A Neural Network Aided Kalman Filtering Approach for SINS/RDSS Integrated Navigation

*HE Xiao-feng, HU Xiao-ping, LU Liang-qing, TANG Kang-hua

College of Mechatronics Engineering and Automation, National Univ. of Defense Technology, Changsha, PR.China(E-mail: hexf_lit@126.com)

Abstract

Kalman filtering (KF) is hard to be applied to the SINS (Strap-down Inertial Navigation System)/RDSS (Radio Determination Satellite Service) integrated navigation system directly because the time delay of RDSS positioning in active mode is random. BP (Back-Propagation) Neuron computing as a powerful technology of Artificial Neural Network (ANN), is appropriate to solve nonlinear problems such as the random time delay of RDSS without prior knowledge about the mathematical process involved. The new algorithm betakes a BP neural network (BPNN) and velocity feedback to aid KF in order to overcome the time delay of RDSS positioning. Once the BP neural network was trained and converged, the new approach will work well for SINS/RDSS integrated navigation. Dynamic vehicle experiments were performed to evaluate the performance of the system. The experiment results demonstrate that the horizontal positioning accuracy of the new approach is 40.62 m (1σ), which is better than velocity-feedback-based KF. The experimental results also show that the horizontal positioning error of the navigation system is almost linear to the positioning interval of RDSS within 5 minutes. The approach and its anti-jamming analysis will be helpful to the applications of SINS/RDSS integrated systems.

Keywords: Integrated Navigation, RDSS, SINS, Kalman Filtering, Neural Network

1. Introduction

A double-star positioning system using two geostationary satellites providing Radio Determination Satellite Service(RDSS) has been established in China since 2000. RDSS combines rapid positioning and short message communication together. But not as GPS and GLONASS, RDSS works in the mode of active positioning, that is, the positioning function is calculated by the center station and transferred by geostationary satellites. The obvious shortcoming of this working mode is that there is a time delay between the positioning request and the answer [1]. The time delay varies random with time so that it is hard to be predicted and compensated and causes larger errors while the user's dynamic increases. Another shortcoming of active mode is that the capacity of the system is limited by the control center's calculation and communication ability, which does not make so many users share the service at the same time.

A SINS is a self-contained positioning and attitude device. In other words, it meets the all-environment requirement. The primary advantage of using an INS for land vehicle navigation applications is that velocity and position of the vehicle can be provided with abundant dynamic information and excellent short term performance.

The last two decades have seen an increasing trend in the use of integrated INS and GPS for a variety of positioning and navigation applications. Chinese scholars such as ZHAO Long[2], LIU Zhun[3], LIN Yueyun[4], HU Guangfeng[5] etc, have researched RDSS/SINS integrated navigation algorithms.

In this paper, new techniques for RDSS/SINS integration based on artificial intelligent are proposed. Firstly, the proposed approach adopts the feedback of velocity to impair the time delay of RDSS. Secondly, BPNN is trained to eliminate the residual time delay and other noise. With the experimental data, the accuracy of the proposed approach is better than the accuracy of velocity-feedback-based KF. The anti-jamming of the system will be analyzed.

2. Design of the BPNN-aided KF

2.1 Inertial state dynamic error model

IMU mechanization is the process of solving the navigation states PVA (position, velocity and attitude) from the measurements of gyros and accelerometers. The axes of the navigation frame are aligned with the directions of east, north and the local vertical (up). The direction of the local vertical is not stable in IMU mechanization [6]. And the receiver of RDSS only exports the latitude, longitude and no information of height. So the direction of the local vertical is not considered. The inertial sensor errors (mainly gyros biases) are very important and will cause the PVA solution to diverge quickly. Therefore, the state vector of the dynamic error model is X in local level frame.

$$X = \left[\delta L, \delta\lambda, \delta V_E, \delta V_N, \phi_E, \phi_N, \phi_U, \varepsilon_E, \varepsilon_N, \varepsilon_U\right]^T \tag{1}$$

where

$\delta L, \delta\lambda$ position (latitude and longitude) error states

$\delta V_E, \delta V_N$ velocity (east and north) error states

ϕ_E, ϕ_N, ϕ_U attitude (east, north and up) error states

$\varepsilon_E, \varepsilon_N, \varepsilon_U$ gyro (east, north and up) drift vectors

The state vector to be solved consists of 10 states adding the gyro residuals. The gyro drift is regarded as Gaussian white noise. Inertial state dynamic error model can be written in the following form:

$$\dot{X}(t) = F(t) \cdot X(t) + G(t) \cdot W(t) \tag{2}$$

where

X error state vector of inertial navigation whose elements include two position errors, two velocity errors, three attitude errors and three gyro drift terms,

F state transition matrix, see reference[6] for details,

G a rectangular matrix,

W a zero-mean Gaussian white noise vector.

2.2 Kalman filtering

The observation equations should be presented in the condition of time synchronization. Actually, the time delay (Δt) of RDSS is more than 0.8 seconds. It is random and hard to be modeled because RDSS works in active positioning mode. The observation equations can be represented in the following form:

$$Z = H \cdot X + V \tag{3}$$

where

$Z = \left[L_I - L_B, \lambda_I - \lambda_B\right]^T$ observation vector,

(L_I, λ_I) position calculated by SINS,

$H = \left[I_{2\times2} \quad 0_{2\times8}\right]$ design matrix relating the system update measurements to the system error states,

V vector of update measurements random noise.

$$L_B = L_B{}' + V_N \bullet \Delta t / R \tag{4}$$

$$\lambda_B = \lambda_B{}' + V_E \bullet \Delta t / R \tag{5}$$

where

(L_B, λ_B) position synchronized to SINS position,

$(L_B{}', \lambda_B{}')$ position by RDSS,

R earth radius,

Δt estimated time delay of RDSS.

We can estimate the time delay (Δt) of RDSS by recording both the time when RDSS receiver asks for the positioning and the time when RDSS receiver accepts the position solution from the centre station through the satellites.

A KF is adopted for the optional estimation of the SINS error state vector components. The state-space error model consists of discrete equations 2 and 3. Therefore, the discrete KF algorithm can be summarized as [7]:

$$\begin{cases} \hat{X}_{k,k-1} = \phi_{k,k-1}\hat{X}_{k-1,k-1} \\ P_{k,k-1} = \phi_{k,k-1}P_{k-1,k-1}\phi_{k,k-1}^T + G_{k-1}Q_{k-1}G_{k-1}^T \\ K_k = P_{k,k-1}H_k^T[H_kP_{k,k-1}H_k^T + R_k]^{-1} \\ \hat{X}_{k,k} = \hat{X}_{k,k-1} + K_k[Z_k - H_k\hat{X}_{k,k-1}] \\ P_{k,k} = [I - K_kH_k]P_{k,k-1} \end{cases} \tag{6}$$

where

$\hat{X}_{k,k-1}$ predicted estimate of the system error state vector at time t_k ,

$\hat{X}_{k,k}$ updated estimate of the system error state vector at time t_k ,

$P_{k,k-1}$ covariance matrix of $\hat{X}_{k,k-1}$,

$P_{k,k}$ covariance matrix of $\hat{X}_{k,k}$,

K_k Kalman gain matrix,

Q_{k-1} covariance matrix of the system input noise

R_k covariance matrix of the measurement noise.

2.3 BPNN-aided KF

Neural network is a nonlinear dynamics system, which possesses strong fault-tolerance performance and robustness in addition to adaptive learning ability [8]. BP neural network is used most abroad in so many applications as a kind of neural network. The proposed approach takes good advantage of BPNN to aid KF in order to overcome the influence of time delay of RDSS, time synchronization of the hybrid navigation system and many other nonlinear factors.

Figure 1 illustrates the structure of the BPNN-aided KF.

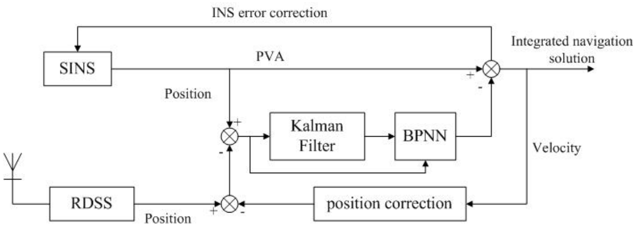

Figure 1. Structure of the BPNN-aided KF

The BPNN-aided KF estimates the state vector by using a form of feedback control. In this case, velocity vector is as a feedback to correct the position error of RDSS, which is represented by equations (4) and (5). INS error will be corrected every filtering cycle.

Jiao Licheng [9] had proved in theory that 3 layers of neural network can realize arbitrarily complicated nonlinear mapping problems. In theory, the more layers network contains, better accuracy will be obtained. But in fact, random noise and inference will be aggrandized as systemic errors if the layers are too much. In this case, neural network may not work well, or even be dispersible. In this paper, we use 3-layer BP neural network to aid KF where the vectors $Z_k - H_k\hat{X}_{k,k-1}$, $\hat{X}_{k,k} - \hat{X}_{k,k-1}$ according to the position information (4 dimensions) are as input layer and the position error (2 dimensions) is as destination layer. The number of hidden layers is 7. The decision function of hidden layers is usually nonlinear and adopted as a TANSIG function provided by MATLAB toolbox. To simplify the BPNN, linear transfer function is adopted as PURELIN function in output layer.

2.4 BP Algorithm

BP algorithm is used to train the neural network of which the learning process includes mode forward and error reverse.

The standard BP algorithm is a simple, fastest declining static optimization algorithm. The weight vector $\omega(k)$ is modified as presented as following.

$$\omega(k+1) = \omega(k) + \eta \bullet \delta x(k) \tag{7}$$

where

$\delta x(k) = -\partial E / \partial \omega(k)$ negative gradient at time k ,

η a momentum factor and $0 \le \eta \le 1$.

In this case, oscillation will occur frequently in the process of study. The network converges slowly.

In this paper, Levenberg-Marquardt (L-M) algorithm is adopted based on the standard BP algorithm and Gauss-Newton algorithm. It takes advantage of fast convergence of Gauss-Newton algorithm and global optimization of the standard BP algorithm. The equation (7) is improved to the following form.

$$\omega(k+1) = \omega(k) - \eta \bullet [J^T J + \lambda I]^{-1} J^T e(x) \tag{8}$$

where

$e(x)$ error vector of the network output,

J Jacobian matrix of $e(x)$,

λ coefficient and $\lambda \geq 0$,

I identity matrix.

λ is an adaptive coefficient. If the sum of error decreases, let $\lambda = \lambda / \varphi$. If the sum of error increased, let $\lambda = \lambda \bullet \varphi$. φ is a factor and satisfies $\varphi > 0$. If $\lambda = 0$, the L-M algorithm equals the Gauss-Newton algorithm. If $\lambda \to +\infty$, the L-M algorithm is close to the standard BP algorithm. We can increase the step by φ times when the convergence is slowly while decrease the step by φ times when extremum is in a local area.

Based on the L-M algorithm, the parameters of the BPNN in this paper are represented as Table 1.

Table 1. Parameters of the BPNN

Number of input nerve cell: 4	Number of output nerve cell: 2
Training epochs: 100	Goal: 0.02
λ : 0.001	φ : 10

3. Experiments

3.1 Experimental system

The experimental system consists of an automobile, RDSS receiver, RDSS antenna, INS, DGPS, industrial computer and power supply. The accuracy of RDSS positioning is 100 m and the minimum interval of positioning is 60 s. INS is a navigation-grade system. The reference position of the experiments is provided by DGPS (Ashetech GG24) of which the position accuracy is better than 2 m and position frequency is 1 Hz.

3.2 Experimental trajectory

Experiments were made in a highway in the suburb of Changsha on Oct 9th, 2005. Figure 2 illustrates the experimental trajectory.

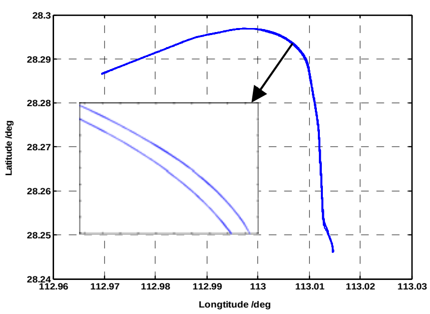

Figure 2. Experimental trajectory

The trajectory is given by DGPS positioning data. The automobile run along the trajectory from the start point to the end point and then turned around back to the start point for three turns. The velocity of the automobile is about 50 km/h.

3.3 Experimental results

The samples of INS are every 0.01 s, but the samples of RDSS are at least every 60 s. The cycle of BPNN-aided KF is at least 60 s in this paper. To build and train the BPNN, we have sampled the data of INS, RDSS and DGPS firstly. Then we evaluate the algorithm offline.

In figure 3, we can see the variation of position errors with respect to time in the BPNN-aided KF approach. The trained BPNN can perform its task well and stably. Figure 4 shows the position errors in the velocity-feedback-based KF case. The horizontal position accuracy of the BPNN-aided KF is 40.62 m while that of the velocity-feedback-based KF is 46.75 m. It is obvious that the performance of the new approach yields significant improvements in performance with reduced RMS errors up to about 13% compared with the velocity-feedback-based KF.

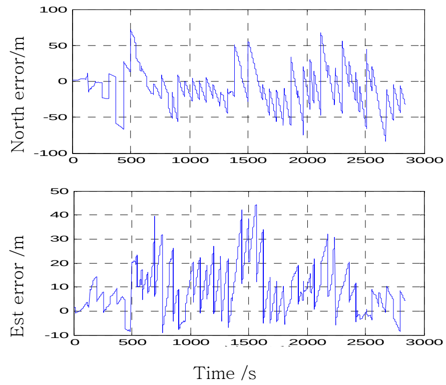

Figure 3. Position error of BPNN-aided KF

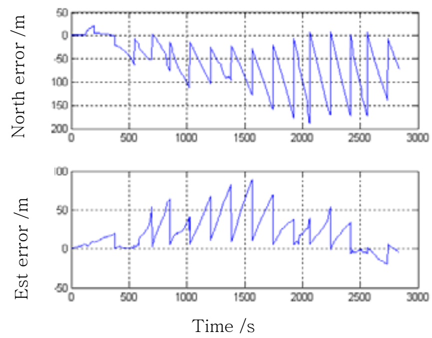

Figure 4. Position error of velocity-feedback-based KF

In order to estimate the performance of the SINS/RDSS integrated navigation system with respect to the positioning interval of RDSS, we analyze the accuracy with different interval of RDSS such as 120 s, 180 s, 240 s, 300 s (Table 2). Obviously, longer the interval is, worse the accuracy will be.

In Table 2, RMS represents the horizontal error of SINS/RDSS system and is calculated by the following equation.

$$RMS = \sqrt{north^2 + east^2} \tag{9}$$

Figure 5 clearly shows that the horizontal positioning error of the SINS/RDSS integrated navigation system is almost linear to the positioning interval of RDSS within 5 minutes.

Table 2. Error comparison of SINS/RDSS with different interval of RDSS positioning.

Error /m	60 s	120 s	180 s	240 s	300 s
North	35.96	73.36	106.07	137.11	166.68
East	18.89	31.56	42.76	53.55	68.14
RMS	40.62	79.86	114.36	147.20	180.07

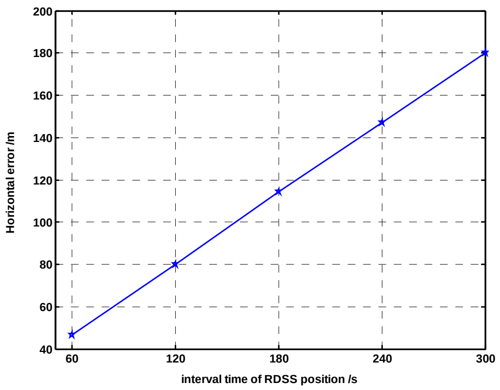

Figure 5. Horizontal error with interval time of RDSS position

4. Conclusion

In this paper we provided an exact solution for the SINS/RDSS integrated navigation system based on BP neural network and KF. Experimental results show that the application of neural network methodology is feasible in the combination of KF and the proposed approach works better than traditional methods. The approach and its anti-jamming analysis based on experimental data will be helpful to the application of SINS/RDSS integrated navigation systems.

Acknowledgement

The authors would like to thank Dr. TANG Yonggang and Dr. LIAN Junxiang for their help in the experiments.

Reference

1. Kazuhiro Kimura, Eihisa Morikawa, “Communication and radio determination system using two geostationary satellites,” *IEEE Transaction on Aerospace and Electronic System*, Vo.32, No.1, 1996, pp. 314-325.
2. ZHAO Long, LIU Zhun and CHEN Zhe, “Recursive damped least square and its application in INS/DS integrated system,” *Journal of Beijing University of Aeronautics and Astronautics*, Vo.29, No.12, 2003, pp. 1136-1138.
3. LIU Zhun, CHEN Zhe, “Research on INS/DS Integrated Navigation Semi-Physical Simulation”, *Journal of System Simulation*, Vo. 16, No. 2, 2004, pp. 313-315.
4. LIN Xueyuan, LIU Jianye and WANG Shuhua, “The Study on Double-Star Position /SINS Integration Navigation System”, *Chinese Space Science and Technology*, Vo. 23, No. 2, 2003, pp. 34-38.
5. HU Guangfeng, MA Xiaoyan, WANG Yandong and FAN Yuezu, “A Method of Real-Time in RDSS/SINS Integrity System”, *Journal of Projectiles, Rockets, Missiles and Guidanc*, Vo. 25, No. 2, 2005, pp. 39-44.
6. YUAN Xin, YU Jixiang and CHEN Zhe, “Navigation System,” 1st Edition, Aviation Industry Press, 1993, pp. 48-54.
7. MOHINDER S.GREWAL, LAWRENCE R.WEILL and ANGUS P.ANDREWS, “Global Positioning System, Inertial Navigation, and Integration”, A John Wiley & Sons, Inc., 2001, pp. 198-200.
8. Yucong HU, Jianmin Xu, and Huiling Zhong, “Fusion Model of Vehicle Positioning with BP Neural Network,” *Proceedings of Intelligent Transportation Systems, IEEE,* 2003, pp. 643-647.
9. JIAO Licheng, “Neural Network theory and system”, Beijing,Xi’an Electronic Science and Technology Press , 1990, pp. 124-128.