GPS Solut (2008) 12:135–146
DOI 10.1007/s10291-007-0073-9

ORIGINAL ARTICLE

An intelligent and autonomous MEMS IMU/GPS integration scheme for low cost land navigation applications

Yun-Wen Huang · Kai-Wei Chiang

Received: 14 February 2007 / Accepted: 4 July 2007 / Published online: 24 July 2007

Abstract An intelligent scheme to integrate inertial navigation system/global positioning system (GPS) is proposed using a constructive neural network (CNN) to overcome the limitations of current schemes, namely Kalman filtering (KF). The proposed CNN technique does not require prior knowledge or empirical trials to implement the proposed architecture since it is able to construct its architecture "on the fly," based on the complexity of the vehicle dynamic variations. The proposed scheme is implemented and tested using Micro-electro-mechanical systems inertial measurement unit data collected in a land-vehicle environment. The performance of the proposed scheme is then compared with the multi-layer feed-forward neural networks (MFNN) and KF- based schemes in terms of positioning accuracy during GPS signal outages. The results are then analyzed and discussed in terms of positioning accuracy and learning time. The preliminary results presented in this article indicate that the positioning accuracy were improved by more than 55% when the MFNN and CNN-based schemes were implemented. In addition, the proposed CNN was able to construct the topology by itself autonomously on the fly and achieve similar prediction performance with less hidden neurons compared to MFNN-based schemes.

Keywords INS · GPS · Kalman filtering · Artificial neural networks

Y.-W. Huang · K.-W. Chiang (✉)
Department of Geomatics,
National Cheng-Kung University,
No.1, Ta-Hsueh Road, Tainan 70101, Taiwan
e-mail: kwchiang@mail.ncku.edu.tw

Abbreviations

ANN	Artificial neural networks
CNN	Constructive neural networks
DGPS	Differential global positioning system
GPS	Global positioning system
IMU	Inertial measurement unit
INS	Inertial navigation system
KF	Kalman filtering
MEMS	Micro-electro-mechanical systems
MFNN	Multi-layer feed-forward neural networks
PUA	Position update architecture
SPP	Single point positioning

Introduction

An integrated navigation system that exploits complementary navigation sensors such as global positioning system (GPS) and inertial navigation system (INS) provide a navigation module that has superior performance in comparison with either a stand-alone GPS or INS. Recently, the progress in micro-electro-mechanical sensors (MEMS) technology enables the development of a complete inertial measuring units (IMU) composed of multiple integrated MEMS accelerometers and gyroscopes. However, due to their noisy measurements and poor stability, the performance of current MEMS-based IMUs does not meet the accuracy requirement of many navigation applications at the present time. For general land-vehicle navigation applications, a system that can provide stable and seamless navigation solutions without suffering the impact of signal obstruction, which can be considered the main limitation of GPS based systems, and overcome the time growing positional errors caused by INS based systems is preferred (Chiang et al. 2004).

The Kalman filtering (KF) approach has been widely recognized as the standard optimal estimation tool for current INS/GPS integration schemes (Gelb 1974); however, it does have limitations that have been reported by several researchers (Brown and Hwang 1992; Vanicek and Omerbasic 1999; Chiang et al. 2004; El-Sheimy and Abdel-Hamid 2004). Several alternative integration schemes have been investigated using different approaches such as Artificial Intelligence, particle filtering, wavelet multi-resolution, adaptive KF, and unscented KF (Ojeda and Borenstein 2002; El-Sheimy and Abdel-Hamid 2004).

Chiang and El-Sheimy (2002) and Chiang et al. (2003) first suggested an INS/GPS integration architecture using the multi-layer feed-forward neural networks (MFNN) for fusing data from differential GPS (DGPS) and either navigation grade IMUs or tactical grade IMUs. Moreover, Chiang and El-Sheimy (2004) introduced the idea of developing the conceptual intelligent navigator using artificial neural networks (ANN) for next generation land vehicular navigation and positioning applications. However, Chiang et al. (2004) indicated that the major limitation of applying the MFNN for alternative INS/GPS integration scheme lies in determining the optimal topology, its effect on the learning time during the update procedure and its effect on the prediction accuracy during GPS signal outages. Once the topology is decided, the synaptic weights are adjusted in a network with a fixed topology.

Limiting factors of MFNN

According to Chiang et al. (2004), the complexity of applying the MFNN varies depending on the complexity of the applications. On the other hand, the complexity of ANN depends on its topology, which consists of the number of hidden neurons (size) and hidden layers (depth) (Ham and Kostanic 2001). The MFNN with an optimal topology is expected to have the generalization ability and provide the best approximation accuracy to the unknown model (Haykin 1999). There are many ways to choose the appropriate number of hidden neurons (Bishop 1995). The common principle indicates that the most appropriate number of hidden neurons is application dependent and can only be decided empirically during the early stage of topology design (Mezard and Nadal 1989).

To avoid the limitations of the MFNN, several methods for successively and automatically constructing a neural network during the learning process have been proposed during the last two decades (Alpaydin 1991). In other words, the networks are able to decide the appropriate topology based on the task given without human intervention (Alpaydin 1991). Among them, the constructive neural network (CNN) invented by Fahlman and Lebiere (1990), has attracted the attention mostly. Therefore, the objectives of this article are to: (a) develop an intelligent and autonomous MEMS IMU/GPS integration scheme using the CNN approach and (b) compare the performance of developed scheme with both the MFNN and KF-based schemes in terms of positioning accuracy during GPS signal outages. Table 1 summarizes the theoretical characteristics of the CNN and MFNN architectures.

The implementation of an intelligent and autonomous Ins/Gps integration scheme

Chiang and El-Sheimy (2002) proposed a position update architecture (PUA), as shown in Fig. 1, to fuse the navigation solutions provided by an INS and GPS simultaneously bridging navigation gaps during GPS signal blockages. In PUA, the input neurons process the INS derived velocity $V_{\mathrm{INS}}(t)$ and azimuth $\phi_{\mathrm{INS}}(t)$ from the INS mechanization. The output neurons generate the two dimensional coordinate differences between two consecutive time epochs along the North and East directions $[\delta N(t) \text{ and } \delta E(t)]$.. The desired (i.e., the reference) outputs $[\delta N_{\mathrm{GPS}}(t),\ \delta E_{\mathrm{GPS}}(t)]$ are provided by GPS during signal availability in either DGPS or single point positioning (SPP) mode of operation. Therefore, when GPS is available, $[\delta N_{\mathrm{GPS}}(t),\ \delta E_{\mathrm{GPS}}(t)]$ are used as the system outputs.

In this study, a modified PUA is implemented, as shown in Fig. 2. The topology of MFNN is then replaced by CNN. The CNN architecture starts with a minimal topology, consisting only of the input and output neurons (Frean 1990). The optimal values of the direct input–output synaptic weight links are computed during the training procedure using GPS derived coordinate differences as updates.

The first step of CNN training procedures begins with the minimal topology for the entire training data set until no further improvement can be achieved. During this process, there is no need to back propagate the output

Table 1 Theoretical comparison between CNN and MFNN

	MFNN	CNN
HN	Empirical trials	OTF
HL	Empirical trials	OTF
Flexibility of topology	Fixed	OTF
Speed of learning	Fast using LM algorithm	Very fast using quickprop algorithm

HN number of hidden neuron, *HL* number of hidden layer, *LM* Levenberg–Marquardt learning algorithm, *OTF* on the fly means the time span during learning process

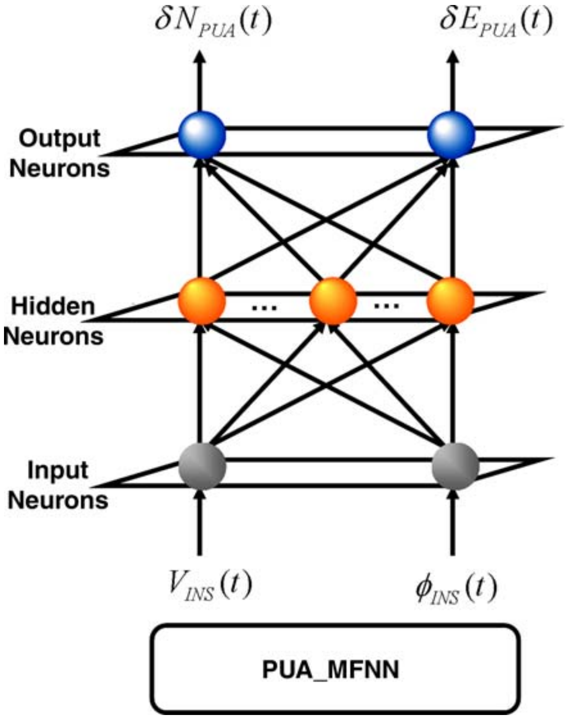


Fig. 1 The topology of MFNN-based PUA

position error (between the network output and the GPS updates) through hidden neurons. In this article, the Quickprop algorithm developed by Fahlman and Lebiere (1990) is implemented due to its simplicity and faster convergence speed (Bishop 1995), as depicted in Fig. 2.

The second step is the recruitment of the first hidden neuron. A pool of candidate neurons that have different sets of randomly initialized weights is applied to reduce the sensitivity of initial weights (Fahlman and Lebiere 1990). The recruitment of the first hidden neuron can be completed in a two-step process. During the first step of recruitment, each candidate neuron is connected to each of the input neurons, but not to the output neurons. All the candidate neurons receive the trainable input connections from the external inputs and from all pre-existing hidden neurons. In addition, all candidate neurons receive the same residual error for each training pattern fed back from the output neurons, as shown in Fig. 2. The weights on connecting the input neurons and candidate neurons are adjusted to maximize the correlation between the output of each candidate neuron and the residual error at the output neuron (see Fig. 3). The pseudo connection shown in Fig. 3 is applied to deliver the error information from the output neurons, but not to forward-propagate the output of candidate neurons. A number of passes over the training data are executed and the inputs of all the candidate neurons are adjusted after each pass.

The goal of this adjustment is to maximize S, the sum over all output neuron (o) of the magnitude of the correlation between (V), the output of a candidate neuron, and (E_o), the residual output error observed at unit (o), as indicated in the following equation.

$$S = \sum_{\mathrm{o}} \left[\sum_{\mathrm{p}} \left(V_\mathrm{p} - \bar{V}\right)\left(E_{\mathrm{p,o}} - \bar{E}_\mathrm{o}\right) \right] \qquad (1)$$

where (o) is the network output at which the error ($E_{\mathrm{p,o}}$) is measured and (p) is the training pattern, the quantities ($\bar{V}$) and ($\bar{E}_\mathrm{o}$) are the values of (V) and (E_o) averaged over all patterns (Fahlman and Lebiere 1990). The (S) indices of all the candidate neurons in the pool are computed simultaneously and the candidate neuron with the highest value of (S) is recruited after all the (S) indices stop improving, as shown in Fig. 4.

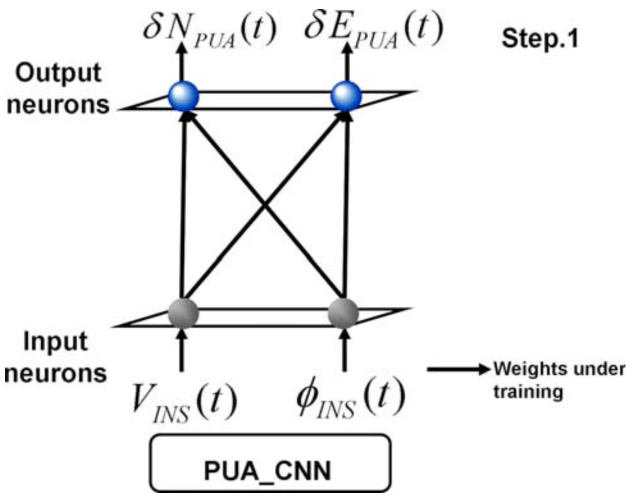


Fig. 2 The initial topology of CNN-based PUA

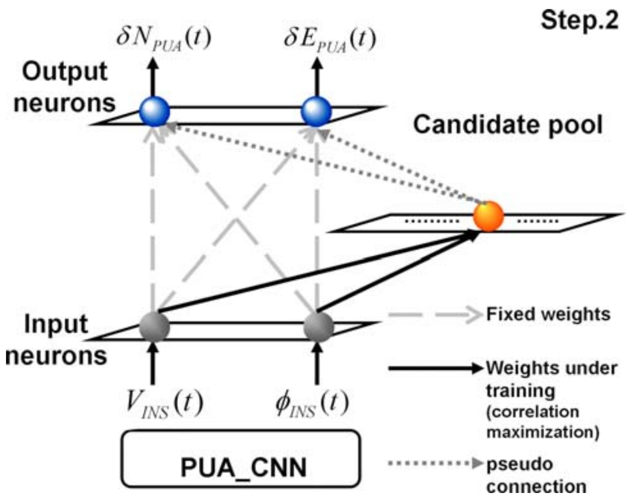


Fig. 3 Step 2, adding first hidden neuron

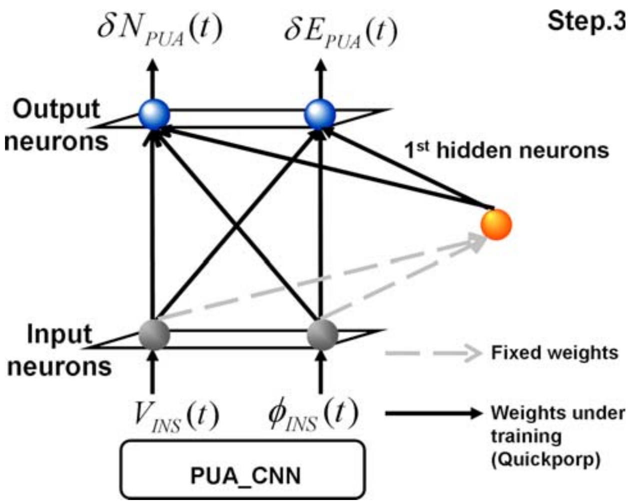


Fig. 4 Step 3, the recruitment of first hidden neuron

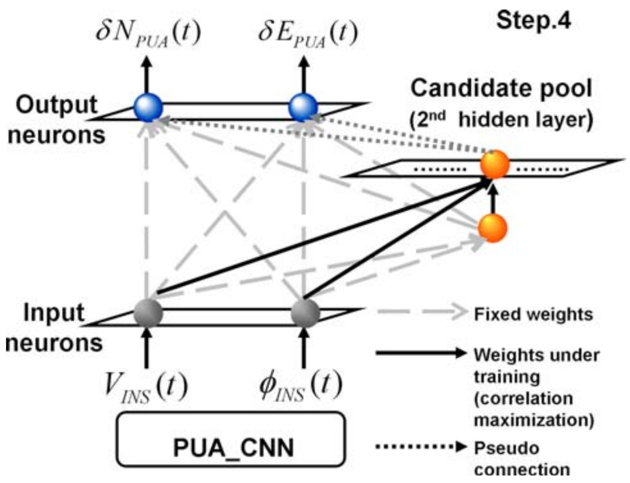


Fig. 5 Step 4, adding second hidden neuron

During the second step of recruitment process shown in Fig. 4, only a single layer of weights are trained. The incoming weights of the winning neurons are frozen and the winner becomes the hidden neuron, which is inserted into the active network in the second step of recruitment. The new hidden neuron is then connected to the output neurons and the weights on connection become adjustable. All connections to the output neurons are trained. In other words, the weights connecting the input neurons and the output neurons are trained again using the quickprop algorithm. On the other hand, the new weights connecting the hidden neurons and output neurons are trained for the first time. Both training steps use the same inputs [$V_{\mathrm{INS}}(t)$ and $\phi_{\mathrm{INS}}(t)$] and outputs [$\delta N(t)$ and $\delta E(t)$], respectively.

The second hidden neuron is then recruited using the same process, as shown in Figs. 5 and 6. This unit receives input signals from both input neurons and previous recruited hidden neurons. All weights connecting input neurons and candidate hidden neurons are adjusted to recruit the second hidden neuron. The values of the weights are then frozen, as soon as the hidden neuron is added to the active network, as illustrated in Fig. 5. All the connections to the output neurons are then established and trained.

The process of recruiting new neurons, training their weights from the input neurons and previously recruited hidden neurons, then freezing the weights and training all connections to the output neurons, is continued until the error reaches the training error goal or the maximum number of epochs. The finalized topology shown in Fig. 7 becomes a modified feed-forward neural network with (n) hidden neurons and (n) hidden layers.

The usage of the CNN for developing an alternative INS/GPS integration scheme poses several advantages over

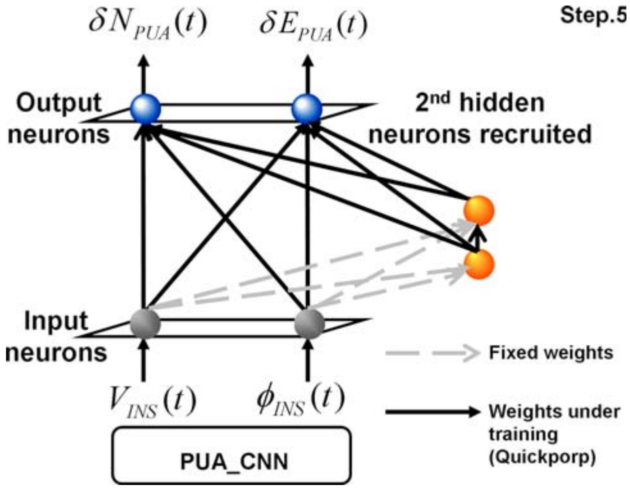


Fig. 6 Step 5, the recruitment of second hidden neuron

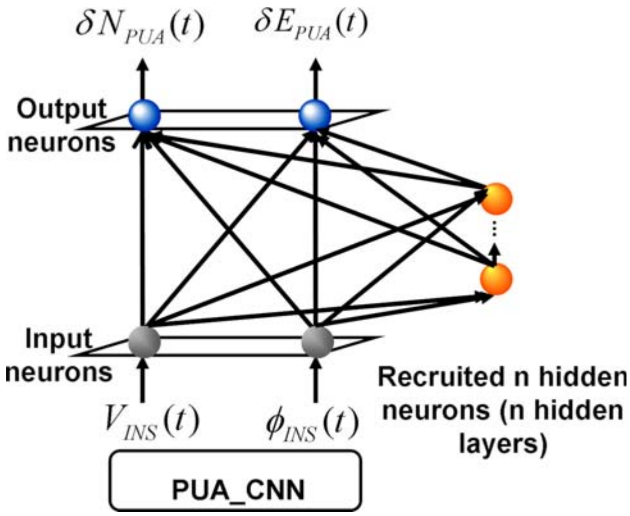


Fig. 7 Finalized CNN-based PUA

the MFNN. First, the best topology can be decided automatically based on the complexity of the applications. There is no need to perform extensive empirical trials to determine the size and depth of the network. Moreover, the learning speed of the CNN is faster than that of MFNN. In the standard back-propagation algorithm, the hidden neurons perform a complex interaction between each other before they settle into a useful role. On the contrary, each of the hidden neurons of the CNN sees a fixed problem and can decisively be used to solve that particular problem. In addition, there is no need to propagate residual error signals through each hidden neuron.

In Fig. 7, since only a single layer of weights is trained (due to the fact that the recruited hidden neurons are treated as additional input neurons), the residual error signal can be delivered to all hidden neurons at the same time. In addition, the CNN is useful for incremental learning, in which new information is added to a previously-trained network. They can reflect the variation of the model complexity by adjusting their weights and topology automatically with aiding information. A segment of the data set (i.e., 120 s) applied in the later section is applied to illustrate the performance of the CNN with MFNN using the PUA mentioned above in theoretical aspects. Figure 8 and Table 2 depict their learning processes and topologies, respectively. The candidate pool contains eight neurons in this case. The goal is to compare the learning speed and accuracy of CNN and MFNN based PUAs.

As indicated in Fig. 8 and Table 2, the CNN applied is able to solve the problem faster and with higher accuracy than the MFNN applied. Figure 8 also illustrates the process of recruiting hidden neurons from the candidate pool. Three hidden neurons are recruited by the CNN to solve this problem; therefore, the CNN is able to construct its topology autonomously with superior performance in terms of the learning time and accuracy.

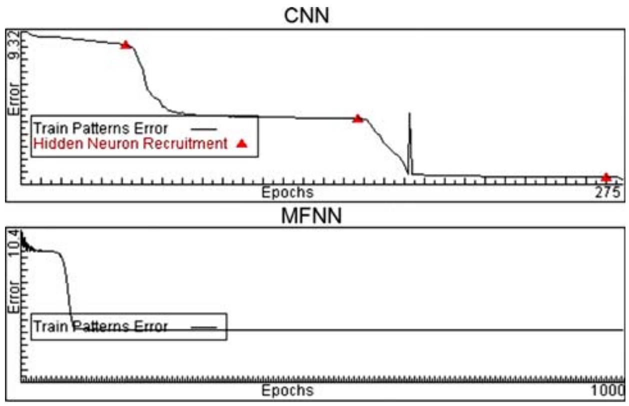


Fig. 8 Learning process of CNN and MFNN

Table 2 Topologies and performance of CNN and MFNN

	MFNN	CNN
Number of inputs	2	2
Number of outputs	2	2
Number of hidden neurons	5 (given)	3 (recruited OTF)
Number of hidden layer	1 (given)	3 (constructed OTF)
Learning time (s)	>60	3.4
Total epoch count	1,000	274
Error value reached	3.7005889	0.4363638

OTF on the fly

According to Fahlman and Lebiere (1990), the fixed topology of the MFNN significantly contributes to this drawback. Consequently, the MFNN lacks the ability to accumulate knowledge incrementally as humans do. In other words, the generalization ability of the MFNN is then limited by its inflexible topology. In contrast, the CNN is able to construct their topologies based on the complexity of the problem encountered and accumulate the knowledge incrementally as humans do. In other words, the CNN has better generalization ability to cope with problems that require flexible topologies.

System implementation

Figure 9 illustrates the INS/GPS system configuration and learning strategy of the proposed integration scheme. Most of the neural network applications apply off-line trained weights to provide prediction of the network output (Haykin 1999). However, in the case of INS/GPS integration for navigation applications, it is required to track directional changes and mimic the motion dynamics using the latest available INS and GPS data. In other words, the synaptic weights should be updated during the navigation process to adapt the network to the latest INS sensor errors and the latest dynamics conditions whenever the GPS signals are available. Therefore, Chiang et al. (2004) proposed a window based weights updating strategy to utilize the synaptic weights obtained during the conventional off-line training procedure (or probably from similar previous navigation missions).

Results and analysis

To verify the performance of the proposed CNN scheme compared to Extended KF, three field tests were conducted in March 2005 using a navigation grade IMU (Honeywell CIMU), the MMSS group MEMS IMU (ADI sensor triad) and two NovATel OEM4 GPS receivers were applied.

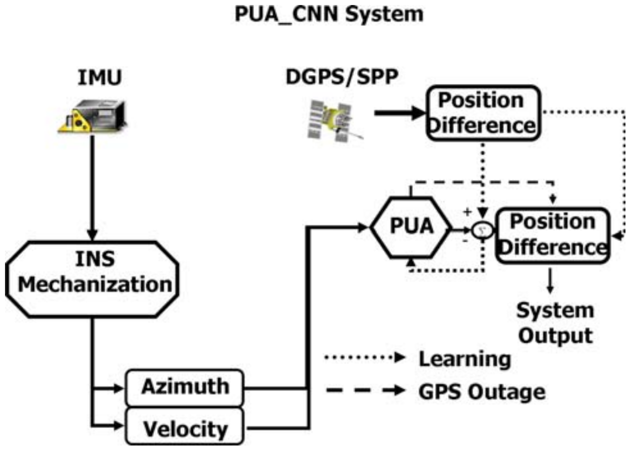


Fig. 9 System configuration

Figure 10a shows the assembly of the sensors used in the field tests.

The reference trajectories were generated with a navigation grade IMU (Honeywell CIMU) and carrier phase DGPS (10 cm level positioning accuracy) with a backward smoothing algorithm. The CIMU can be considered one of the most accurate IMUs available for civil applications. On the other hand, since the proposed algorithm integrates a low cost IMU and GPS in SPP mode, therefore, the reference trajectories used in this study can be considered appropriate. The IMU and GPS measurements obtained in the first and second field tests were used to generate the stored navigation knowledge (training). The third field test trajectory was used as the test trajectory for the proposed scheme.

The first field test is composed of a large loop, see Fig. 10b. The duration of this field test is 1,200 s. The second field test (Fig. 10c) consists of a straight line segment and a large loop. The duration of this field test is 1,000 s.

The stored weights were acquired using measurements obtained through the first two field tests in the traditional off-line training mode. As shown in Fig. 11, 16 simulated GPS outages marked in circles with dot lines from the first and second trajectories were used to acquire stored weights. Four of them were 30 s, two of them were 60 s, and four of them were 120 s. The dynamics variations experienced by the test vehicle during those outages include acceleration/deceleration along straight lines, slow turns and sharp turns. They are considered to be typical maneuvers that could be experienced by land-vehicles.

Furthermore, the window based weights updating strategy was applied to update the navigation knowledge during the availability of the GPS signal. It is worth mentioning that the final topology of the CNN after off-line training consisted of 41 hidden neurons after applying the two training trajectories.

Fig. 10 Sensor assembly and field test trajectories

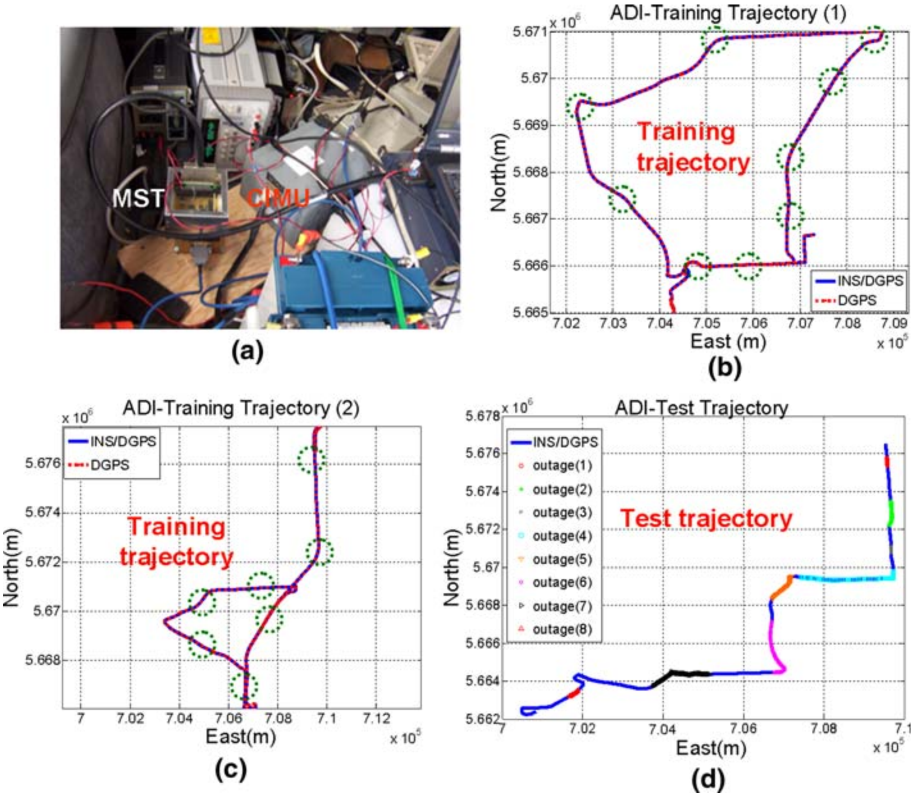
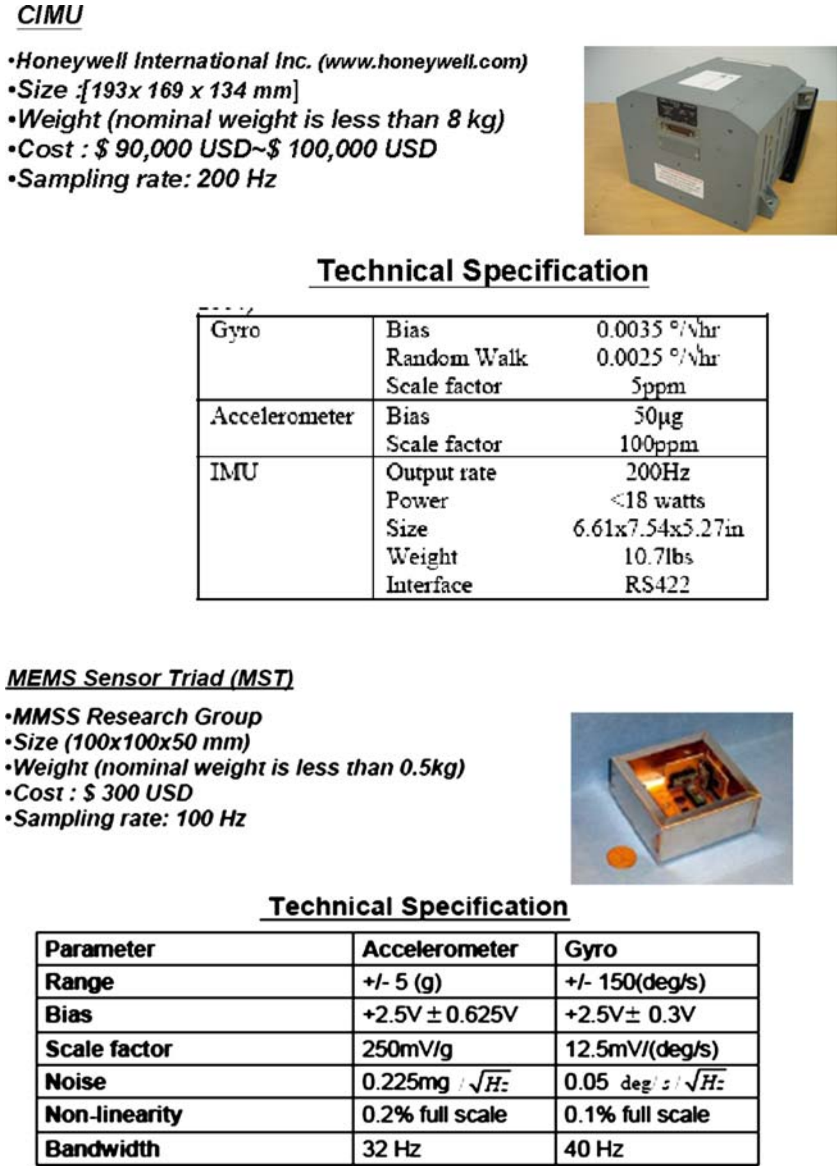


Fig. 11 The specifications of IMUs



CIMU

- Honeywell International Inc. (www.honeywell.com)
- Size :[193x 169 x 134 mm]
- Weight (nominal weight is less than 8 kg)
- Cost : $ 90,000 USD~$ 100,000 USD
- Sampling rate: 200 Hz

Technical Specification

Gyro	Bias Random Walk Scale factor	0.0035 °/√hr 0.0025 °/√hr 5ppm
Accelerometer	Bias Scale factor	50μg 100ppm
IMU	Output rate Power Size Weight Interface	200Hz <18 watts 6.61x7.54x5.27in 10.7lbs RS422

MEMS Sensor Triad (MST)

- MMSS Research Group
- Size (100x100x50 mm)
- Weight (nominal weight is less than 0.5kg)
- Cost : $ 300 USD
- Sampling rate: 100 Hz

Technical Specification

Parameter	Accelerometer	Gyro
Range	+/- 5 (g)	+/- 150(deg/s)
Bias	+2.5V ± 0.625V	+2.5V± 0.3V
Scale factor	250mV/g	12.5mV/(deg/s)
Noise	0.225mg / $\sqrt{Hz}$	0.05 deg / s / $\sqrt{Hz}$
Non-linearity	0.2% full scale	0.1% full scale
Bandwidth	32 Hz	40 Hz

The third field test (shown in Fig. 10d) consists of six straight-line segments and eight sharp turns. The duration of this field test is 1,800 s. Eight actual GPS signal outages, including the impact from intermittent signal reception or no signal reception, took place along this trajectory. The trajectories experienced by the vehicle during these outages are shown in Fig. 10d. In addition, the dynamics experienced by the vehicle during those outages are summarized in Table 3. The test data can be considered independent from those training data sets as there were collected independently in terms of spatial and temporal relationships. In addition, the test GPS outages were caused by true GPS signal obstructions; therefore, screening certain dynamics variations for testing is not possible. The specifications of those IMUs are shown in Fig. 11.

To test the performances, the different schemes were implemented using the CNN and MFNN approaches. The number of hidden neurons and layers of the MFNN were decided empirically. The MFNN was designed to have a nonlinear hidden layer and linear output layer. On the other hand, the topology of the CNN-based scheme was decided automatically and all the hidden neurons were nonlinear (e.g., hyperbolic tangent sigmoid). A segment of the second field test data (1,200 s) was applied first to evaluate the learning performance of CNN and MFNN-based schemes in terms of the required learning time (no. of epochs). The initial weights of each scheme were given randomly and the pool size of candidate hidden neurons was set at four. Figure 12 illustrates the learning time required by both the CNN and MFNN-based schemes, respectively.

Table 3 The dynamic variations during test GPS outages

	1	2	3	4	5	6	7	8
Length (s)	30	60	30	120	60	120	120	60
Motion	Straight line (south)	Straight line (south)	Straight line (south)	Sharp turn and straight line (west)	Smooth turn	Straight line (south) and sharp turn	Straight line and sharp turn (west)	Straight line (south-west)
Distance (m)	612	1,260	585	1,800	720	1,750	1,835	1,220

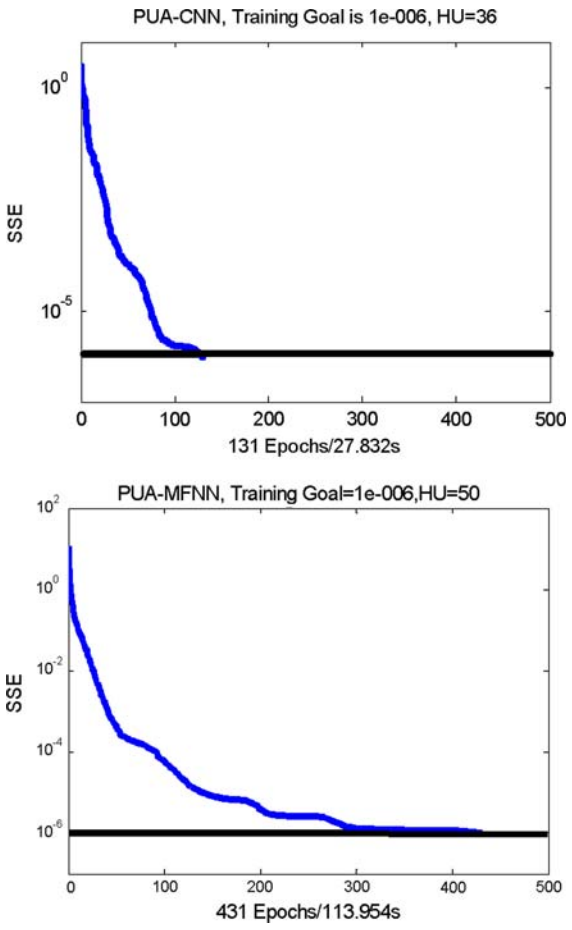


Fig. 12 Learning speed comparison

Figure 12 clearly indicates that the CNN-based scheme converged twice as fast when compared to the MFNN-based scheme with the same training dataset and error goal. The number of recruited hidden neurons was 36, which was less than the number of hidden neurons decided empirically for the MFNN-based scheme (50 in this case). Based on the results presented in Fig. 12, the CNN-based scheme is able to reach the same training goal with less training time and a simpler architecture when compared to the MFNN-based scheme. In addition, the CNN-based scheme distinguishes itself from the MFNN-based scheme since it can decide its topology "on the fly" based on vehicle's dynamic variations and presented INS sensor errors. Figures 13 and 14 show the accumulated positional errors for different INS/GPS SPP integration schemes during several GPS signal outages used for testing the proposed scheme. Table 4 illustrates the performance summary of all GPS signal outage periods. Only half of the results are presented in figures (only four outages). However, Table 4 does show all the results of this study.

Figures 13 and 14 clearly show that the KF positional errors increase, as the function of the length of the GPS signal outage period. This time dependent growth is typical for the KF when operating in prediction mode without GPS updates. On the other hand, the positional errors of both PUA methods (MFNN and CNN) behaved differently. In general, if taking into account the relationship between the vehicle's dynamic variations and positional error accumulation, one can see the strong correlation between them (e.g., significant dynamic variations and large positional errors). In other words, the error behaviors of the proposed schemes are affected by the vehicle's maneuvers. Chiang et al. (2004) developed a performance analysis index to study the impact of vehicle's dynamic variations on positional error accumulations of PUA (MFNN) and the results did show similar correlation presented in this study.

As indicated in Table 4, both MFNN and CNN schemes were able to reduce the positional error during most of the GPS signal outage periods. Since CNN is one of the variants of MFNN and both of them were trained with the same training data sets, it is reasonable that both of them exhibited similar performance in position domain. The overall improvements reached 55 and 69%, respectively. Figures 13 and 14 clearly depict that the KF was able to provide better positioning accuracy during the first 5–10 s of GPS signal outage periods in most scenarios by zooming in Figs. 13 and 14 and looking into the positional error accumulation for the first 10 s of each outage. In contrast, both MFNN and CNN schemes outperformed the KF when the length of the outages extended to 20 s or longer.

In contrast, although the MFNN and CNN schemes suffered from minor degradations in terms of positional errors in some scenarios due to the impact of dynamic variations, they were able to reduce the time dependant errors and provide relatively stable solutions. As indicated in Fig. 10d, there were slow but noticeable heading chan-

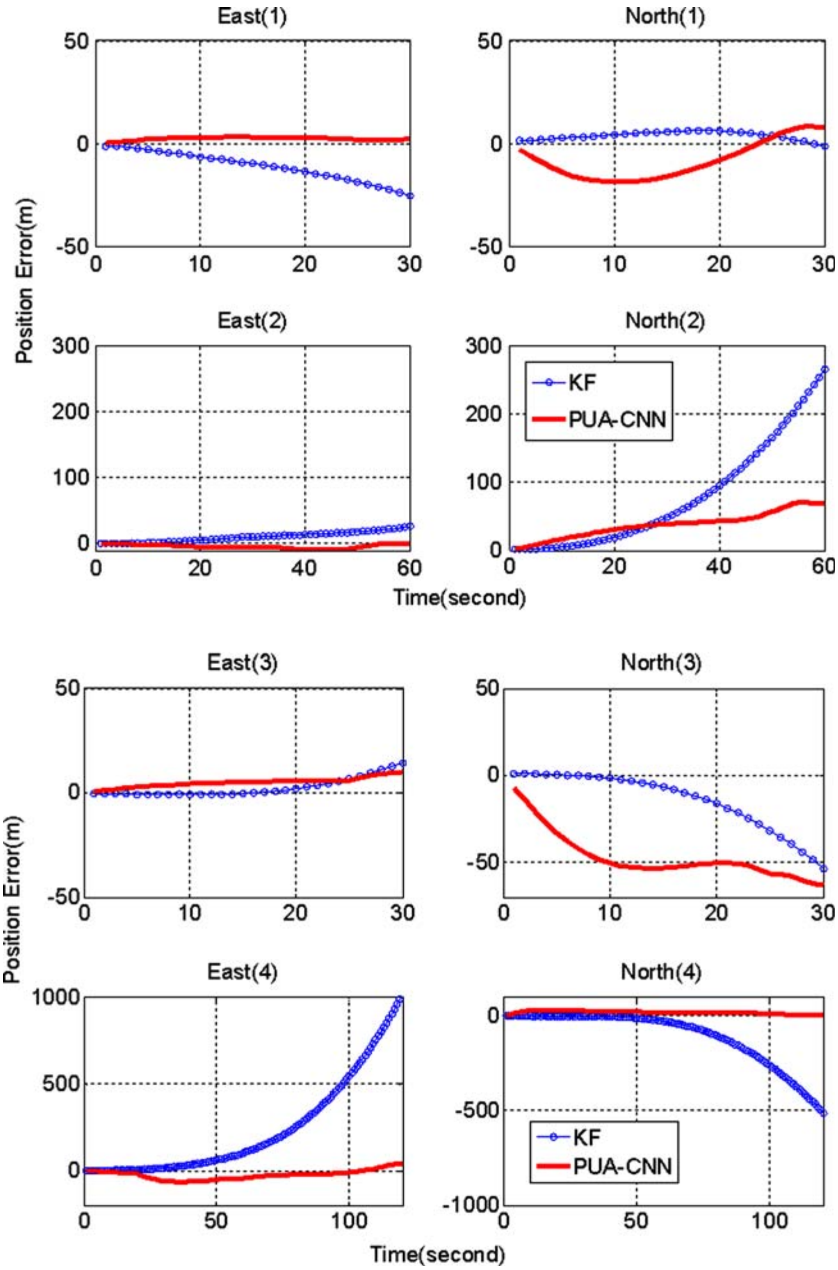


Fig. 13 Positional errors during 1st ~ 4th GPS outages (PUA_CNN)

ges in the middle of the first and second GPS outages. For example, the vehicle first drove southward, then experienced a sharp turn toward the east, then turned southward again during the 1st GPS outage. Similarly, there were rapid heading changes at the beginning of the third GPS outage. Therefore, the impact of sudden heading changes was responsible for the curve shapes shown in Figs. 13 and 14. Comparing Figs. 13 and 14, the performance of the MFNN and CNN-based schemes in the position domain was similar in general. However, the proposed CNN-based scheme was able to construct the topology by itself based on the complexity of the application encountered.

The obtained results indicate that the CNN-based scheme was able to achieve similar performance when the prediction task was applied with less hidden neurons. In addition, the scheme applied was expected to reflect the impact of new information to capture the latest dynamic variations and sensor error variations. In neural networks terminology, such requirements can be fulfilled by continuously learning to adjust the weights and proper variations of the topology when applicable. With this requirement, a self-growing or constructive network such as the CNN used in this study can provide certain advantages over a fixed topology network such as the MFNN.

The proposed schemes are able to form an internal error compensation scheme of the inputs which are contaminated by numerous error sources, including the impact of improper inertial sensor stochastic error models and the lack

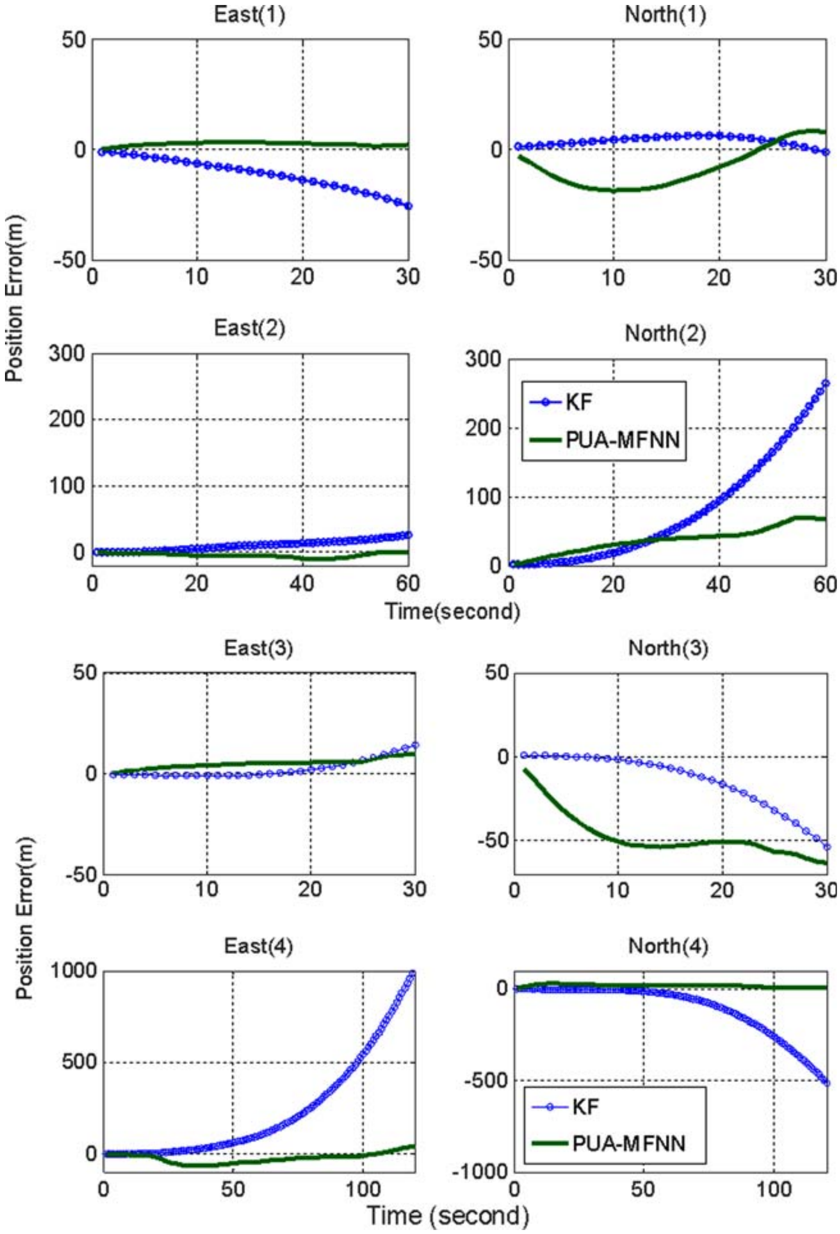


Fig. 14 Positional errors during 1st ~ 4th GPS outages (PUA_MFNN)

of observability by feeding the networks using the desired outputs provided by SPP during the availability of GPS signal. Once the signal blockage occurs, the proposed methods are expected to provide compensation to the inputs and provide corrected outputs with performance similar to the desired outputs during training implicitly.

Conclusions

This study proposed an intelligent and autonomous INS/SPP integration scheme for low-cost MEMS IMUs using the CNN. Compared to fixed topology networks, such as the MFNN, the CNN provide more flexibility. They can develop alternative schemes and decide upon the network topology including the size and depth based on the complexity of the application. The intensive empirical trials needed for the MFNN to decide upon an optimal topology can be avoided. The results presented in this study demonstrate that the KF-based scheme was able to provide better positioning accuracy during the first 5–10 s of GPS signal outage periods in some scenarios when a low-cost MEMS IMU/SPP system was used. However, due to the limitations of bias stability and noisy measurements of MEMS IMUs, the corresponding KF positioning accuracy degraded rapidly with time. On the other hand, when the MFNN or CNN schemes were implemented, the results of KF were improved by more than 55%. Although the per-

Table 4 Positioning errors during GPS signal blockages (PUA_CNN)

Blockage no.	Blockage Length(s)	MAX_N (m)	MAX_E (m)	RMSE Total (m)	Improvement (%) (against KF)
KF (INS/SPP)					
1	30	6.34	25.48	13.76	–
2	60	246.85	25.83	110.54	–
3	30	53.48	14.26	22.23	–
4	120	511.57	1016.6	490.80	–
5	60	85.66	102.17	57.08	–
6	120	1406.6	597.60	559.43	–
7	120	1491.3	1698.1	878.02	–
8	60	39.15	247.66	106.08	–
PUA_CNN (INS/SPP)					
1	30	8.8	7.2	11.37	17.4
2	60	11.23	12.64	16.91	84.7
3	30	20.07	12.56	23.68	–6.52
4	120	35.07	66.35	35.04	92.86
5	60	68.58	15.66	53.43	6.39
6	120	44.48	67.64	89.95	83.92
7	120	34.62	35.97	49.92	94.31
8	60	24.19	29.03	20.63	80.55
PUA_MFNN (INS/SPP)					
1	30	3.60	18.04	18.40	–33.72
2	60	9.80	69.94	42.01	62.00
3	30	9.8	63.10	49.27	–121.64
4	120	32.49	67.46	41.89	91.46
5	60	22.55	135.27	107.36	–88.09
6	120	75.62	47.31	44.45	92.05
7	120	19.80	46.47	19.95	97.73
8	60	31.97	13.78	15.85	85.06

–Left blank intentionally

formance of MFNN and CNN-based schemes in the position domain was similar in general, the proposed CNN-based scheme was able to construct the topology by itself autonomously on the fly. In addition, the CNN-based scheme was able to achieve similar prediction performance with less hidden neurons.

Finally, the INS/GPS integration applications require a more flexible approach to track the dynamic variation of the vehicle and the INS sensor error variations. This is especially true when a low-cost IMU is used. In neural network terminology, such a requirement can be fulfilled by continuously adjusting the weights and topology if applicable. With this requirement, a self-growing or constructive network such as the CNN might be able to provide certain advantages over fixed topology networks such as the MFNN.

Acknowledgments The authors acknowledge the financial support by the National Science Council of the Executive Yun, ROC (Taiwan) (NSC 95-2221-E-006 -335 -MY2). The authors thank Prof. Naser El-Sheimy from the Department of Geomatics Engineering, the University of Calgary for providing the field test data sets applied in this research. AINS toolbox developed by the MMSS group at the Department of Geomatics Engineering, the University of Calgary is used.

References

Alpaydin E (1991) CAL: neural networks that grow when they learn and shrink when they forgot. Technical report-91-032, International Computer Science Institute, Berkeley

Bishop CM (1995) Neural networks for pattern recognition. Oxford University Press, Oxford

Brown RG, Hwang PYC (1992) Introduction to random signals and applied Kalman filtering. Wiley, New York

Chiang KW, El-Sheimy N (2002) INS/GPS integration using neural networks for land vehicle navigation applications. Proceedings of ION GPS 2002 meeting, Portland

Chiang KW, El-Sheimy N (2004) Artificial neural networks in direct georeferencing. Photogramm Eng Remote Sens 70(7):765–768

Chiang KW, Noureldin A, El-Sheimy N (2003) Multi-sensors integration using neuron computing for land vehicle navigation. GPS Solut 6(3):209–218

Chiang KW, El-Sheimy N, Noureldin A (2004) A new weights updating method for neural networks based INS/GPS integration architectures. Meas Sci Technol 15(10):2053–2061

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

Fahlman SE, Lebiere C (1990) The cascade learning architecture. In: Touretzky D (ed) Advances in neural information processing system 2. Morgan Kaufmann, Denver

Frean M (1990) The upstart algorithm: a method for constructive and training feed-forward neural networks. Neural Comput 2:198–209

Gelb A (1974) Applied optimal estimation. MIT Press, Cambridge

Ham FM, Kostanic I (2001) Principles of neurocomputing for science and engineering. McGraw-Hill, New York

Haykin S (1999) Neural networks: a comprehensive foundation, 2nd edn. Prentice-Hall, New Jersey

Mezard M, Nadal JP (1989) Learning in feed-forward layered networks: the tiling algorithm. J Phys A Math Gen 22:2191–2203

Ojeda L, Borenstein J (2002) FLEXnav: fuzzy logic expert rule-based position estimation for mobile robots on rugged terrain. Proceedings of the 2002 IEEE international conference on robotics and automation, Washington

Vanicek P, Omerbasic M (1999) Does a navigation algorithm have to use Kalman filter? Can Aeronaut Space J 45(3):292–296