International Global Navigation Satellite Systems Society
IGNSS Symposium 2006

Holiday Inn Surfers Paradise, Australia
17 – 21 July 2006

Designing a Neural Network for GPS/INS/PL Integration

***Jianguo Jack Wang, Jinling* Wang**
School of Surveying and Spatial Information Systems
University of New South Wales, Sydney, NSW 2052, Australia
Tel: +61-2-9385 4185 Fax: +61-2-9313 7493
Email: jianguo.wang@student.unsw.edu.au

David Sinclair, Leo Watts
QASCO Surveys Pty. Limited,
41 Boundary St. South Brisbane, Qld, 4101
Tel: Ph. 07-3331 3000 Fax: 07-3217 7777
Email: David.Sinclair@qasco.com.au

Jianguo Jack Wang

ABSTRACT

Although Kalman filtering is an optimal real-time data fusion method for GPS/INS integration, it has some limitations in terms of stability, adaptability and observability, etc. A Kalman filter can perform optimally only when its dynamic model is correctly defined and the noise statistics for the measurement and process are completely known. As a Kalman filter's prediction diverges without measurement update, the performance of a GPS/INS integrated system could degrade rapidly if GPS signals are unavailable. Neural networks, on the other hand, can map input-output relationships without apriori knowledge about them - hence this technique can be applied to overcome some limitations of Kalman filtering.

This paper presents the design of a high accuracy GPS/INS/PL (pseudolite) integrated airborne georeferencing system. It combines Kalman filtering and neural networks to improve navigation solutions between GPS/PL sampling and during GPS/PL receiver dropouts caused by antenna shading or other reasons. Kalman filtering is the primary integration method when GPS signals are available. A multi-layer neural network is trained to map INS measurements with the corresponding Kalman filter predicted errors, such as accelerometer and gyro errors and navigation parameter errors. Then the neural network can be used to improve Kalman filter estimated navigation solutions when only INS measurements are available. Field test data are processed to evaluate the proposed hybrid method.

KEYWORDS: hybrid, Kalman filter, training, prediction, navigation

1. INTRODUCTION

GPS/INS integrated systems have been becoming a popular tool to direct georeference airborne surveying platforms. Carrier phase DGPS can provide accurate positioning solutions at a relatively low data rate. Ground based pseudolites (PLs) can be used to strengthen GPS measurement geometry and further improve positioning accuracy and reliability, especially in the vertical component (Wang et al., 2004). At the same time, INS measures attitude, velocity and position at a high data rate, with GPS/PL frequently correction using Kalman filter (KF) or other real-time data fusion method. The performance of an integrated system depends not only on the quality of each component but also the data fusion method. It is a challenging task to develop optimal real-time data fusion methods for GPS/INS integration that can improve system performance and reduce the cost (especially INS).

KF is the core of most INS/GPS integrated systems implemented to date. It can optimally estimate the position, velocity and attitude of a moving vehicle by using precise GPS measurements to update the INS frequently. KF is computationally efficient, which is especially useful for real-time processing applications. On the other hand, KF has some shortcomings. The system dynamics need to be completely known, and the statistics of the system error and the observation error are known to be normally distributed. But actually many systems can hardly meet these requirements fully. Another problem with the KF is the drift during prediction mode when GPS/PL signals are lost. The next state vector of the vehicle is estimated based on the inputs from the inertial sensors and previous filter predicted states. KF has a limited 'memory' in this mode. In most cases a first order Gauss Markov assumption is made which means that the current estimates depend solely on the previous estimates. So if the previous estimates have any errors associated with them, they are propagated to the current estimate and summed with new errors to accumulate an even larger error (Goodall et al., 2005). This lack of 'memory' causes larger drift for longer time period is an inherent disadvantage of Kalman filter predictions.

Neural networks (NNs) have been proposed to replace KF as the multi-sensor integrator (Chiang and El-Sheimy, 2004; El-Sheimy and Abdel-Hamid, 2004). It is well known that NNs are capable of adapting themselves to learn input-output relationships. This means that no initial dynamic or noise models need to be set as these are learned over time. If the models or vehicle dynamic changes then the NNs adapt to the changes. At the same time, however, the NN approach also has some shortcomings. Its estimation accuracy is not ideal and depends on the artificial experience. At current stage, Kalman Filter still remains at the forefront of INS/GPS integration.

Combining KF with NN to circumvent their inherent problems and improve the overall performances of INS/GPS/PL integration systems is a potential solution. A NN aided adaptive extended KF method was proposed (Jwo and Huang, 2004). A NN based approach for tuning

KF is developed (Korniyenko et al., 2005). NN and KF were combined together to bridge GPS outages (Goodall et al., 2005). NN model was used for de-noising MEMS-based inertial data (El-Rabbany and El-Diasty, 2004).

This paper proposes a new hybrid method that combines KF and NN to overcome the shortcomings of each. An Extended Kalman filter estimates the accelerometer and gyro measurement errors, plus velocity and attitude errors etc., and gives precise position, velocity and attitude solutions while DGPS signals are available. At the same time, a multi-layer feed-forward back-propagation NN is trained to map the vehicle dynamics with corresponding KF predictions. The inputs of the NN are the platform attitude and dynamic changes ($\Delta\theta$, ΔV), and the outputs are Kalman filter predicted error states. The NN is employed to map these input-output relationships. After the NN is trained to meet a similarity threshold, its output can be used to compensate KF drifts and improve navigation solutions when only INS measurements are available.

The remainder of the paper is organized as follows. Section 2 describes the architecture of the proposed hybrid system. Section 3 analyzes the KF states and their relationship, and defines the inputs and outputs of the NN. Section 4 describes the design of the NN. Section 5 presents the test results and discussions. The concluding remarks are given in Section6.

2. SYSTEM ARCHITECTURE

2.1 INS/GPS/PL Integration with KF

A tightly coupled Kalman filtering scheme is applied for PL augmented GPS/INS integration, which makes it possible to update the filter even with less than four GPS/PL signals, and can provide better accuracy and less sensitivity to satellite dropouts than a loosely coupled one does. A multilayer NN is combined in the system to improve KF predictions.

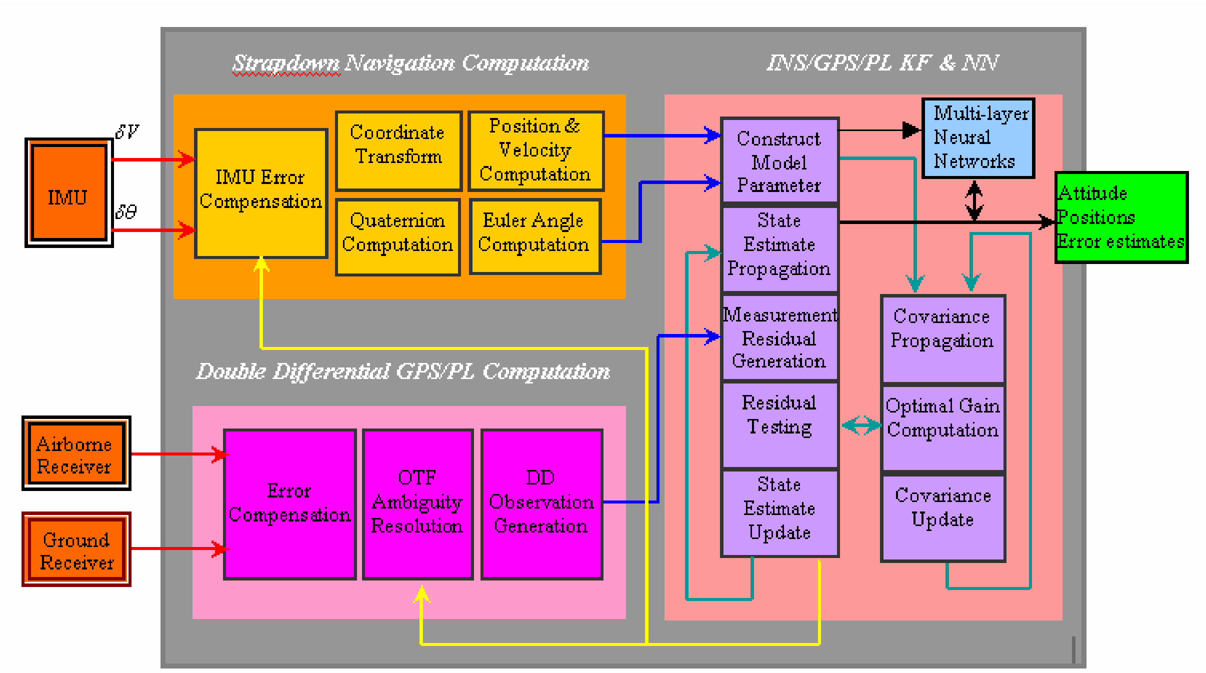

Figure 1. Architecture of tightly integrated INS/GPS/PL System with NN

As shown in Figure 1, after the initial ambiguity resolution, a tightly coupled KF is employed to process the INS, GPS and PL data. INS row measurements processed by a strapdown navigation computation and double differenced GPS/PL carrier phase measurements are put into the KF. The GPS/PL integer ambiguities are fixed by an OTF searching method before the filter measurement update. INS data processed by the filter provides the relative movement information and is used to detect and repair GPS/PL cycle slips. In addition, the close-loop update technique is used to correct the INS errors. The estimated errors are fed back to update the inertial solutions to limit the inertial position errors to centimeter level with 1 Hz GPS/PL measurements available.

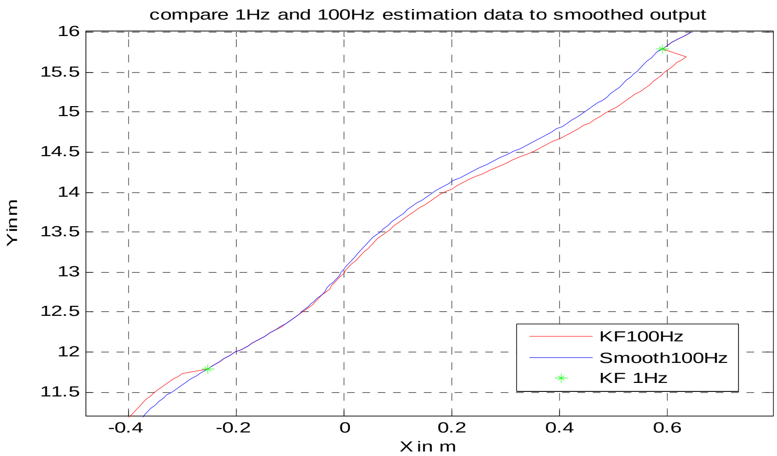

Figure 2. Iinertial positioning errors between GPS corrections

As shown in Figure 2, however, a trajectory produced by the integrated system with KF is not smooth due to inertial positioning drift caused by the KF prediction error between GPS/PL measurement updates. It is well known that INS suffers from time-dependent growth of systematic errors that quickly exceed the accuracy specification of many applications if no frequently GPS updating is applied to limit these errors.

2.2 GPS Outage Bridging

Many smoothing methods are developed to improve the KF based INS/GPS geo-referencing solution, especially for bridging GPS outages (Nassar et al., 2005). In backward smoothing, an optimal smoothed estimate of the state vector at an epoch k is obtained by combining the forward and backward estimates. The forward estimate is obtained using all measurements up to k. The backward estimate is obtained using all or some of the measurements after k. The backward smoothing in general can improve KF estimates during GPS outages. However, its solution is only available in post-mission.

A near real time smoothing method called INS backward error modeling (BEM) is proposed to bridge GPS Outages (Nassar and Schwarz, 2001). This method is based on the fact that the error at the beginning and end of a GPS outage can be determined accurately with the GPS measurement updates. Based on the errors at both ends of GPS outages and the relation between KF parameters, a quadratic model during GPS outages could be derived with only one constant acceleration error parameter. The blue trajectory in the Figure 2 is the smoothing result with this method, and the red one is the KF output.

This simple INS bias modeling method performs better than backward smoothing methods for near straight-line trajectories, but does not work well for the curved trajectories. This is because the assumption of a constant acceleration error parameter is not accurate enough for smoothing long term GPS outage with varying platform dynamics. More delicate models are needed to handle the change of system parameters during the GPS outages.

2.3 NN Aided Solution

A new approach is developed for real time GPS/INS outputs correction (smoothing) by combining NN into the KF based integration system. As shown in Figure 1, a multi-layer feed-forward back-propagation NN is trained on-line to map the platform dynamics with corresponding KF prediction errors, such as accelerometer and gyro bias and navigation parameter errors, at the rate of GPS/PL measurement update. Then the NN can be used to improve KF estimated navigation solutions when only INS measurements are available, for both between the (1Hz) GPS/PL samplings and during GPS outages. Similar to the BEM, the principle of this method is that the KF predicted navigation error at epoch *k+1* could be used to correct the system output between epoch *k* and *k+1* in near real time, based on the following equation.

$$\begin{aligned} &\hat{x}_{k+\tau j}^{Nav} = \hat{x}_{k}^{Nav} + f(j/\tau)\,(\hat{x}_{k+1}^{Nav} - \hat{x}_{k}^{Nav}) \\ &\tau = \mathrm{f}_{\mathrm{IMU}}/\mathrm{f}_{\mathrm{GPS}} \ \text{ and } j = 0,1,\ldots,\tau \end{aligned} \tag{1}$$

where f_{GPS} and f_{IMU} are the sampling rates of GPS and IMU respectively, and *f* is the dynamic relations of the KF predicted navigation errors.

If the NN can be trained on line to map the platform dynamic changes between epoch *k* and *k+1* with KF predicted navigation parameter errors at epoch *k+1*, the system output can be corrected in real time. A few issues need to be handled before this method can be implemented. Proper inputs and outputs of the NN have to be selected first. Then an optimal NN architecture needs to be designed for the online training. After that the NN recalls the training results and merges them into the system output. The next section describes the details of the KF states and their relationship, and the definition of the NN inputs and outputs.

3. KF AND NN PARAMETERS ANALYSIS

3.1 KF States Relationship

The error states (instead of whole-value filter states) are chosen for the KF. The complexity of the INS error model depends on the model for IMU sensor measurement errors, as well as the gravity uncertainty (Da et al., 1996). Here, a 24-state model includes the following variables:

$$\begin{aligned} &\mathbf{x}_{Nav} = [\delta r_N, \delta r_E, \delta r_D, \delta v_N, \delta v_E, \delta v_D, \delta\psi_N, \delta\psi_E, \delta\psi_D]^T \\ &\mathbf{x}_{Acc} = [\nabla_{bx}, \nabla_{by}, \nabla_{bz}, \nabla_{fx}, \nabla_{fy}, \nabla_{fz}]^T \\ &\mathbf{x}_{Gyro} = [\varepsilon_{bx}, \varepsilon_{by}, \varepsilon_{bz}, \varepsilon_{fx}, \varepsilon_{fy}, \varepsilon_{fz}]^T \\ &\mathbf{x}_{Grav} = [\delta g_N, \delta g_E, \delta g_D]^T \end{aligned} \tag{2}$$

where $\mathbf{x}_{Nav}$, $\mathbf{x}_{Acc}$, $\mathbf{x}_{Gyro}$ and $\mathbf{x}_{Grav}$ are the navigation solution error vector, the accelerometer measurement error vector, the gyro measurement error vector and gravity uncertainty respectively. Subscript b stands for bias and subscript f stands for scaling factor.

The following complete terrestrial INS psi-angle error model is adopted in the system.

$$
\begin{aligned}
\delta \dot{\mathbf{v}} &= -(\omega_{ie} + \omega_{in}) \times \delta \mathbf{v} - \delta \psi \times \mathbf{f} + \delta \mathbf{g} + \nabla \\
\delta \dot{r} &= -\omega_{en} \times \delta \mathbf{r} + \delta \mathbf{v} \\
\delta \dot{\psi} &= -\omega_{in} \times \delta \psi + \varepsilon
\end{aligned}
\tag{3}
$$

where $\boldsymbol{\delta v}$, $\boldsymbol{\delta r}$, and $\delta\psi$ are the velocity, position, and attitude error vectors respectively; ∇ is the accelerometer error vector; $\boldsymbol{\delta g}$ is the error in the computed gravity vector; and $\boldsymbol{\varepsilon}$ is the gyro drift vector.

It is important to develop proper dynamic and stochastic models for the system errors to understand their effect on the navigation solution, and to estimate these errors using external measurements. The strap-down INS navigation computation diagram is expressed in Figure 3. All these computations are applied in the proposed integration system.

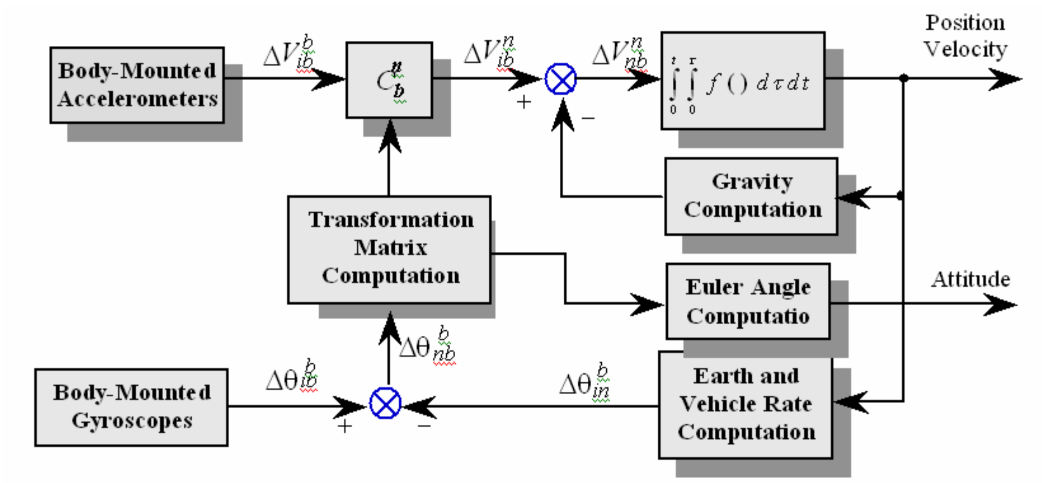

Figure 3. Strap-down INS navigation computation diagram

where ΔV_{ib}^{b} is delta velocity from accelerometers, $\Delta\theta_{ib}^{b}$ is delta □ from gyros (angular rates) and C_{n}^{b} is the direction cosine matrix from b-frame to n-frame.

3.2 NN Inputs and Outputs

As mentioned about, the principal idea of the proposed NN and KF hybrid method is using NN to map the relations between platform dynamic changes during the KF measurement update and the KF predicted error states after each update. This NN training procedure is processed at the GPS sampling rate. Then the well-trained NN can be used to improve the KF prediction at the system output rate (up to the IMU sampling rate). The input parameters selected in this approach are the changes of platform velocity and attitude from epoch k to k+1 and the average attitude in the epoch, as listed in Equation (4). It should be noticed that the heading angle is not included in the input but its change rate is included.

$$NN_{in} = [\Delta v_N, \Delta v_E, \Delta v_D, \Delta\psi_N, \Delta\psi_E, \Delta\psi_D, \psi_N, \psi_E] \quad (4)$$

As the heading angler ψ_D (green curve in the Figure 4) is limited to change between $-\pi$ and π, its changing rate $\Delta\psi_D$ has big jumps when the heading angle has jumps, as the red curve shown in the figure. These jumps will disturb the NN training, and need to be removed. The blue curve in the figure is $\Delta\psi_D$ after the jumps are removed.

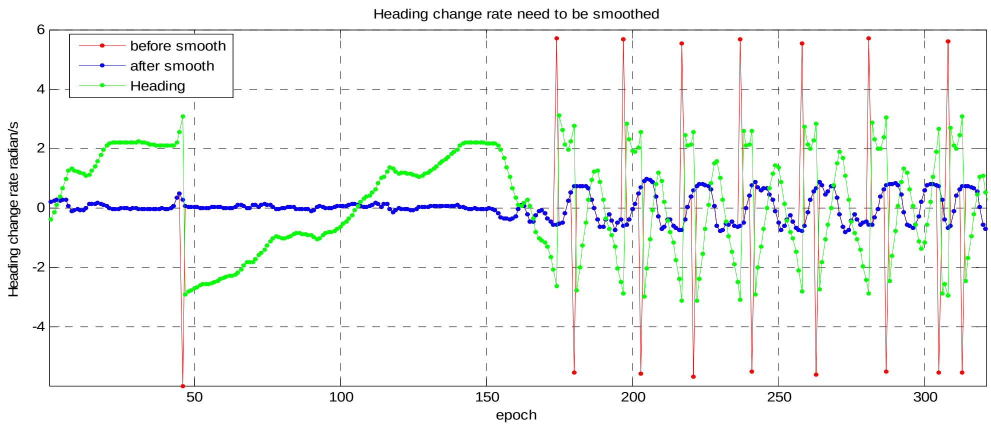

Figure 4. Smooth heading change rate.

The NN outputs/targets are selected as the KF error states that largely impact the system navigation solution, which are listed as follows:

$$NN_{out} = [\nabla_{bx}, \nabla_{by}, \nabla_{bz}, \varepsilon_{bx}, \varepsilon_{by}, \varepsilon_{bz}, \delta r_N, \delta r_E, \delta r_D, \delta v_N, \delta v_E, \delta v_D, \delta\psi_N, \delta\psi_E, \delta\psi_D] \quad (5)$$

These states are accelerometer and gyro bias, position, velocity and orientation errors, respectively. There is a pattern in the NN output parameters, as shown in Figure 5, which is largely caused by the platform dynamics. The top curve in the figure is the platform heading change rate, and the bottom one is the KF estimated corresponding orientation error. This indicates that some relationships between the NN input and output parameters exist, and a properly trained NN could map these relationships.

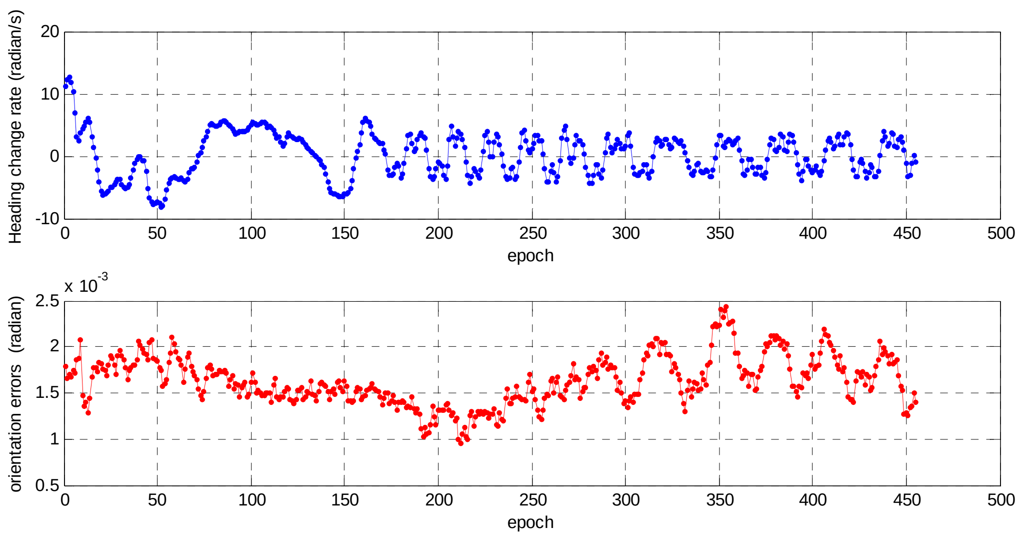

Figure 5. NN input and output Samples

4. NN DESIGN

4.1 NN Supervised Learning

NN has a history of more than fifty years, but has found solid application only in the past two decades, and this field is still developing rapidly. Today NN can be designed and trained to perform complex functions and solve problems that are difficult for conventional computers or human beings.

Neural networks are composed of simple elements operating in parallel. These elements are inspired by biological nervous systems. As in nature, the network function is determined largely by the connections between elements. A NN can be trained to perform a particular function by adjusting the values of the connections (weights) between elements so that a particular input leads to a specific target output. Such a situation is shown below. The NN is adjusted, based on a comparison of the output and the target, until the network output matches the target. The procedure of supervised learning for NN is shown in Figure 6. Given an unknown model or an unknown functional relationship with its inputs ***x*** and observed outputs ***d***. A neural network learns to fit the unknown model or functional relationship by comparing the output from a neural network ***y*** with the observed output ***d***. It then adjusts the value of its internal weighted links ***w*** iteratively until the error ***e*** between ***y*** and ***d*** meet a predefined accuracy.

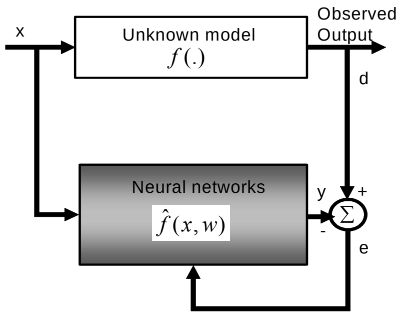

Figure 6. NN learning procedure(Chiang and El-Sheimy, 2004)

Typically many such input/target pairs are used to train a network. Batch training of a network proceeds by making weight and bias changes based on an entire set (batch) of input vectors. Incremental training changes the weights and biases of a network as needed after presentation of each individual input vector. Incremental training is sometimes referred to as "on line" or "adaptive" training.

3.2 Multi-layer Feed-forward NN Architecture

The neuron model and the architecture of a NN describe how a network transforms its input into an output. A network can have several layers. Each layer has a weight matrix *W*, a bias vector *b*, and an output vector *a*. To distinguish between the weight matrices, output vectors, etc. for each of these layers, the number of the layers is appended as a superscript to the variable of interest, as shown in the three-layer network and the equations in Figure 7.

The layers of a multi-layer network play different roles. A layer that produces the network output is called an output layer. All other layers are called hidden layers. The three-layer network shown in Figure 7 has one output layer (layer 3) and two hidden layers (layer 1 and layer 2). The neurons in the hidden layer gather values from all input neurons and pass the net input to an activation function that calculates the output for each neural node.

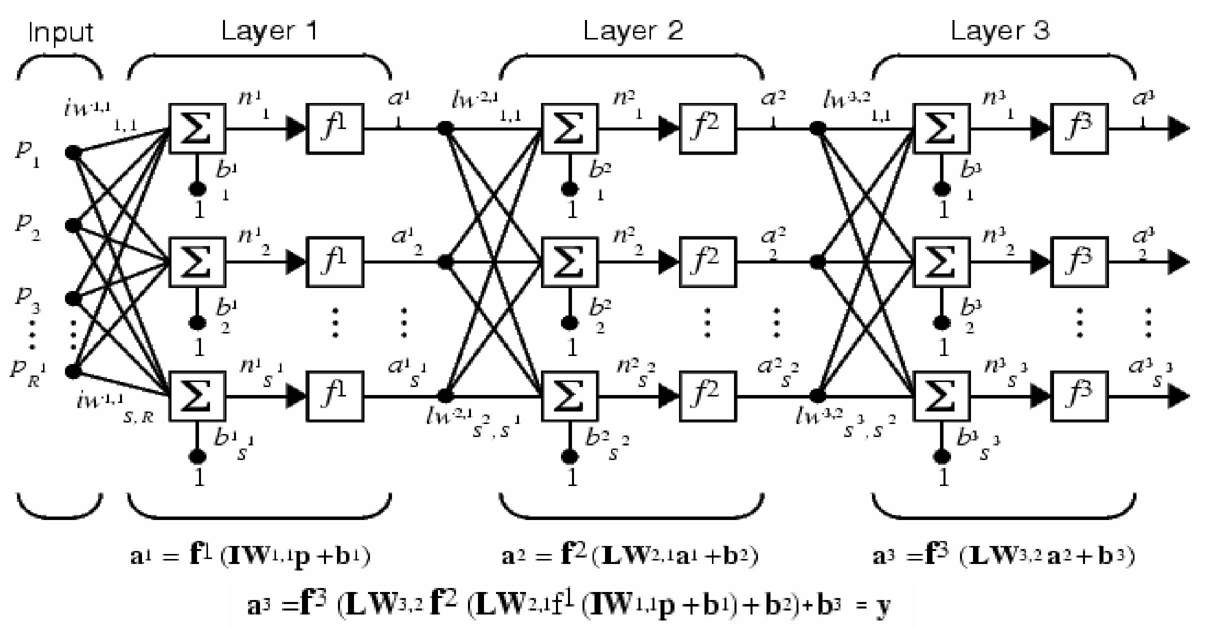

Figure 7. Three layer neural network (Demuth and Beale, 2004)

Multiple-layer networks are quite powerful. For instance, a network of two layers, where the first layer is sigmoid and the second layer is linear, can be trained to approximate any function (with a finite number of discontinuities) arbitrarily well. More details about neuron model and the architecture of NN and can be found in the Matlab Neural Network Toolbox (Demuth and Beale, 2004).

A three-layer feed-forward NN is employed in this approach. The functions of the first and second layers are sigmoid and the third layer is linear. They have 16, 16 and 15 neurons respectively.

5. TEST RESULTS

Field test data were collected for processing. The test system comprises two sets of Leica 530 GPS receiver and one set of C-MIGITS II (DQI-NP) INS system, which gyro and accelerometer bias is 5 deg/hr and 500 μg respectively. Another Micro Tracker GPS receiver is used to synchronize the INS time tagging with GPS time. One of the Leica receivers was set up as a reference station. Another one was the mover receiver with its antenna next to the INS unit. 1 Hz GPS data were saved in GPS receiver PCMCIA card and 100 Hz IMU data were stored in a notebook PC for processing.

The data were processed with the modified AIMS™ software and NN algorithms to test the proposed KF and NN hybrid algorithm for GPS/INS integration. The AIMS™ software was developed by the Center for Mapping at the Ohio State University (OSU) for direct geo-referencing large scale mapping and precise positioning applications (Brzezinska and Toth, 1999). First the data were processed by AIMS™ to generate reference navigation solutions and KF error states data. These data were then processed with the proposed NN algorithm.

5.1 NN Training Results

The NN was trained with an incremental batch method. A small set (batch) of input vectors are applied to train the NN by making its weight and bias changes. Then the next set of input vectors are applied for training. The NN network parameters are changed after each set input vector.

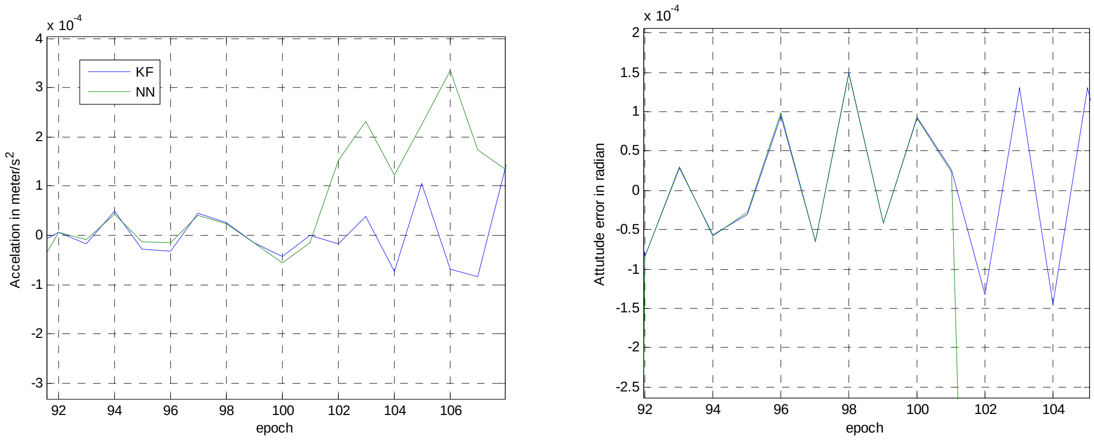

Figure 8. NN training results

The training results of two parameters are shown in Figure 8. The NN output is very close to the target in the training window (epoch 92 to 100 in the figures), but diverges soon after the window. Only one epoch after the window (epoch 101) the output is still similar to the target. This means that NN after training can make reasonable prediction for next GPS epoch, which is useful to improve navigation solutions between GPS/PL sampling but not reliable during GPS signal outages.

5.2 Hybrid Navigation Results

The hybrid navigation results are produced by applying the NN training results at epoch *k+1* to modify the KF predicted navigation results between epochs *k* and *k+1*.

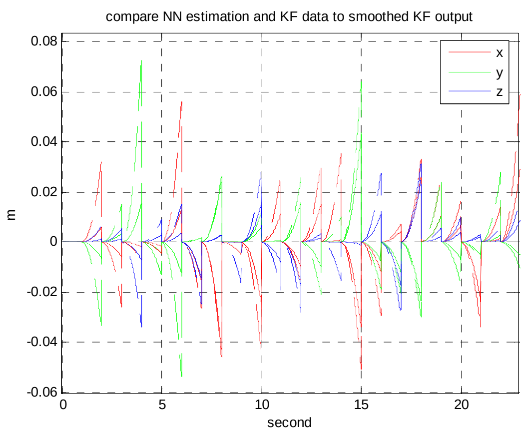

Figure 9. Differences of the positioning results

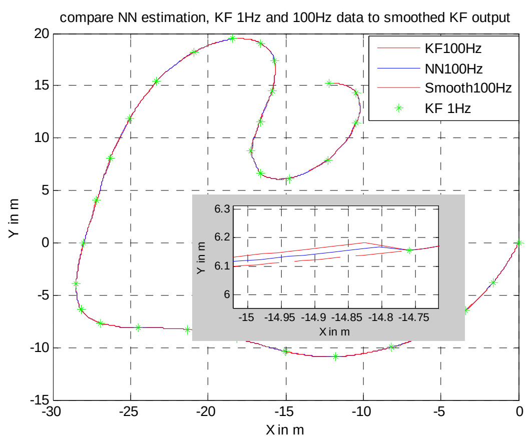

Figure 10. Trajectory comparison

Figure 9 is the comparison results of the difference between the KF output and the near real time BEM smoothed results (dashed line), and the difference between the NN output and BEM smoothed results (solid line). The jumps at the measurement update of the NN outputs are smaller than the direct KF output. Figure 10 is a section of the trajectory with an enlarged segment near a GPS measurement update. The red dashed line is the smoothed trajectory, red solid line is KF solution and blue line is the NN corrected output. The '*' marks the GPS measurement updated points. The test results show that the NN corrected output (blue line) is smoother than the KF output (red solid line).

However, as the NN after training only works well around the training window, its output can only predict one epoch after the window. It is not reliable to correct KF prediction in long term. Further investigation is needed to develop a more effective NN algorithm to improve KF estimates during longer GPS outages.

6. CONCLUDING REMARKS

Primary test results have indicated that there are some relationships between platform dynamic changes during KF measurement update (NN input) and the KF predicted error states (NN output). NN can map these relationships in a short period after training. NN training procedure is processed at the GPS sampling rate, and then properly trained NN can be used to improve the KF prediction at the system output rate (up to IMU sampling rate) in real time. The proposed method can improve navigation solution between the GPS updates. Further research is needed to develop a NN for use during longer GPS outages.

ACKNOWLEDGEMENTS

This research is supported by an ARC (Australian Research Council) research project on 'Integration of GPS/Pseudolite/INS to Geo-reference Airborne Surveying and Mapping Sensors.

REFERENCES

Brzezinska, D.A.G. and Toth, C.K., 1999. AIMS™: An Alternative Tool for Coastal Mapping, Marine Geodesy. International Journal for Marine Geodesy, 22(2): 129-137.

Chiang, K.-W. and El-Sheimy, N., 2004. Performance analysis of a neural network based INS/GPS integration architecture for land vehicle navigation, CD Proceedings of the 4th International Symposium on Mobile Mapping Technology, Kunming, China.

Da, R., Dedes, G. and Shubert, K., 1996. Design and Analysis of a High-Accuracy Airborne GPS/INS System, GPS/GNSS. ION, Kansas City, Missouri, pp. 955-964.

Demuth, H. and Beale, M., 2004. Neural Network Toolbox. The MathWorks, Inc.

El-Rabbany, A. and El-Diasty, M., 2004. An efficient neural network model for de-noising of MEMS-based inertial data. Journal of Navigation(57): 407-415.

El-Sheimy, N. and Abdel-Hamid, W., 2004. An adaptive neuro-fuzzy model to bridge GPS outages in MEMS-INS/GPS land vehicle navigation, GNSS 2004, ION, Long Beach, California, USA.

Goodall, C., El-Sheimy, N. and Chiang, K.-W., 2005. The development of a GPS/MEMS INS integrated system utilizing a hybrid processing architecture, ION GNSS 18th International Technical Meeting of the Satellite Division, Long Beach, CA.

Jwo, D.-J. and Huang, H.-C., 2004. Neural network aided adaptive extended Kalman filtering approach for DGPS positioning. Journal of Navigation: 449-463.

Korniyenko, O.V., Sharawi, M.S. and Aloi, D.N., 2005. Neural network based approach for tuning Kalman filter, IEEE Electro/Information Technology Conference (EIT 2005), Lincoln - Nebraska, May 22-25,, pp. 1-5.

Nassar, S. and Schwarz, K.-P., 2001. Bridging DGPS outages in kinematic applications using a simple algorithm for INS bias modeling, The International Symposium on Kinematic Systems in Geodesy, Geomatics and Navigation (KIS 2001).

Nassar, S., Shin, E.-H., Niu, X. and El-Sheimy, N., 2005. Accurate INS/GPS positioning with different inertial systems using various algorithms for bridging GPS outages, ION GNSS 18th International Technical Meeting of the Satellite Division, pp. 1401-1410.

Wang, J.G., Wang, J., Kearsley, A.H.W. and Lee, H.K., 2004. Pseudolite Augmentation for GPS Aided Aerial Photogrammetry: An Analysis of Systematic Errors. Geomatics Research Australasia(81): 30-44.