

A new source difference artificial neural network for enhanced positioning accuracy

This content has been downloaded from IOPscience. Please scroll down to see the full text.

2012 Meas. Sci. Technol. 23 105101

(<http://iopscience.iop.org/0957-0233/23/10/105101>)

View [the table of contents for this issue](#), or go to the [journal homepage](#) for more

Download details:

IP Address: 193.52.245.81

This content was downloaded on 15/05/2017 at 14:54

Please note that [terms and conditions apply](#).

You may also be interested in:

[A new weight updating method for INS/GPS integration architectures based on neural networks](#)

Kai-Wei Chiang, Aboelmagd Noureldin and Naser El-Sheimy

[Bridging GPS outages using neural network estimates of INS position and velocity errors](#)

Lorinda Semeniuk and Aboelmagd Noureldin

[HEPF for integrated inertial navigation and global positioning systems](#)

Priyanka Aggarwal, Zainab Syed and Naser El-Sheimy

[A neuro-wavelet method for multi-sensor system integration for vehicular navigation](#)

Aboelmagd Noureldin, Ahmed Osman and Naser El-Sheimy

[A reliable modeless mobile multi-sensor integration technique based on RLS-lattice](#)

Mahmoud El-Gizawy, Aboelmagd Noureldin and Naser El-Sheimy

[A novel fusion methodology to bridge GPS outages for land vehicle positioning](#)

Wei Chen, Xu Li, Xiang Song et al.

[A new multi-position calibration method for MEMS inertial navigation systems](#)

Z F Syed, P Aggarwal, C Goodall et al.

[Foot mounted inertial system for pedestrian navigation](#)

S Godha and G Lachapelle

A new source difference artificial neural network for enhanced positioning accuracy

Deepak Bhatt¹, Priyanka Aggarwal¹, Vijay Devabhaktuni¹
and Prabir Bhattacharya²

¹ EECS Department, University of Toledo, MS 308, 2801 W. Bancroft St., Toledo, OH 43606, USA

² School of Computing Sciences and Informatics, University of Cincinnati, Cincinnati, OH 45221, USA

E-mail: deepak.bhatt@rockets.utoledo.edu, Priyanka.Aggarwal@utoledo.edu,
Vijay.Devabhaktuni@utoledo.edu and prabir_bhattacharya@yahoo.com

Received 12 February 2012, in final form 11 June 2012

Published 1 August 2012

Online at stacks.iop.org/MST/23/105101

Abstract

Integrated inertial navigation system (INS) and global positioning system (GPS) units provide reliable navigation solution compared to standalone INS or GPS. Traditional Kalman filter-based INS/GPS integration schemes have several inadequacies related to sensor error model and immunity to noise. Alternatively, multi-layer perceptron (MLP) neural networks with three layers have been implemented to improve the position accuracy of the integrated system. However, MLP neural networks show poor accuracy for low-cost INS because of the large inherent sensor errors. For the first time the paper demonstrates the use of knowledge-based source difference artificial neural network (SDANN) to improve navigation performance of low-cost sensor, with or without external aiding sources. Unlike the conventional MLP or artificial neural networks (ANN), the structure of SDANN consists of two MLP neural networks called the coarse model and the difference model. The coarse model learns the input–output data relationship whereas the difference model adds knowledge to the system and fine-tunes the coarse model output by learning the associated training or estimation error. Our proposed SDANN model illustrated a significant improvement in navigation accuracy of up to 81% over conventional MLP. The results demonstrate that the proposed SDANN method is effective for GPS/INS integration schemes using low-cost inertial sensors, with and without GPS.

Keywords: artificial neural network, global positioning system, inertial navigation system, source difference artificial neural network

(Some figures may appear in colour only in the online journal)

1. Introduction

Most of the current land vehicles are equipped with global positioning system (GPS) to provide absolute position and velocity information over extended periods of time, covering any part of the world during day or night [1]. However, standalone GPS receivers do not work in all environments and may experience signal outages or deteriorated performance because of the multipath effects. Inertial navigation technology, on the other hand, overcomes the shortcomings of

GPS by providing continuous position, velocity and attitude information based on the self-contained measurements derived from the inertial sensors [2]. However, its solution accuracy decreases with time because of inherent sensor errors (biases, scale-factor errors, noises and drifts) that may render the uncorrected measurements useless, especially for low-cost sensors. The inertial navigation system (INS) error growth can be limited by utilizing external aiding sources like GPS derived position and velocity data, with bounded errors [1, 3]. Further, INS can bridge GPS signal gaps, assist

in signal reacquisition after an outage and reduce the search domain for detecting and correcting GPS cycle slips [4, 5]. Therefore, an integrated INS/GPS system combines the advantages of both the techniques by reducing INS errors and continuously provides seamless navigation solution [2].

To integrate INS and GPS data, various Bayesian filtering techniques such as the Kalman filter (KF), the extended Kalman filter (EKF) and the particle filter (PF) have been developed and implemented [6–17]. The KF is an optimal filter for linear systems with Gaussian noise, but is not applicable to nonlinear systems [6]. To deal with the system nonlinearity, EKF has been developed, which is based on linearization of the nonlinear models. However, the linearization process is often complicated, time consuming and may cause filter divergence [7, 8]. PF is implemented to overcome these limitations of Kalman filtering approaches by a number of researchers [13–15]. In PF, the posterior distribution is represented by a cluster of random particles rather than a linearized function as in EKF [14]. However, basic PF may require a large number of particles and might not be applicable in real-time applications [16, 17].

To overcome the limitations of the existing Bayesian techniques, artificial intelligence (AI) method such as artificial neural networks (ANN) [18] are developed. ANNs have emerged as intelligent and powerful tools to solve nonlinear systems [19]. The basic principle behind the ANN technique is to mimic the latest vehicle dynamics through appropriate network training during GPS signal availability. In the training procedure, ANN captures the inherent input–output nonlinear relationship relating INS and GPS data. During the period of GPS outages, the trained module estimates reliable navigation solution based on the established nonlinear model. A common supervised neural network is a multi-layer perceptron (MLP) network with three layers (i.e. MLP-3), which has been applied to different grades of INS [20, 21]. However, for low-cost INS units the complexity of the input–output relationship to be modeled is enormous. This affects the neural network learning capability, which leads to degradation of their prediction accuracy during outages [21]. Therefore, the MLP-3 neural network exhibits poor solution accuracy when used to integrate low-cost microelectromechanical system (MEMS), INS and GPS data, where INS sensor errors (like turn-on to turn-on biases, in-run biases and scale factor drifts) are more prominent and difficult to model. The limited performance of the conventional ANN can be improved by a more sophisticated ANN referred to as the knowledge-based neural network (KBNN) [22, 23], which has excellent extrapolation capabilities. KBNNs can be used to efficiently model systems with a relatively higher degree of nonlinearity as exhibited by MEMS INS units, where conventional MLP networks typically fail.

For the first time, in this paper a particular type of KBNN called the source difference artificial neural network (SDANN) is implemented for GPS/INS integration in order to improve prediction accuracy of neural network models. It consists of the existing neural network, also called the coarse model (i.e. MLP-3 neural network) and an additional neural network (called the difference model). This difference model enhances the overall prediction accuracy by embedding

additional knowledge about the coarse model estimation error into the system.

The paper is organized into the following sections. Section 2 gives an introduction to the ANN modeling process. Section 3 provides a brief review of the different ANN-based architectures, namely: position update architecture (PUA) and position and velocity update architecture (PVUA). Section 4 explains the detailed implementation of the proposed SDANN algorithm. Section 5 presents the results of the SDANN model and its comparison with the existing ANN techniques while section 6 presents the concluding remarks.

2. Artificial neural network

In recent years, ANNs have emerged as a powerful tool capable of modeling any nonlinear complexity with ease [19–23]. The individual units (called neurons) are connected through the communication channels along with their associated weights. For supervised ANNs, the network parameters, like weights and biases, are adjusted through the combination of input and error signals, where the error is the difference between the desired output and the current network output. Primitive ANNs contained one input layer and one output layer. Then, multi-layer feed-forward networks, like MLP networks, evolved to capture the inherent functional relationship between the input and the output. MLP constitutes of input, hidden and output layers and can have more than one hidden layer. Generally, MLP networks have one hidden computational layer, according to the universal approximation theorem [19]. This theorem states that standard multilayer feed-forward networks with a single hidden layer containing several hidden neurons are universal approximators [19] and can proficiently model any nonlinearity. Experience helps in selecting the number of hidden neurons in the hidden layer while training data decide the neurons in the input and output layer. Using too few hidden neurons leads to under-learning, while using too many hidden neurons leads to over-learning [19]. Under-learning occurs when the network is unable to properly learn the training data and gives large training errors (difference between the network and the desired output). On the other hand, over-learning is a situation where the network memorizes the training data but fails to generalize the relationship.

Now consider the case where INS and GPS data needs to be fused together. Generally, GPS data form the output layer as these provide accurate solutions over extended periods of time while INS data form the input layer. Let x_1, x_2, \dots, x_n be the model/problem inputs and y the desired model output:

$$y = f(\mathbf{x}), \quad (1)$$

where $\mathbf{x} = [x_1, x_2, \dots, x_n]^T$ represents the vector of all inputs, n is the total number of inputs while f represents the relationship between \mathbf{x} and y . The MLP-3 network is trained to derive a representative ANN model (f_{ANN}) that closely emulates f when provided with appropriate training data as given in equation (2) by iteratively adjusting the neuron weights \mathbf{w} :

$$\{(\mathbf{x}_p, \mathbf{d}_p), p = 1, 2, \dots, N\}, \quad (2)$$

where N is the number of training samples.

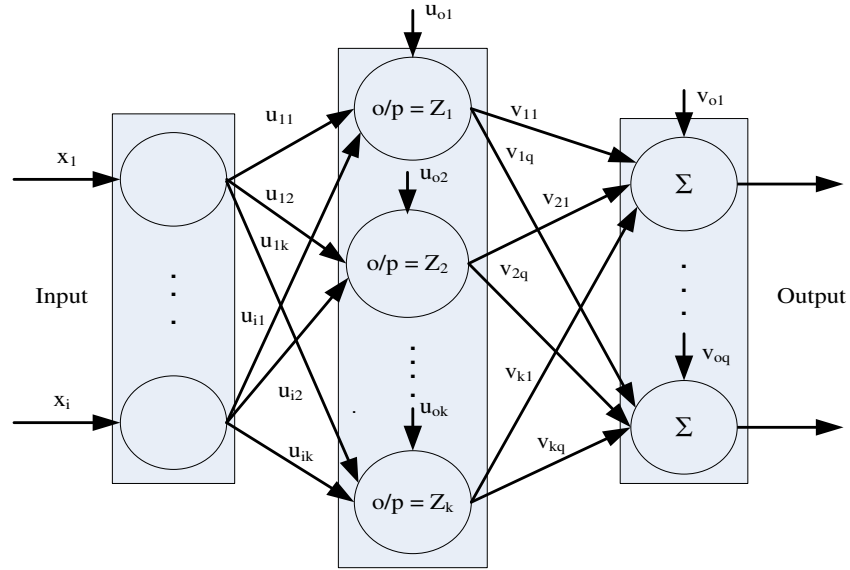


Figure 1. Multi-layer perceptron network.

The main focus of this iterative ANN training is to determine the weight vector \mathbf{w} such that the difference between model outputs and the desired output is minimized as given in the following equation:

$$E(\mathbf{w}) = \frac{1}{2} \sum_{p=1}^N \sum_{q=1}^{N_y} (y_{pq} - \hat{y}_{pq})^2, \quad (3)$$

where N_y represents the number of output layer neurons and y_{pq} and \hat{y}_{pq} are the desired and the predicted output of the modeled neural network. In MLP neural network, \hat{y}_{pq} is computed using equation (4) where v_{oq} is the bias weight parameter of the q th neuron in the output layer, v_{kq} represents the weight of the link between the k th neuron in the hidden layer (with m hidden neurons) and the q th neuron while z_k represents the output of the k th hidden layer neuron given by equation (5).

$$\hat{y}_{pq} = \sum_{k=1}^m (z_k \times v_{kq}) + v_{oq}, \quad (4)$$

$$z_k = \sigma \left(\sum_{i=1}^n x_i \times u_{ik} + u_{ok} \right). \quad (5)$$

In equation (5), u_{ik} represents the weight of the link between the i th input neuron (out of a total of n input neurons) and the k th hidden neuron, x_i is the i th input neuron, u_{ok} is the bias parameter for the k th hidden neuron and $\sigma(\cdot)$ is a hidden neuron activation function as illustrated in figure 1.

3. Overview of ANN-based INS/GPS integration algorithms

ANN forms a parallel data processing system that allows the modeling of highly complex and stochastic problems with elevated levels of noise, such as correlated noise, bias instability, white noise, random walks, etc, which cannot be solved by conventional algorithmic approaches [21–27]. The conventional PUA and PVUA utilizing MLP-3 neural network have previously been applied to integrate INS and GPS data [21, 28] as explained in the following sections.

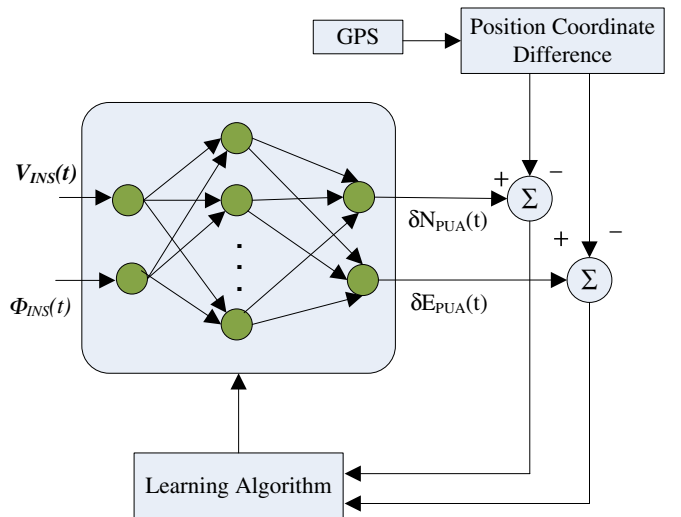


Figure 2. The PUA MLP network.

3.1. Position update architecture

Position update architecture (PUA) consists of a MLP-3 neural network to update the position components along the North and East directions as shown in figure 2 [21, 28, 29]. Each of the three layers (i.e. input layer, hidden layer and output layer) utilizes different activation function (i.e. relay, nonlinear functions like sigmoid and linear functions). The PUA network processes the INS velocity $\{V_{INS}(t)\}$ and azimuth $\{\Phi_{INS}(t)\}$ to produce the position coordinate differences between two consecutive epochs at the output. Position differences are used instead of the position components to simplify the training process. The magnitudes of position differences are usually less than 200 m and hence the relationship between inputs (INS velocity and azimuth) and output (position differences) is easier to learn [28]. The PUA output is compared with the desired consecutive position coordinate differences taken from the GPS as shown in figure 2 and the weights are adjusted accordingly to reduce the mean square error as given

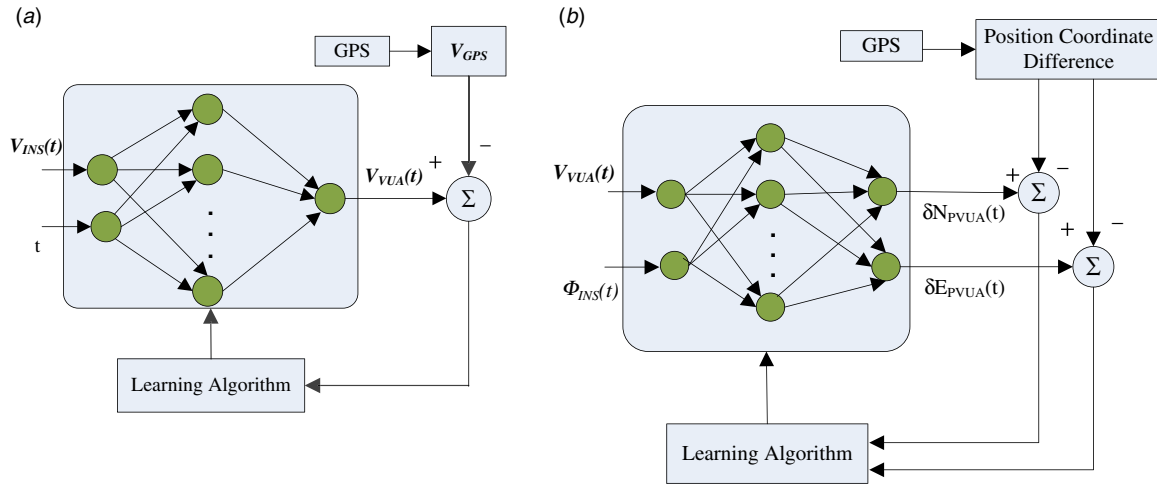


Figure 3. (a) Velocity update architecture and (b) position update architecture.

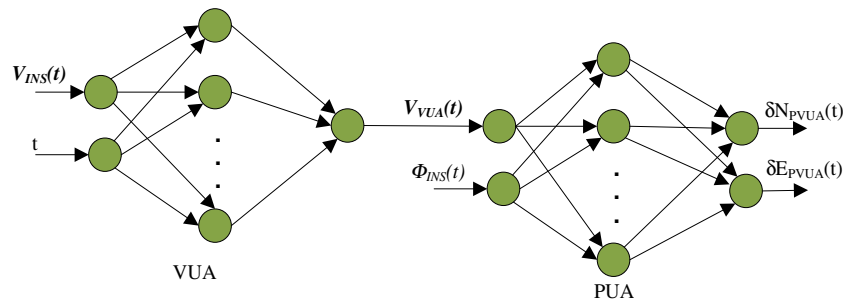


Figure 4. MLP-3 neural network of PVUA in prediction phase.

in equation (3). This iterative process of comparison and adjustments of the neural network weights continues as long as GPS signals are available, i.e. during the training module. During GPS outages, the trained neural network utilizes INS derived velocity and azimuth components to predict the position coordinates {i.e. $\delta N_{PVUA}(t)$ and $\delta E_{PVUA}(t)$ } and correct INS errors.

3.2. Position and velocity update architecture

Position and velocity update architecture (PVUA) (or modified PUA) contains two processing architectures: velocity update architecture (VUA) and PUA each utilizing MLP-3 neural network as shown in figure 3 [21, 28, 29]. The PUA and VUA are trained during GPS availability so as to adapt to the latest vehicle dynamics. The VUA network utilizes $V_{INS}(t)$ and time t as inputs and produces the updated velocity $V_{VUA}(t)$. During the training stage, the VUA network parameters are continuously adjusted by comparing its output with the GPS velocity as illustrated in figure 3(a). The PUA of PVUA utilizes VUA updated velocity $V_{VUA}(t)$ and $\Phi_{INS}(t)$ as inputs with position coordinate differences as the outputs (figure 3(b)), like the PUA model illustrated in subsection 3.1. During any GPS outage, the training stops and the trained PUA processes the trained VUA model velocity components along with INS azimuth to predict the position difference components (figure 4).

The implementation of PUA and PVUA utilizing conventional ANN is summarized in algorithm 1.

Algorithm 1: ANN implementation for PUA and PVUA

Step 1: Identify the integration architecture along with inputs and the desired outputs;

Step 2: Train MLP-3 neural network using the inputs and outputs identified in step 1;

Step 3: Compare the ANN model output with the desired value to obtain the estimation error;

Step 4: Optimally adjust the network parameters to reduce the estimation error;

Step 5: Repeat steps 2–4 until a point where GPS outage occurs; and

Step 6: Utilize the trained model to obtain the navigational information during GPS outage.

The existing PUA and PVUA utilizing MLP-3 neural networks show limited performance in the case of low-cost INS. This limited performance is mainly due to the increased complexity of the input–output functional relationship [30]. However, to effectively model such a highly nonlinear relationship and to improve the neural network prediction accuracy, a knowledge-based neural network has been proposed in this paper. The study utilizes a particular type of knowledge-based model to improve the accuracy of the existing PUA and PVUA architectures as discussed next.

4. Proposed SDANN methodology

The source difference artificial neural network model enhances the prediction accuracy of the existing conventional ANN

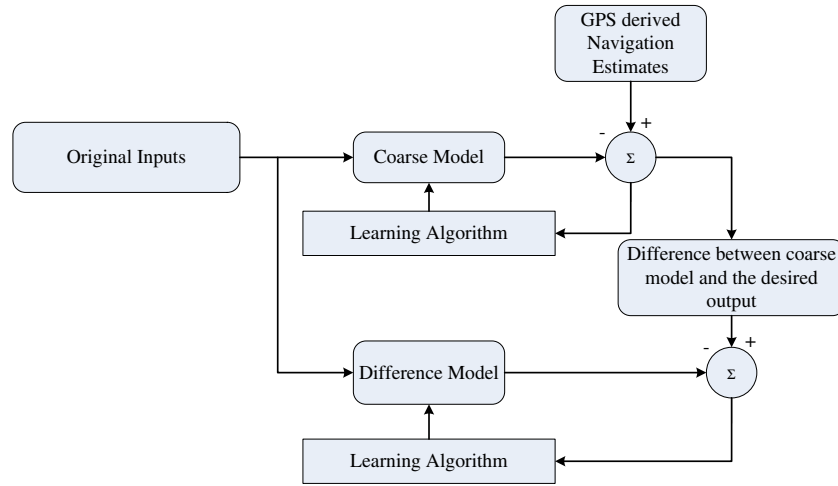


Figure 5. Source difference based training model.

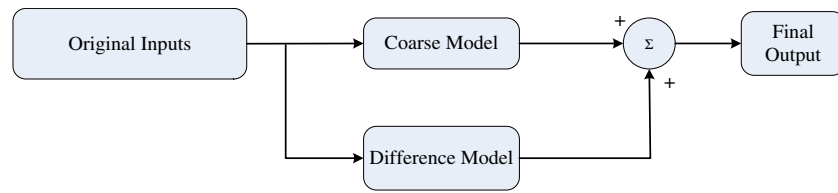


Figure 6. Source difference based prediction model.

model by incorporating additional knowledge about the coarse model estimation error [22, 23]. The difference model embeds this additional knowledge into the system thereby further mitigating the inertial sensor errors. In a recent study [22], it has been shown that compensating for the estimation errors associated with the coarse model significantly enhances the solution accuracy. Hence, for the first time, SDANN models of PUA and PVUA architectures are implemented as an alternate INS/GPS fusion method for very low-cost MEMS inertial measurement units (IMUs).

4.1. Source difference artificial neural network method

The source difference method proposed by Watson and Gupta [23] consists of two MLP-3 neural networks, a coarse model which is considered as an approximate model and a difference model that represents the difference between the actual data and approximate model output. The coarse model represents the original modeling problem as given by equation (1) and is trained using an input–output sample pair as defined in equation (2). However, to start with difference model training, the best coarse model needs to be identified and is explained below.

Let us assume that for an input–output functional relationship modeling problem defined in equation (1), we train the different coarse models with varying number of hidden layer neurons defined as

$$\begin{aligned} \mathbf{y}_{\text{coarse},1} &= f_1(\mathbf{x}, w_1) \\ \mathbf{y}_{\text{coarse},2} &= f_2(\mathbf{x}, w_2) \\ &\vdots \\ \mathbf{y}_{\text{coarse},n} &= f_n(\mathbf{x}, w_n). \end{aligned} \quad (6)$$

In the above equation, $\mathbf{y}_{\text{coarse},i}$ represents the i th coarse model output when presented with input vector \mathbf{x} , where $1 \leq i \leq n$. Among a pool of different coarse model candidates defined in equation (6) with different weights, the best coarse model is identified as the one having least validation error. Once the optimal coarse model is identified, along with the number of hidden neurons, the difference model is trained utilizing the same input vector \mathbf{x} and the desired output vector evaluated as

$$\mathbf{d}_{\text{diff}} = \mathbf{d}_{\text{coarse}} - \mathbf{y}_{\text{coarse}}. \quad (7)$$

In equation (7), $\mathbf{y}_{\text{coarse}}$ represents the best coarse model output when presented with input vector \mathbf{x} , and $\mathbf{d}_{\text{coarse}}$ represents the corresponding desired output vector. Similar to the coarse model the best difference model is identified. The trained difference model helps in fine tuning the coarse model output and thus gives better performance. The coarse and the difference models are trained during GPS availability using suitable input and output data such that their mean square training error (as shown in figure 5) is minimized. This study proposes a quasi-Newton learning algorithm to minimize the error function with respect to the parameters of the network because of its faster convergence ability [31]. For details related to quasi-Newton please refer [31, 32].

Once the training stops, i.e. GPS outages occur, the final output of the SDANN model is predicted by summing the coarse and the difference model output as given by equation (8) and illustrated in figure 6:

$$\mathbf{y}_{\text{final}} = \mathbf{y}_{\text{coarse}} + \mathbf{y}_{\text{diff}}. \quad (8)$$

The detailed implementation of the proposed technique for two architectures, namely PUA and PVUA, is as follows.

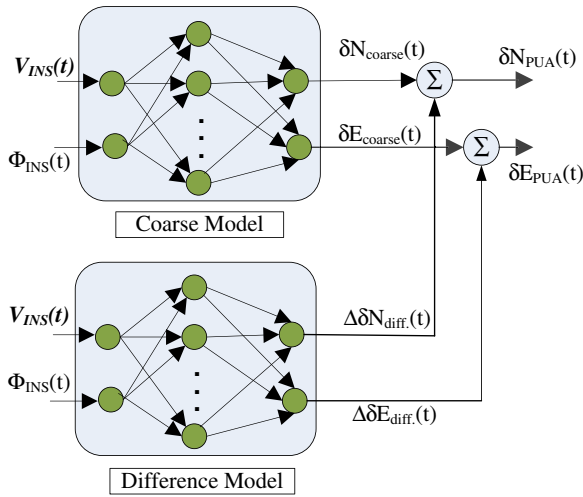


Figure 7. Proposed SDANN-based PUA approach.

4.2. SDANN-based PUA architecture

Position Update: The PUA based on the SDANN model consists of two MLP-3 neural networks as shown in figure 7.

In figure 7, δ depicts the difference in position coordinates between two consecutive epochs/time-points and Δ represents the difference between coarse model outputs and desired outputs. The detailed implementation of the proposed PUA architecture based on SDANN model is given by algorithm 2.

Algorithm 2: Proposed SDANN based PUA training procedure

Step 1: Identify the input and output for neural network training according to the PUA model; **Repeat**

Step 2: Train the coarse model with INS velocity $\{V_{INS}(t)\}$ and the azimuth angle $\{\Phi_{INS}(t)\}$ as inputs. Desired outputs are GPS position coordinates difference between two consecutive epochs $\{\delta N_{GPS}(t), \delta E_{GPS}(t)\}$;

Step 3: Evaluate the coarse model outputs $\{\delta N_{coarse}(t), \delta E_{coarse}(t)\}$ along North and East directions;

Step 4: Evaluate the estimation error i.e., difference between the GPS position coordinates $\{\delta N_{GPS}(t), \delta E_{GPS}(t)\}$ and the coarse model outputs $\{\delta N_{coarse}(t), \delta E_{coarse}(t)\}$;

Step 5: Train the difference model using INS velocity $\{V_{INS}(t)$, and $\Phi_{INS}(t)\}$ as inputs and the estimation error evaluated in step 4 as the output;

Until GPS outage occurs; and Final output is the sum of the coarse and the difference model.

As explained, the objective of training is to estimate the network parameter in order to reduce the minimum square training error. The training procedure outlined by algorithm 2 is adopted during GPS availability. In the case of GPS outages the output is the trained PUA model output $\{\delta N_{PUA}(t), \delta E_{PUA}(t)\}$ which is evaluated as the sum of the coarse and the difference model outputs, given as

$$\delta N_{PUA}(t) = \delta N_{coarse}(t) + \Delta \delta N_{diff}(t), \quad (9)$$

and

$$\delta E_{PUA}(t) = \delta E_{coarse}(t) + \Delta \delta E_{diff}(t). \quad (10)$$

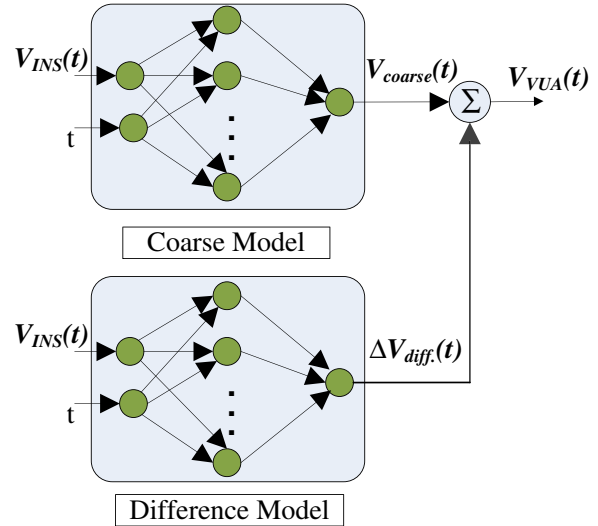


Figure 8. Velocity update strategy utilizing SDANN model.

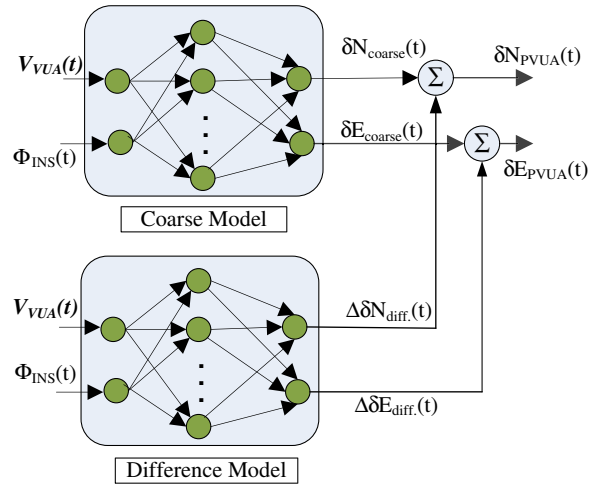


Figure 9. Position update of PVUA based on SDANN model during training.

4.3. SDANN-based PVUA model

Implementation of the proposed PVUA is done at two different levels, at the velocity update (figure 8) and position update (figure 9) stages. Each of the two updates utilizes SDANN model.

Velocity update: The training in this architecture includes the algorithm 3 steps given below.

Algorithm 3: Proposed SDANN based PVUA training procedure

Step 1: Identify the input and output for neural network training according to the VUA model; **Repeat**

Step 2: Train the coarse model using INS velocity $\{V_{INS}(t)\}$, and time $\{t\}$ as inputs and GPS velocity $\{V_{GPS}(t)\}$ as the desired output;

Step 3: Evaluate the trained coarse model output $\{V_{coarse}(t)\}$;

Step 4: Calculate the estimation error between GPS velocity $\{V_{GPS}(t)\}$ and coarse model output $\{V_{coarse}(t)\}$; and



Figure 10. The field test trajectory used to assess performance of the proposed SDANN approach.

Step 5: Train the difference model with the inputs $\{V_{INS}(t)\}$ and time $\{t\}$ and the estimation error evaluated in step 4 as the output;

Until GPS outage occurs; and

Final output is the sum of the coarse and the difference model.

The final VUA updated velocity is calculated as

$$V_{VUA}(t) = V_{coarse}(t) + \Delta V_{diff}(t), \quad (11)$$

Position update: The training of the position update of PVUA is similar to that of the PUA. However, it utilizes a different input vector, i.e. the VUA network velocity $\{V_{VUA}(t)\}$ and INS azimuth $\{\Phi_{INS}(t)\}$, to estimate the GPS position coordinate between two consecutive epochs. During GPS outages, the trained PUA network processes the VUA updated velocity and $\Phi_{INS}(t)$ as inputs to predict the positional coordinates (see figure 9).

The validity of the proposed SDANN method is tested by real-field test data obtained using a MEMS grade unit, high grade IMU and differential GPS (DGPS) solution under both GPS outages and no outages conditions.

5. Results

In this paper, the performance of the SDANN method is examined using two different architectures, PUA and PVUA, for real-field test data. A CAD tool (i.e. NeuroModeler, Department of Electronics, Carleton University, Ottawa, Canada) is used for training the neural network [33]. The field test data were collected by installing various equipment in a test vehicle [30]. This equipment includes a Crossbow IMU 300CC-100, reference high grade IMU by Honeywell (HG 1700), Novatel OEM GPS receivers and computer. The reference solution is obtained by combining Honeywell IMU with the DGPS solution. The specifications of the Crossbow IMU and HG1700 are provided in table 1 where the data were

Table 1. Characteristics of crossbow IMU and HG 1700.

	Crossbow IMU 300CC	HG 1700
Gyroscope		
Bias	$< \pm 2.0^\circ \text{ s}^{-1}$	1.0° h^{-1}
Scale factor	$< 1\%$	150 ppm
Random walk	$< 2.25^\circ \text{ h}^{-1/2}$	$0.12^\circ \text{ h}^{-1/2}$
Accelerometer		
Bias	$\pm 30.0 \text{ mg}$	1.0 mg
Scale factor	$< 1\%$	300 ppm
Random walk	$< 0.15 \text{ m s}^{-1} \text{ h}^{-1/2}$	$0.019 \text{ m s}^{-1} \text{ h}^{-1/2}$

collected at 100 Hz [35]. A GPS receiver provides position and velocity information at 1 Hz and the ANN model is trained at 1 Hz for the trajectory of 52 min duration, spanning over a distance of 22 km (figure 10).

The test trajectory included all the real-life scenarios encountered by a typical land vehicle which includes high speed highway section, suburban roads with hills, trees and winding turns, urban streets with frequent stops and sudden vehicle accelerations/decelerations. Throughout the test, a minimum of seven satellites were visible, except for several short natural GPS signal outages. For testing purposes, several short and long GPS outages (figure 10) are intentionally picked to compare the performance of the proposed SDANN algorithm under diverse conditions such as straight portions, turns, slopes, high speed, slow speeds and stops. Typically, short GPS outages correspond to 10–50 s, while long GPS outages correspond to 200 s for navigational grade INS and up to 50–100 s for tactical grade INS [35]. As such, in our case of low-cost INS, outages up to 100 s are considerably long. In this paper, the first outage of 30 s duration occurs when the vehicle is stationary, the second outage of 70 s during a winding curve, and the third outage of 90 s during straight line motion. All the three outages of varying duration (i.e. 30, 70 and 90 s) in this study were intentionally inserted as a check of the proposed system's accuracy over conventional ANN model.

Table 2. Comparison of position errors for PUA based on SDANN and the conventional ANN.

PUA	North position error				East position error			
	MAE (m)		RMSE (m)		MAE (m)		RMSE (m)	
	ANN	SDANN	ANN	SDANN	ANN	SDANN	ANN	SDANN
Outage 1 (30 s)	17.20	15.20	20.80	18.76	5.69	3.95	6.17	4.47
Outage 2 (70 s)	76.25	74.15	83.25	82.31	132.70	67.68	159.08	80.27
Outage 3 (90 s)	153.58	89.30	181.51	103.74	147.95	123.31	168.05	140.76

The position predicted by the SDANN approach during these outages is compared with the reference solution to calculate mean absolute error (MAE) and root mean square error (RMSE) as

$$\left| \text{MAE} = \frac{\sum_{p=1}^N |\hat{y}_p(x_i, w) - y_p|}{N} \right|_{N=30,70 \text{ and } 90}, \quad (12)$$

and

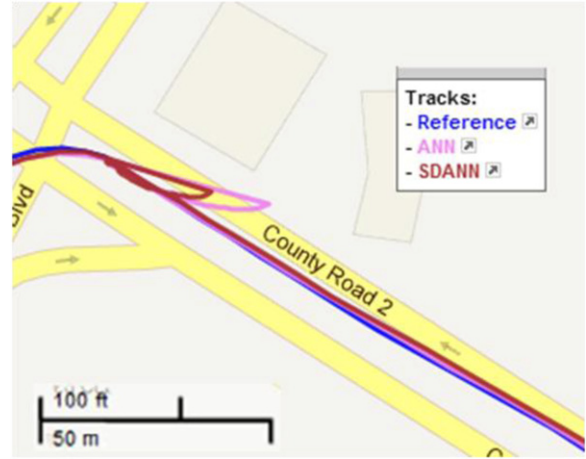
$$\text{RMSE} = \sqrt{\frac{\sum_{i=1}^N [\hat{y}_p(x_i, w) - y_p]^2}{N}}. \quad (13)$$

The performance of the SDANN module tested using PUA and PVUA architectures is reported and compared with the existing ANN-based PUA and PVUA using the error parameter defined by equations (12) and (13).

5.1. PUA model based on SDANN approach

The PUA utilizing SDANN model is trained using algorithm 2 as described in section 4. The training error is less than 10^{-2} with the maximum number of steps set to be 200. It was observed that the coarse and the difference models, containing 15 and 20 hidden neurons, respectively, were able to achieve the desired accuracy during the simulated outages. Duration of these outages is randomly selected and occurs under varied vehicle dynamics. For the first outage of 30 s (outage 1), the proposed SDANN algorithm reduced the positional drift by 11% compared to conventional ANN technique (see figure 11).

The second GPS outage (outage 2) was particularly challenging as it consisted of a curved turn preceded by a straight stretch. The SDANN model successfully reduced the east direction position error from 132.80 to 67.68 m and

**Figure 11.** PUA model performance during GPS outage 1.

demonstrated an overall improvement of 36% as shown in figure 12.

From figure 12, it is evident that SDANN method significantly reduced the positional error when the vehicle encountered a sharp turn as opposed to the straight portion of the trajectory. For the third outage of 90 s (outage 3), shown in figure 13, a 29% improvement in performance by SDANN method is observed.

Table 2 lists the quantitative MAE and RMSE values for all three outages of randomly selected duration.

These results demonstrate the effectiveness of SDANN in reducing the impact of the time growing positional errors for both short and long periods. The main reason for the reduction of positional errors can be attributed to the additional knowledge contained in the difference model, in the form of an estimation of the errors of the coarse model.

**Figure 12.** PUA model performance during GPS outage 2.

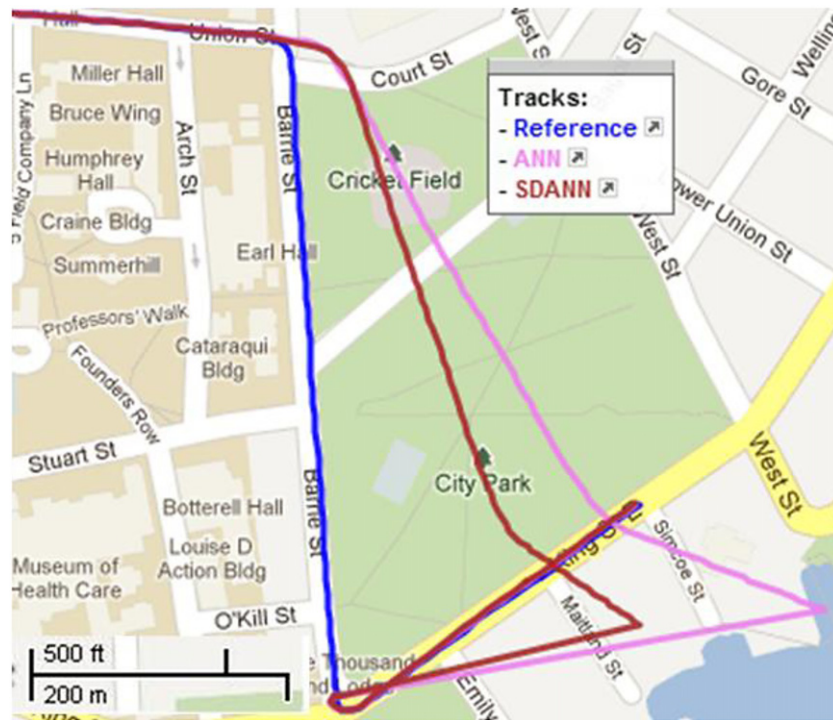


Figure 13. PUA model performance during GPS outage 3.

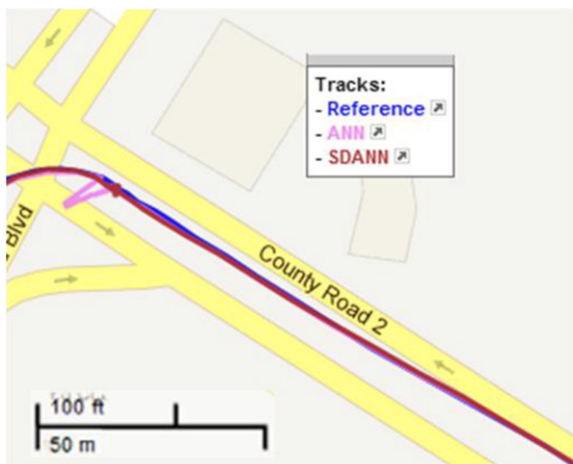


Figure 14. PVUA results during GPS outage 1.

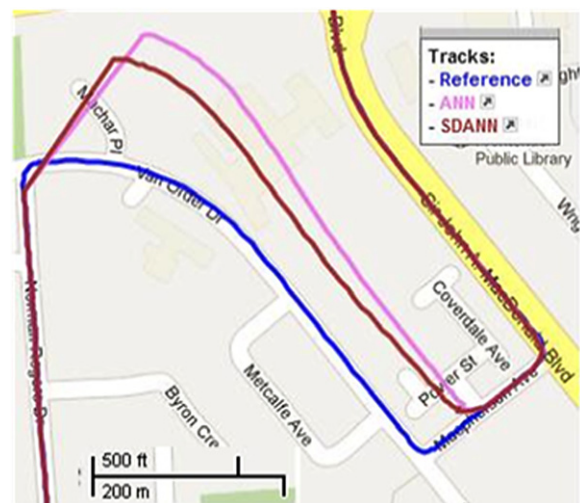


Figure 15. PVUA performance during GPS outage 2.

5.2. PVUA model based on SDANN approach

The training criterion of PVUA is set similar to the PUA with the error tolerance of less than 10^{-2} for a maximum of 200 steps. It was found that for the VUA portion, the coarse and the difference models with 20 and 25 hidden neurons, respectively, provided desired accuracy. On the other hand, for the PUA portion of the PVUA model, the optimal model consisted of 20 hidden neurons for the coarse and 15 hidden neurons for the difference model. For outage 1, we observed an impressive 81% improvement in the solution accuracy using SDANN-based PVUA algorithm over PVUA based on the conventional ANN (figure 14).

The performance of the developed model is then tested with another intentionally introduced 70 s GPS outage

(outage 2). The SDANN model showed 22% improvement in the positioning accuracy as illustrated in figure 15.

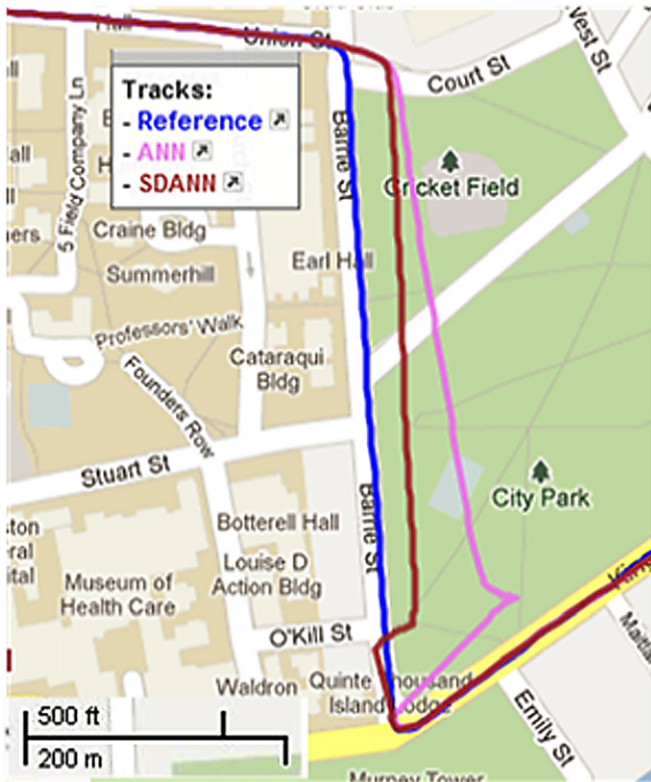
The percentage improvement by SDANN approach for outage 3 is 32% as illustrated by figure 16.

A quantitative comparison of the accumulated position errors obtained using our proposed SDANN algorithm in comparison to the conventional ANN-based PVUA is given in table 3.

Although we have not implemented the EKF in this paper, we hypothesize that our SDANN algorithm will exhibit better performance, when GPS outages occur at high dynamic regions such as swaying, curving and turning roads. During high dynamic outages, EKF may suffer from filter divergence as it is based on the linearization of the nonlinear models.

Table 3. Position errors for the proposed PVUA model and the conventional PVUA model.

PVUA	North position error (m)				East position error (m)			
	MAE		RMSE		MAE		RMSE	
	ANN	SDANN	ANN	SDANN	ANN	SDANN	ANN	SDANN
Outage 1 (30 s)	5.82	0.27	6.72	0.38	4.78	1.46	5.68	1.60
Outage 2 (70 s)	41.84	30.55	50.75	38.14	113.01	90.63	126.74	101.66
Outage 3 (90 s)	41.72	15.61	43.41	18.23	99.05	70.49	113.37	80.64

**Figure 16.** PVUA performance during GPS outage 3.

Further, the proposed SDANN algorithm seems advantageous because it does not require the explicit definition of a model, nor does it require the models to be linearized (unlike EKF).

5.3. Discussion

Both PUA and PVUA models based on the proposed SDANN algorithm have outperformed their conventional counterparts. The performance of SDANN approach is superior because of the difference model, which fine-tunes the approximate output of the coarse model by providing additional knowledge to the overall system. The difference model acquires and stores the knowledge of the coarse model estimation error during GPS availability. In the case of GPS outages, the difference model corrects the coarse model output by utilizing the learnt knowledge. It is evident from figures 11–16 that the proposed SDANN and conventional ANN model trajectories are nearly aligned with the reference trajectory in the absence of GPS outages. However during outages, the conventional model starts to deviate from the reference solution because of the effect of sensor errors. During these periods, the PVUA (i.e.

modified PUA) utilizing SDANN model provides improved navigation solution when compared with PUA. This is mainly due to the two tier approach adopted by the PVUA, i.e. velocity updates followed by position updates. In this two tier approach the velocity updates are fed as an input to the PUA network of PVUA which are then incorporated to update the position components. Thus utilization of updated velocity as inputs which is considered to be more accurate, unlike PUA (which utilizes INS velocity), improves the overall PVUA network learning capability and results in enhanced solution accuracy. Thus, the proposed SDANN-based model demonstrated the increased potential of reducing the time growing positional error for both architectures, i.e. PUA and PVUA. Future work includes incorporating other KBNN methods and different training algorithms to further enhance the navigation accuracy of the GPS-aided low-cost INS systems.

6. Conclusions

For the first time, this paper brings to the forefront a knowledge-based SDANN model that successfully enhances the positioning accuracies especially in the case of very low-cost MEMS grade sensors. Compared to conventional ANNs, the proposed SDANN model-based architectures (PUA and PVUA) were able to reduce the accumulated positional errors significantly. Specifically, the PUA offers 11–36% reduction and the PVUA offers 32–81% reduction in the position errors. The percentage improvement in the navigation accuracy varied depending upon the vehicle dynamics and duration of the GPS outages. The dramatically improved accuracy is attributed to the additional knowledge in the form of a difference model. In conclusion, the proposed SDANN significantly improved the navigation solution accuracy and provided continuous navigation, both with and without GPS signals.

Acknowledgments

The authors would like to thank Dr Aboelmagd Noureldin, Royal Military College of Canada, for sharing the field test data. Financial support from the Air Force Office of Scientific Research (grant no FA9550-01-1-0519) and from the EECS Department, University of Toledo, in the form of graduate assistantship is thanked. PB and VD gratefully acknowledge the encouragement and support of Dr Jon Sjogren (program manager; sensing, surveillance, navigation; Physics and Electronics Directorate, Air Force Office of Scientific Research).

References

- [1] Lobo J, Lucas P, Dias J and Traca de Almeida A 1995 Inertial navigation system for mobile land vehicles *Proc. IEEE Int. Symp. on Ind. Electron.* vol 2 pp 843–8
- [2] Grewal M S, Weill L R and Andrews A P 2007 *Global Positioning Systems Inertial Navigation, and Integration* 2nd edn (Hoboken, NJ: Wiley)
- [3] Misra P and Enge P 2010 *Global Positioning System: Signals, Measurements and Performance* (Lincoln, MA: Ganga-Jamuna Press)
- [4] El-Rabbany A 2002 *Introduction to GPS—The Global Positioning System* (Norwood, MA: Artech House)
- [5] Wong R V C, Schwarz K P and Cannon M E 1988 High-accuracy kinematic positioning by GPS–INS *J. Inst. Navig.* **35** 275–87
- [6] Grewal M and Andrews A P 2001 *Kalman Filtering: Theory and Practice Using Matlab* (New York: Wiley)
- [7] Eun-Hwan S 2001 Accuracy improvement of low-cost INS/GPS for land application *MSc Thesis* Department of Geomatics Engineering, University of Calgary, UCGE Report No. 20156
- [8] Eun-Hwan S 2005 Estimation techniques for low-cost inertial navigation *PhD Thesis* Department of Geomatics Engineering University of Calgary, UCGE Report No. 20219
- [9] Julier S J, Uhlmann J K and Durrant H F 2000 A new approach for nonlinear transformations of means and covariances in filters and estimators *IEEE Trans. Autom. Control* **45** 477–82
- [10] Julier S J and Uhlmann J K 1997 New extension of the Kalman filter to nonlinear systems *Proc. SPIE* **3068** 182–93
- [11] Wan E A and Van der Merwe R 2000 The unscented Kalman filter for nonlinear estimation *Proc. IEEE Adaptive Systems for Signal Processing, Communications and Control Symp. (Lake Louise, Alberta, Canada)* pp 153–8
- [12] Nordlund P J 2002 Sequential Monte Carlo filters and integrated navigation *Thesis* No. 945, Linköping University, Linköping, Sweden
- [13] Gordon N, Salmond D J and Smith A F M 1993 Novel approach to nonlinear/non-Gaussian Bayesian state estimation *IEEE Proc. F Rader Signal Process.* **140** 107–13
- [14] Doucet A, De Freitas N and Gordon N 2001 *Sequential Monte Carlo Methods in Practice* (New York: Springer)
- [15] Gustafsson F, Gunnarsson F, Bergman N, Forssell U, Jansson J, Karlsson R and Nordlund P J 2002 Particle filters for positioning, navigation and tracking *IEEE Trans. Signal Process.* **50** 425–37
- [16] Arulampalan S, Maksell S, Gordon N and Clapp T 2002 A tutorial of particle filters for online nonlinear/non-Gaussian Bayesian tracking *IEEE Trans. Signal Process.* **50** 174–88
- [17] Aggarwal P, Syed Z F and El-Sheimy N 2008 Hybrid extended particle filter (HEPF) for integrated civilian navigation system *Proc. IEEE/ION PLANS (Monterey, CA)* pp 984–92
- [18] Forrest M, Spracklen T and Ryan N 2000 An inertial navigation data fusion system employing an artificial neural network as the data integrator *Proc. ION NTM (Anaheim, CA)* pp 153–8
- [19] Zhang Q J and Gupta K C 2000 *Neural Networks for RF and Microwave Design* (Boston, MA: Artech House)
- [20] Semeniuk L and Noureldin A 2006 Bridging GPS outages using neural network estimates of INS position and velocity errors *Meas. Sci. Technol.* **17** 2783–98
- [21] Chiang K W 2005 INS/GPS integration using neural networks for land vehicular navigation applications *PhD Thesis* Department of Geomatics Engineering, University of Calgary, UCGE Report No. 20209
- [22] Akkala A, Bhatt D, Devabhaktuni V and Kumar A 2012 Knowledge based neural network approaches for modeling and estimating radon concentrations *Environ. Prog. Sustainable Energy* at press doi:[10.1002/ep.11617](https://doi.org/10.1002/ep.11617)
- [23] Watson P M and Gupta K C 1996 EM-ANN models for microstrip vias and interconnects in multilayer circuits *IEEE Trans. Microw. Theory Tech.* **44** 2495–503
- [24] Haykin S 1994 *Neural Networks—A Comprehensive Foundation* (Piscataway, NJ: IEEE Press)
- [25] Bishop C M 1996 *Neural Networks for Pattern Recognition* 1st edn (London: Oxford University Press)
- [26] Ham M and Kostanic I 2001 *Principles of Neurocomputing for Science and Engineering* (New York: McGraw-Hill)
- [27] Abhijit S P and Robert B M 1996 *Pattern Recognition With Neural Networks in C++* (Piscataway, NJ: IEEE Press)
- [28] El-Sheimy N, Chiang K W and Noureldin A 2006 The utilization of artificial neural networks for multisensor system integration in navigation and positioning instruments *IEEE Trans. Instrum. Meas.* **55** 1606–15
- [29] Chiang K W and El-Sheimy N 2002 INS/GPS integration using neural networks for land vehicle navigation applications *Proc. US Institute of Navigation (ION), GPS Oregon Convention Centre (Portland, OR, USA)* (CD) 2002
- [30] Noureldin A, Karamat T B, Eberts M D and El-Shafie A 2009 Performance enhancement of MEMS-based INS/GPS integration for low-cost navigation applications *IEEE Trans. Veh. Technol.* **58** 1077–96
- [31] Dennis J E and Schnabel R B 1983 *Numerical Methods for Unconstrained Optimization and Nonlinear Equations* (Englewood Cliffs, NJ: Prentice-Hall)
- [32] Likas A and Stafylopatis A 2000 Training the random neural network using quasi-Newton methods *Eur. J. Oper. Res.* **126** 331–9
- [33] Zhang Q J 1999 NeuroModeler, Dep. Electron, Carleton University, 1125 Colonel By Drive, Ottawa, K1S5B6 Ontario, Canada