A LOW COST INS/GPS NAVIGATION SYSTEM INTEGRATED WITH A MULTILAYER FEED FORWARD NEURAL NETWORK

Edmundo A. Marques Filho

INPE - Divisão de Mecânica Espacial e Controle – DMC
São José dos Campos, Brasil
E-mail: amf@yahoo.com.br

Atair Rios Neto

INPE - Divisão de Mecânica Espacial e Controle – DMC
São José dos Campos, Brasil
E-mail: atairrn@uol.com.br

Helio Koiti Kuga

INPE - Divisão de Mecânica Espacial e Controle – DMC
São José dos Campos, Brasil
E-mail: hkk@dem.inpe.br

Abstract: *This article investigates the use of a multilayer feedforward artificial neural network into a GPS integrated low cost inertial navigation system based on MEMS sensors. The neural network is applied as an alternative of integration technique, with the purpose of providing better navigation solutions, during the lack of information in GPS outages portions of time. An input-output neural network signals model is proposed, based on a set of simplified terrestrial vehicle navigation equations. Also an adaptive Kalman filter training methodology is tested with real navigation data. Preliminary simulated numerical results are presented, based on urban vehicular positioning application data trials, acquired from low cost Crossbow CD400-200 IMU and an Ashtech Z12 GPS receiver.*

Keywords: *INS/GPS/Neural Network navigation, low cost navigation, IMU, INS, GPS, neural networks, MEMS.*

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, is suitable for sensor integration and capable of providing accurate and low cost navigation.

The need for alternative source of POS/NAV information arises because GPS does not work properly in all environments. The GPS receiver can suffer 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 and the integration of GPS with inertial navigation systems (INS), by using stochastic parameter estimation techniques, such as Kalman filters, are frequently used.

Micro-electromechanical systems (MEMS) based inertial sensors application for navigation purposes, has been developed due to its low price, small size, lightweight, and lower power consumption El-Sheimy and Niu (2008). 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 IMU [2,3,4,5]. 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 Chiang and El-Sheimy (2008).

In what follows the use of an artificial neural network scheme to integrate GPS with low cost MEMS inertial sensors INS, based on an adaptive ANN training Kalman filtering methodology is presented and preliminarily tested with urban vehicular positioning application data. Section 2 presents the inertial navigation equations constrained to the land vehicle type of motion. Section 3 briefly introduces the use of ANN and presents the

Presented at the VI Simpósio Brasileiro de Engenharia Inercial, Rio de Janeiro, Brazil, October, 2010

proposed Kalman filter based methodology. In section 4 the results of preliminary testing are presented. Finally in section 5 a few conclusions are drawn.

2 Constrained inertial navigation equations

Constraints on the motion of land vehicles can be defined and used to derive a reduced set of equations of motion. Brandt and Gardner (1998) define the following constraints: (i) direction of the vehicle's velocity coincides with direction of the vehicle's longitudinal axis; (ii) pitch and roll angles of the of the vehicle's body relative to the Earth surface are small; and (iii) vehicle always remains on the Earth surface. Under these constraints, a set of equations of motion can be defined Brandt and Gardner (1998):

$$\dot{v}^f = f_u^b - g \cdot sin\theta \quad (1)$$

$$v^f \cdot r = f_v^b + g \cdot sin\phi \cdot cos\theta \quad (2)$$

$$v^f \cdot w = -f_w^b - g \cdot cos\phi \cdot cos\theta \quad (3)$$

Where v^f is the forward velocity, defined in the direction of movement; g is the gravity acceleration; ϕ, θ, ψ are roll, pitch and yaw vehicle attitude angles; p, q, r are the gyros outputs measured angular velocities and f_u^b, f_v^b, f_w^b are the accelerometer outputs measured specific forces. The attitude angles can be computed as:

$$\dot{\phi} = p + (sin\phi \cdot tan\theta) \cdot q + (cos\phi \cdot tan\theta) \cdot r \quad (4)$$

$$\dot{\theta} = cos\phi \cdot q - sin\phi \cdot r \quad (5)$$

$$\dot{\psi} = \frac{sin\phi}{cos\theta} \cdot q + \frac{cos\phi}{cos\theta} \cdot r \quad (6)$$

These differential equations can be further simplified for usual vehicle urban use, and will define the ANN input/output model:

$$\dot{v}^f \approx f_u^b \quad (7)$$

$$\dot{\psi} \approx r \quad (8)$$

3 Artificial neural networks and Kalman filter methodology

A. 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. (9)) by adjustment of its synaptic weights, via a supervised learning process Haykin (2001).

$$f \in C : x \in D \subset \Re^n \to y \in \Re^m \quad (9)$$

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:

$$\{(\mathbf{x}(t), \mathbf{y}(t)): \quad \mathbf{y}(t) = f(\mathbf{x}(t)), \quad t = 1,2 \cdots L\} \quad (10)$$

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

$$\hat{\mathbf{y}}(t) = \hat{f}(\mathbf{x}(t), \mathbf{w}(t)) \quad (11)$$

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 $\mathbf{R}^{-1}$ Neto (1997):

$$\mathrm{J}(\mathbf{w})=\frac{1}{2}\left[(\mathbf{w}-\overline{\mathbf{w}})^{\mathrm{T}}\overline{\mathbf{P}}^{-1}(\mathbf{w}-\overline{\mathbf{w}})+\sum_{t=1}^{L}\left(\left(\mathbf{y}(t)-\hat{f}(\mathbf{x}(t),\mathbf{w})\right)^{\mathrm{T}}\mathbf{R}^{-1}\left(\mathbf{y}(t)-\hat{f}(\mathbf{x}(t),\mathbf{w})\right)\right)\right] \tag{12}$$

B. Kalman filtering training method

Singhal and Wu (1989) have proposed the use of the extended Kalman filtering as training algorithm, viewing ANN as a stochastic parameter estimation problem. Rios Neto (1997) has further explored this concept and proposed an algorithm that features parallel processing, in order to avoid large computational charges, 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 (1997) proposed solution is implemented in this paper, adapted to an inertial navigation problem, and is summarized in what follows.

The mapping from Eq. (11) 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. (12):

$$\alpha(t)\cdot[\mathbf{y}(t)-\overline{\mathbf{y}}(t,\overline{\mathbf{w}})]+\mathbf{H}(t,\overline{\mathbf{w}})\cdot\overline{\mathbf{w}}=\mathbf{H}(t,\overline{\mathbf{w}})\cdot\mathbf{w}(t) \tag{13}$$

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:

$$\mathbf{H}(t,\overline{\mathbf{w}})=\hat{f}_{w}(\mathbf{x}(t),\overline{\mathbf{w}})=\left.\frac{\partial\hat{f}(\mathbf{x}(t),\mathbf{w})}{\partial\mathbf{w}}\right|_{\mathbf{w}=\overline{\mathbf{w}}} \tag{14}$$

And, in a compact notation:

$$\mathbf{z}(t)=\alpha(t)\cdot[\mathbf{y}(t)-\overline{\mathbf{y}}(t,\overline{\mathbf{w}})]+\mathbf{H}(t,\overline{\mathbf{w}})\cdot\overline{\mathbf{w}} \tag{15}$$

Then a stochastic linear estimation problem can formulated as:

$$\overline{\mathbf{w}}(t)=\mathbf{w}(t)+\overline{\mathbf{e}} \tag{16}$$

$$\mathbf{z}(t)=\mathbf{H}(t)\cdot\mathbf{w}(t)+\mathbf{v}(t) \tag{17}$$

$$\mathbf{v}(t)=\mathrm{N}(\mathbf{0},\mathbf{R}) \tag{18}$$

$$\overline{\mathbf{e}}=\mathrm{N}(\mathbf{0},\overline{\mathbf{P}}) \tag{19}$$

Where **R** and $\overline{\mathbf{P}}$, respectively, are the covariance matrices of $\mathbf{v}$ and $\overline{\mathbf{e}}$ error vectors. A sequential Kalman filter solution, for the estimation problem, with $t = 1, \cdots, L$, is given by:

$$\mathbf{K}(t)=\overline{\mathbf{P}}(t)\mathbf{H}^{\mathrm{T}}(t,\overline{\mathbf{w}})\cdot\left[\mathbf{R}(t)+\mathbf{H}(t,\overline{\mathbf{w}})\overline{\mathbf{P}}(t)\mathbf{H}^{\mathrm{T}}(t,\overline{\mathbf{w}})\right]^{-1} \tag{20}$$

$$\hat{\mathbf{w}}(t)=\overline{\mathbf{w}}(t)+\mathbf{K}(t)\cdot[\mathbf{z}(t)-\mathbf{H}(t,\overline{\mathbf{w}})\overline{\mathbf{w}}(t)] \tag{21}$$

$$\mathbf{P}(t)=\overline{\mathbf{P}}(t)-\mathbf{K}(t)\mathbf{H}(t,\overline{\mathbf{w}})\overline{\mathbf{P}}(t) \tag{22}$$

$$\overline{\mathbf{P}}(t+1)=\mathbf{P}(t)+\mathbf{Q}(t) \tag{23}$$

$$\overline{\mathbf{w}}(t+1) = \hat{\mathbf{w}}(t) \tag{24}$$

For the above equations, **K** is the Kalman gain and **Q** is the noise process covariance matrix. At the end of the each iteration, t = L. If a stopping criterion is satisfied, then the a priori values for a new data set are $\overline{\mathbf{w}} = \hat{\mathbf{w}}(L)$ 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 $\overline{\mathbf{P}}(1) = \overline{\mathbf{P}}_0$.

C. Adaptive Kalman Filter solution for ANN training

Due to algorithm bad numerical behavior and observation model errors, divergence may occur as a large data set is processed. In this situation the neural network can loose its capacity of keeping learning as new data are processed. Rios Neto (1997) has proposed an adaptive procedure based on a criterion of statistical consistency to balance a priori information priority with that of new learning information:

$$\beta \cdot E\left[v_j^2(t)\right] = \mathbf{H}_j(t,\overline{\mathbf{w}}) \cdot \left[\mathbf{P}(t) + \mathbf{Q}(t)\right] \cdot \mathbf{H}_j^T(t,\overline{\mathbf{w}}) \tag{25}$$

Where j = 1, …, m observations, and β is to be adjusted. When β = 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 more value than the stored one.

Equation (25) can be considered as an observation and expanded into the following associated exact estimation problem Pinto and Neto (1990), to be processed with a Kalman filter algorithm:

$$0 = \mathbf{q}(t) + \overline{\mathbf{e}}^q(t) \tag{26}$$

$$\mathbf{z}^q(t+1,\beta) = \mathbf{H}^q(t+1) \cdot \mathbf{q}(t) + \mathbf{v}^q(t+1) \tag{27}$$

$$E\left[\overline{\mathbf{e}}^q\right] = 0\ , \quad E\left[\overline{\mathbf{e}}^q \cdot \overline{\mathbf{e}}^{q^T}\right] = \mathbf{I}_{n_w} \tag{28}$$

$$\begin{aligned} &E\left[\mathbf{v}^q(t+1)\right] = 0\ , \\ &E\left[\mathbf{v}^q(t+1) \cdot \mathbf{v}^{q^T}(t+1)\right] = \mathbf{R}^q(t+1) = 0 \end{aligned} \tag{29}$$

Where:

$$q_k(t) = \begin{cases} 0 & \text{se} \quad \hat{q}_k < 0 \\ \hat{q}_k & \text{se} \quad \hat{q}_k \geq 0 \end{cases} \tag{30}$$

$$\mathbf{Q}(t) = diag\left[\mathbf{q}(t)\right] \tag{31}$$

D. Proposed ANN architecture for sensor integration

The proposed ANN architecture uses a three layer feedforward, with the adaptive extended Kalman filtering learning algorithm. 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 vehicle dynamics and the possible observations from GPS. Fig. 1 shows the selected ANN:

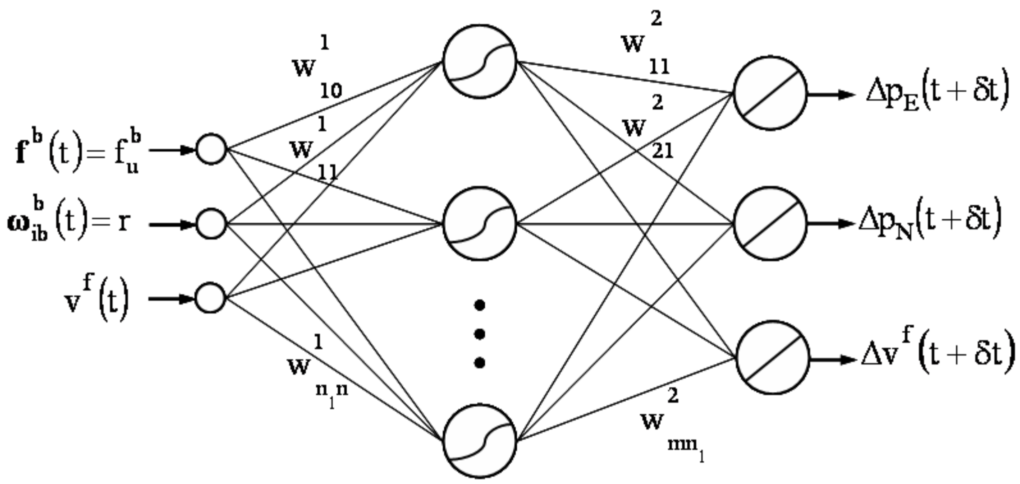

Figure 1. One hidden layer ANN input/output signals

The input layer signals are 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 $(t+\delta t)$, where δt is the IMU output frequency, and v^f can be observed from:

$$v_{GPS}^f = \sqrt{\left(v_N^{GPS}\right)^2 + \left(v_E^{GPS}\right)^2} \quad (32)$$

The ANN navigation solution is given by:

$$\mathbf{p}^n = \begin{bmatrix} p_E & p_N & v^f \end{bmatrix}^T \quad (33)$$

While in training mode, an error vector $\mathbf{e} = [\mathbf{e}_{POS} \quad \mathbf{e}_{VEL}]$ is generated for supervised training:

$$\mathbf{e}_{POS}(t+\delta t) = \Delta\mathbf{p}_{GPS}^n(t+\delta t) - \Delta\mathbf{p}_{ANN}^n(t+\delta t) \quad (34)$$

During the prediction mode, the ANN increments output are added to the last GPS navigation solution, $\mathbf{p}^n(t_u) = \mathbf{p}_{GPS}^n(t_u) = \mathbf{p}_u^n$, where t_u denotes the last time signal before GPA outage. Also, in this mode, the velocity v^f input comes from a feedback output summation.

$$\mathbf{p}_{NAV}^n(t_u + i\delta t) = \mathbf{p}_u^n + \sum_i \Delta\mathbf{p}_{RNA}^n(t_u + i\delta t), \; i = 1, 2, ... \quad (35)$$

4 Simulations methodology and preliminary results

A. Field test procedure

Field tests were conducted in a land vehicle, for urban usage, at INPE campus, in São José dos Campos (23.2113^{o}S, 314.1408^{o}W). A low cost Crossbow CD400-200 IMU, based on MEMS technology, and an Ashtech Z12 GPS receiver, with single antenna, were used to acquire data. All information was post-processed to test proposed training algorithms and methods, by numerical simulations. To validate 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 urban land vehicle navigation purposes El-Sheimy and Niu (2008).

B. Methods of training

In order to explore the adaptive Kalman filtering training algorithm, methods of dealing with data patterns are revised and some modifications are proposed. For all cases, an MLP was used with a 20 neurons hidden layer, with sigmoidal activation functions; and 40 iterations, or epochs, as a stopping criterion.

First method is the simplest one. Data from IMU and GPS, when available, are processed and stored sequentially in time, into data sets, or windows of data, of some size. When each data set is completed, a training procedure starts, with initial weights obtained by random initializations of small values [-0.5, 0.5]. The resulting weights are stored as the best solution, if GPS outages occur, while a new data set is being completed.

Second method is the same of first, with one modification. After the first data set has being trained, the next training procedures have initial weights obtained from previous stored solution Chiang *et al.*(2004). The resulting weights are stored.

Third method is the same of second, with the following proposed modification: 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. 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 the following: after the first data set training is completed, the stored weights are only updated by filtering pattern-by-pattern new data, during some determined time interval, or until a GPS outage starts, when the resulting updated weights are used. The process is resumed after the GPS outage, or after the time interval, when a new data set is completed.

C. Simulation and preliminary results

For the first experiment, the vehicle was in static position. Information from GPS and IMU are shown in Fig. 2 and 3. The position errors for GPS outages, starting at t = 115[s] and t = 165[s], are shown in Fig. 4 and Fig. 5, where methods 1 to 4 were described in section *B*. During the 45 seconds simulated GPS outages, the vehicle prediction position error, with respect to north and east GPS information, is given by:

$$e_{pred_{POS}} = \sqrt{e^2_{pred_{EAST}} + e^2_{pred_{NORTH}}} \tag{36}$$

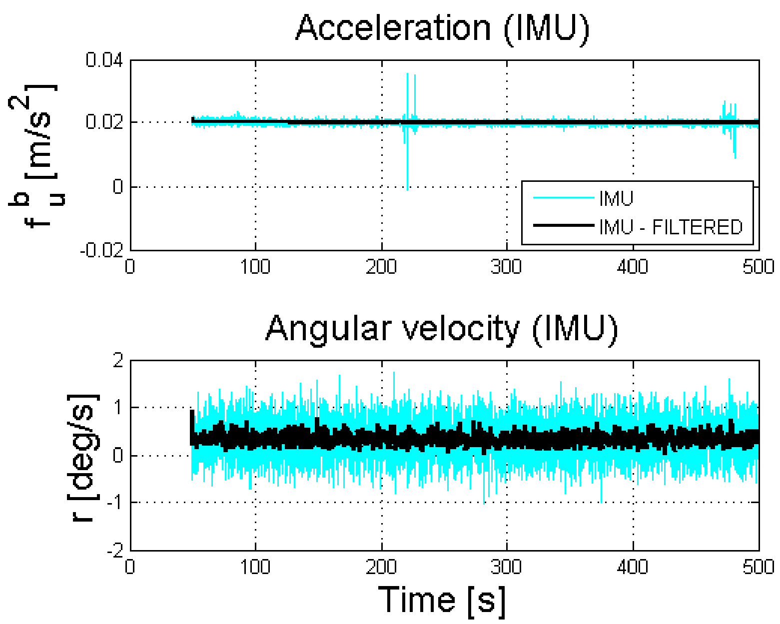

Figure 2. IMU output for first experiment

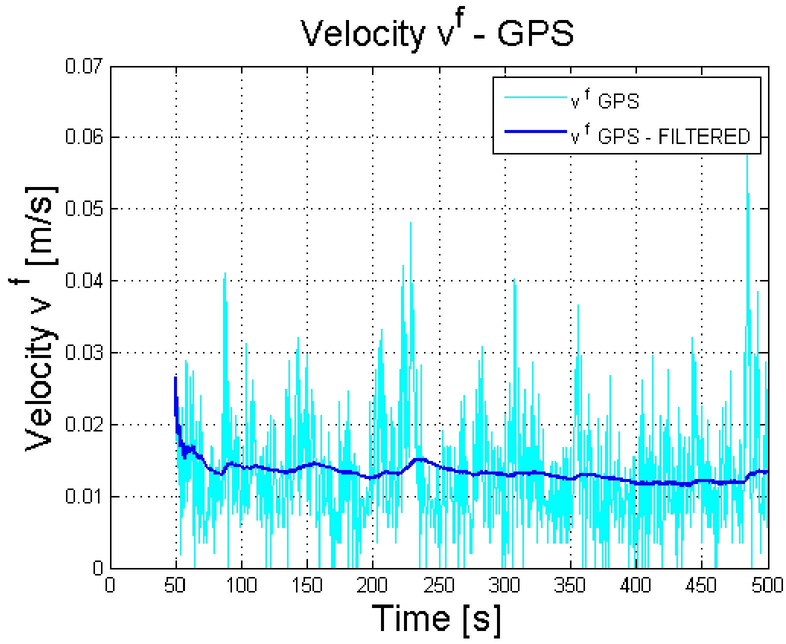

Figure 3. Forward velocity for first experiment

It should be noted that, for the first experiment, simulation starts around t = 50[s]. Since data set length is 20[s], method 4 starts to filtering 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 45[s] and 95[s], respectively. For methods 1 to 3, completed data set were trained two more time, until first outage, and four more times until second outage, when compared to method 4, with one data set training completed.

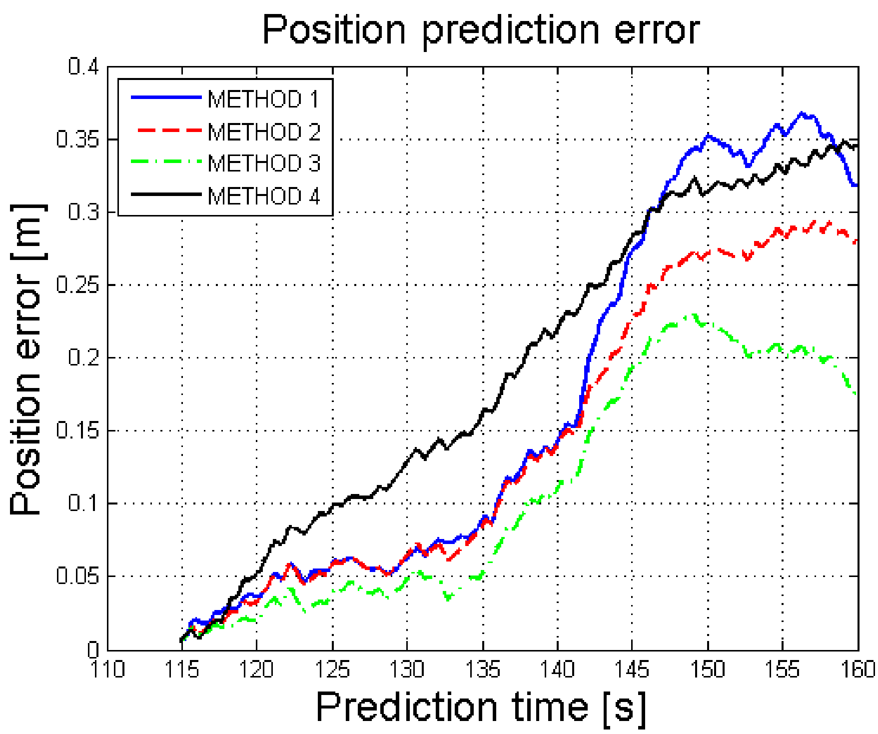

Figure 4. Prediction error for GPS outage at t = 105[s]

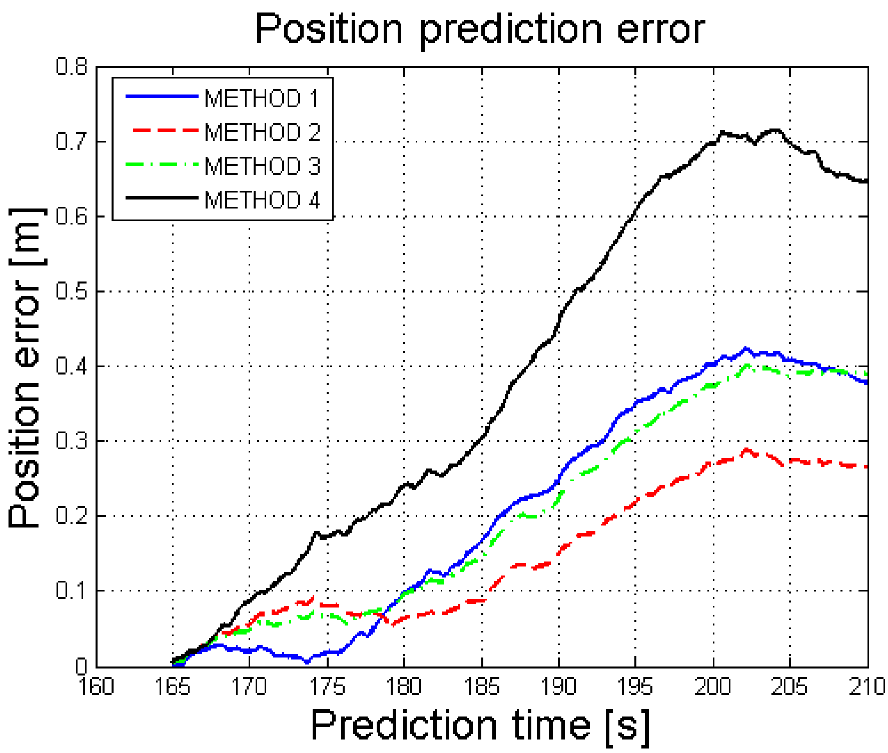

Figure 5. Prediction error for GPS outage at t = 155[s]

In the second experiment, the vehicle is in movement. Information from GPS and IMU are shown in Fig. 6 and 7. The position errors for GPS outages, starting at t = 715[s] and t = 765[s], are shown in Fig. 8 and Fig. 9.

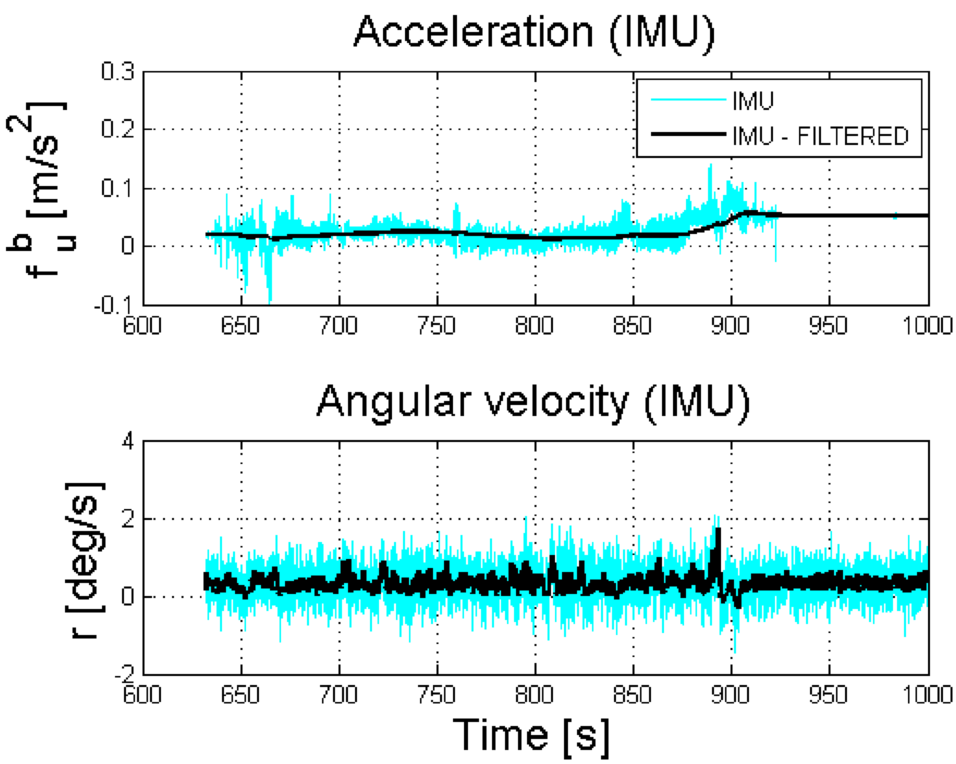

Figure 6. IMU output for second experiment

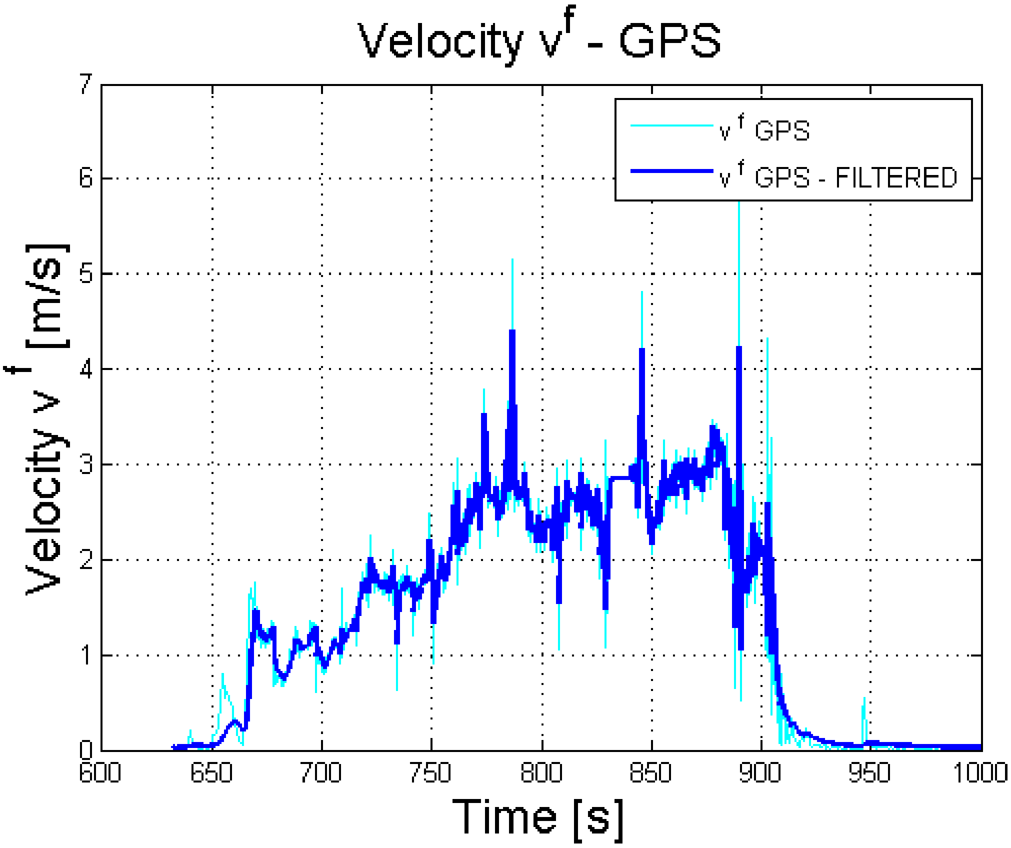

Figure 7. Forward velocity for second experiment

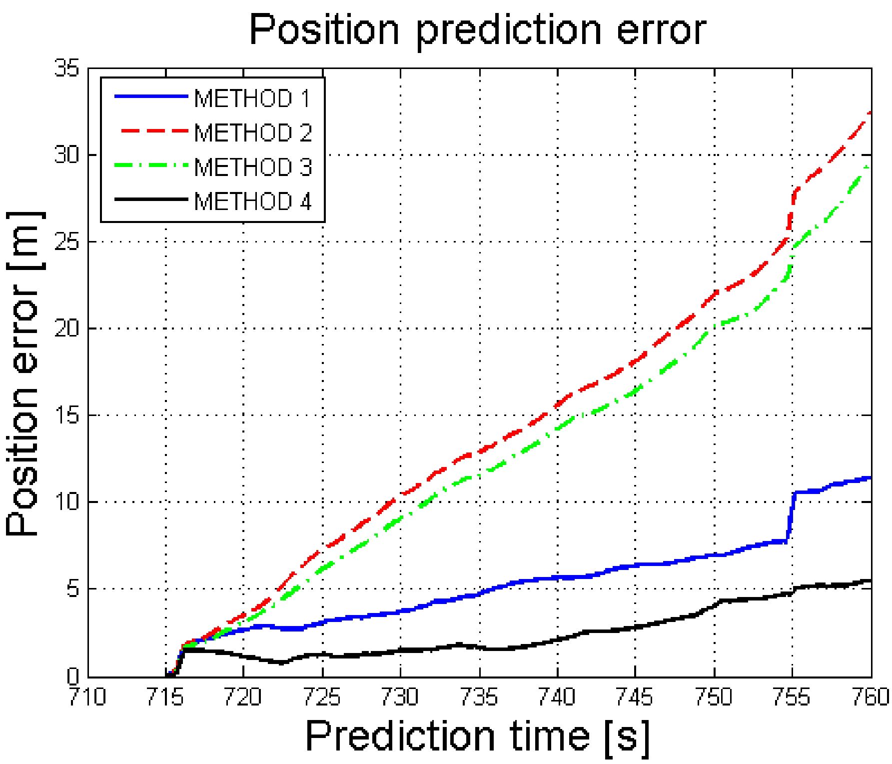

Figure 8. Prediction error for GPS outage at t = 705[s]

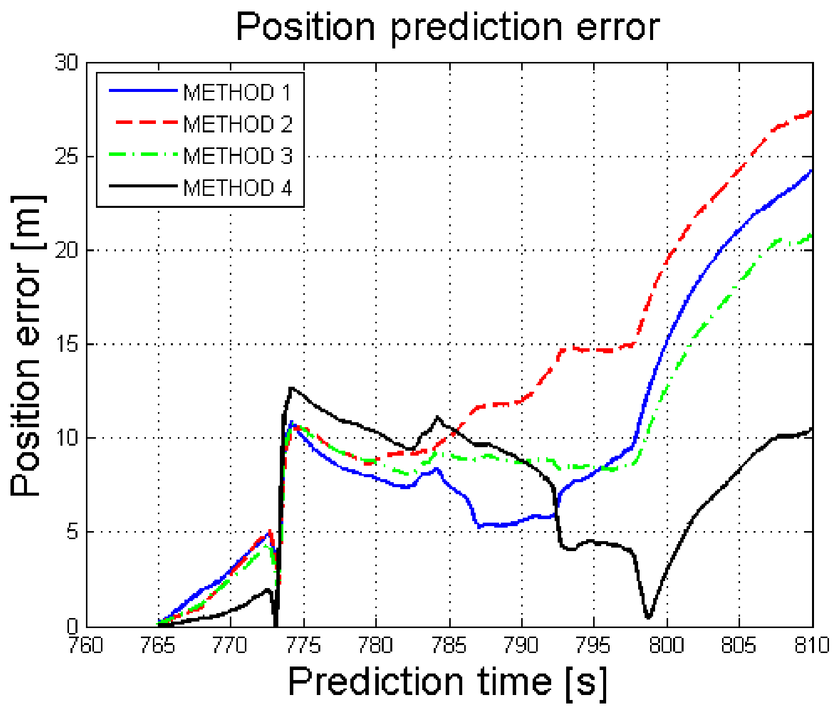

Figure 9. Prediction error for GPS outage at t = 755[s]

Again, it should be noted that, for the second experiment, simulation starts around t = 630[s]. Since data set length is 20[s], method 4 starts to filtering individual training data pairs around t = 650[s], after the first data set is completed. Then, the total filtering time, until the GPS simulated outages, is about 65[s] and 115[s], respectively. For methods 1 to 3, completed data set were trained three more times, until first outage, and five more times until second outage, when compared to method 4, with one data set training completed.

5 Conclusions

Methods of training ANN, with an adaptive Kalman filtering algorithm, were tested with real navigation data, during GPS simulated outages. The proposed method, with capacity to continually extract information, by filtering individual pattern-by-pattern of training data, explored 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 issue 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, offline training may not be useful to modeling the vehicle dynamics based on previous runs.

The Kalman adaptive solution, proposed to give some balance between the priori information and new learning information, aids the neural network to preserve some previous knowledge, acting as a time fading memory.

Simulated results, with different vehicle dynamic situations, suggest 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.

6 References

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 … Philadelphia, AACC-IEEE, 1998, p. 1848-1852.

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.

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 … Portland: ION, 2002. p. 535-544.

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.

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.

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.

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.

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

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

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.

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.