Prediction of SINS/GPS Navigation Information by ELM Algorithm during GPS outages

Yang Cao, Fangxiu Jia*, Xun Jiang, Qing Zhang
School of Mechanical and Engineering Nanjing University of Science and Technology Nanjing, China
*jiafangxiu@126.com

Abstract. SINS/GPS integrated navigation has received great attention recently due to its impressive navigation accuracy on missile-borne platforms. However, it remains an ongoing challenge to address the problem of GPS signal loss. Furthermore, the limited space and power consumption of a projectile have made it challenging to carry out time-consuming iterative calculations. In this paper, ELM is proposed for the first time to predict the navigation information on the missile-borne platform for its simpler neural network structure and no iterative calculation. Comparing it with the algorithms of BP and GABP, the simulation results reveal that the ELM algorithm is more time-saving with higher prediction accuracy than the others. More importantly, the number of parameters needed to be set is much less than those of the other two algorithms, which is beneficial for its practical application.

1. Introduction
All-weather navigation systems with strong anti-jamming performance can fulfill the needs of project guidance [1] [2]. However, the flight time of conventional guided projectiles is usually several tens of seconds, and the missile-borne computer needs to solve the attitude of the projectile in a limited time and generate control instructions to adjust the attitude of the projectile. In addition, for the missile-borne platform, limited space and power consumption constraint its computing power, therefore it is necessary to find a time-saving navigation algorithm with high precision, less computational intensity, and strong generalization ability.

A Strap-down Inertial Navigation System/Global Position System (SINS/GPS) has been spotlighted as an excellent navigated system. It reasonably combines two navigation systems to improve the overall accuracy and reliability of the navigation system and better compensate for the defects of the navigation performance of a single system [3]. Despite these considerable advantages, the practical application of SINS/GPS is still not perfect. Due to various abnormal conditions such as changes in the atmospheric environment and satellite interference, the user's location information may be lost, which may cause the output of the navigation system to diverge rapidly.

To date, considerable researches have been conducted to predict the navigation information by fitting the error curve of the SINS [4] and building a neural network to predict and compensate for errors [5-12] during the GPS outages. However, the method based on curve fitting is generally applicable to the static error characteristic system and is not suitable for the time-varying dynamic system. The combination between the wavelet transform and the neural network to assist the Kalman system is
discussed by Zhang and Xu [5]. In the paper [6]-[11], the information fusion of radial basis function neural network (RBFNN) with Kalman filtering is referred, and Noureldin et al. [12] adopted the dynamic neural networks to predict the system errors of GPS/INS. All these are aimed to predict the SINS position and velocity errors. However, most of the designed neural network algorithms and structures are complex, requiring large amounts of data and time-consuming calculation, which leads to poor real-time system performance. More importantly, the application scenes are constantly concentrated on ground mobile carriers or civil manned aircraft, which are not suitable for low-cost guided projectiles. But it may well encounter GPS signal missing in flight. Besides that, the traditional neural network algorithms (such as the BP algorithm) present performances with slow training speed and sensitive learning rate, and it is easy to fall into a local optimal solution [13]-[15].

The Extreme Learning Machine (ELM) is a new algorithm for the Single-hidden Layer Feedforward Neural Network (SLFN). Compared with the traditional training algorithm, ELM shows the advantages of fast learning speed and good generalization performance. In the ELM algorithm framework, the weight that connects the input layer and the hidden layer, and the bias of the hidden layer neurons are randomly generated, which are not required to be adjusted during the training process, and it enables the calculation time to be shortened. Besides that, this algorithm avoids the selection of a large number of parameters and only needs to set the number of neurons in the hidden layer to obtain the global optimal solution, which is beneficial to its application on a missile-borne platform. However, in the traditional learning algorithm such as BP algorithm, the learning rate has a great influence on the performance of the neural network, and an inappropriate learning rate may even cause the training process not converge, which may cause fatal damage to the guidance process of the projectile.

In this work, we proposed the ELM to predict the SINS navigation errors on the missile-borne platform during the GPS outages for the first time. To explore its error prediction ability, we established a model in MATLAB with a part of the navigation data of a projectile, and the algorithm was compared with the BP neural network and the genetic algorithm optimization BP (GABP) neural network algorithm. The content of this paper is distributed as follows: the SINS/GPS integrated navigation and the Kalman model, training algorithm, simulation results and analysis, conclusion.

2. THE SINS/GPS integrated navigation and the Kalman model

The SINS/GPS integrated navigation adopts loose coupling indirect mode with the GPS and SINS working independently and not interfering with each other. A 15-dimensional state vector is chosen as follows:

\[
X = \begin{bmatrix}
\phi_{E,N,U} & \delta V_{E,N,U} & \delta P_{I,\lambda,h} & \epsilon_{x,y,z} & \nabla_{x,y,z}
\end{bmatrix}^T
\]  

(1)

where \( \phi_{E} \), \( \phi_{N} \) and \( \phi_{U} \) represent attitude errors of the calculated platform in the local Cartesian coordinates coordinate system; \( \delta V_{E} \), \( \delta V_{N} \) and \( \delta V_{U} \) denote triaxial velocity errors; \( \delta \lambda \), \( \delta h \) and \( \delta \) are the longitude error, latitude error, and height error; \( \epsilon_{x} \), \( \epsilon_{y} \), \( \epsilon_{z} \) are the gyro constant drift errors in the body coordinate system and \( \nabla_{x} \), \( \nabla_{y} \), and \( \nabla_{z} \) representing the accelerometer constant offset errors.

The equation of state of the system is written as follows:

\[
\dot{X} = FX + \Phi W
\]  

(2)

The system process noise matrix is defined as \( w \) which includes random errors of gyros and accelerometers (excluding random constant errors). The system noise variance matrix is selected based on the inertial noise level of the SINS/GPS integrated navigation system.

\[
W = \begin{bmatrix}
W_{\epsilon_{x}} & W_{\epsilon_{y}} & W_{\epsilon_{z}} & W_{\nabla_{x}} & W_{\nabla_{y}} & W_{\nabla_{z}}
\end{bmatrix}^T
\]  

(3)
The differences between the position and the speed of the SINS and the GPS are selected as the measurement information of the Kalman filter, and the measurement equation is:

\[ Z(t) = H(t)X(t) + V(t) = \begin{bmatrix} H_v \\ H_p \end{bmatrix} X(t) + \begin{bmatrix} V_v(t) \\ V_p(t) \end{bmatrix} \]  

(4)

The measurement vector \( Z(t) = [\delta V_E, \delta V_N, \delta V_U, \delta l, \delta \lambda, \delta h]^T \), where \( \delta V_E, \delta V_N, \) and \( \delta V_U \) represent the speed difference, while \( \delta l, \delta \lambda, \) and \( \delta h \) denote the position difference between SINS and GPS. The measurement noise vector is given by (5), and the measurement noise variance matrix is chosen by the GPS position and velocity noise level.

\[ V(t) = \begin{bmatrix} v_{\delta V_E} \\ v_{\delta V_N} \\ v_{\delta V_U} \\ v_{\delta L} \\ v_{\delta \lambda} \\ v_{\delta h} \end{bmatrix} \]  

(5)

3. Training Algorithms

In recent years, theoretical and experimental investigations of the BP, GABP, and ELM in SLFNs have increased tremendously [13-15]. These studies have shown that they perform well in fitting nonlinear functions. It’s interesting to note that the weights of the input layer to the hidden layer and the hidden layer to the output layer in the BP algorithm all need to be solved iteratively (with the gradient descent method), and

![Figure 1. ELM algorithm modeling](image-url)

The GABP optimizes the initial weights and biases of the BP. However, the weight of the input layer to the hidden layer in the ELM algorithm is randomly assigned. When the weight between the input layer and the hidden layer is set, the weight between the hidden layer and the output layer can be obtained by the least square method, which greatly improves the training speed of the neural network.

GABP uses a genetic algorithm to optimize the weights and bias of BP neural network. In GABP algorithm, each individual in the group contains all the weights and bias of the network. The fitness values of individuals are calculated by the fitness function, and the genetic algorithm determines the individuals corresponding to the optimal fitness values by selection, crossover and mutation operations. Then, the BP neural network uses the optimal individual obtained by the genetic algorithm as the initial weights and bias of the network. After a long iteration training process, the network outputs predictions.
Compared with the GABP and the BP, the ELM is a simple algorithm without iteration, and the framework of the ELM algorithm is shown in Fig. 1.

Suppose $X$ represents the input training matrix with $Q$ samples and the output matrix is $Y$, where $X \in \mathbb{R}^{m \times Q}$ and $Y \in \mathbb{R}^{n \times Q}$. If the activation function of hidden layer neurons is $g(x)$, the output $Y$ of the network is:

$$ Y = \begin{bmatrix} y_1, y_2, \cdots, y_Q \end{bmatrix}_{m \times Q} \quad (6) $$

Where

$$ y_j = \begin{bmatrix} y_{1j} \\ y_{2j} \\ \vdots \\ y_{mj} \end{bmatrix}_{m \times 1} = \begin{bmatrix} \sum_{i=1}^{Q} \beta_{1i} g(\omega_i x_j + h) \\ \sum_{i=1}^{Q} \beta_{2i} g(\omega_i x_j + h) \\ \vdots \\ \sum_{i=1}^{Q} \beta_{mi} g(\omega_i x_j + h) \end{bmatrix}_{m \times 1} \quad (j = 1, 2, \cdots, Q) \quad (7) $$

Where

$$ \omega_i = [\omega_{i1}, \omega_{i2}, \cdots, \omega_{in}]^T, x_j = [x_{1j}, x_{2j}, \cdots, x_{mj}]^T $$

And the formula (4) can be expressed as

$$ H \beta = Y \quad (8) $$

Where

$$ H(\omega_1, \omega_2, \cdots, \omega_l, b_1, b_2, \cdots, b_1, x_1, x_2, \cdots, x_Q) = $$

$$ \begin{bmatrix} g(\omega_1 x_1 + b_1) g(\omega_2 x_1 + b_2) \cdots g(\omega_l x_1 + b_l) \\ g(\omega_1 x_2 + b_1) g(\omega_2 x_2 + b_2) \cdots g(\omega_l x_2 + b_l) \\ \vdots \\ g(\omega_1 x_Q + b_1) g(\omega_2 x_Q + b_2) \cdots g(\omega_l x_Q + b_l) \end{bmatrix}_{Q \times l} \quad (9) $$

It is proved that [14] when the activation function $g(x)$ is infinitely differentiable, the parameters of the SLFN requires no adjustments. The weight $\omega$ between the input layer and the hidden layer, and the bias $b$ of neurons in the hidden layer can be selected randomly and it remains the same condition during the training process. Hence, it greatly reduces the iterative computation time that is required in other traditional algorithms. The weight $\beta$ between the hidden layer and the output layer is obtained by the least square solution described in (10). In the solution $\hat{\beta} = H^+T^*$, $H^*$ is the Moore-Penrose generalized inverse of the hidden layer output matrix $H$. 

4
The training procedures based on the ELM algorithm mainly includes 3 steps. Firstly, determine the number of neurons in the hidden layer, with randomly setting the connection weight $\omega$ between the input layer and the hidden layer, and the bias $b$ of the hidden layer neurons. Secondly, an infinitely differentiable function is selected as the activation function of the hidden layer neurons, and then the hidden layer output matrix $H$ is calculated. Finally, calculate the weight of the output layer. Clearly, without iteration and complicated parameters, the algorithm greatly reduces the calculation time in the training process and also improves the robustness of the network.

4. Simulation Results And Analysis
In order to demonstrate the superiority of the ELM algorithm in the missile-borne platform, comparative analysis is carried out in three aspects: algorithm time consumption, the practicability of neural network structure and prediction accuracy. To compare the three algorithm prediction performances, the BP, GABP, and ELM models are established for the navigation system in MATLAB respectively. The navigation data come from a part of the flight record of a projectile.

\[
\min_{\beta} \| H \beta - T \| \tag{10}
\]

\begin{figure}
\centering
\includegraphics[width=\textwidth]{diagram.png}
\caption{Two GPS/INS loosely coupled integrated navigation modes}
\end{figure}

\begin{table}
\centering
\caption{The computation time of the three algorithms}
\begin{tabular}{|c|c|}
\hline
Algorithm & Computation time \\
\hline
BP & 58s \\
GABP & 12min \\
ELM & 0.48ms \\
\hline
\end{tabular}
\end{table}
Figure 3. the tracking of the UAV’s position and speed using different algorithms: (a) longitude-latitude trajectory; (b) upper velocity; (c) south velocity; (d) north velocity

Table 2. Model parameters

| Parameter       | Value | Parameter       | Value       |
|-----------------|-------|-----------------|-------------|
| BP              |       | GABP            |             |
| Number of iterations | 100  | Evolution algebra | 100         |
| Learning rate   | 0.1   | Population size | 10          |
| Hidden layer neurons | 10   | Crossover probability | 0.4        |
| ELM             |       | Mutation probability | 0.2        |

The SINS/GPS integrated navigation system based on the neural network mainly works in two modes (Fig. 2). When the GPS works normally, the output position $P_{SINS}$ and velocity $V_{SINS}$ of the SINS system are taken as the inputs of the neural network, and the position $P_{SINS/GPS}$ and velocity $V_{SINS/GPS}$ obtained by the Kalman filter are taken as the ideal outputs of the neural network for the training neural network to continuously optimize its parameters. During the GPS outages, the input position $P_{SINS}$, and velocity $V_{SINS}$ obtained through the SINS system behave as the input of the trained neural network, and the network can output the optimized position $P_{NN}$ and velocity $V_{NN}$. In the simulation, the computation time based on three different neural networks is as follow (Table 1): the BP neural network takes about 1 minute, the GABP occupies more than 10 minutes and the ELM needs just less than 1 second. Among the three algorithms, the ELM displays outstanding time-saving performance.

Suppose the GPS signal is interrupted after 2880s, and the UAV’s position and speed are tracked by using different algorithms as shown in Fig. 3. The parameters in the simulation are set in Table 2.

Overall, the position and velocity calculated from SINS and SINS/GPS (with Kalman filter) are markedly different due to the accumulation of inertial errors. In Fig. 3, it is clear that when the GPS is available, the calculation results of SINS are quite different from the ideal output of SINS/GPS, which means the calculation results of SINS display great errors. Furthermore, during the GPS outages (after 2880s), these three algorithms considerably improve the tracking performance of the GPS signals. It is
worth noting that there are still minor differences among the three algorithms. A compelling reason is that the performance of the BP network is greatly affected by its parameters such as the number of hidden layer nodes and hidden layer layers, node transfer function, iteration times and the learning rate. In addition, the performance of the GABP network is also related to some other parameters such as the population size, evolution times, crossover probability and mutation probability. By adjusting the parameters, the performance of the trained GABP network might well be better than that of the BP neural network. However, the simulation shows that a slight change in parameters may cause the GABP algorithm to be not only time-consuming but also slightly worse than the BP algorithm. Hence, the performance of the BP and the GABP are very sensitive to their parameters, which increase their practical application difficulty.

To further explore the differences in prediction accuracy of the three algorithms during the GPS outages, the position and the velocity errors are analyzed in detail as shown in Fig. 4. In general, the average errors of position and velocity prediction under the three algorithms are close to zero, and the variation of the three algorithms are mainly determined by the standard deviation. Fig. 4. (a)(b)(c) Illustrate the latitude, longitude and height errors. The three algorithms are more accurate in predicting the latitude and longitude, with the standard deviation of the ELM algorithm (about ± 0.002° and ± 0.001° respectively) being much smaller than that of the other two algorithms (around ± 0.003° and ± 0.007° respectively). Above-mentioned algorithms have a noticeable deviation in the prediction of the height with a deviation of 36 meters. The reason may be that when the GPS signal is interrupted, the SINS works independently and has no damping in height, which causes the value of the height to diverge rapidly and become invalid. The prediction of the speed (Fig. 4. (d)(e)(f)) by the three algorithms are showing the similarity of their prediction of the position (Fig. 4. (a)(b)(c)). In the north and the east directions, the prediction value under the ELM algorithm are more aggregated, and the standard deviation difference with the maximum error being around 0.25 m/s is approximately 1/4 of that of the other two algorithms. However, the standard deviation of the ELM algorithm is slightly larger than that of the other two algorithms in the upper speed.

![Figure 4](image-url)

**Figure 4.** Error prediction of position and velocity under three different algorithms: (a) latitude error; (b) longitude error; (c) height error; (d) north velocity error; (e) south velocity error; (f) upper-velocity error

5. Conclusion

In summary, the ELM presents a surprising prediction ability in the SINS/GPS integrated navigation system during GPS outages. The ELM algorithm serves as a new way to address the challenge of the
complex iterative computation of the traditional algorithms on the missile-born platform, which is available to reduce the power consumption of projectiles. Furthermore, the simulation reveals that it has the remarkable advantages of much fewer parameters and simpler network structure than the BP and the GABP, which enables the ELM algorithm to be more time-saving and present stronger practicability. However, the ELM still needs further improvement when predicting the position of the height direction. To be more important, this paper is based on the offline training and prediction of the neural networks, while in practical application, the problems of the real-time performance need to be considered.

Acknowledgments
This work was supported by the National Natural Science Foundation Project (31201391). Fangxiu Jia is the corresponding author.

References
[1] Gross J, Gu Y, Gururajan S, et al. A comparison of Extended Kalman Filter, Sigma-Point Kalman Filter, and Particle Filter in GPS/INS sensor fusion [C]// Aiaa Guidance, Navigation, & Control Conference. 2013.
[2] Park D B, Shin D H, Oh S H, et al. Development of a GPS/INS system for precision GPS guided bombs [J]. IEEE Aerospace and Electronic Systems Magazine, 2012, 27 (3): 31 - 39.
[3] Wang L J, Zhao H C, Xiaonu Y. The modeling and simulation for GPS/INS integrated navigation system [C]// International Conference on Microwave & Millimeter Wave Technology. IEEE, 2008.
[4] Li X, Xie L, Chen J, et al. A ZUPT method based on SVM regression curve fitting for SINS [C]// 2014 33rd Chinese Control Conference (CCC). IEEE, 2014.
[5] Zhang T, Xu X. A new method of seamless land navigation for GPS/INS integrated system [J]. Measurement, 2012, 45 (4): 691 - 701.
[6] Wang X, Chen J X, Ni W. A hybrid prediction method and its application in the distributed low-cost INS/GPS integrated navigation system [C]// International Conference on Information Fusion. IEEE, 2015.
[7] Malleswaran M, Deborah S A, Manjula S, et al. Integration of INS and GPS using radial basis function neural networks for vehicular navigation [C]// International Conference on Control Automation Robotics & Vision. IEEE, 2011.
[8] Sharaf R, Noureldin A, Osman A, et al. Online INS/GPS integration with a radial basis function neural network [J]. IEEE Aerospace & Electronic Systems Magazine, 2005, 20 (3): 8 - 14.
[9] Noureldin A, Sharaf R, Osman A, et al. INS/GPS data fusion technique utilizing radial basis functions neural networks [C]// Position Location & Navigation Symposium. IEEE, 2004.
[10] Semeniuk L, Noureldin A. Bridging GPS outages using neural network estimates of INS position and velocity errors [J]. Measurement Science and Technology, 2006, 17 (10): 2783 - 2798.
[11] Chen L, Fang J. A Hybrid Prediction Method for Bridging GPS Outages in High-Precision POS Application [J]. IEEE Transactions on Instrumentation and Measurement, 2014, 63 (6): 1656 - 1665.
[12] Noureldin A, El-Shafie A, Bayoumi M. GPS/INS integration utilizing dynamic neural networks for vehicular navigation [J]. Information Fusion, 2011, 12 (1): 48 - 57.
[13] Zhu Q Y, Qin A K, Suganthan P N, et al. Evolutionary extreme learning machine [J]. Pattern Recognition, 2005, 38 (10): 1759 - 1763.
[14] Huang G B, Zhu Q Y, Siew C K. Extreme learning machine: Theory and applications [J]. Neurocomputing, 2006, 70 (1-3): 489 - 501.
[15] Huang G B, Chen L, Siew C K. Universal Approximation Using Incremental Constructive Feedforward Networks With Random Hidden Nodes [J]. IEEE Transactions on Neural Networks, 2006, 17 (4): 879 - 89.