



# IMU Data and GPS Position Information Direct Fusion Based on LSTM

Xingxing Guang <sup>1,\*</sup>, Yanbin Gao <sup>1</sup>, Pan Liu <sup>2</sup> and Guangchun Li <sup>1</sup>

<sup>1</sup> College of Intelligent System Science and Engineering, Harbin Engineering University, Harbin 150001, China; gaoyanbin@hrbeu.edu.cn (Y.G.); lgc\_67@hrbeu.edu.cn (G.L.)

<sup>2</sup> Beijing Institute of Control and Electronic Technology, Beijing 100032, China; liupan003@hrbeu.edu.cn

\* Correspondence: guangxingxing1008@163.com

**Abstract:** In recent years, the application of deep learning to the inertial navigation field has brought new vitality to inertial navigation technology. In this study, we propose a method using long short-term memory (LSTM) to estimate position information based on inertial measurement unit (IMU) data and Global Positioning System (GPS) position information. Simulations and experiments show the practicability of the proposed method in both static and dynamic cases. In static cases, vehicle stop data are simulated or recorded. In dynamic cases, uniform rectilinear motion data are simulated or recorded. The value range of LSTM hyperparameters is explored through both static and dynamic simulations. The simulations and experiments results are compared with the strapdown inertial navigation system (SINS)/GPS integrated navigation system based on kalman filter (KF). In a simulation, the LSTM method's computed position error Standard Deviation (STD) was 52.38% of what the SINS computed. The biggest simulation radial error estimated by the LSTM method was 0.57 m. In experiments, the LSTM method computed a position error STD of 23.08% using only SINSs. The biggest experimental radial error the LSTM method estimated was 1.31 m. The position estimated by the LSTM fusion method has no cumulative divergence error compared to SINS (computed). All in all, the trained LSTM is a dependable fusion method for combining IMU data and GPS position information to estimate position.

**Keywords:** long short-term memory; IMU data; hyperparameter evaluation; positioning



**Citation:** Guang, X.; Gao, Y.; Liu, P.; Li, G. IMU Data and GPS Position Information Direct Fusion Based on LSTM. *Sensors* **2021**, *21*, 2500. <https://doi.org/10.3390/s21072500>

Academic Editors: Antonio Moschitta and Javier Bajo

Received: 12 January 2021

Accepted: 31 March 2021

Published: 3 April 2021

**Publisher's Note:** MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



**Copyright:** © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

## 1. Introduction

INS is one of the important means for realizing vehicle positioning. It uses dead reckoning technology, based on IMU data, to compute the vehicle position information. SINS functions are a group of nonlinear equations. The position error, calculated based on inertial sensor data, includes two aspects, sensor measurement error and calculation error. When the IMU is confirmed, the position calculation method will directly affect the position estimation error. The nonlinear SINS functions should not cause divergent position error [1–3].

Today, deep learning is widely used in various fields, such as image processing, semantic understanding, speech processing, and data processing [4–10]. In recent years, deep learning has also been popular in the inertial navigation field [11]. Deep learning methods, like LSTM network and recurrent neural networks (RNN), have many advantages over forward networks for nonlinear system modeling [12,13]. The nonlinear autoregressive neural network is combined with an Unscented Kalman Filter (UKF) with external inputs to improve the position and velocity precision of the INS/Global Navigation Satellite System (GNSS) during GNSS outages [14]. RNN is used to improve the UKF for estimating and compensating the random drift of inertial sensors in real time [15]. To overcome the problem caused by overrange inertial measurement data, the faster region-based convolutional neural network (RCNN) method is proposed to realize robust autonomous navigation [16]. To improve the navigation ability of inertial/GPS integrated navigation system, LSTM is

used to improve the system's error prediction ability [17]. LSTM is used to estimate the com-cop inclination angle during walking based on inertial sensors [18]. LSTM-RNN is also proposed to denoise the MEMS IMU output signals [19]. A number of studies have shown that the application of deep learning to the inertial navigation field brings about new vitality to inertial technology [20–22].

In this study, a method is proposed that uses LSTM to estimate the position information based on the IMU data and GPS position information. Aiming at the structure of LSTM, the value range of LSTM hyperparameters is explored through the simulation. The results of LSTM estimation are compared with those computed with only SINS and those computed with a SINS/GPS loosely coupled KF during static positioning and dynamic positioning.

## 2. LSTM Net

LSTM takes sequences of information and uses recurrent mechanisms and gate techniques [23]. The structure, training/testing, and hyperparameters of LSTM are described in this section.

### 2.1. LSTM Net Structure

As shown in Figure 1, the rounded rectangle in the LSTM cell is the neural network layer of the LSTM [18,24]. The combination of a rounded rectangle and a multiplication operation is a gate. Gates are used to control the state of the LSTM cell. The repeating module in an LSTM cell contains four interacting layers: a forgetting gate layer, an input gate layer, a new candidate layer, and an output gate layer. In the LSTM net structure, the gate is realized by a sigmoid function:

$$\sigma(x) = 1/1 + e^{-x}. \quad (1)$$

- The forgetting gate layer is made by a sigmoid layer [25]:

$$f(t) = \sigma(W_f[h(t-1), x(t)] + b_f). \quad (2)$$

- The input gate layer decides what new information is going to be stored in the cell state. It has two steps. First, a sigmoid layer decides which values are to be updated. Next, a hyperbolic tangent function (tanh) layer creates a vector of new candidate values,  $\tilde{C}(t)$ , that could be added to the state:

$$i(t) = \sigma(W_i[h(t-1), x(t)] + b_i) \quad (3)$$

$$\tilde{C}(t) = \tanh(W_C[h(t-1), x(t)] + b_C). \quad (4)$$

- The new candidate layer is used to update the old cell state,  $C(t-1)$ , into the new cell state  $C(t)$ , as in the below equation:

$$C(t) = f(t) \times C(t-1) + i_i \times \tilde{C}(t). \quad (5)$$

- The output gate layer: the output is decided in this layer. First, it is going to be forgotten that the net decided to forget earlier. Then, we scale the new candidate values. Finally, we decide the output. The output is based on the cell states. The output computation is as in the below equations:

$$O(t) = \sigma(W_O[h(t-1), x(t)] + b_O) \quad (6)$$

$$h(t) = O(t) \times \tanh(C(t)). \quad (7)$$

As to the LSTM structure, the LSTM can use  $i(t)$  to decide when to keep or override information in memory cell  $C(t)$ , and  $O(t)$  to decide when to access memory cell  $C(t)$

and when to prevent other units from being perturbed by  $C(t)$ . The  $O(t)$  is the output information and the  $h(t)$  is the output about the LSTM structure.

Due to the forgetting gate in the structure, the LSTM net can realize the conditional prediction based on time. In another words, during the learning of a large amount of IMU data and GPS position data, the LSTM learns the position information characteristics related to time. The SINS computes the current inertial position information based on the last step. The SINS/GPS integrated navigation system using KF to estimate the position information is also based on the last step estimation at the current moment. Those processed can be regarded as a conditional prediction based on time. Therefore, we can use LSTM to estimate the position information based on IMU data and GPS position information. This means that the position information is defined as the features of IMU data. To learn the position features from IMU data, we also input the GPS position data to the LSTM net as the target.

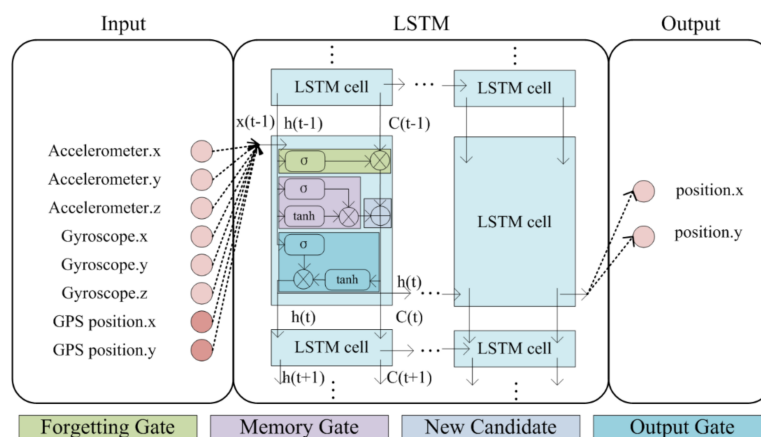


Figure 1. The LSTM net structure of inertial position estimation.

### 2.2. LSTM Training and Testing Settings

As shown in Figure 2, in this study, the training rate is 0.85. This means that 85% of the input dataset is the training set and the other 15% is the test set. Input and output size are based on the practical application of the LSTM.

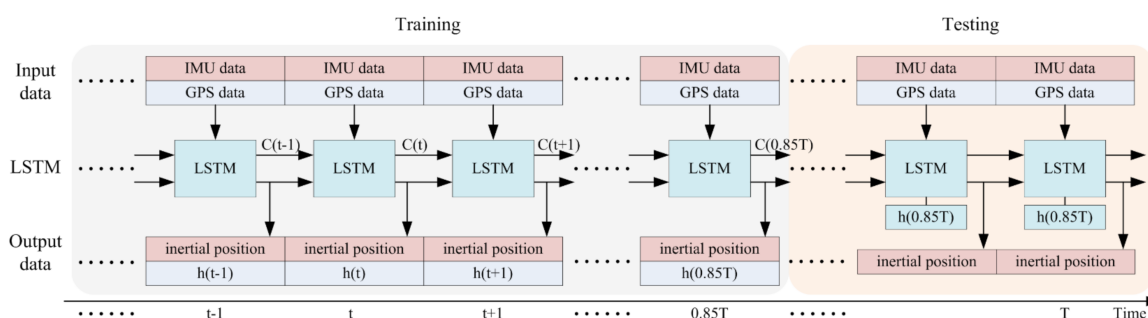


Figure 2. Training and testing the LSTM to estimate position information.

As shown in Figures 1 and 2, the input set includes two parts: IMU data, including three dimensions of gyroscope measurement data and three dimensions of accelerometer measurement data; and GPS data, including two dimensions of position information. The IMU data are the input of the LSTM, and the GPS data are the training target data. The output dataset includes two parts: the predicted inertial position, in two dimensions, and the output of the LSTM, with the number of dimensions based on the LSTM structure.

In LSTM, the purpose of net training is to optimize the weights and biases. The state activation function is the tanh, while the optimizer function is the adaptive moment

estimation (Adam) stochastic gradient descent method with high convergence performance [18,26,27].

We use the root mean stand error (RMSE) to describe the training cost:

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n (y_{pi} - y_i)^2}, \quad (8)$$

where,  $y_{pi}$  is the target GPS position data and  $y_i$  is the inertial position predicted by LSTM.

### 2.3. Hyperparameters of LSTM

Hyperparameters need to be adjusted during LSTM training to make sure of the training cost in a confidence interval, including the time step, cell size, batch size, learning rate, and forgetting rate.

- Time step: in LSTM, the purpose of net training is to optimize the weights and biases, to make sure the output errors are within a certain range. It is worth noting that the calculation of weights and biases is independent of the time step. In other words, the time step of LSTM can be set to any number. In this study, we expect that when the input IMU and target GPS data are at the same frequency, each frame of IMU data can be calculated to obtain the position information. Therefore, the time step in this study is 1.
- Cell size: the hidden layer unit size, also called the hidden size. Cell size defines the deep degree of LSTM.
- Batch size: also called mini-batch size. In the mini-batch gradient descent method, the data are divided into several batches and the parameters are updated according to the batches. In this way, the data in a batch jointly determine the direction of the gradient, so that the descent is not easy to determine, and the randomness is reduced. On the other hand, because the sample size of the batch is much smaller than the whole dataset, the computation burden will be reduced.
- The learning rate and forgetting rate control the speed of adjusting weights and biases of neural network based on the loss gradient.

## 3. Simulation

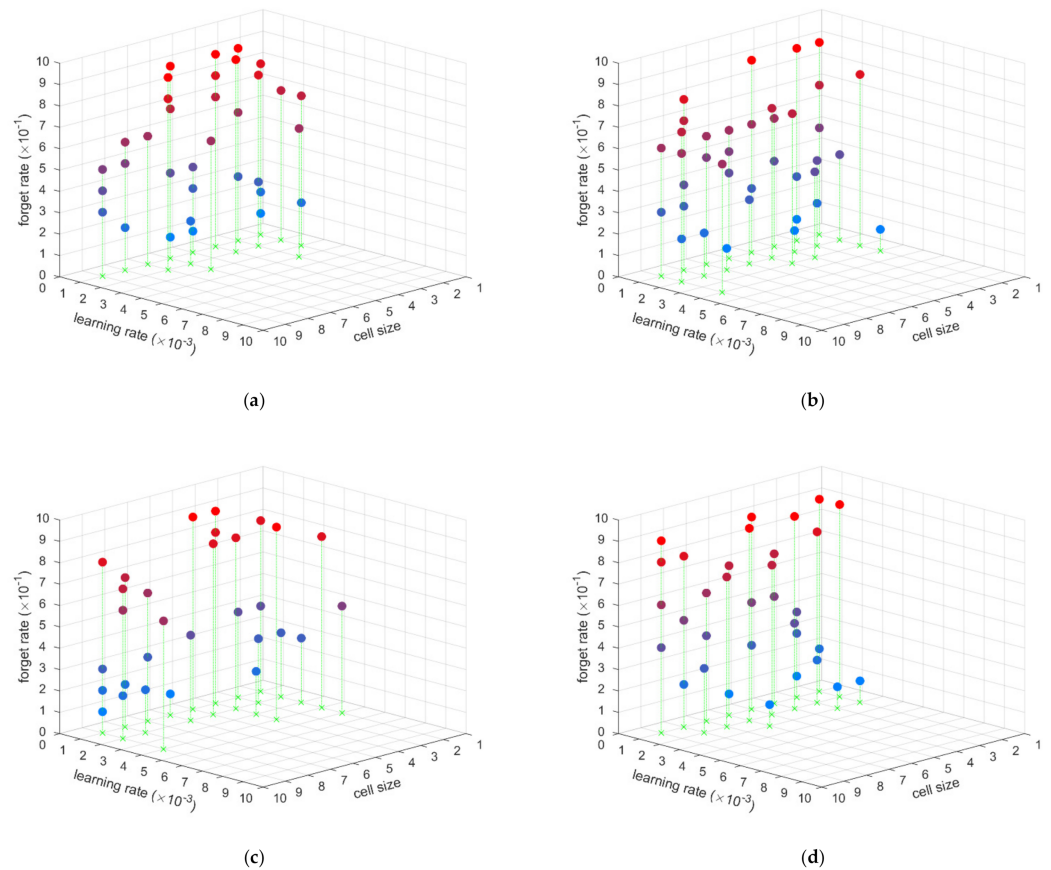
To select appropriate hyperparameters such as the batch size, cell size, learning rate, and forgetting rate, we performed many simulations with different hyperparameters in this section.

### 3.1. Hyperparameter Evaluation

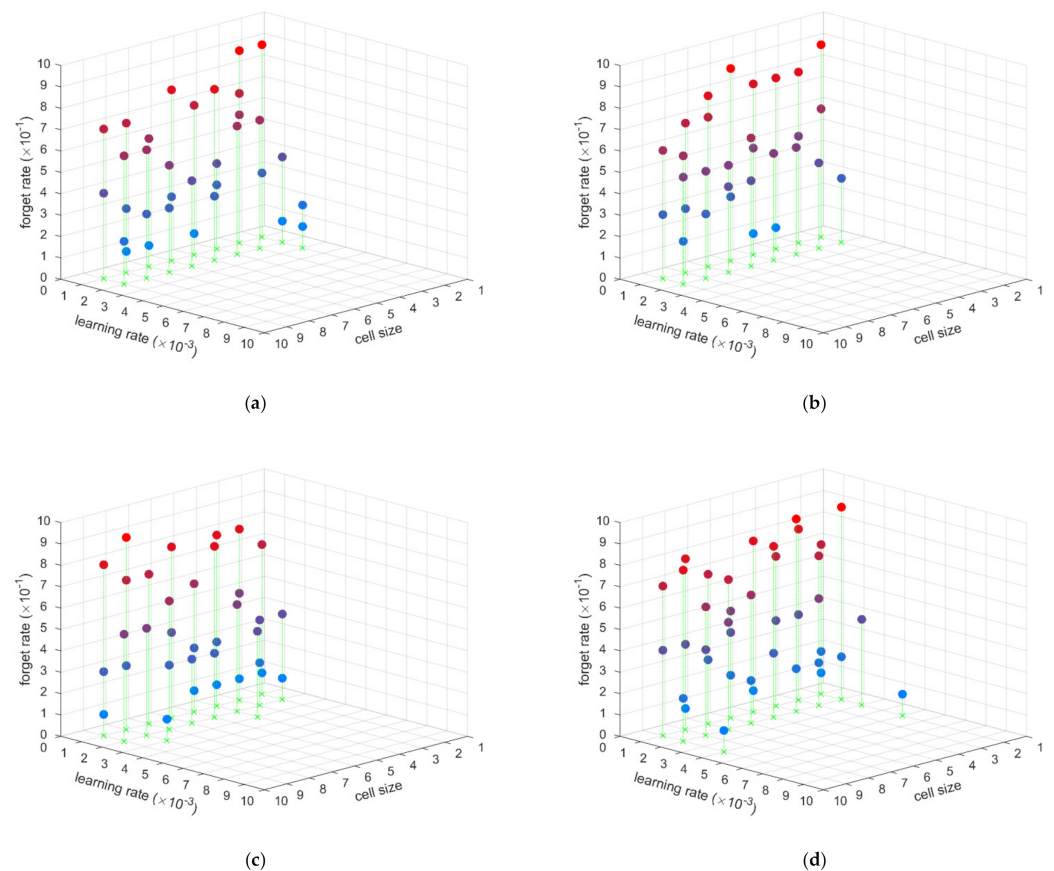
There are two simulation cases, static position estimation and dynamic position estimation. The IMU data used in the simulation are generated by the inertial simulator. The latitude and longitude of the initial location are set to 35° N and 108° E, respectively. The gyro constant drifts and random drifts are set to 0.02°/h and 0.001°/h, respectively. The accelerometer's constant bias and random bias are  $1 \times 10^{-4} g$  and  $1 \times 10^{-5} g$ , respectively, where  $g$  is the local gravitation. In the static case, the simulation time is 60 s. In the dynamic case, the simulation time is 780 s, and the velocity is  $[1, 1, 0]^T$  m/s. For each hyperparameter set, there are 100 training and testing events based on the simulated IMU data. We performed statistical tests on the hyperparameters used to train the LSTM to estimate the location based on the IMU data and GPS position information. The hyperparameters in the evaluation distribution point are the set that never cause overfitting or underfitting when training the LSTM to estimate the inertial position. They are recoded as shown in the following figures.

As each subgraph shows in Figures 3 and 4, the hyperparameter evaluation distribution is similar when the batch size is different, and the hyperparameter evaluation distribution is also similar with different cell sizes and forgetting rates. However, different learning rates obviously affected the hyperparameter evaluation. Thus, the forgetting rate,

cell size, and batch size have little influence on the LSTM used to estimate the position based IMU data. The important hyperparameter is the learning rate. From those simulations, when the learning rate is between 0.001 and 0.002, the LSTM is able to estimate the position based IMU data well. This hyperparameter evaluation applies to both static and dynamic situations. In another words, the static position estimation LSTM net and dynamic position estimation LSTM net have the same hyperparameter range. This means that an LSTM net with the same hyperparameters can be trained to estimate position based IMU data, no matter whether the conditions are static or dynamic.



**Figure 3.** The hyperparameter evaluation for static position estimation: (a) batch size is 5, (b) batch size is 10, (c) batch size is 15, (d) batch size is 20.

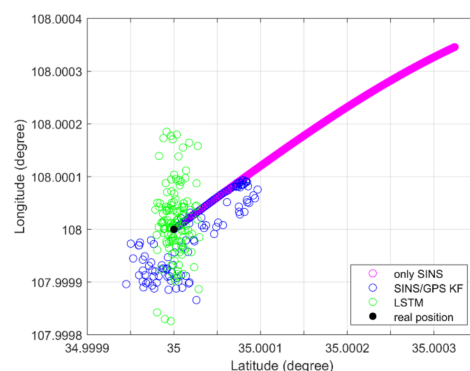


**Figure 4.** The hyperparameter evaluation for dynamic position estimation: (a) batch size is 5, (b) batch size is 10, (c) batch size is 15, (d) batch size is 20.

### 3.2. Position Estimation

In this section, the position information estimated by LSTM and the position information obtained by the inertial system functions are compared in static conditions and moving conditions. The uniform rectilinear motion data are simulated in the dynamic case.

As shown in Figure 5, an LSTM net can fuse the IMU and GPS data to estimate the position information, as with the SINS/GPS loosely coupled based KF, in static simulation. Both an LSTM net and a loosely coupled KF-based system can restrain the divergence of only SINS. The position error STDs of only SINS, KF, and LSTM are 1.47 m, 0.88 m, and 0.77 m, respectively. The KF method computed position error STD is 59.86% for only SINSs, and the LSTM method estimated position error STD is 52.38% for only SINSs. The LSTM estimation position is more convergent than the KF method estimated.



**Figure 5.** Static position information of simulation.

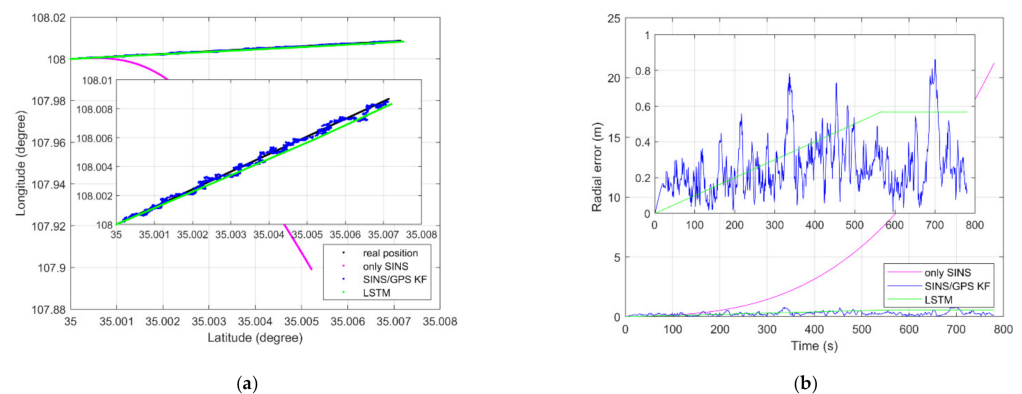
The position error in the dynamic case is described by radial error. The radial error is defined by

$$e_{radial} = \sqrt{((L_e - L_r)R_N)^2 + ((\lambda_e - \lambda_r)R_M \cos L_r)^2}, \quad (9)$$

where  $L_e$  and  $\lambda_e$  are the estimation latitude and estimation longitude, respectively, and  $L_r$  and  $\lambda_r$  are the reference latitude and reference longitude, respectively.  $R_N = R_e / \sqrt{1 - e^2 \sin^2 L}$ ,  $R_M = R_e(1 - e^2) / (1 - e^2 \sin^2 L)^{3/2}$ ,  $R_e$  is the ellipse major axis, and  $e$  is the ellipse eccentricity.

As shown in Figure 6a, LSTM net can fuse the IMU and GPS data to estimate the position information, like the SINS/GPS loosely coupled KF-based system, in a dynamic simulation. Both the LSTM net and the loosely coupled KF-based system can restrain the divergence of only SINS. As shown in Figure 6b, the biggest radial error of only SINS, KF method, and LSTM method are 21.21 m, 0.86 m, and 0.57 m, respectively. As Figure 6a,b shows, the position information obtained by only SINS has a nonlinear divergence error. The radial error of only SINS will continue to grow over time. However, the position error of KF and LSTM does not increase over time. The radial error of KF oscillates around 0. The radial error of LSTM is piecewise linear. Before 580 s, the position information estimated by LSTM has a linear divergence error. After 580 s, it becomes a constant error. LSTM can enforce constant error through “constant error carousels” within the forgetting gate during enough training steps [23]. The radial error of the LSTM estimation position accords with this conclusion.

As the simulation results show, the LSTM net can fuse the IMU and GPS data to estimate the position information and restrain the divergence error of only SINS, as with the SINS/GPS loosely coupled KF-based system.



**Figure 6.** (a) Dynamic position estimation of simulation; (b) dynamic position radial error of simulation.

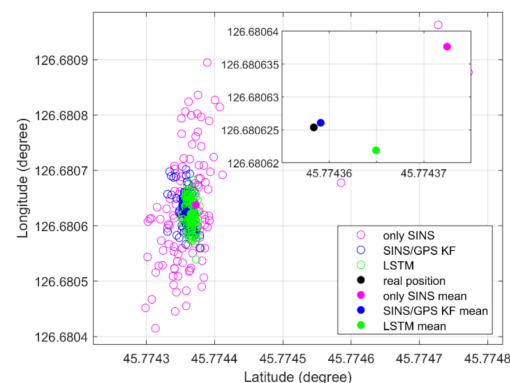
#### 4. Experiments

An experimental system was assembled to evaluate the proposed approach, as shown in Figure 7. The experimental equipment included an IMU, a GPS, and a power system on the vehicle. The IMU constituted a three-axis fiber-optic gyroscope with three accelerometers on each gyro-axis. The GPS was UR370 form UNICORE. The IMU, GPS receiver, and power system are in the vehicle trunk. The IMU is fixed on the vehicle via a steel plate that is parallel to the under panel of the vehicle. During the experiment, the IMU and GPS data were recorded. Both IMU data and GPS data included the GPS time. In this study, the GPS provided the position information target. The GPS data and IMU data were synchronized by their GPS times. There were two cases in the experiment, one static and the other dynamic. In static conditions, the vehicle stops 2 min. In dynamic conditions, the vehicle was driving in a uniform rectilinear motion along the road. The driving distance was 450 m and the driving speed was 2 km/s.



**Figure 7.** The experimental equipment.

As shown in Figure 8, an LSTM net can fuse the IMU and GPS data to estimate the position information, like the SINS/GPS loosely coupled KF-based system, in a static experiment. Both the LSTM net and loosely coupled KF-based system can restrain the divergence of only SINS. The position error STDs of only SINS, KF method, and LSTM method are 1.30 m, 0.33 m, and 0.30 m, respectively. The KF method computed position error STD is 25.38% for only SINSs, and the LSTM method computed position error STD is 23.08% for only SINSs. The LSTM estimation position is more convergent than the KF method estimated.

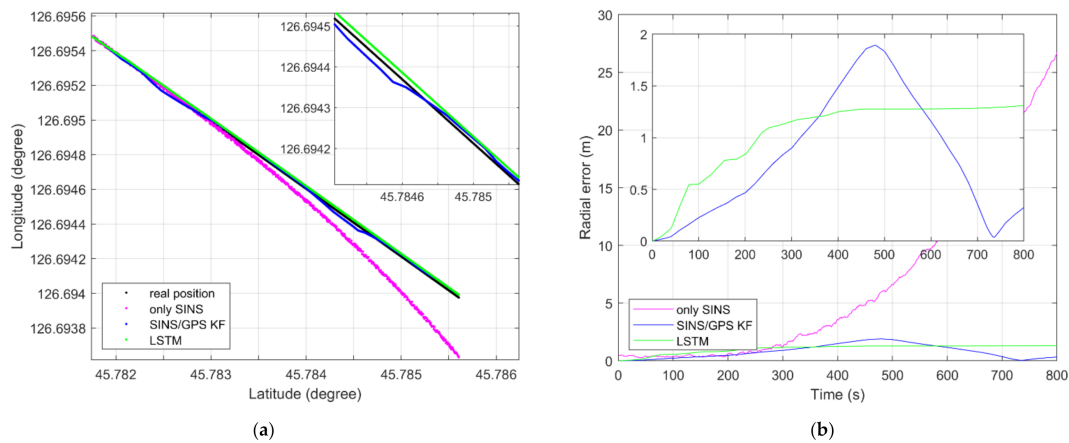


**Figure 8.** Static position of experiment.

As shown in Figure 9a, an LSTM net can fuse the IMU data and GPS position information to estimate the position information, like the SINS/GPS loosely coupled KF-based system, in a dynamic experiment. Both the LSTM net and loosely coupled based KF can restrain the divergence of only SINS. As shown in Figure 9b, the largest radial error of only SINS, KF method, and LSTM method are 26.71 m, 1.89 m, and 1.31 m, respectively. As Figure 6a,b shows, the position information obtained by only SINS has nonlinear divergence error. The radial error of only SINS will continue to grow over time. However, the position error of KF method and LSTM method does not increase over time. The radial error of the KF method oscillates around 0. The radial error of LSTM method is piecewise linear. Before 450 s, the position information estimated by LSTM has linear divergence error. After 450 s, it becomes constant error. LSTM can enforce constant error through “constant error carousels” within the forgetting gate during enough training steps [23]. The radial error of the LSTM estimation position accords with this conclusion.

As the experimental results show, the LSTM net can fuse the IMU and GPS data to estimate the position information and restrain the divergence error of only SINS, like the SINS/GPS loosely coupled KF-based system.





**Figure 9.** (a) Dynamic position estimation of experiment; (b) dynamic position radial error of experiment.

## 5. Discussion

As the results of the simulations and experiments have shown, the effectiveness of the proposed method, using an LSTM net to fuse the IMU data and GPS to estimate the position information, has been proven in both static and dynamic conditions. Using the LSTM net to fuse IMU data and GPS position information can restrain the divergence of only SINS, like the SINS/GPS loosely coupled KF-based navigation system can.

The position error of only SINS includes sensor measurement error and calculation error. In this study, the simulation and experiment are based on the same IMU data and the same GPS position information. Thus, different position errors were found based on the calculation methods.

The model of position information calculation determines the character of the computational error. For the only SINS, as Equations (A3) and (A6) in Appendix A show, the position information and position error of only SINS are related to the inertial velocity computation and the inertial velocity error. The velocity information is integrated by geometric acceleration, and we get the position information by integrating the velocity information. The SINS functions and SINS error functions are nonlinear. The nonlinear error will accelerate the divergence of the position error through integration.

For the SINS/GPS loosely coupled KF-based navigation system, the system fusion the GPS position information and position, velocity and attitude information computed by only SINS, as Equations (A7) to (A9) in Appendix A show. The position information computed by only SINS diverges over time. The GPS position computing is without integral, so that GPS position information has almost no drift. Thus, based on KF, GPS position information can restrain the divergence of only SINS.

However, as Figure 2 shows, the IMU data are directly fused with GPS position information, without an inertial computer. Thus, there is no accumulation of error due to integration in the proposed method. As Equation (6) shows, the relationship between LSTM input information and LSTM output information is linear. The linear parameters are  $W_O$  and  $b_C$ . For a trained LSTM net, the structural parameters are constant. The error of LSTM estimation position is dependent on the parameters. The LSTM structure determines that the error of LSTM estimation position is linear. It is well known that suppression of linear error is easier than suppression of nonlinear error. Therefore, the LSTM position estimation method, which only produces linear error, makes the application of inertia information more convenient.

Both the KF method and the LSTM method can fuse GPS and an inertial sensor to restrain the error divergence of only SINS. There are some differences between the KF method and the LSTM method. For the KF information fusion method, it is a “white box.” It is applicable to the same navigation systems with the same system equation and same measurement equation. The system function and fusion function are definite.

The system noise and measurement noise must be Gaussian white noise. With the same initial value, the position computed by the KF method is the same in every time. For the LSTM information fusion method, it is a “black box.” It is not required to give the specific equations of the system. The trained LSTM method model is only applicable to the system that provides the training and testing data. The LSTM output is based on the gate threshold value and the training cost function. There are no specific requirements for system noise or measurement noise. With the LSTM information fusion method, the position estimation output is different even with the same hyperparameters in every time. The outputs are distributed in an accuracy range.

## 6. Conclusions

In this study, we proposed a method using an LSTM net to estimate position information based on IMU data and GPS position information, because of its “memory” layer. Simulation and experimental results have shown that the proposed method is effective. The range of LSTM hyperparameters is explored through simulations. The LSTM information fusion method can restrain the position divergence of only SINS. In simulations, the LSTM method computed position error STD is 52.38% of only SINSs. The biggest simulation radial error estimated by the LSTM method is 0.57 m. In the experiments, the LSTM method computed position error STD at 23.08% for only SINSs. The biggest experimental radial error the LSTM method estimated was 1.31 m.

The trained LSTM fusion model is not applicable to all IMUs. It is only applicable to the IMU which provides the training data. The LSTM fusion method is used to estimate position information based on IMU data and GPS position, which only produces linear error, which makes the subsequent application of navigation information more convenient.

All in all, the trained LSTM is a dependable fusion method for combining IMU data and GPS position information to estimate position.

**Author Contributions:** Conceptualization, methodology, investigation, software, data curation, and visualization, X.G.; validation and formal analysis, X.G. and P.L.; resources, Y.G. and G.L.; writing—original draft preparation, X.G.; writing—review and editing, X.G. and P.L.; supervision, project administration, and funding acquisition, Y.G. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research was funded by the National Natural Science Foundation of China, grant number 61803118; a Special Project of Chongqing Technology Innovation and Application Development, grant number Cstc2019jscx-Msxmx0423; and the Science and Technology Research Program of Chongqing Municipal Education Commission, grant number KJZD-K201804701.

**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Not applicable.

**Data Availability Statement:** The data presented in this study are available on request from the corresponding author. The data are not publicly available due to ethics.

**Acknowledgments:** The authors would like to thank Shutong Li, Fan Zhang, and Yunlong Sun for helping with the experiments.

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

## Appendix A

IMU data are computed into positioning/navigation information by a SINS algorithm [1]. The pose, velocity, and position information updating functions are shown below:

$$\dot{C}_b^n = C_b^n \left( \omega_{ib}^b \times \right) - \left( \omega_{in}^n \times \right) C_b^n, \quad (\text{A1})$$

where  $\omega_{ib}^b$  is the IMU measurement angular velocity and  $\omega_{in}^n$  is the rotation of navigation coordinate relative to the inertial coordinates.

$$\dot{v}^n = C_b^n f_{sb}^b - (2\omega_{ie}^n + \omega_{en}^n) \times v^n + g^n, \quad (A2)$$

where  $\dot{v}^n$  is the geometric acceleration of the vehicle in the navigation coordinate system.  $f_{sb}^b$  is the IMU measurement accelerated velocity, while  $g^n$  is the equatorial gravitation.  $-(2\omega_{ie}^n + \omega_{en}^n) \times v^n$  is the harmful acceleration. Equation (A2) is called the inertial navigation specific force equation. We can get the velocity information by integrating  $\dot{v}^n$  once, and the position information by integrating  $\dot{v}^n$  twice. This is the base equation of the SINS algorithm:

$$\dot{p} = M_{pv}v^n, \quad (A3)$$

$$\text{where } M_{pv} = \begin{bmatrix} 0 & 1/R_{Mh} & 0 \\ \sec L/R_{Nh} & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}.$$

In the navigation coordinates, the position error equations in dynamic conditions are as follows:

$$\dot{\phi} = M_{aa}\phi + M_{av}\delta v^n + M_{ap}\delta p - C_b^n \varepsilon^b, \quad (A4)$$

where

$$M_{aa} = -(\omega_{in}^n \times) = \begin{bmatrix} 0 & \omega_U + v_E \tan L/R_{Nh} & -\omega_N - v_E/R_{Nh} \\ -\omega_U - v_E \tan L/R_{Nh} & 0 & -v_E/R_{Nh} \\ \omega_N + v_E/R_{Nh} & v_N/R_{Mh} & 0 \end{bmatrix},$$

$$M_{av} = \begin{bmatrix} 0 & -1/R_{Mh} & 0 \\ 1/R_{Nh} & 0 & 0 \\ \tan L/R_{Nh} & 0 & 0 \end{bmatrix},$$

$$M_{ap} = \begin{bmatrix} 0 & 0 & v_N/R_{Mh}^2 \\ -\omega_{ie} \sin L & 0 & -v_E/R_{Nh}^2 \\ \omega_{ie} \cos L + v_E \sec^2 L/R_{Nh} & 0 & -v_E \tan L/R_{Nh}^2 \end{bmatrix},$$

where  $L$ ,  $\lambda$ , and  $h$  denote the local latitude, longitude, and height, respectively;  $v$  defines the body velocity vector;  $\phi$  describes the body pose errors;  $\varepsilon$  is the constant gyroscope drifts; the subscripts  $E$ ,  $N$  and  $U$  denote the projection on the east, north, and up axes of the local coordinates, respectively.  $R_{Nh} = R_N + h$ ,  $R_{Mh} = R_M + h$ ,  $R_N = R_e/\sqrt{1 - e^2 \sin^2 L}$ , and  $R_M = R_e(1 - e^2)/(1 - e^2 \sin^2 L)^{3/2}$ , where  $R_e$  is the ellipse major axis and  $e$  is the ellipse eccentricity.

The velocity error equations in the dynamic condition are as follows:

$$\delta \dot{v}^n = M_{va}\phi + M_{vv}\delta v^n + M_{vp}\delta p + C_b^n \nabla^b, \quad (A5)$$

$$\text{where } M_{vp} = \begin{bmatrix} 2(v_N\omega_N + v_U\omega_U) + v_E v_N \sec^2 L/R_{Nh} & 0 & v_E(v_U - v_N \tan L)/R_{Nh}^2 \\ -v_E(2\omega_N + v_E \sec^2 L/R_{Nh}) & 0 & v_E v_N/R_{Mh}^2 + v_E^2 \tan L/R_{Nh}^2 \\ 2\omega_U v_E + g_e \sin 2L(\beta - 4\beta_1 \cos 2L) & 0 & -v_E^2/R_{Mh}^2 - v_N^2/R_{Nh}^2 + \beta_2 \end{bmatrix},$$

$$M_{vv} = \begin{bmatrix} (v_N \tan L - v_U)/R_{Nh} & 2\omega_U + v_E \tan L/R_{Nh} & -2\omega_N - v_E/R_{Nh} \\ -2\omega_U - 2v_E \tan L/R_{Nh} & -v_U/R_{Mh} & -v_N/R_{Mh} \\ 2\omega_N + 2v_E/R_{Nh} & 2v_N/R_{Mh} & 0 \end{bmatrix},$$

$$M_{va} = \begin{bmatrix} 0 & -f_U & f_N \\ f_U & 0 & -f_E \\ -f_N & f_E & 0 \end{bmatrix},$$

where  $\nabla$  denotes the accelerometer zero biases,  $f$  is the specific force, and  $\beta$  is the gravitation flattening, with  $\beta \approx 1/188.6$ ,  $\beta_1 \approx 1/8(2\beta f + f^2)$ , and  $\beta_2 \approx 3.08 \times 10^{-6} s^{-2}$ .

The position error equations in the dynamic condition are as follows:

$$\delta \dot{p} = M_{pv} \delta v^n + M_{pp} \delta p, \quad (A6)$$

$$\text{where } M_{pp} = \begin{bmatrix} 0 & 1/R_{Mh} & -v_N/R_{Mh}^2 \\ v_E \sec L \tan L/R_{Nh} & 0 & -v_E \sec L/R_{Nh}^2 \\ 0 & 0 & 0 \end{bmatrix}.$$

SINS information and GPS information can be fused by a KF [28]. For the loosely coupled method, the state vector is  $X = [\delta p, \delta v^n, \phi, \nabla^b, \varepsilon^b]^T$  and the system equation is

$$\dot{X} = FX + GW, \quad (A7)$$

where,  $F$  is the system matrix,  $G$  is the system noise matrix, and  $W$  is the system noise.

$$F = \begin{bmatrix} M_{pp} & M_{pv} & 0_{3 \times 3} & 0_{3 \times 3} & 0_{3 \times 3} \\ M_{vp} & M_{vv} & M_{va} & C_b^n & 0_{3 \times 3} \\ M_{ap} & M_{av} & M_{aa} & 0_{3 \times 3} & -C_b^n \\ 0_{3 \times 3} & 0_{3 \times 3} & 0_{3 \times 3} & 0_{3 \times 3} & 0_{3 \times 3} \\ 0_{3 \times 3} & 0_{3 \times 3} & 0_{3 \times 3} & 0_{3 \times 3} & 0_{3 \times 3} \end{bmatrix}, \quad G = \begin{bmatrix} 0_{3 \times 3} & 0_{3 \times 3} \\ 0_{3 \times 3} & 0_{3 \times 3} \\ 0_{3 \times 3} & 0_{3 \times 3} \\ I_{3 \times 3} & 0_{3 \times 3} \\ 0_{3 \times 3} & I_{3 \times 3} \end{bmatrix}, \quad W = \begin{bmatrix} w_g \\ w_a \end{bmatrix}, \quad (A8)$$

The measurement vector is

$$Z = \begin{bmatrix} (L_{sins} - L_{GPS})R \\ (\lambda_{sins} - \lambda_{GPS})R \cos L \end{bmatrix} = \begin{bmatrix} R \delta L + N_n \\ R \cos L \delta \lambda + N_e \end{bmatrix} = HX + V, \quad (A9)$$

where,  $H$  is the system measurement matrix,  $V = [N_n \quad N_e]^T$  is the measurement noise, and  $N_n$  and  $N_e$  are the GPS measurement noise of North position and East position, respectively.

$$H = \begin{bmatrix} R & 0 & 0_{13 \times 2} \\ 0 & R \cos L & 0_{13 \times 2} \end{bmatrix}. \quad (A10)$$

The KF includes a one-step state prediction equation, a state estimation equation, a filtering gain equation, a one-step prediction mean square error equation, and an estimated mean square error equation. They are listed below:

$$\hat{X}_{k+1,k} = XF_{k+1,k} \hat{X}_k \quad (A11)$$

$$\hat{X}_{k+1} = \hat{X}_{k+1,k} + K_k (Z_k - H_k \hat{X}_{k+1,k}) \quad (A12)$$

$$K_{k+1} = P_{k+1,k} H_k^T (H_k P_{k+1,k} H_k^T + R_k)^{-1} \quad (A13)$$

$$P_{k+1,k} = F_{k+1,k} P_k F_{k+1,k}^T + G_{k+1,k} Q_k G_{k+1,k}^T \quad (A14)$$

$$P_{k+1} = (I - K_k H_k) P_k (I - K_k H_k)^T + K_k R_k K_k^T, \quad (A15)$$

where,  $\hat{X}_{k+1,k}$  denotes the prediction of the state vector from  $t_k$  to  $t_{k+1}$ ,  $\hat{X}_k$  denotes the prediction of the state vector at  $t_k$ ,  $K_k$  denotes filtering gain matrix,  $P_{k+1,k}$  denotes one-step estimated mean square error matrix from  $t_k$  to  $t_{k+1}$ ,  $P_k$  denotes estimated mean square error matrix,  $Q_k$  is system noise covariance matrix, and  $R_k$  denotes measurement noise covariance matrix.

## References

1. Qin, Y.Y. Strapdown inertial navigation system. In *Inertial Navigation*, 2nd ed.; Science Press: Beijing, China, 2014; pp. 298–315.
2. Savage, P.G. Analytical Modeling of Sensor Quantization in Strapdown Inertial Navigation Error Equations. *J. Guid. Control Dyn.* **2002**, *25*, 833–842. [\[CrossRef\]](#)
3. Qing, Y.Y.; Zhang, H.Y.; Wang, S.H. The error model of inertial navigation system. In *Principles of Kalman Filtering and Integrated Navigation*, 2nd ed.; Northwestern University of Technology Press: Xi'an, China, 2012; pp. 316–334.

4. Zhang, W.; Wu, Q.M.J.; Yang, Y.; Zhang, H. A Width-growth Model with Subnetwork Nodes and Refinement Structure for Representation Learning and Image Classification. *IEEE Trans. Ind. Inform.* **2020**, *17*, 1562–1572. [[CrossRef](#)]
5. Taghipour-Gorjikoalie, M.; Sadri, J.; Razavi, S.M. Deep Adaptive Feature Enrichment. *Expert Syst. Appl.* **2020**, *162*, 113780. [[CrossRef](#)]
6. Kubany, A.; Ishay, S.B.; Ohayon, R.S.; Shmilovici, A.; Rokach, L.; Doitshman, T. Comparison of state-of-the-art deep learning APIs for image multi-label classification using semantic metrics-ScienceDirect. *Expert Syst. Appl.* **2020**, *161*, 113656. [[CrossRef](#)]
7. Dokuz, Y.; Tufekci, Z. Mini-batch sample selection strategies for deep learning based speech recognition. *Appl. Acoust.* **2020**, *171*, 107573. [[CrossRef](#)]
8. Yao, K.; Peng, B.; Zhang, Y.; Shi, Y. Spoken language understanding using long short-term memory neural networks. In *Spoken Language Technology Workshop*; IEEE: New York, NY, USA, 2015.
9. Choi, J.Y.; Lee, B. Combining LSTM Network Ensemble via Adaptive Weighting for Improved Time Series Forecasting. *Math. Probl. Eng.* **2018**, *2018 Pt 9*, 1–8. [[CrossRef](#)]
10. Ghosh, A.; Grolinger, K. Edge-Cloud Computing for IoT Data Analytics: Embedding Intelligence in the Edge with Deep Learning. *IEEE Trans. Ind. Inform.* **2020**, *99*, 1. [[CrossRef](#)]
11. Fang, W.; Jiang, J.G.; Lu, S.Q.; Gong, Y.L.; Tao, Y.F.; Tang, Y.N.; Yan, P.H.; Luo, H.Y.; Liu, J.N. See the Future: A Semantic Segmentation Network Predicting Ego-Vehicle Trajectory With a Single Monocular Camera. *IEEE Robot. Autom. Lett.* **2020**, *5*, 3066–3073.
12. Yu, W.; Gonzalez, J.; Li, X. Fast training of deep LSTM networks with guaranteed stability for nonlinear system modeling. *Neurocomputing* **2021**, *422*, 85–94. [[CrossRef](#)]
13. Gonzalez, J.; Yu, W. Non-linear system modeling using LSTM neural networks. In Proceedings of the Second IFAC Conference on Modelling, Identification and Control of Nonlinear Systems, Guadalajara, Mexico, 20–22 June 2018; Volume 51, pp. 485–489.
14. Bitar, N.A.; Gavrilov, A. A new method for compensating the errors of integrated navigation systems using artificial neural networks. *Measurement* **2020**, *168*, 108391. [[CrossRef](#)]
15. Li, D.; Zhou, J.; Liu, Y. Recurrent-neural-network-based unscented Kalman filter for estimating and compensating the random drift of MEMS gyroscopes in real time-ScienceDirect. *Mech. Syst. Signal Process.* **2021**, *147*, 107057. [[CrossRef](#)]
16. Yang, S.; Xing, L.; Liu, W.; Qian, W.; Zhu, Y. Robust Navigation Method for Wearable Human-Machine Interaction System Based on Deep Learning. *IEEE Sens. J.* **2020**, *20*, 14950–14957. [[CrossRef](#)]
17. Shen, C.; Zhang, Y.; Guo, X.; Liu, J. Seamless GPS/Inertial Navigation System Based on Self-Learning Square-Root Cubature Kalman Filter. *IEEE Trans. Ind. Inform.* **2020**, *68*, 499–508. [[CrossRef](#)]
18. Choi, A.; Jung, H.; Mun, J.H. Single Inertial Sensor-Based Neural Networks to Estimate COM-COP Inclination Angle during Walking. *Sensors* **2019**, *19*, 2974. [[CrossRef](#)]
19. Jiang, C.H.; Chen, S.; Chen, Y.W.; Zhang, B.Y.; Feng, Z.Y.; Zhou, H.; Bo, Y.M. A MEMS IMU De-Noising Method Using Long Short Term Memory Recurrent Neural Networks (LSTM-RNN). *Sensors* **2018**, *18*, 3470. [[CrossRef](#)]
20. Zhang, Y. A Fusion Methodology to Bridge GPS Outages for INS/GPS Integrated Navigation System. *IEEE Access* **2019**, *7*, 61296–61306. [[CrossRef](#)]
21. Du, S.; Zhang, S.; Gan, X. A hybrid fusion strategy for the land vehicle navigation using MEMS INS, odometer and GNSS. *IEEE Access* **2020**, *8*, 152512–152522. [[CrossRef](#)]
22. Fang, W.; Jiang, J.G.; Lu, S.Q.; Gong, Y.L.; Tao, Y.F.; Tang, Y.N.; Yan, P.H.; Luo, H.Y.; Liu, J.N. A LSTM Algorithm Estimating Pseudo Measurements for Aiding SINS during GNSS Signal Outages. *Remote Sens.* **2020**, *12*, 256. [[CrossRef](#)]
23. Hochreiter, S.; Schmidhuber, J. Long Short-Term Memory. *Neural Comput.* **1997**, *9*, 1735–1780. [[CrossRef](#)] [[PubMed](#)]
24. Smagulova, K.; James, A.P. A survey on LSTM memristive neural network architectures and applications. *Eur. Phys. J. Spec. Top.* **2019**, *228*, 2313–2324. [[CrossRef](#)]
25. Understanding LSTM Networks. Available online: <http://colah.github.io/posts/2015-08-Understanding-LSTMs/> (accessed on 1 November 2020).
26. Choi, A.; Yun, T.S.; Suh, S.W.; Yang, J.H.; Park, H.; Lee, S.; Roh, M.S.; Kang, T.G.; Mun, J.H. Determination of input variables for the development of a gait asymmetry expert system in patients with idiopathic scoliosis. *Int. J. Precis. Eng. Manuf.* **2013**, *14*, 811–818. [[CrossRef](#)]
27. Kingma, D.P.; Ba, J. Adam: A Method for Stochastic Optimization. In Computer Science. In Proceedings of the 3rd International Conference on Learning Representations (ICLR 2015), San Diego, CA, USA, 7–9 May 2015.
28. Qing, Y.Y.; Zhang, H.Y.; Wang, S.H. The integrated navigation system is designed based on basic filtering theory. In *Principles of Kalman Filtering and Integrated Navigation*, 2nd ed.; Northwestern University of Technology Press: Xi'an, China, 2012; pp. 362–372.