10th Brazilian Congress on Computational Intelligence (CBIC'2011), November 8 to 11, 2011, Fortaleza, Ceará Brazil

A LOW COST INS/GPS NAVIGATION SYSTEM INTEGRATED BY AN ADAPTIVE NEURAL NETWORK TRAINING KALMAN FILTERING METHODOLOGY

Edmundo A. Marques Filho, Atair Rios-Neto and Helio Koiti Kuga

INPE/ETE/DMC – São José dos Campos, S.P., Brazil
edmundo_amf@yahoo.com.br, atairrn@uol.com.br, hkk@dem.inpe.br

Abstract – This paper presents the development and testing of an alternative approach to make it feasible to use low cost inertial measurements units (IMU) in integrated Inertial Navigation System (INS) and Global Positioning System (GPS) applied to positioning and navigation (POS/NAV). Low cost inertial measurement units (IMU) based navigation systems have the disadvantage of accumulating increasingly continuous errors in great extension, leading to poor system performance. Moreover GPS does not work in all environments, or can not provide reliable solutions, under certain circumstances, during some time interval. The integration of both systems can handle and overcome their limitations. There are different solutions to fulfill information during GPS blockage and integrated systems with inertial sensors and GPS are frequently used with stochastic parameter estimation techniques. This work investigates the use of artificial neural network (ANN), to learn and compensate for IMU errors such as to provide better NAV/POS solutions, during the lack of information in GPS outages portion of time. It is proposed to model the input-output ANN signals based on a set of constrained land vehicle navigation equations and the use of an adaptive ANN training Kalman filtering methodology. Numerical simulation results are presented based on urban vehicular positioning application data, acquired from low cost Crossbow CD400-200 inertial measurement unit and an Astech Z12 GPS receiver. Comparison with equally simulated results of a current more standard Kalman filter INS/GPS integration scheme gives a indication of feasibility and potential of the proposed approach.

Keywords – Low cost navigation, ANN, inertial navigation, GPS, IMU, Kalman filter.

1 Introduction

The global positioning system NAVSTAR-GPS, or simply GPS, is a one-way, real-time and world-wide radio navigation system based on a satellite constellation. The system provides a globally available signal that allows a dedicated receiver to compute accurately, in real time, its position (longitude, latitude and altitude). It is also usable for accurate time transfer and velocity estimation. GPS is portable, has low power consumption, and is suitable for sensor integration.

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 (KF), is frequently used. However, there is the necessity of modeling the sensor error dynamics and usually linearization is used with consequent poor approximations of navigation errors compensation and correction.

Application of micro-electromechanical systems (MEMS) based inertial sensors, for navigation purposes, has been developed due to its low price, small size and lightweight, and lower power consumption [1]. But, 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.

Recently, 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 mostly 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 [2].

This paper is an extension from previous works [6,7] 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 [8].

In what follows, closely to what was developed in [6,7], Section 2 presents the inertial navigation equations constrained to a land vehicle type of motion and Section 3 briefly introduces the use of ANN and presents the proposed Kalman filter based methodology. In section 4 the results of testing are presented and compared with a current more standard Kalman filter INS/GPS integration scheme. Finally in section 5 a few conclusions are drawn.

2 Inertial Navigation and Constrained Motion Equations

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.

To assess the simulations, the proposed method results are compared to a 15 state Sigma-Point Kalman Filter (SPKF) INS/GPS integration loosely coupled scheme, under the same IMU and GPS data set. The SPKF state vector is given by [8]:

$$\mathbf{x} = \begin{bmatrix} \mathbf{r} & \mathbf{v} & \boldsymbol{\theta} & \mathbf{b}_a & \mathbf{b}_g \end{bmatrix}^T$$

Where:

$\mathbf{r} = \begin{bmatrix} \varphi & \lambda & h \end{bmatrix}^T$ are the geodetic latitude, longitude and height,

$\mathbf{v} = \begin{bmatrix} v_N & v_E & v_D \end{bmatrix}^T$ are the north, east and down velocity components,

$\boldsymbol{\theta} = \begin{bmatrix} \phi & \theta & \psi \end{bmatrix}^T$ are the roll, pitch and yaw attitude angles,

$\mathbf{b}_a = \begin{bmatrix} b_u & b_v & b_w \end{bmatrix}^T$ are the accelerometers x,y and z axes biases,and

$\mathbf{b}_g = \begin{bmatrix} b_p & b_q & b_r \end{bmatrix}^T$ are the gyroscopes x,y and z axes biases.

In a loosely coupled system, the KF GPS receiver 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 position, velocity and attitude navigation solution, as shown in Figure 1.

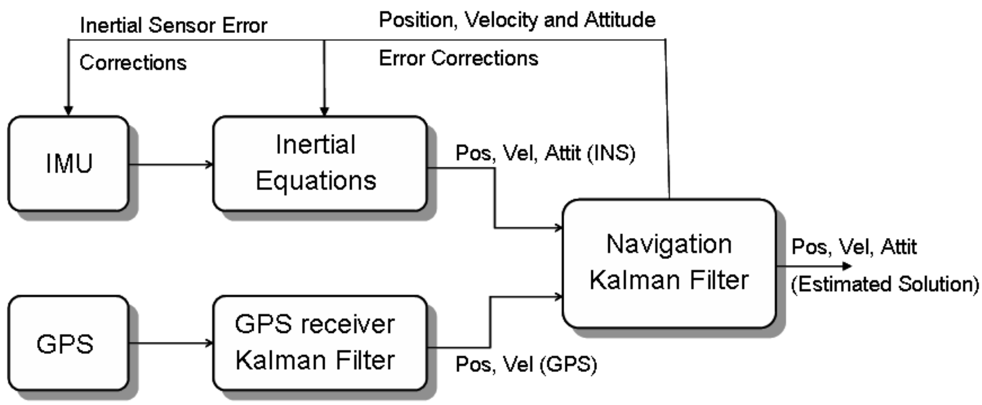

Figure 1 – Conventional loosely coupled INS/GPS scheme

Constraints on the movement of land vehicles can be defined and used to derive a reduced set of equations of motion. Brandt and Gardner [9] 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 equations of motion can be defined [9]:

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

$$v^f \cdot r = f_v^b + g \cdot \sin\phi \cdot \cos\theta \tag{2}$$

$$v^f \cdot w = -f_w^b - g \cdot \cos\phi \cdot \cos\theta \tag{3}$$

Where v^f is the forward velocity, defined in the direction of movement; g is the gravity acceleration; $\boldsymbol{\Phi} = \begin{bmatrix} \phi & \theta & \psi \end{bmatrix}^T$ are roll, pitch and yaw vehicle attitude angles; $\boldsymbol{\omega}_{ib}^b = \begin{bmatrix} p & q & r \end{bmatrix}^T$ are the angular velocity gyro output and $\mathbf{f}^b = \begin{bmatrix} f_u^b & f_v^b & f_w^b \end{bmatrix}^T$ are the specific forces accelerometer output. The attitude angles can be computed as:

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

$$\dot{\theta} = \cos\phi \cdot q - \sin\phi \cdot r \tag{5}$$

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

These differential equations can be further simplified for usual vehicle urban use:

$$\dot{v}^{f} \approx f_{u}^{b} \tag{7}$$

$$\dot{\psi} \approx r \tag{8}$$

The equations (7) and (8) will define the ANN input/output model later on.

3 Artificial Neural Networks 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. (9)) by adjustment of its synaptic weights, via a supervised learning process [10].

$$f \in C : x \in D \subset \Re^{n} \rightarrow y \in \Re^{m} \tag{9}$$

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:

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

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 $\hat{\mathbf{y}}(t)$:

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

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 $\mathbf{R}^{-1}$ [12]:

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

3.2 Kalman filtering training method

Singhal and Wu [11] have proposed the use of the extended Kalman filtering as training algorithm, viewing ANN as a stochastic parameter estimation problem. Rios Neto [12] 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 [12] proposed solution is implemented in this paper, adapted to an inertial navigation problem, and is summarized in what follows.

The mapping in 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):

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

Or, in a compact notation:

$$\mathbf{z}(t) = \alpha(t) \cdot [\mathbf{y}(t) - \overline{\mathbf{y}}(t, \overline{\mathbf{w}})] + \mathbf{H}(t, \overline{\mathbf{w}}) \cdot \overline{\mathbf{w}} \tag{14}$$

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:

$$\mathbf{H}(\mathrm{t},\overline{\mathbf{w}}) = \hat{\mathrm{f}}_{\mathrm{w}}(\mathbf{x}(\mathrm{t}),\overline{\mathbf{w}}) = \left.\frac{\partial \hat{\mathrm{f}}(\mathbf{x}(\mathrm{t}),\mathbf{w})}{\partial \mathbf{w}}\right|_{\mathbf{w}=\overline{\mathbf{w}}} \tag{15}$$

Then a stochastic linear estimation problem can formulated as:

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

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

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

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

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 $\mathrm{t} = 1, \cdots, \mathrm{L}$, is given by:

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

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

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

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

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

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, the training is finished, then $\overline{\mathbf{w}} = \hat{\mathbf{w}}(\mathrm{L})$ and $\overline{\mathbf{P}} = \hat{\mathbf{P}}(\mathrm{L})$. Otherwise a new iteration starts, with the initial conditions $\overline{\mathbf{w}}(1) = \hat{\mathbf{w}}(\mathrm{L})$ and $\overline{\mathbf{P}}(1) = \overline{\mathbf{P}}_0$.

3.3 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 [12] has proposed an adaptive procedure based on a criterion of statistical consistency to balance a priori information priority with that of new learning information:

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

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 less value than the stored one. Equation (25) can be considered as an observation and expanded into the following associated exact estimation problem [13], to be processed with a Kalman filter algorithm:

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

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

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

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

Where:

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

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

3.4 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. Figure 2 shows the selected ANN:

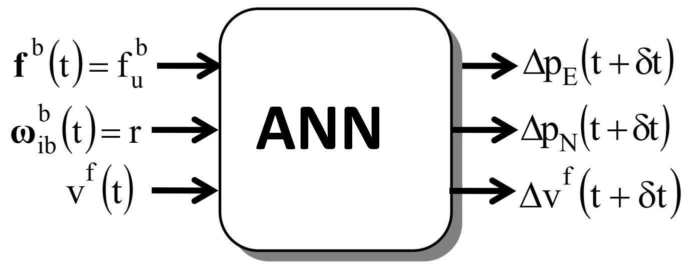

Figure 2 - ANN input/output signals

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

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

The ANN navigation solution is given by:

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

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

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

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.

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

4 Simulation Methodology and Preliminary 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). 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. Figures 3 and 4 show the interval [50s, 815s] used for simulations.

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 [1].

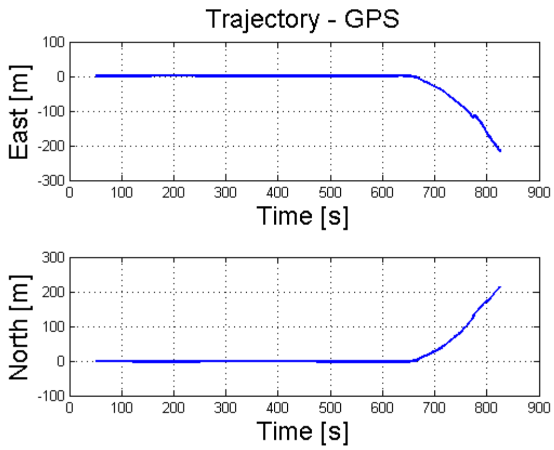

Figure 3: North-East GPS trajectory

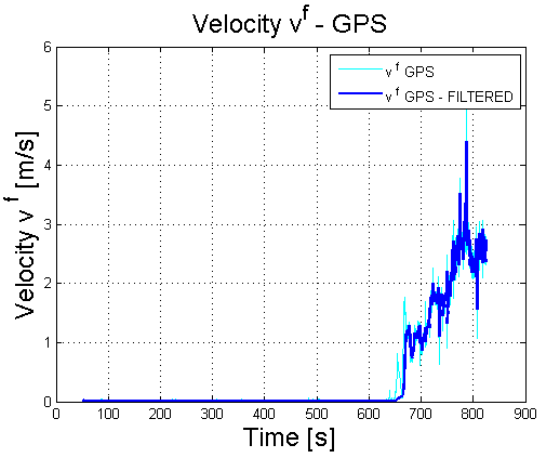

Figure 4: Forward vehicle GPS velocity

4.2 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 [14]. 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 is 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.

4.3 Proposed Adaptive Kalman filter training method verification

The proposed adaptive extended Kalman filter (AEKF) training solution, presented in Section 3, can be verified in an off-line function approximation. A simulation of 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, and the AEKF method is compared to traditional Levenberg-Marquardt (LM).

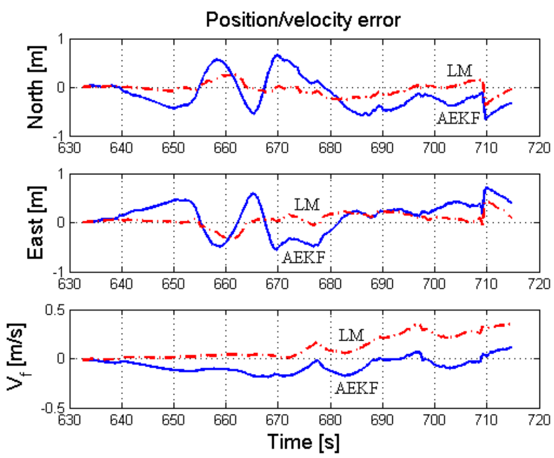

Figure 5: Position and velocity errors comparison

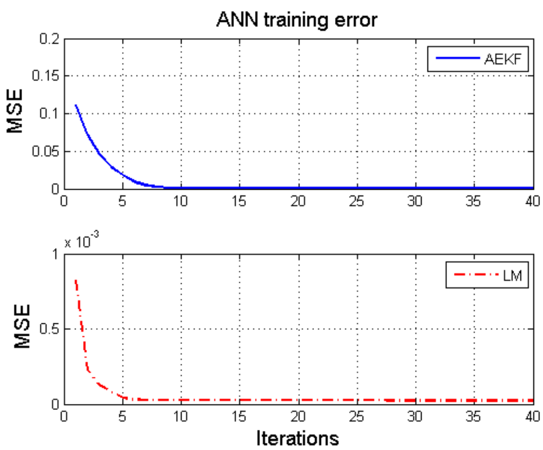

Figure 6: ANN MSE training errors

It can be observed from Figure 5 that North and East positions and velocity errors have the same magnitude for both methods, and Figure 6 shows that LM method converges in fewer iterations but AEKF method has a lower final mean squared error

(MSE) after the iterations. Also it should be noted that sequential training patterns is only allowed for AEKF method (LM is a batch method).

4.4 Simulation and preliminary results

For the first experiment, the vehicle was in static position. Information from GPS and IMU [50s, 200s] interval are used. The position errors for GPS outages, starting at t = 105 [s] and t = 155[s], are shown in Figures 7 and 8, where methods 1 to 4 were described in section 4.2. During the 45 seconds simulated GPS outages, the vehicle prediction position error, with respect to north and east GPS information, is given by:

$$e_{pred_{POS}} = \sqrt{e^2_{pred_{EAST}} + e^2_{pred_{NORTH}}} \quad (37)$$

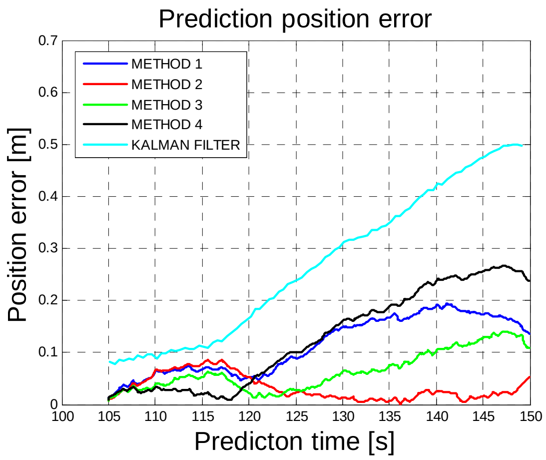

Figure 7 - Error for GPS outage at t = 105[s]

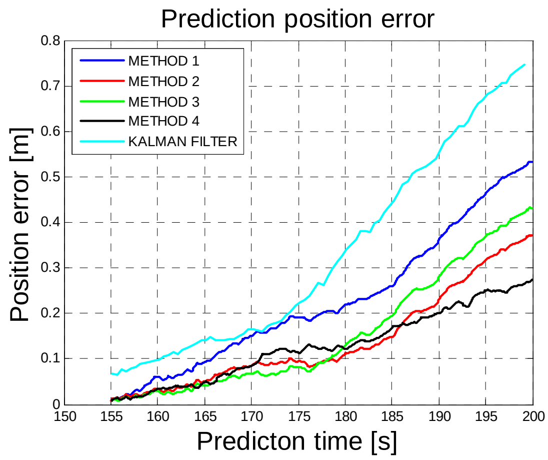

Figure 8 - Error for GPS outage at t = 155[s]

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 35 [s] and 85 [s], respectively. 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 is in movement. Information from GPS and IMU [630s, 800s] interval are used. The position errors for GPS outages, starting at t = 705 [s] and t = 755[s], are shown in Figures 9 and 10.

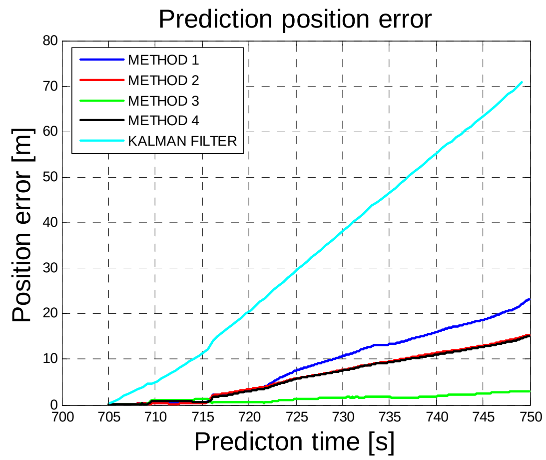

Figure 9 - Error for GPS outage at t = 705[s]

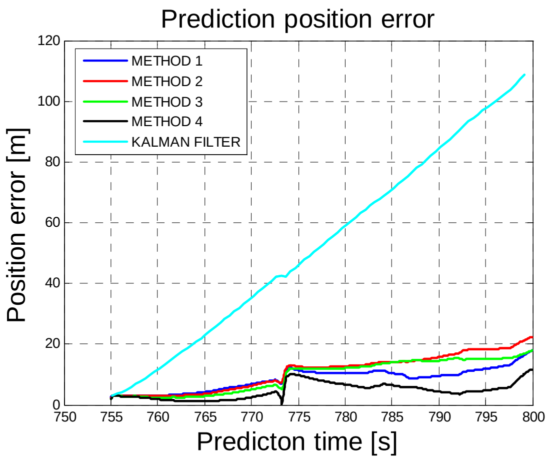

Figure 10 – Error for GPS outage at t = 755[s]

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 20 [s], method 4 starts to filter 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 55 [s] and 105 [s], respectively. For methods 1 to 3, completed data set were trained two 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, off-line and during GPS simulated outages. The proposed method, 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, offline 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.

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. 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.

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** … 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 veicular 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** … ICNPAA 2010.

[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. In: Simpósio Brasileiro de Engenharia Inercial 2010, 20-22 october, IPqM, Rio de Janeiro, Brazil. **Proceedings** … SBEIN 2010.

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

[9] 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.

[10] HAYKIN, S. **Redes Neurais: princípios e práticas**. Porto Alegre: Bookman, 2001. 900 p.

[11] 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.

[12] 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.

[13] 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.

[14] 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.