Learning and Nonlinear Models (L&NLM) – Journal of the Brazilian Computational Intelligence Society, Vol. 10, Iss. 4, pp. 227-241, 2012

LOW COST GPS/INS NAVIGATION SYSTEMS WITH ERROR COMPENSATION BY ARTIFICIAL NEURAL NETWORKS

Edmundo Alberto Marques Filho
Atair Rios Neto
Helio Koiti Kuga

National Institute for Space Research - INPE
Division of Space Mechanics and Control
edmundo_amf@yahoo.com.br; atairrn@uol.com.br; hkk@dem.inpe.br
Av. dos Astronautas, 1758 - CP 515
CEP 12227-010, São José dos Campos, SP, Brazil

Abstract - This paper addresses the use of artificial neural networks (ANN) to compensate for errors in inertial measurement units (IMU) of global positioning system (GPS) aided inertial navigation systems (INS). The GPS technology dominates, nowadays, the positioning and navigation (POS/NAV) market, and alternative POS/NAV systems are only needed because GPS does not work in all environments, or can not provide reliable solutions during some time interval. There are different solutions to fulfill information during GPS blockage and integrated inertial sensors systems with GPS are frequently used. However, low cost inertial sensors have the disadvantage of accumulating continuous errors in great extension, leading to poor system performance. In this context, ANN is applied to provide better NAV/POS solutions during the lack of information in GPS outages. This work introduces a review of the main used concepts and techniques, an approach to define input-output ANN signals based on a reduced set of inertial navigation equations, and the ANN prediction and training operation modes. It also presents a training algorithm, based on adaptive Kalman filtering approach and proposes a method for ANN training with the characteristic of alternating the training patterns from batch mode, with a constant data set size, to sequential mode, by filtering individual pattern-by-pattern of training data, which gives to the method some real time training capacity. Finally, numerical simulation results are assessed from urban vehicular positioning application, with data acquired from an MEMS IMU Crossbow CD400_200 and an Ashtech Z12 GPS receiver. The proposed methods were tested with different land vehicle dynamic situations and the position errors, computed in prediction mode or simulated GPS outage, were assessed. When compared to a conventional INS/GPS system, integrated by a Kalman filter and operating without GPS updates, the ANN position errors have lower magnitudes. These results indicate that ANN was more capable to learn the vehicle's kinematics, for a certain time interval, than the modeling presented by the conventional navigation system.

Keywords- GPS aided inertial navigation; low cost navigation; positioning and navigation systems; artificial neural networks; adaptive Kalman filter.

1 Introduction

The global positioning system NAVSTAR-GPS, or simply GPS, is a one-way, almost all-weather, real-time and world-wide radio navigation system based on a satellite constellation. The main purpose of the system is to provide a signal that allows a dedicated receiver to compute accurately and in real time its position (longitude, latitude and altitude). The system is also usable for accurate time transfer and velocity estimation. Besides being globally available, GPS is portable, has low power consumption, suitable for sensor integration and capable of providing accurate and low cost navigation.

The need for alternative source of positioning and navigation (POS/NAV) information arises because GPS does not work properly in all environments. The GPS receiver can suffer from signal blockage, or interference, due to weather and or environment obstacles, which may deteriorate the overall system performance. Different solutions have been proposed to fulfill the lack of information during the GPS outages by using stochastic parameter estimation techniques, such as Kalman filters, in order to integrate GPS with inertial navigation systems (INS).

Micro-electromechanical systems (MEMS) based inertial sensors application for navigation purposes has been developed due to its low price, small size, light weight, and lower power consumption [1]. However, low cost inertial sensors have the disadvantage of accumulating continuous errors in great extension, leading to poor system performance, when operating in a stand-alone mode.

Alternative solutions for GPS/INS integration, based on artificial neural network (ANN), have been proposed, most of them for general land vehicle applications, using field data collected from tactical and navigation grade inertial measurement units

(IMU) [2, 3, 4, 5, 17]. ANN method does not rely on prior knowledge or dependencies such as dynamics and sensor error modeling or linearization, and can learn from the existing data [2].

This paper is an extension from previous works [6, 7, 8] where a scheme using an artificial neural network to integrate GPS with low cost MEMS inertial sensors INS, based on an adaptive ANN training Kalman filtering methodology, is presented. The results of tests on real data of an urban vehicular position application are compared against those obtained by using a 15 state Sigma-Point Kalman Filter (SPKF) INS/GPS integration loosely coupled scheme, under the same IMU and GPS data set [9].

In what follows, closely to what was developed in [6, 7, 8], Section 2 presents the usual Kalman Filter (KF) GPS aided inertial navigation loosely coupled scheme and Section 3 briefly introduces the use of ANN and presents the proposed Kalman filter based methodology for a land vehicle type of motion. In Section 4 the results of testing are presented and compared with a current more standard Kalman filter. Finally in Section 5 some conclusions are drawn.

2 Usual KF GPS aided inertial navigation

INS is a self-contained mounted vehicle device that estimates its position and attitude by processing higher frequency information from IMU sensors, accelerometers and gyroscopes, while the GPS receiver estimates position and velocity by processing lower frequency signals from, at least, four satellites and can be used to correct accumulated errors in the INS.

The usual Kalman Filter (KF) GPS aided inertial navigation loosely coupled scheme process satellite signals to estimate position and velocity, which are combined with the INS to form positions and velocities errors and sent to the navigation KF. This filter corrects the navigation equations errors and the inertial sensors biases to provide the navigation solution vectors: position, velocity and the attitude Euler Angles, as shown in Figure 1. The KF state vector is given by [9]: $\mathbf{r} = [\varphi \quad \lambda \quad \mathrm{h}]^{\mathrm{T}}$ the geodetic latitude, longitude and height; $\mathbf{v} = [\mathrm{v_N} \quad \mathrm{v_E} \quad \mathrm{v_D}]^{\mathrm{T}}$, the north, east and down velocity components; $\boldsymbol{\theta} = [\phi \quad \theta \quad \psi]^{\mathrm{T}}$, the roll, pitch and yaw attitude angles; $\mathbf{b}_{\mathrm{a}} = [\mathrm{b_u} \quad \mathrm{b_v} \quad \mathrm{b_w}]^{\mathrm{T}}$, the accelerometers x, y and z axes biases; and $\mathbf{b}_{\mathrm{g}} = [\mathrm{b_p} \quad \mathrm{b_q} \quad \mathrm{b_r}]^{\mathrm{T}}$, the gyroscopes x, y and z axes biases.

This GPS aided inertial navigation loosely coupled scheme with a Sigma Point Kalman Filter (SPKF) under the same IMU and GPS data set is taken as a benchmark and used to assess the proposed method.

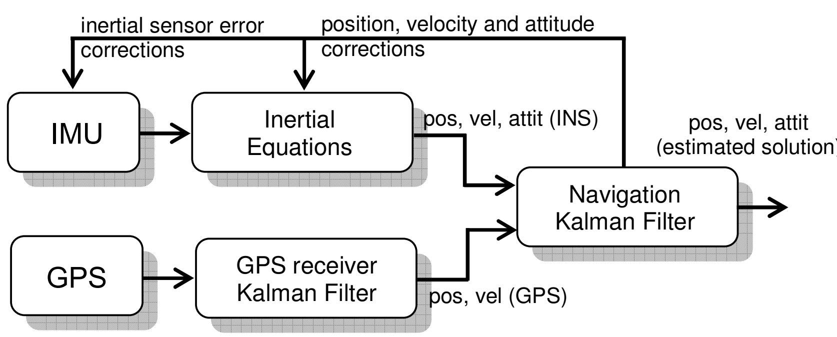

Figure 1: Conventional loosely coupled INS/GPS scheme.

3 Artificial neural network and Kalman filter methodology

3.1 Introduction

A Multilayer of perceptrons (MLP) type of ANN is made up of layers of basic artificial neurons (perceptrons) connected forward and can learn nonlinear mappings (Eq. (1)) by adjustment of its synaptic weights, via a supervised learning process [11].

$$\mathrm{f} \in \mathrm{C} : \mathrm{x} \in \mathrm{D} \subset \Re^{\mathrm{n}} \rightarrow \mathrm{y} \in \Re^{\mathrm{m}} \tag{1}$$

The MLP training, by supervised learning, can be done by estimating the weight parameters in order to fit the neural network model to a set of L input-output patterns [13]:

$$\{(\mathbf{x}(t), \mathbf{y}(t)): \quad \mathbf{y}(t) = f(\mathbf{x}(t)), \quad t = 1,2 \cdots L\} \tag{2}$$

The MLP can be viewed and treated as a parameterized mapping of the input data, $\mathbf{x}(t)$, to neural network output $\hat{\mathbf{y}}(t)$:

$$\hat{\mathbf{y}}(t) = \hat{f}(\mathbf{x}(t), \mathbf{w}(t)) \tag{3}$$

The supervised training process can be solved by minimizing, with respect to the vector of weights w, the following functional, given the input-output data set, a priori estimate $\overline{\mathbf{w}}$, and the weight matrices $\overline{\mathbf{P}}^{-1}$ and $\mathbf{R}^{-1}$ [13]:

$$J(\mathbf{w}) = \frac{1}{2}\left[(\mathbf{w} - \overline{\mathbf{w}})^T \overline{\mathbf{P}}^{-1} (\mathbf{w} - \overline{\mathbf{w}}) + \sum_{t=1}^{L} \left((\mathbf{y}(t) - \hat{f}(\mathbf{x}(t), \mathbf{w}))^T \mathbf{R}^{-1} (\mathbf{y}(t) - \hat{f}(\mathbf{x}(t), \mathbf{w}))\right)\right] \tag{4}$$

Minimizing **J** is the solution of a linear Least Square problem, where the $\overline{\mathbf{P}}$ matrix can be chosen diagonal, since the diagonal terms are variances from the *priori* information errors, typically assessed empirically. **R** is the covariance matrix of errors from the data used for training.

When available, GPS and IMU data are acquired, recorded and processed to create a training data set, with position and velocity increments output being linearly interpolated at IMU higher rate. A MLP can then be trained in a correspondent GPS aided inertial navigation loosely coupled scheme to learn and compensate for errors in the IMU in a scheme as depicted in Figure 2.

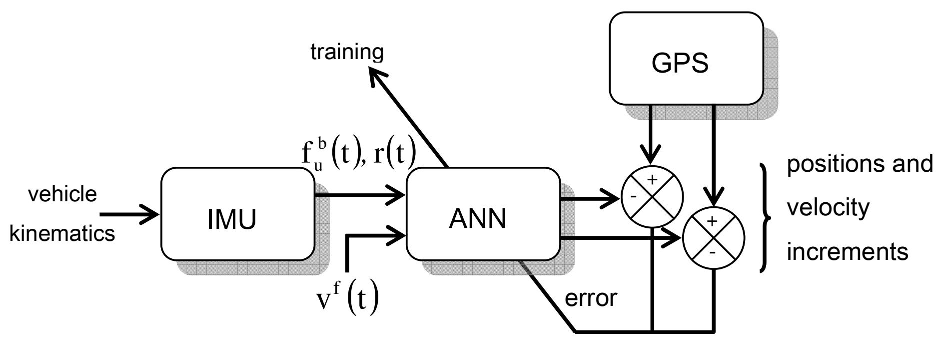

Figure 2: ANN training in a loosely coupled INS/GPS scheme.

3.2 Kalman filtering training method

Singhal and Wu [12] have proposed the use of the extended Kalman filtering as training algorithm, viewing ANN as a stochastic parameter estimation problem. Rios Neto [13] has further explored this concept and proposed an algorithm that features parallel processing, in order to avoid large computational loads, and an adaptive state noise estimation to prevent the Kalman filtering based ANN parameter estimators from loosing the capacity of distributing the extraction of information to all training data. The Rios Neto [13] proposed solution is implemented in this paper, adapted to an inertial navigation problem, and is summarized in what follows.

The mapping in Eq. (3) can be expanded in a Taylor series, and in a typical ith iteration, a linear perturbation is adopted to approximate the functional given by Eq. (4) [13]:

$$\alpha(t)[\mathbf{y}(t) - \overline{\mathbf{y}}(t, \overline{\mathbf{w}})] + \mathbf{H}(t, \overline{\mathbf{w}})\overline{\mathbf{w}} = \mathbf{H}(t, \overline{\mathbf{w}})\mathbf{w}(t) \tag{5}$$

Or, in a compact notation:

$$\mathbf{z}(t) = \alpha(t)[\mathbf{y}(t) - \overline{\mathbf{y}}(t, \overline{\mathbf{w}})] + \mathbf{H}(t, \overline{\mathbf{w}})\overline{\mathbf{w}} \tag{6}$$

Where $\overline{\mathbf{w}}$ is the priori estimate of **w** coming from the previous iteration, α is an adjustable parameter to guarantee the hypothesis of linear perturbation, $0 < \alpha(t) \leq 1$, and the **H** is defined as:

$$\mathbf{H}(t, \overline{\mathbf{w}}) = \hat{f}_w(\mathbf{x}(t), \overline{\mathbf{w}}) = \left.\frac{\partial \hat{f}(\mathbf{x}(t), \mathbf{w})}{\partial \mathbf{w}}\right|_{\mathbf{w}=\overline{\mathbf{w}}} \tag{7}$$

With these linearized approximations the minimization of the functional in Eq. (4) can be iteratively solved, and in a given iteration, for t= 1.2. …,L, treated as a stochastic linear estimation problem, which can be formulated as:

$$\overline{\mathbf{w}}(t) = \mathbf{w}(t) + \overline{\mathbf{e}} \tag{8}$$

$$\mathbf{z}(t) = \mathbf{H}(t)\mathbf{w}(t) + \mathbf{v}(t) \tag{9}$$

$$\mathbf{v}(t) = N(\mathbf{0}, \mathbf{R}),\ \overline{\mathbf{e}} = N(\mathbf{0}, \overline{\mathbf{P}}) \tag{10}$$

Where the weighting matrices $\mathbf{R}$ and $\overline{\mathbf{P}}$, respectively, are now viewed and treated as the covariance matrices of $\mathbf{v}$ and $\overline{\mathbf{e}}$ error vectors. KF can then be applied if the **w(t)** are modeled as a stochastic process with **ω**(t) white noise:

$$\mathbf{w}(t+1) = \mathbf{w}(t) + \boldsymbol{\omega}(t) \tag{11}$$

A sequential KF solution, for the estimation problem, with $t = 1, \cdots, L$, is given by

$$\mathbf{K}(t) = \overline{\mathbf{P}}(t)\,\mathbf{H}^T(t, \overline{\mathbf{w}})\left[\mathbf{R}(t) + \mathbf{H}(t, \overline{\mathbf{w}})\overline{\mathbf{P}}(t)\mathbf{H}^T(t, \overline{\mathbf{w}})\right]^{-1} \tag{12}$$

$$\hat{\mathbf{w}}(t) = \overline{\mathbf{w}}(t) + \mathbf{K}(t)\left[\mathbf{z}(t) - \mathbf{H}(t, \overline{\mathbf{w}})\overline{\mathbf{w}}(t)\right] \tag{13}$$

$$\mathbf{P}(t) = \overline{\mathbf{P}}(t) - \mathbf{K}(t)\mathbf{H}(t, \overline{\mathbf{w}})\overline{\mathbf{P}}(t) \tag{14}$$

$$\overline{\mathbf{P}}(t+1) = \mathbf{P}(t) + \mathbf{Q}(t) \tag{15a}$$

$$\overline{\mathbf{w}}(t+1) = \hat{\mathbf{w}}(t) \tag{15b}$$

For the above equations, **K** is the Kalman gain and **Q** is the $\boldsymbol{\omega}(t)$ process noise covariance matrix. At the end of the each iteration, t = L. If a stopping criterion is satisfied, the training is finished, then $\overline{\mathbf{w}} = \hat{\mathbf{w}}(L)$ and $\overline{\mathbf{P}} = \hat{\mathbf{P}}(L)$. Otherwise a new iteration starts, with the initial conditions $\overline{\mathbf{w}}(1) = \hat{\mathbf{w}}(L)$ and $\overline{\mathbf{P}}(1) = \overline{\mathbf{P}}_0$.

3.3 Adaptive Kalman filter solution for ANN training

Kalman filtering estimators divergence may occur as a large data set is processed, due to both algorithm bad numerical behavior and observation model errors. In this situation, the neural network can lose its capacity of keeping learning as new data are processed. Rios Neto [13] has proposed an adaptive procedure based on a simple criterion of statistical consistency to balance a priori information priority with that of new learning information:

$$\beta E\left[v_j^2(t)\right] = \mathbf{H}_j(t, \overline{\mathbf{w}})\left[\mathbf{P}(t)+\mathbf{Q}(t)\right]\mathbf{H}_j^T(t, \overline{\mathbf{w}}) \tag{16}$$

Where j = 1, …, m observations, and β is to be adjusted. When $\beta = 1$, new information has the same value, when compared to that one stored in trained weights; and with $\beta < 1$, but close to 1, new processed information, from new pattern, has less value than the stored one. Eq. (16) can be considered as an observation and expanded into the following associated exact estimation problem, according to Freitas Pinto and Rios Neto [14], to be processed with a Kalman filtering algorithm:

$$0 = \mathbf{q}(t) + \overline{\mathbf{e}}^q(t) \tag{17}$$

$$\mathbf{z}^q(t+1,\beta) = \mathbf{H}^q(t+1)\mathbf{q}(t) + \mathbf{v}^q(t+1) \tag{18}$$

$$E\left[\overline{\mathbf{e}}^q\right] = 0, \; E\left[\overline{\mathbf{e}}^q \cdot \overline{\mathbf{e}}^{q^T}\right] = \mathbf{I}_{n_w} \tag{19}$$

$$E\left[\mathbf{v}^q(t+1)\right] = 0, \; E\left[\mathbf{v}^q(t+1)\cdot \mathbf{v}^{q^T}(t+1)\right] = \mathbf{R}^q(t+1) = 0 \tag{20}$$

Where:

$$q_k(t) = \begin{cases} 0 & \text{se} \quad \hat{q}_k < 0 \\ \hat{q}_k & \text{se} \quad \hat{q}_k \geq 0 \end{cases} \tag{21}$$

$$\mathbf{Q}(t) = diag\left[\mathbf{q}(t)\right] \tag{22}$$

3.4 Proposed ANN architecture for sensor integration

Without any loss of generality the proposed solution with an ANN is developed for the case of land vehicle navigation. This choice is justified by its great interest together with the facility it offers for generating testing data. It has been the choice made for application and testing in most of the alternative solutions for GPS/INS integration based on artificial neural network (ANN) that have been proposed [2, 3, 4, 5].

Constraints on the movement of land vehicles can be defined and used to derive a reduced set of equations of motion. Brandt and Gardner [10] define the following constraints: direction of the vehicle's velocity coincides with direction of the vehicle's longitudinal axis; pitch and roll angles of the of the vehicle's body relative to the Earth surface are small and vehicle always remains on the Earth surface. Under these constraints, a set of simplified equations of motion can be defined [10]:

$$\dot{v}^f = f_u^b - g\,sin\theta \tag{23}$$

$$v_N = cos\theta\, cos\psi\; v^f \tag{24}$$

$$v_E = cos\theta\, sin\psi\; v^f \tag{25}$$

Where v^f is the forward velocity, defined in the direction of movement; g is the gravity acceleration; v_N and v_E are, respectively, north and east vehicle velocities defined in the navigation coordinate frame; $\mathbf{\Phi} = [\phi \quad \theta \quad \psi]^T$ are roll, pitch and yaw vehicle attitude angles; and f_u^b is the specific force accelerometer output in the direction of movement. The attitude angles can be computed as [10]:

$$\dot{\phi} = \mathrm{p} + (sin\phi tan\theta)\mathrm{q} + (cos\phi tan\theta)\mathrm{r} \tag{26}$$

$$\dot{\theta} = cos\phi \mathrm{q} - sin\phi \mathrm{r} \tag{27}$$

$$\dot{\psi} = \frac{sin\phi}{cos\theta}\mathrm{q} + \frac{cos\phi}{cos\theta}\mathrm{r} \tag{28}$$

Where $\boldsymbol{\omega}_{ib}^{b} = [\mathrm{p} \quad \mathrm{q} \quad \mathrm{r}]^T$ is the vector of angular velocities from gyros output. Considering the previous constraints differential equations (23) and (28) can be further simplified for usual vehicle urban use:

$$\dot{\mathrm{v}}^f \approx \mathrm{f}_u^b \tag{29}$$

$$\dot{\psi} \approx \mathrm{r} \tag{30}$$

The proposed ANN uses a three layer feedforward architecture with an input/output model based on Eq. (29) and (30), as shown in the Figure 3. The ANN works in two modes: training mode, while GPS information is available, and prediction mode, during GPS outages. Selection of input/output signals takes into account the nature of the land vehicle dynamics and the possible observations from GPS.

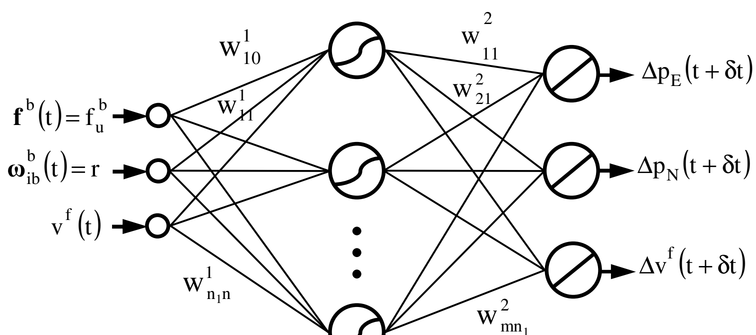

Figure 3: ANN input/output signals.

The input layer signals came from accelerometer and gyro measurements, and the forward velocity v^f, at instant time t. The output layer signals are increments of velocity v^f, east and north positions at instant time $(\mathrm{t} + \delta\mathrm{t})$, where $\delta\mathrm{t}$ is the IMU output frequency, and v^f can be observed from:

$$\mathrm{v}_{GPS}^f = \sqrt{(\mathrm{v}_N^{GPS})^2 + (\mathrm{v}_E^{GPS})^2} \tag{31}$$

The ANN navigation solution is given by:

$$\mathbf{p}^n = [\mathrm{p}_E \quad \mathrm{p}_N \quad \mathrm{v}^f]^T \tag{32}$$

If v_N and v_E are needed, it can be computed from Eq. (24) and (25). While in training mode, an error vector $\mathbf{e}(\mathrm{t} + \delta\mathrm{t})$ is generated for supervised training:

$$\mathbf{e}(\mathrm{t} + \delta\mathrm{t}) = \Delta\mathbf{p}_{GPS}^n(\mathrm{t} + \delta\mathrm{t}) - \Delta\mathbf{p}_{ANN}^n(\mathrm{t} + \delta\mathrm{t}) \tag{33}$$

During the prediction mode, the ANN increments output are added to the last GPS navigation solution, $\mathbf{p}^n(t_u) = \mathbf{p}^n_{GPS}(t_u) = \mathbf{p}^n_u$, where t_u denotes the last time signal before GPS outage and $\mathbf{p}^n$ is described by Eq. (32). Also, in this mode, the velocity v^f input comes from a feedback output summation, as shown in Figure 4.

$$\mathbf{p}^n_{NAV}(t_u + i\delta t) = \mathbf{p}^n_u + \sum_i \Delta\mathbf{p}^n_{RNA}(t_u + i\delta t),\ i = 1, 2, \ldots \tag{34}$$

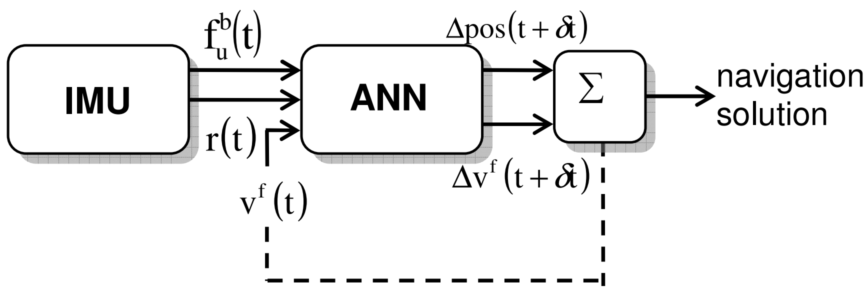

Figure 4: ANN in prediction mode.

3.5 Methods of ANN training for the navigation problem

In order to explore the adaptive Kalman filtering training algorithm, methods of dealing with data patterns are revised and some modifications are proposed. Figures 5 to 8 show the training methods.

First method is a simple version of Chiang et al. [15] proposal. Data from IMU and GPS, when available, are preprocessed and stored sequentially in time, into data sets $\mathbf{J}_i$, or windows of data, of some size. When each $\mathbf{J}_i$ data set is completed, a $\mathbf{T}_i$ training procedure starts, with initial weights obtained by random initializations of small values [-0.5, 0.5]. After completing the ANN training, the resulting weights $\mathbf{W}_i$ are stored as the best available solution if GPS outages occur, while a new data set $\mathbf{J}_{i+1}$ is being completed. After that a new training procedure is repeated.

Second method is the Chiang et al. [15] proposal. It is the same of first method, with one modification. After the first data set has being trained, the next training procedures have initial weights $\mathbf{w}_0$ obtained from previous stored solution, or $\mathbf{w}_{0_i} = \mathbf{W}_{i-1}$. The resulting $\mathbf{W}_i$ weights are stored as the best available solution.

Third method is the same of second method, with the following modification proposed by the authors: after the first data set training, the stored weights are updated by filtering pattern-by-pattern new data, while a new data set is being completed. After that a new training procedure starts with $\mathbf{w}_0 = \mathbf{W}_{i-1_{UPDATED}}$. It is a tentative to bring the previous weight solution as close as possible to a GPS outage, since it can occur far from the previous training procedure.

Fourth method, proposed by the authors, is the following: After the $\mathbf{J}_1$ first data set training is completed, the $\mathbf{W}_1$ stored weights solution are only updated by filtering pattern-by-pattern new data, during some predetermined time interval criterion, or until a GPS outage starts, when the resulting $\mathbf{W}_{1_{UPDATED}}$ updated weights are used. The process is resumed after the GPS outage, or after the time interval criterion is met, when a new data set is completed and the method is repeated.

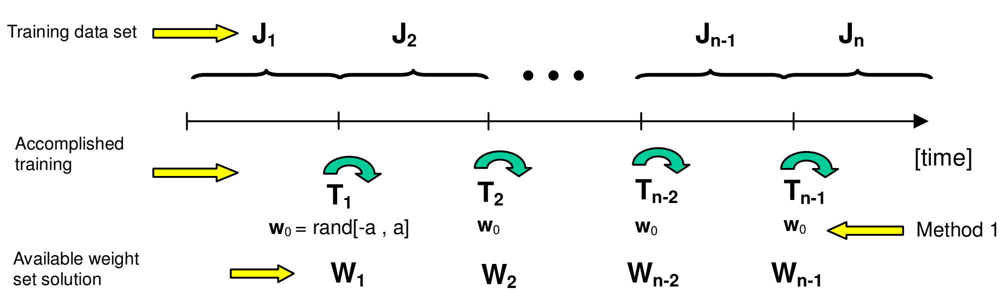

Figure 5: Training method 1.

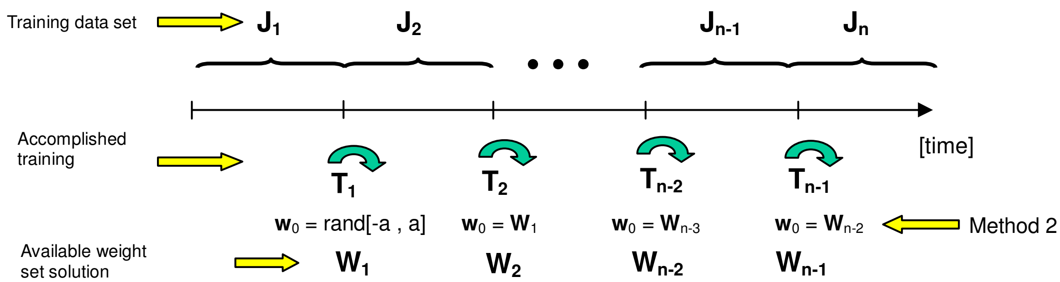

Figure 6: Training method 2.

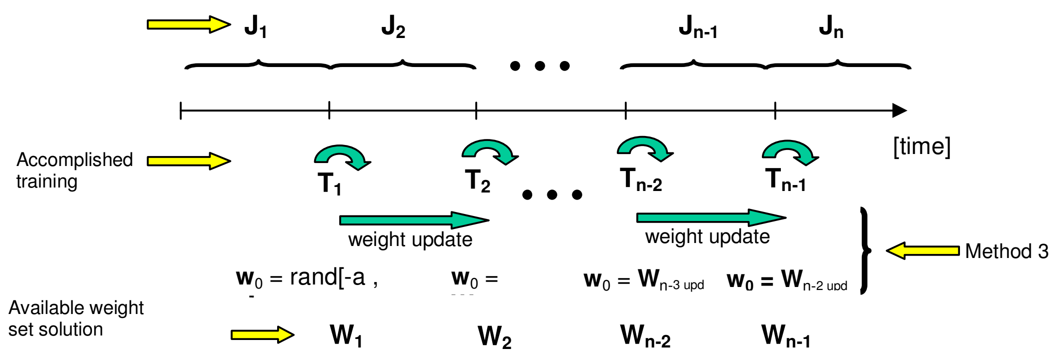

Figure 7: Training method 3.

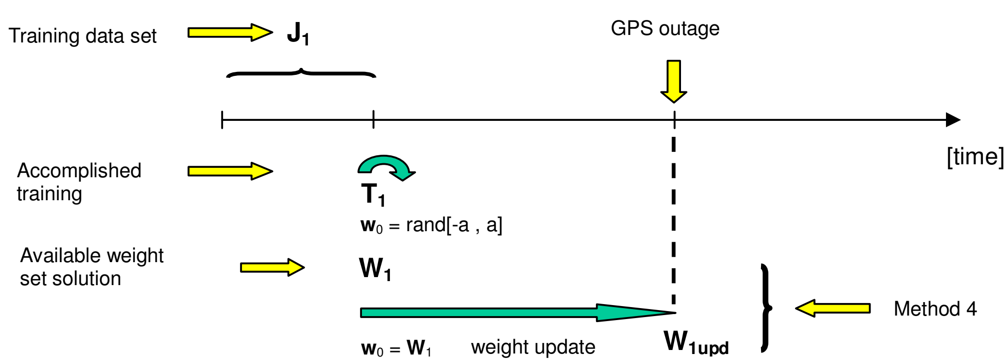

Figure 8: Training method 4.

4 Simulation methodology and results

4.1 Field test procedure

Field tests were conducted in a land vehicle, for urban usage, at INPE campus in São José dos Campos (23.2113° S, 314.1408° E), characterized by streets and avenues relatively flat, with surrounding medium sized buildings and large stature trees. It was respected internal signaling with maximum allowed velocity of 40km/h. A low cost Crossbow CD400-200 IMU, based on MEMS technology, and an Ashtech Z12 GPS receiver, with single antenna, were used to acquire two data sets. IMU data was acquired at 20Hz and GPS data at 2Hz and 1Hz, respectively. GPS different output set up was used in order to assess the influence of higher rate information interpolation on the training data set capacity to capture changes on vehicle's kinematics.

Figures 9 and 10 show the interval [50s, 815s] from a first data set used for simulations. In order to produce a reliable vehicle's kinematics modeling, the ANN should have correct training patterns. Considering the necessity to extract accurate information from training data set, it has been prefiltered by using a simple KF technique mainly due to low cost inertial sensors noise characteristics [16].

All information was post-processed to test proposed training algorithms and methods, by numerical simulations. To evaluate the proposed methods, a prediction error of 20 meters, in 30 seconds of simulated GPS outage, was set as the experiment goal, considering it acceptable for general low performance land vehicle navigation purposes [1].

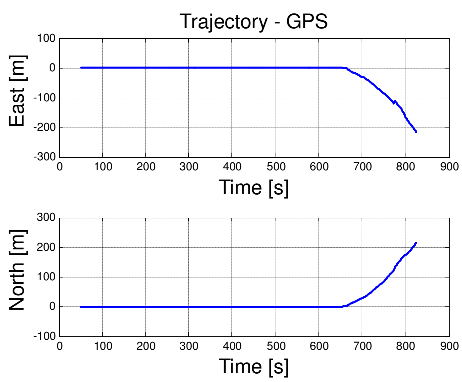

Figure 9: North-East GPS trajectory.

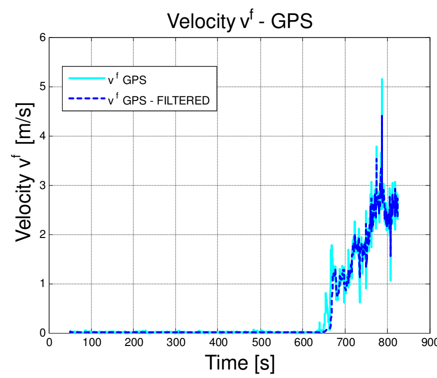

Figure 10: Forward vehicle GPS velocity.

4.2 Proposed adaptive Kalman filter method verification

The proposed adaptive extended Kalman filter (AEKF) training solution, presented in Section 3, can be verified in an off-line function approximation problem. A simulation of vehicle's North and East positions, and forward velocity from [635s, 715s] data interval is tested. The trained ANN output is compared to GPS solution to generate errors. In order to assess the proposed solution performance the AEKF method is compared to the known and efficient Levenberg-Marquardt (LM) method.

For all cases an MLP was used with a 20 neurons hidden layer, with sigmoidal/linear activation functions; 40 iterations, or epochs, as a stopping criterion; α = 0,2 e β = 0,9. Entire pattern data set was divided into 30% to test and verification, and 70% to training. After 100 independent runs, it can be observed from Figure 11 that North and East positions and velocity errors have almost the same magnitude for both methods, and Figure 12 shows that LM method converges in fewer iterations but AEKF method has a slightly lower final mean squared error (MSE) after the iterations. It should be also noted that sequential training patterns is only allowed for AEKF method, since LM is a batch method.

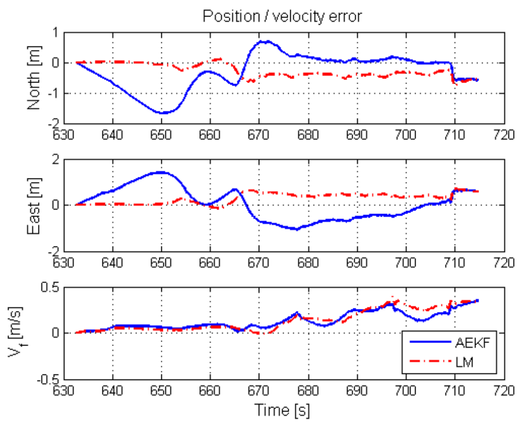

Figure 11: Position and velocity errors comparison.

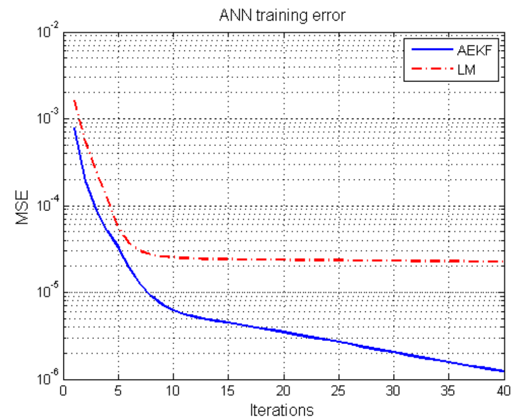

Figure 12: ANN MSE training errors

4.3 Hidden neurons and training iterations considerations

The choice of neural network architecture was partially defined in the Section 3. Since there is not a general rule upon which to decide the exact number of hidden neurons for a mapping or approximation problem [11], some heuristic procedure may be useful to give some empirical insight about the neurons number's choice. The R correlation index, from linear regression analysis, can be used to give some hint about ANN generalization. R index close to unity indicates a good matching between ANN output and desired response values. As an example Figures 13 and 14 shows R index for the testing data set, east and north increments, from the previous section simulation with the same ANN architecture and stopping criterion.

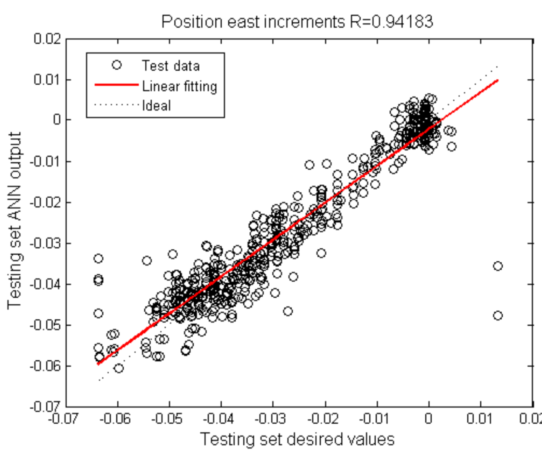

Figure 13: R index for testing set – East increments.

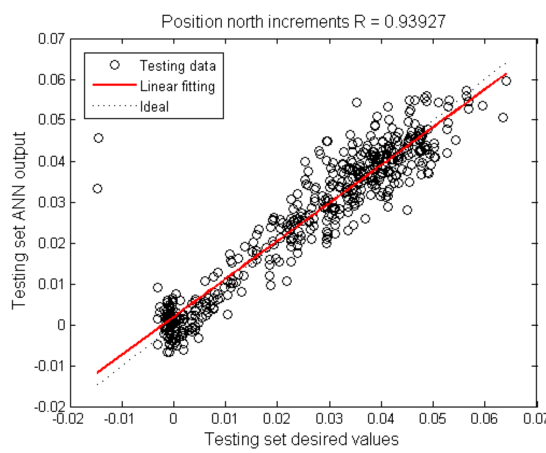

Figure 14: R index for testing set – North increments.

After 5 independent runs, Figure 15 shows R index, training and test error mean values in function of hidden neurons. It can be observed that R values do not improve from 20-25 neurons. Another similar simulation is repeated with iteration number and with 20 hidden neurons chosen as shown in Figure 16. It can be observed that after 40 iterations the R values do not improve.

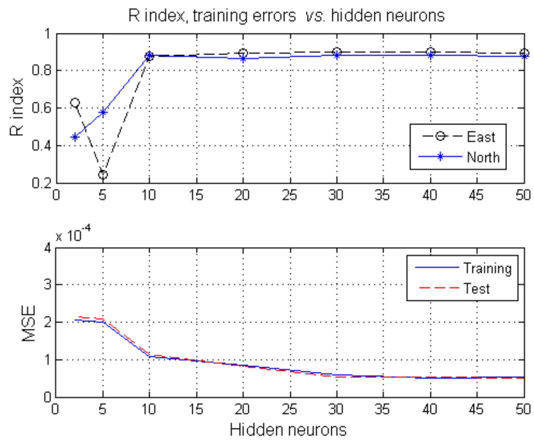

Figure 15: R index and training errors *vs*. hidden neurons.

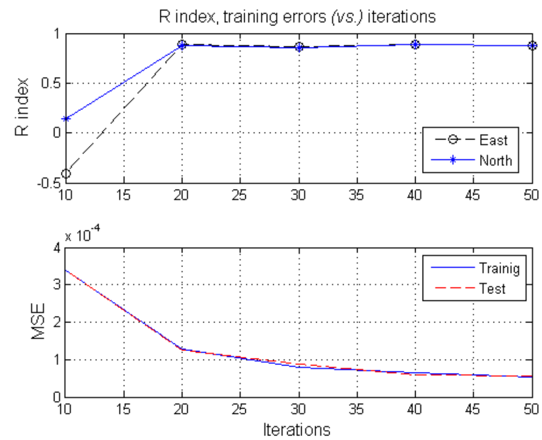

Figure 16: R index and training errors *vs*. iterations.

4.5 Simulations and results

First data set: A field test campaign to acquire data was taken by using the GPS, at 2Hz rate, and the IMU at 20 Hz.

For a first experiment, the vehicle was in static position. Information from GPS and IMU [50s, 200s] interval were used. The position errors for GPS outages, starting at t = 155[s], are shown in Figures 17 and 18, where methods 1 to 4 were tested with different data set lengths, J_1, J_2, … J_n , according to Figures 5-8. Chosen lengths are 20s, 40s and 60s. During the 45 seconds simulated GPS outages, the vehicle prediction position error, with respect to north and east GPS information, is given by:

$$e_{pred_{POS}} = \sqrt{e^2_{pred_{EAST}} + e^2_{pred_{NORTH}}} \tag{35}$$

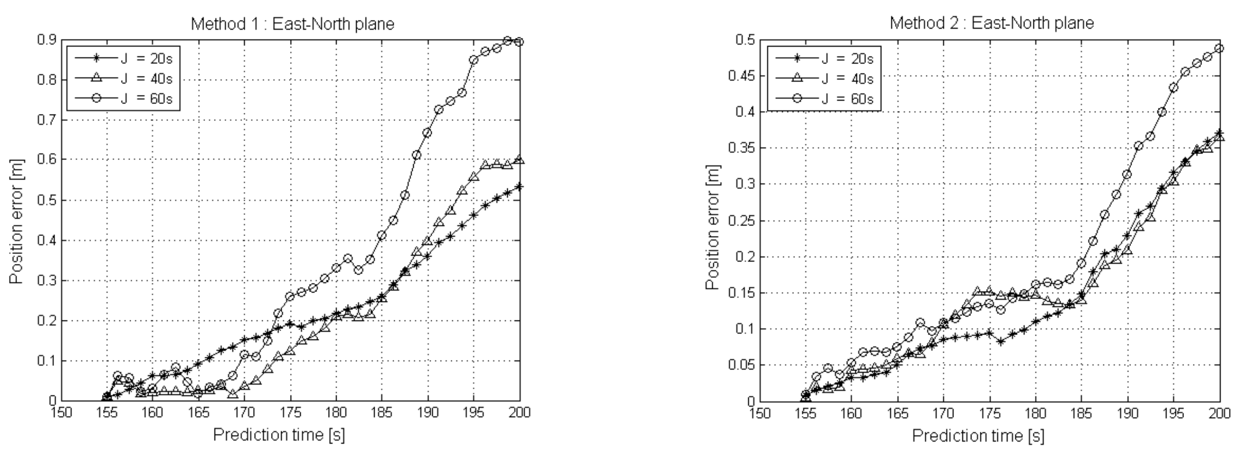

Figure 17: Error for GPS outage at t = 155[s]. a) Method 1, b) Method 2.

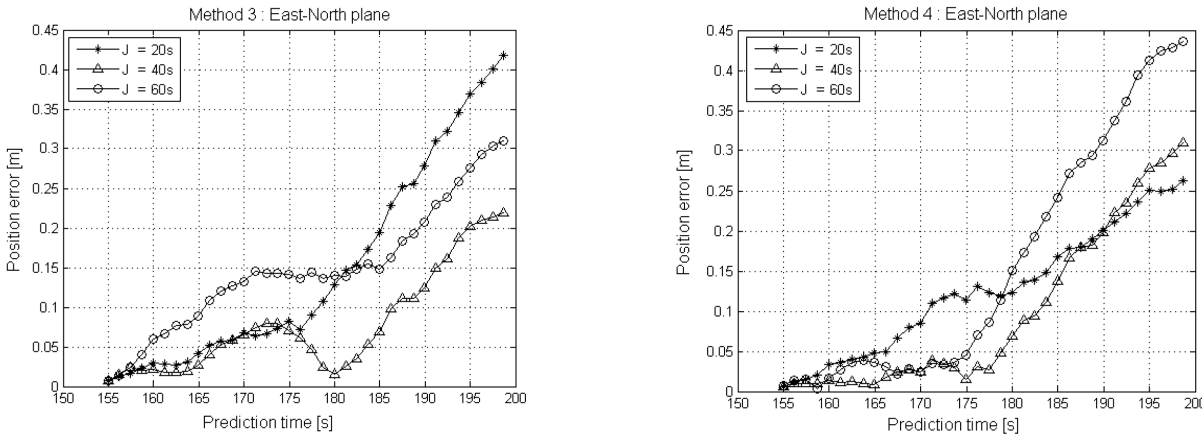

Figure 18: Error for GPS outage at t = 155[s]. a) Method 3, b) Method 4.

It can be observed for static positioning that data set different size has little influence on prediction errors. To compare the proposed training methods a 20s data set was chosen. Figure 19 explains de differences among the training methods and Figure 20 shows the prediction errors including the SPKF INS/GPS integration scheme also operating in prediction mode

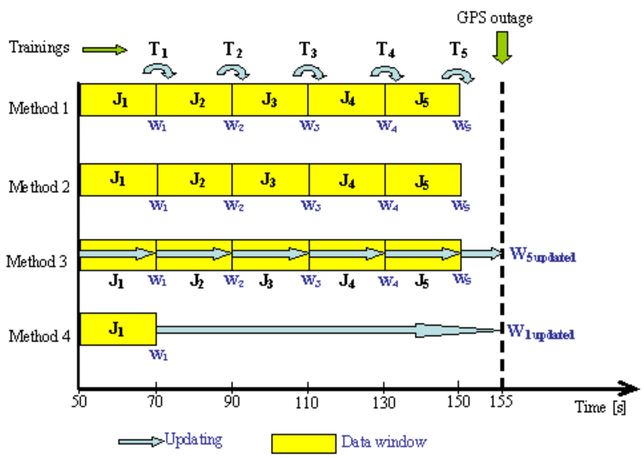

Figure 19: Training methods diagram for t_{pred} = 155[s].

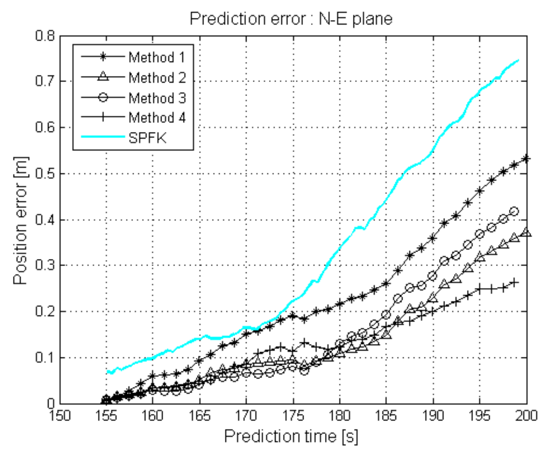

Figure 20: Error for GPS outage at t_{pred} = 155[s].

It should be noted that, for the first experiment, simulation starts around t = 50 [s], including the SPKF scheme. Since data set length is 20 [s], method 4 starts to filter individual training data pairs around t = 70 [s], after the first data set is completed. Then, the total filtering time, until the GPS simulated outages, is about 85 [s]. For methods 1 to 3, completed data set were trained one more time, until first outage, and four more times until second outage, when compared to method 4, with one data set training completed.

In the second experiment, the vehicle was in movement. Information from GPS and IMU [630s, 800s] interval were used and the previous simulation procedures were repeated. A data set length of 60 [s] showed best results. The position errors for GPS outages, starting at t = 705 [s] and t = 755[s], are shown in Figures 21 and 22. Again, it should be noted that, for the second experiment, simulation starts around t = 630 [s], also including the SPKF scheme. Since data set length is 60 [s], method 4 starts to filter individual training data pairs around t = 690 [s], after the first data set was completed. Then, the total filtering time, until the GPS simulated outages, is about 15 [s] and 65 [s], respectively. For methods 1 to 3, completed data set was trained one more time, until second outage, when compared to method 4, with one data set training completed.

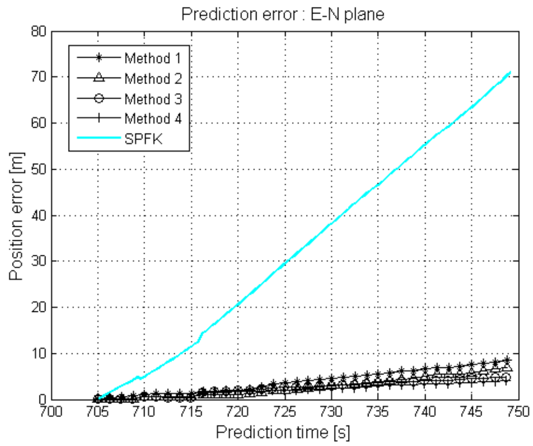

Figure 21: Error for GPS outage at t = 705[s].

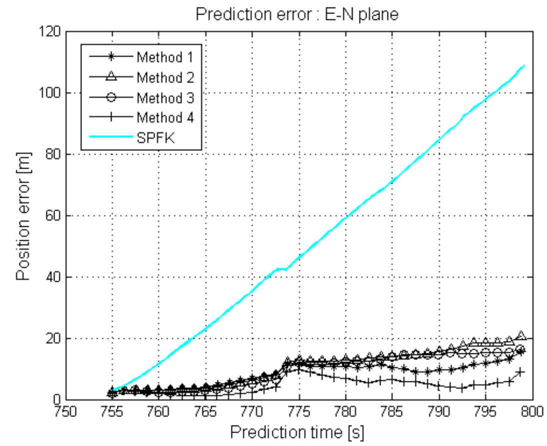

Figure 22: Error for GPS outage at t = 755[s].

Second data set: A second independent field test campaign to acquire data was taken by using the same GPS, at 1 Hz rate, IMU at 20 Hz, and another model of a small urban vehicle. The trajectory was the same of previous simulations at INPE campus, with similar velocities and accelerations, same initial orientation and starting position. Figures 23 and 24 show the interval [0s, 200s] from this data set used for simulations.

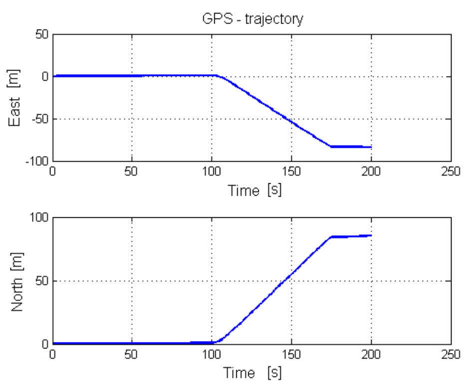

Figure 23: North-East GPS trajectory.

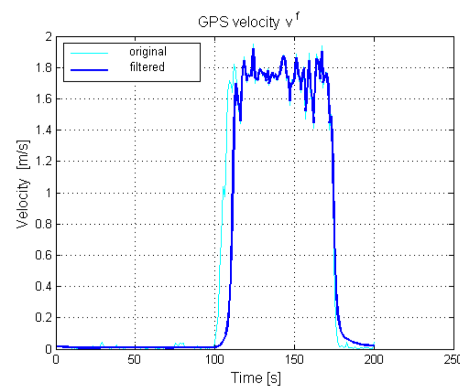

Figure 24: Forward vehicle GPS velocity.

The same simulation procedure from first data set was repeated and a 20 hidden neurons ANN with 40 iterations as stopping criterion was also chosen. Again a 20s, 40s and 60s data set length was assessed with similar results. Figure 25 shows a first experiment with a 20s data set, and the vehicle was in static position until t = 100s approximately, and then moving on. Figure 26 shows a second experiment, with a 60s data set, and the vehicle was in movement. The results obtained from both data set simulations demonstrate that ANN keeps similar modeling capacity independently of IMU turn on-off action and different GPS sampling rate. These results were expected since the vehicle only has been operated at very low velocities in straight line and curves, then the 1Hz and 2Hz GPS output rates have similar influence on vehicle's kinematic modeling.

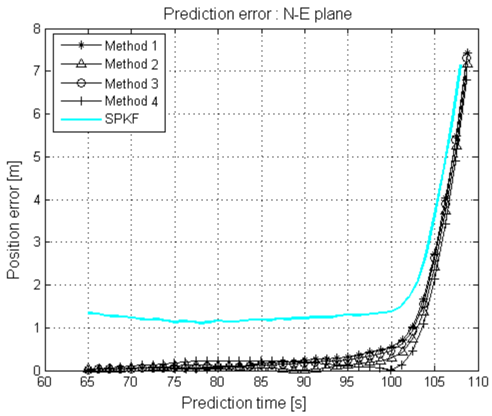

Figure 25: Error for GPS outage at t = 65[s].

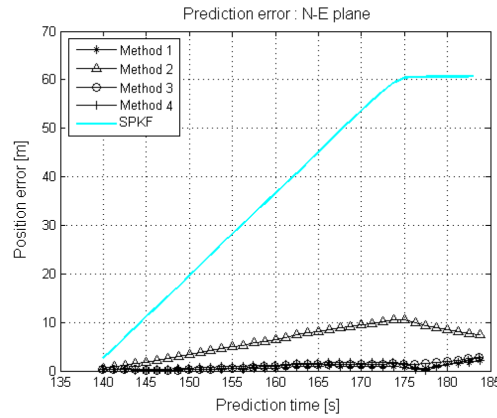

Figure 26: Error for GPS outage at t = 140[s].

5 Conclusions

Methods of training ANN, with an adaptive Kalman filtering algorithm, were tested with real navigation data, off-line and during GPS simulated outages. The proposed method 4, with capacity to continually extract information, by filtering individual pattern-by-pattern of training data, explores the possibility of updating the previously trained weights, during a defined time interval, giving to the neural network aided navigation system some real time training capacity. This is an important characteristic when using low cost IMU, based on MEMS sensors, since its noise characteristics may vary from one run to other, leading to large start-up errors. Hence, off-line training may not be useful to modeling the vehicle dynamics based on previous runs. Also, considering the Kalman adaptive solution, which is proposed to give some balance between the priori information and new learning information, the method aids the neural network to preserve some previous knowledge, acting as a time fading memory.

The proposed methods and ANN architecture were tested by using data acquired from a low cost MEMS IMU Crossbow CD400_200 and an Ashtech Z12 GPS receiver. Simulated results, with different vehicle dynamic situations, show that the proposed fourth method has positions errors of same order, when compared to other methods and achieves the position target error, of 20 meters in 30 seconds of simulated GPS outage. Also when compared to SPKF, operating only in prediction phase, the ANN schemes have the same magnitude position error when the vehicle is static and seem to have better performance when vehicle is in motion. These results indicate that the ANN was more capable of learning the vehicle's kinematics, for a certain time interval, than the modeling presented by the conventional KF based navigation system.

6 References

[1] El-Sheimy, N.; Niu X. "The promise of MEMS to the navigation community". Inside GNSS, march/april, 46-56 p., 2007. Available in: <http://www.insidegnss.com>. Last access: 12 Aug 2008.

[2] Chiang, K.-W.; El-Sheimy, N. "An alternative low cost MEMS IMU/GPS integration scheme". Coordinates, September, p.10-13, 2005. Available in: <www.coordinates.org>. Last access: 27 Mar 2008.

[3] Chiang, K.-W.; El-Sheimy, N. "INS/GPS integration using neural networks for land vehicle navigation applications". In: ION GPS 2002 - Institute of Navigation. Portland, OR. Proceedings of ION GPS 2002 Portland: ION, 2002. p. 535-544.

[4] Chiang, K.-W.; Huang Y.-W. "An intelligent navigator for seamless INS/GPS integrated land vehicle navigation applications". Applied Soft Computing. Elsevier B.V.: n. 8, p. 722-733, 2008.

[5] Noureldin et al. "GPS/INS integration utilizing dynamic neural networks for vehicular navigation". Information Fusion. Elsevier B.V. doi: 10.1016/j.inffus .2010.01.003. 2010. Available in: <www.elsevier.com/locate/inffus>.

[6] Marques Filho, E. A.; Rios Neto, A.; Kuga, H. K. "ANN applied to low cost INS/GPS integrated navigation systems". In: 8th International Conference on Mathematical problems in Engineering, Aerospace and Sciences, june 30 - july 3, INPE,

São José dos Campos, Brazil. Proceedings of ICNPAA 2010, p. 485-493. ISBN 978-1-908106-18-6, CD-ROM, 2011 Cambridge Scientific Publishers Ltd.

[7] Marques Filho, E. A. Rios Neto, A.; Kuga, H. K. “A low cost INS/GPS navigation system integrated with a multilayer feedforward neural network”. Journal of Aerospace Engineering, Sciences and Applications, Vol. II, No 2, p. 26-36, May – Aug. 2010. ISSN 2236-577X, Brazilian Aeroespace Association (AAB). Available at: < http://www.aeroespacial.org.br/jaesa/editions2.html>. Last acess: 06 jun 2012. Presented at: Simpósio Brasileiro de Engenharia Inercial 2010, 20-22 october, IPqM, Rio de Janeiro (SBEIN 2010).

[8] Marques Filho, E. A. Rios Neto, A.; Kuga, H. K. “A low cost INS/GPS navigation system integrated by an adaptive neural network training Kalman filtering methodology”. In: 10th Brazilian Congress on Computational Intelligence, 8-11 november, Fortaleza, Ceará, Brazil. CBIC 2011,CD-ROM.

[9] Kuga, H. K. et al. “Experimental static calibration of an IMU (Inertial Measurement Unit) based on MEMS”. In: 19th International Congress of Mechanical Engineering, 5-9 november, Brasília, DF Brazil. Proceedings of ABCM COBEM 2007.

[10] Brandt, A.; Gardner, J. F. “Constrained navigation algorithms for strapdown inertial navigation systems with reduced set of sensors”. In: American Control Conference, 24-26 june, 1998. Philadelphia, PA, USA. Proceedings of AACC-IEEE, 1998. p. 1848-1852.

[11] Haykin, S. Redes Neurais: princípios e práticas. Porto Alegre: Bookman, 2001. 900 p.

[12] Singhal, S.; Wu, L. “Training multilayer perceptrons with the extended Kalman algorithm”. In: TSOURETZKY D. S. Advances in Neural Information 1. San Mateo, CA. Morgan Kaufman: p. 133-140, 1989.

[13] Rios Neto, A. “Stochastic optimal linear parameter estimation and neural nets training is system modeling”. RBCM – Journal of the Braz. Soc. Mechanical Sciences, v. XIX, n. 2, p. 138-146, 1997.

[14] Freitas Pinto, R. L. U.; Rios Neto, A. “An optimal linear estimation approach to solve systems of linear algebraic equations”. Journal of Computational and Applied Mathematics, n. 33, p. 261-268, 1990.

[15] Chiang, K.-W.; Noureldin, A.; El-Sheimy, N. “A new weight updating method for INS/GPS integration architectures based on neural networks”. Measurement Science and Technology. Institute of Physics Publishing Ltd, n. 15, p. 2053-2061, 2004.

[16] Marques Filho, E. A. Sistema de navegação inercial GPS/INS de baixo custo com compensação de erros por redes neurais artificiais (in Portuguese). 2011. 147 p. (sid.inpe.br/mtc-m19/2011/11.07.22.08-TDI). Thesis (Doctorate in Space Mechanics and Control) - National Institute for Space Research, São José dos Campos, SP, Brazil, 2011. Available at: <http://urlib.net/8JMKD3MGP7W/3AP6SB5>. Last acess: 08 jun. 2012.

[17] Noureldin, A.; El-Shafie, A.; Bayoumi, M. “GPS/INS integration utilizing dynamic neural networks for vehicular navigation”. Information Fusion, Special Issue on Inteligent Transportation systems, Elsevier B.V., vol 12, issue 1, p. 48-57, 2011.