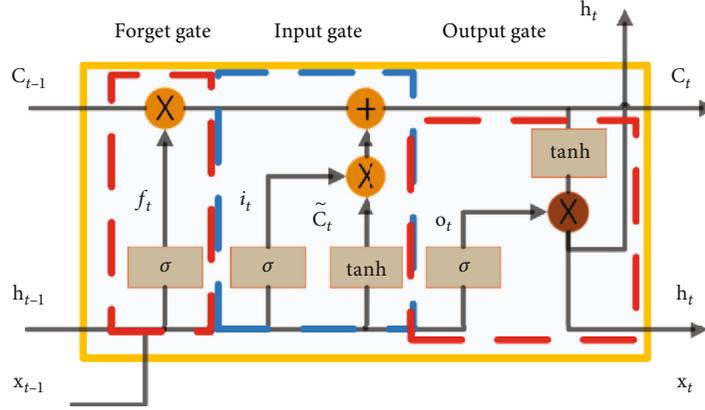


FIGURE 5: Long Short-Term Memory (LSTM) unit.

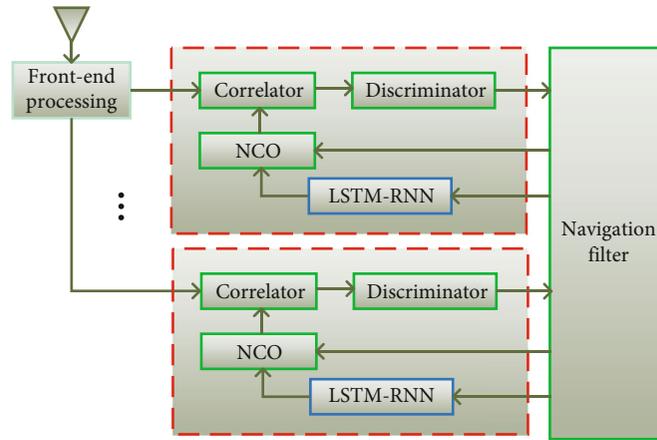


FIGURE 6: LSTM-RNN-assisted VTL.

The second is the “forget gate,” which is responsible for regulating the previous information. This operation is usually carried out using a sigmoid function. Assuming the input vector is \mathbf{I}_t , then the function is given as

$$\mathbf{f}_t = \sigma(\mathbf{W}\mathbf{e}\mathbf{i}_f \cdot [\mathbf{h}_{t-1}, \mathbf{I}_t] + \mathbf{b}_f), \quad (18)$$

where, in Equation (18), $\sigma(\cdot)$ is the sigmoid function, $\mathbf{W}\mathbf{e}\mathbf{i}_f$ is the updating weights vector, and \mathbf{b}_f is the bias vector. The output of this gate \mathbf{f}_t is a vector that contains numbers between 0 and 1, which represents the keeping degree of the corresponding values.

The last gate is the output gate; the output gate is employed for regulating the outputs. As listed in Figure 5, two functions (a sigmoid function and a tanh function) are included in this gate to carry out the operation. This sigmoid function is listed in Equation (19), and the tanh function is listed in Equation (20).

$$\mathbf{o}_t = \sigma(\mathbf{W}\mathbf{e}\mathbf{i}_o \cdot [\mathbf{h}_{t-1}, \mathbf{x}_t] + \mathbf{b}_o), \quad (19)$$

$$\mathbf{h}_t = \mathbf{o}_t \cdot \tanh(C_t). \quad (20)$$

Finally, the cell state output is

$$\mathbf{C}_t = \mathbf{f}_t \cdot \mathbf{C}_{t-1} + \mathbf{i}_t \cdot \tilde{\mathbf{C}}_t. \quad (21)$$

3.2. LSTM-RNN-Assisted VTL Implementation. The designed LSTM-RNN-assisted VTL is illustrated in Figure 6. The LSTM-RNN is integrated with the navigation filter; the past innovative sequences are the input to the LSRM-RNN to assist the VTL. Figure 5 is the just a LSTM-RNN unit, and Figure 7 is the LSTM-RNN from time $t-1$ to $t+1$. The information exchanging is also illustrated in Figure 7. The innovative sequence of the VTL navigation filter is employed as the inputs of the LSTM-RNN, and the innovation is written as

$$\mathbf{innovation} = \mathbf{Z}_{k+1} - \mathbf{H}_{k+1} \mathbf{all} \tilde{\mathbf{X}}_{k+1}^-. \quad (22)$$

While the signal is normal, the LSTM-RNN is trained for determining some parameters, and then, the LSTM-RNN will output the predicted information to aid the VTL during a signal outage. As listed in Figure 6, each channel has a unique LSTM-RNN which obtains the information from the navigation filter. Considering the computation load and

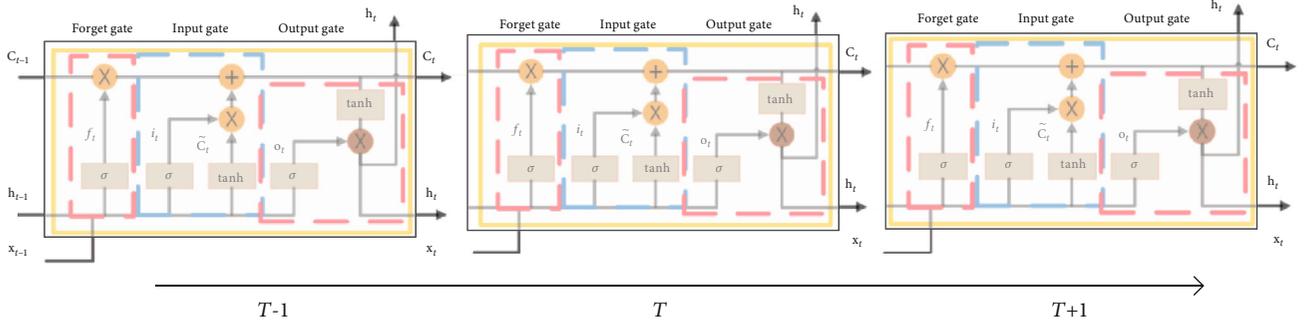


FIGURE 7: LSTM-RNN.

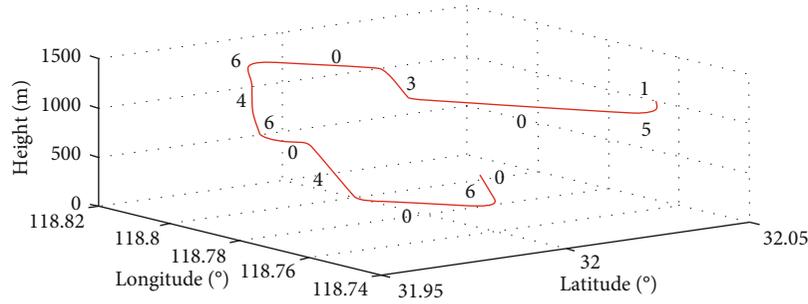


FIGURE 8: 3D plot of the employed dynamic trajectory.

TABLE 1: Motion details of the trajectory.

Task	Motion
0	Uniform flight
1	Acceleration
2	Deceleration
3	Up
4	Down
5	Turn right
6	Turn left

TABLE 2: Acquisition results.

Channel	PRN	Doppler	Code offset
1	4	-2594	9356
2	5	5579	5530
3	13	1441	4628
4	26	3024	8266
5	7	4311	3688
6	8	3272	7368
7	25	-2288	8989
8	2	1688	79

the prediction performance, past 50 innovation sequence in each channel is employed in this VTL.

4. Simulation and Results

Section 2 introduces the basic formulas and the details of the proposed method. In this section, a simulation will be carried out using a flexible Matlab software receiver. The LSTM-RNN was implemented in Matlab and incorporated with the VTL. An intermediate frequency (IF) data was collected from a hardware signal simulator. Through the GUI of the signal simulator, each satellite could be blocked or opened during the simulation. This section is divided into two subsections: (1) firstly, the setting up of this simulation is given in detail, including the trajectory details, the VTL working flow, and LSTM-RNN setting parameters; (2) secondly, the positioning results and the related analysis are given, including the position error comparison with a common VTL during signal outages. (3) SVM and RBFNN were

employed to compare with the LSTM-RNN in terms of the positioning errors.

4.1. Setting Up. Figure 8 shows the 3D dynamic trajectory. The numbers presented in the figure represent different flying task or motion; the specific meanings of all the numbers are listed in Table 1. There are seven different flying tasks in the trajectory. Then, the trajectory file is input to a GNSS signal hardware simulator; an IF data collector is employed for collecting the GNSS signal. The sampling frequency is 10 MHz, and the IF frequency is 3.02 MHz.

Before operating the VTL, some initial information is necessary, for instance, position, velocity, signal tracking parameters, and ephemeris. Table 2 lists the acquisition results including the satellites RPN, Doppler, and code offset, which are essential for initializing the VTL. As revealed in Table 2, eight satellites are included in the simulation. Particularly, GPS is included in this simulation for representing the GNSS. In aspects of the employed LSTM-RNN, considering

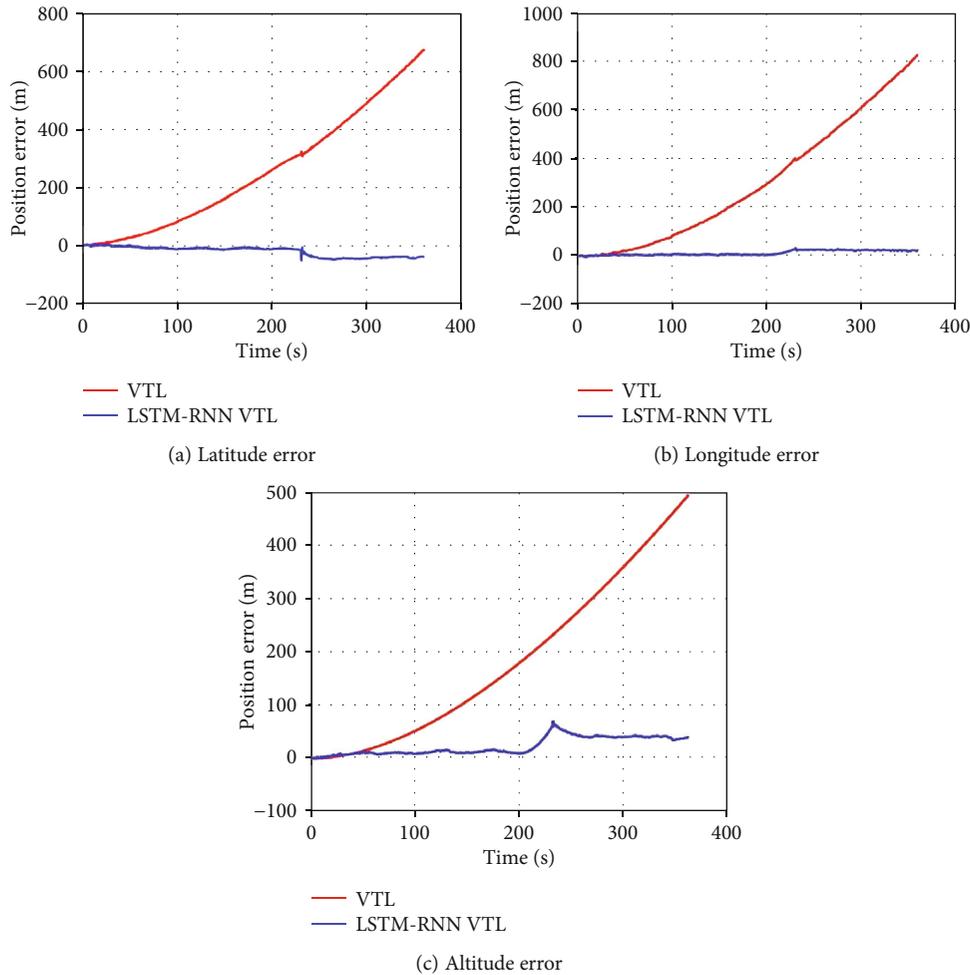


FIGURE 9: Position errors (two available satellites).

the computation load, the LSTM-RNN is a single-layer structure, the learning rate is 0.01, the training epoch is 100, and the batch size is 30.

4.2. LSTM-RNN VTL Results and Analysis. Firstly, with the proper settings, only two satellites are included in the VTL to estimate the navigation solutions, which aims to evaluate the performance of the LSTM-RNN VTL under the condition that part of the signals is blocked at the beginning of the trajectory. As listed in Table 2, eight satellites are acquired, and we randomly select two satellites (PRN 04 and PRN 08) in this simulation. Figure 9 presents the positioning errors including the comparison between VTL and LSTM-RNN-assisted VTL. As shown in Figure 8, the VTL three-axis errors are over 800 m, 800 m, and 500 m (red line presents the VTL positioning errors in Figures 9(a)–9(c)). Blue lines in Figures 9(a)–9(c) represent the positioning errors from the LSTM-RNN-assisted VTL; the three-axis positioning error maximum values are approximately 50 m, 40 m, and 80 m. The results obviously demonstrated the effectiveness of the LSTM-RNN-assisted VTL in positioning error reduction during a long-time signal outage.

Moreover, Figure 10 reveals the position error comparison for signal outages without any available satellites. Figure 10(a)

presents the latitude error; Figures 9(b) and 9(c) plot the longitude and altitude errors. In Figure 10, the LSTM-RNN VTL is compared with a common VTL with available satellites. As plotted in Figure 9, while there are only two satellites employed in VTL, the positioning errors diverge quickly and over 500 m. Therefore, we selected the VTL with enough satellites as the reference.

Figure 10 presents the positioning errors with three different signal outages, and each outage continues approximately 20 seconds. In aspects of the latitude errors (Figure 10(a)), with the LSTM-RNN assistance, the errors still grow during the signal outage. For the outage 1, the maximum errors reach to 15 m; the outage 2 and outage 3 reach to approximately 23 m. In aspects of the longitude errors (Figure 10(b)), the maximum positioning errors are 8 m, 19 m, and 20 m, respectively, for the three individual signal outages. At last, Figure 10(c) gives the results of the altitude errors, the maximum value happens during signal outage 1, and it reaches approximately 26 m, but during the other two signal outage, the errors are both kept below 20 m.

In this simulation, the signal outages all last for 20 seconds, and the results reveal that most of the positioning errors are below 20 m, which is a somewhat satisfying result. As presented in these figures, we can find that the errors of

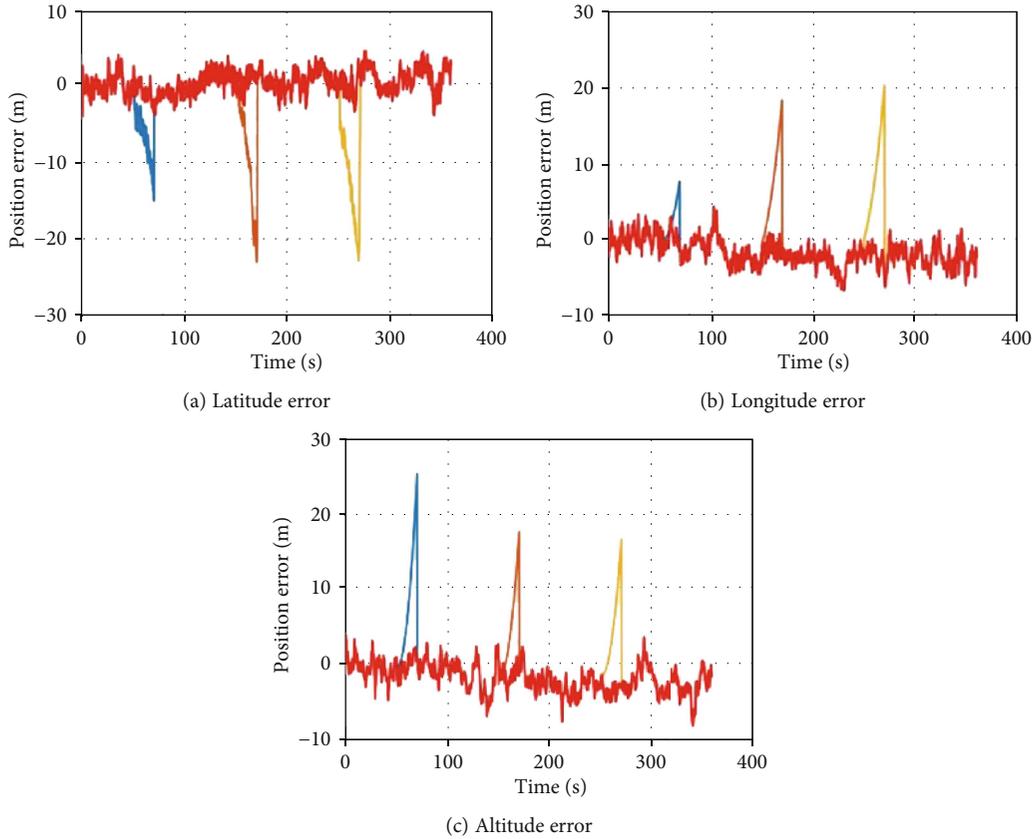


FIGURE 10: Position errors (no available satellites during a signal outage).

the LSTM-RNN will grow further if the signal outage will last for a longer time. LSTM-RNN is effective to reduce or slow down the errors growing. Specifically, compared with Figure 8, in Figure 10, the positioning errors are smaller; since there are two available satellites during the signal outage, this can account why the LSTM-RNN obtains different performance with the same trajectory.

4.3. Comparisons of LSTM-RNN with SVM, RBF-NN. Subsection 4.2 presents the results and analysis of the LSTM_RNN performance during signal outages. For further evaluating the performance of the LSTM-RNN in VTL signal outage, two conventional machine learning methods Support Vector Machine (SVM) and Radial Basis Function Neural Network (RBF-NN) are employed for comparing with LSTM-RNN. The implementation of the SVM and RBFNN heavily references the literature [18, 19]. The testing trajectory is the same as that presented in Figure 8. The comparison is composed of two parts: outage with two available satellites and outages with no available satellites.

Firstly, the positioning results of the long-time outage with two available satellites are presented in Figure 11. The yellow lines represent the results from the SVM, and the green lines represent that from the RBF-NN. As presented, the LSTM-RNN performs the smallest positioning errors during the testing time. The RBF-NN also performs smaller positioning errors than the SVM.

Secondly, the positioning error comparison results between the three methods during the signal outages are presented in Figure 12. The green lines represented the positioning errors from SVM, the red lines presented the positioning errors from LSTM-RNN, and the blue lines represent that from RBFNN. While the signal is normal, the system worked under the integration model; therefore, only blue lines are visible in Figure 12 (excluding the signal outages). It is obvious that the LSTM-RNN suppresses the positioning errors effectively and obtains the best performance compared with SVM and RBFNN.

5. Conclusions

In this paper, a LSTM-VTL was investigated for improving the VTL navigation accuracy during signal outages. A dynamic trajectory with signal blockages was employed to evaluate the performance of the proposed LSTM-RNN-assisted VTL. With the LSTM-RNN aiding, the position errors were reduced, which supported the conclusion that LSTM-RNN was effective for improving the VTL during signal outages. However, we thought there were still some limitations as follows:

- (1) The LSTM-RNN should be trained before its deployment in the VTL; the training was time-consuming,

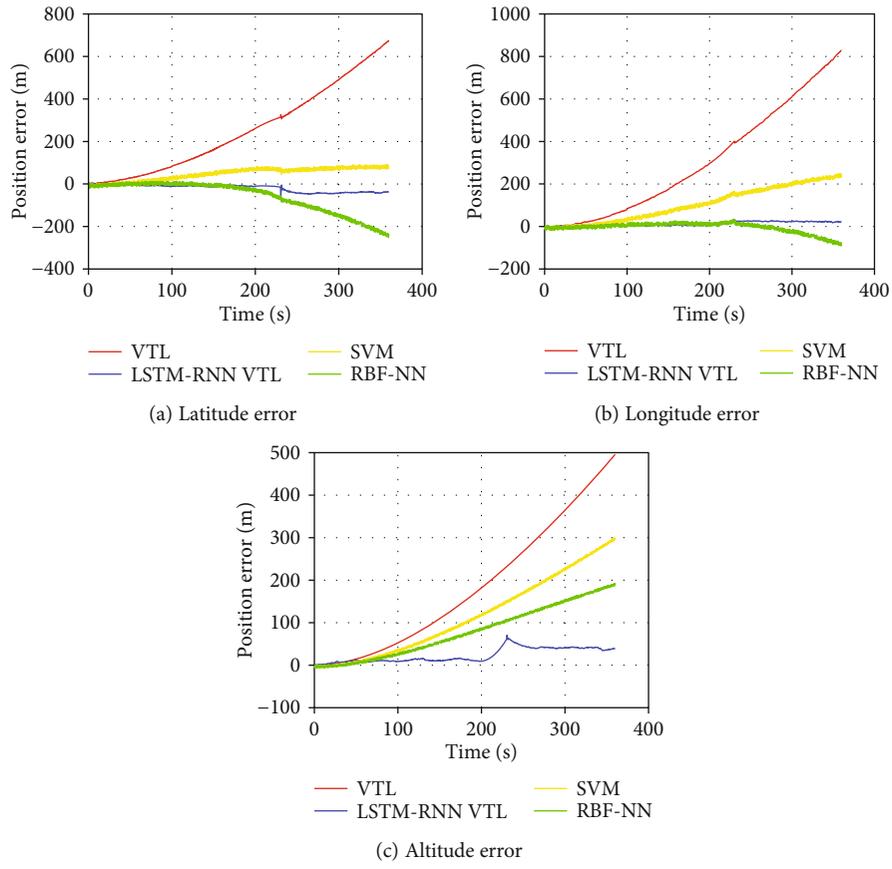


FIGURE 11: Position errors (two available satellites).

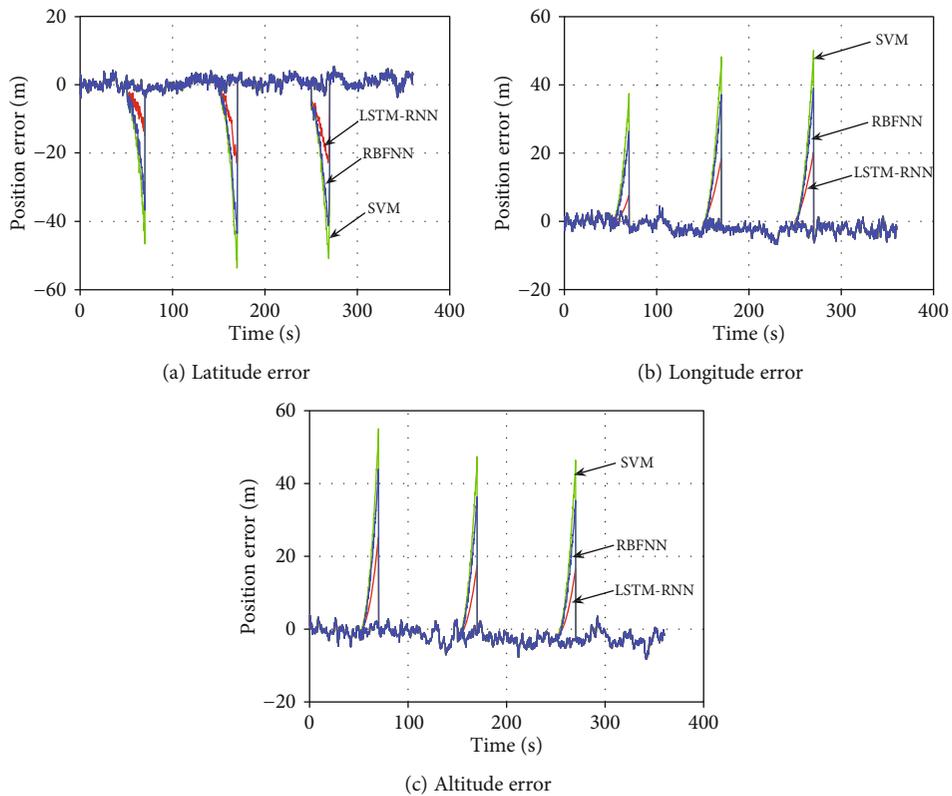


FIGURE 12: Position errors (no available satellites).

which might increase the computation load of the processor

- (2) As presented in Figure 12, position errors of the LSTM-RNN-aided VTL still diverge over time during a signal outage; longer time signal outage will bring about larger positioning errors

In the future, we think the following work is worth further investigation:

- (1) Better or more approximate neural networks are beneficial for improving the performance of the VTL during a signal outage. It is interesting to investigate some other variants of RNN in this application, for instance, Gated Recurrent Unit (GRU)
- (2) A field test is necessary for evaluating the performance of the proposed method, for instance, city canyon, and it is worthwhile for improving positioning accuracy for VTL in signal challenging environments
- (3) For a ground vehicle or UAV, there are other sensors equipped, and it is of significance to investigate the integration of VTL with these sensors for more robust positioning performance

Data Availability

The data and code used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare no conflicts of the paper publication.

Authors' Contributions

Di Liu and Qing Xia carried out the simulation and wrote the first version of this paper; Changhui Jiang proposed this idea; Chaochen Wang collected the data and reviewed the paper; Yuming Bo guided the paper writing and reviewed the paper.

Acknowledgments

This work was supported by the National Natural Science Foundation (NNSF) of China under Grant 61503180, the Six Talent Peaks Project in Jiangsu Province of China under Grant 2016-JXQC-015, the Science Foundation of Higher Education of Jiangsu Province under Grants 15KJA460007 and 18KJB510016, and The Fund of Nanjing Institute of Technology under Grants CKJA201605, CKJB201702, and CKJA201804.

References

- [1] L. Eldredge, *Alternative positioning, navigation & timing (PNT) study*, International Civil Aviation Organisation Navigation Systems Panel (NSP) Working Group Meetings, Montreal, Canada, 2010.
- [2] C. Jiang, S. Chen, Y. Chen, Y. Bo, C. Wang, and W. Tao, "Performance analysis of GNSS vector tracking loop based GNSS/CSAC integrated navigation system," *Journal of Aeronautics, Astronautics and Aviation*, vol. 49, no. 4, pp. 289–297, 2017.
- [3] Z. Zhu, Y. Bo, and C. Jiang, "A MEMS gyroscope noise suppressing method using neural architecture search neural network," *Mathematical Problems in Engineering*, vol. 2019, Article ID 5491243, 9 pages, 2019.
- [4] C. Jiang, S. Chen, Y. Chen, and Y. Bo, "Research on a chip scale atomic clock aided vector tracking loop," *IET Radar, Sonar & Navigation*, vol. 13, no. 7, pp. 1101–1106, 2019.
- [5] C. Jiang, S. Chen, Y. Bo, Z. Sun, and Q. Lu, "Implementation and performance evaluation of a fast relocation method in a GPS/SINS/CSAC integrated navigation system hardware prototype," *IEICE Electronics Express*, vol. 14, no. 6, pp. 20170121–20170121, 2017.
- [6] P. Groves, *Principles of GNSS, Inertial, and Multisensor Integrated Navigation Systems*, Artech house, 2013.
- [7] J. J. Spilker, *Vector delay lock loop processing of radiolocation transmitter signals*, no. article 5398034, 1994US Patent, 1994.
- [8] M. Lashley, D. M. Bevely, and J. Y. Hung, "A valid comparison of vector and scalar tracking loops," in *IEEE/ION Position, Location and Navigation Symposium*, pp. 464–474, Indian Wells, CA, USA, 2010.
- [9] M. Lashley and D. M. Bevely, "Vector delay/frequency lock loop implementation and analysis," in *Proceedings of the 2009 International Technical Meeting of the Institute of Navigation (ITM 2009)*, vol. 26–28, pp. 1073–1086, Anaheim, CA, USA, January 2009.
- [10] M. Lashley, D. M. Bevely, and J. Y. Hung, "Performance analysis of vector tracking algorithms for weak GPS signals in high dynamics," *IEEE Journal of Selected Topics in Signal Processing*, vol. 3, no. 4, pp. 661–673, 2009.
- [11] M. Lashley and D. M. Bevely, "Analysis of discriminator based vector tracking algorithms," in *Proceedings of the Institute of Navigation (NTM)*, San Diego, CA, USA, January, 2007.
- [12] T. Pany and B. Eissfeller, "Use of a vector delay lock loop receiver for GNSS signal power analysis in bad signal conditions," in *2006 IEEE/ION Position, Location, And Navigation Symposium*, pp. 893–903, San Diego, CA, USA, April 2006.
- [13] J. H. Won and B. Eissfeller, "Effectiveness analysis of vector-tracking-loop in signal fading environment," in *2010 5th ESA Workshop on Satellite Navigation Technologies and European Workshop on GNSS Signals and Signal Processing (NAVITEC)*, Noordwijk, Netherlands, December 2010.
- [14] K. Ji, H. Zhou, and Y. Fan, "GPS vector tracking loop enhancement using a robust cubature Kalman filter," *Journal of Aeronautics, Astronautics and Aviation*, vol. 51, no. 4, pp. 345–353, 2019.
- [15] K.-H. Kim, G.-I. Jee, and S.-H. Im, "Adaptive vector-tracking loop for low-quality GPS signals," *International Journal of Control, Automation and Systems*, vol. 9, no. 4, pp. 709–715, 2011.
- [16] E. Wycoff and G. Grace, "A Python software platform for cooperatively tracking multiple GPS receivers," in *Proceedings of the 27th International Technical Meeting of the Satellite Division of the Institute of Navigation (ION GNSS+2014)*, pp. 1417–1425, Tampa, Florida, September 2014.
- [17] C. Jiang, S. Chen, Y. Chen, and Y. Bo, "Research on a chip scale atomic clock driven GNSS/SINS deeply coupled navigation

- system for augmented performance,” *IET Radar, Sonar & Navigation*, vol. 13, no. 2, pp. 326–331, 2019.
- [18] D. J. Jwo, Z. M. Wen, and Y. C. Lee, “Vector tracking loop assisted by the neural network for GPS signal blockage,” *Applied Mathematical Modelling*, vol. 39, no. 19, pp. 5949–5968, 2015.
- [19] J. Zhao, X. Zhao, D. Li, D. Deng, and C. Han, “GPS/BDS VTL-assisted by the NN for complex environments,” *IET Communications*, vol. 12, no. 4, pp. 473–479, 2018.
- [20] C. Jiang, S. Chen, Y. Chen et al., “A MEMS IMU de-noising method using long short term memory recurrent neural networks (LSTM-RNN),” *Sensors*, vol. 18, no. 10, 2018.
- [21] F. A. Gers and E. Schmidhuber, “LSTM recurrent networks learn simple context-free and context-sensitive languages,” *IEEE Transactions on Neural Networks*, vol. 12, no. 6, pp. 1333–1340, 2001.
- [22] F. Ordóñez and D. Roggen, “Deep convolutional and lstm recurrent neural networks for multimodal wearable activity recognition,” *Sensors*, vol. 16, no. 1, 2016.