Window-Updated Method for Strapdown Inertial Navigation Systems Based on Adaptive Neuro-Fuzzy Inference System

* Ahmed Mudher Hassan

Received on : 26 / 3 /2007
Accepted on :30 / 6 /2008

Abstract

The last two decades have shown an increasing trend in the use of navigation technologies such as Strapdown Inertial Navigation Systems(SDINS) in several applications including land vehicles and automated car navigation. On the other hand it can cause large position errors over short time, due to the low quality of the Inertial Measurement Unit (IMU). These errors determine the performance and the navigation accuracy of the INSs. Although the huge efforts to improve SDINS in terms of its mechanization equations, it could not cover the remaining drawbacks of SDINS; such as the impact of INS short term errors, model dependency, prior knowledge dependency, sensor dependency, and computational errors.

This paper proposed an intelligent navigator to overcome the limitations of existing INS algorithms. The intelligent navigator is based on Adaptive Neuro-Fuzzy Inference System (ANFIS).

The proposed conceptual intelligent navigator consisted of SDINS architecture that was developed using adaptive fuzzy system networks to acquire the navigation knowledge. In addition, a navigation information DataBase, and a window-based learned parameters updating method were implemented to store and accumulate navigation knowledge.

Keywords: vehicular navigation, Inertial Navigation System (INS), DataBase, Adaptive Neuro-Fuzzy Inference System (ANFIS).

الخلاصة:

العقدين الاخيرين شهد أرتفاعا ملحوظا في استخدام تقنيات الملاحة مثل منظومة الملاحة بعزم القصور الذاتي في مختلف التطبيقات بضمنها المركبات الارضية ومركبات الملاحة الاوتوماتيكية. ومن ناحية اخرى فان منظومات الملاحة قد تسبب خطأ كبير في حساب الموقع خلال زمن قصير، تبعا لردائه نوعية وحدة قياس العزم. هذه الاخطاء الناتجة تحدد أداء ودقة الملاحة لمنظومات الملاحة بعزم القصور الذاتي. مع ذلك فالجهود الكبيرة المبذولة لتحسين منظومة الملاحة بعزم القصور الذاتي من ناحية معادلاتها الميكانيكية، لم تشمل كل المساوئ لمنظومات الملاحة بعزم القصور الذاتي، مثل التاثيرات الرجعية لمنظومة الملاحة بعزم القصور الذاتي خلال فترة زمنية قصيرة، الاعتمادية على الموديل الرياضي للمنظومة، الاعتمادية على المعلومات المسبقة، الاعتمادية على نوع المتحسسات، والاخطاء الناتجة عن الحسابات الرياضية.

هذا البحث يقترح ملاح ذكي لتجاوز المحددات الموجودة في خوارزمية منظومة الملاحة بعزم القصور الذاتي. الملاح الذكي مبني على أساس (ANFIS).

ان مضمون الملاح الذكي يتضمن منظومة ملاحة بعزم القصور الذاتي والذي طور بأستخدام شبكة النظام الضبابي المتكيف لاكتساب معلومات الملاحية. بالاضافة الى ذلك، قاعدة بيانات للمعلومات الملاحية و طريقة لتحديث المؤشرات التعليمية مبنية على أساس النافذة قد نفذت لخزن و جمع المعلومات الملاحية.

Control and System Eng. Dept. University of Technology

1. Introduction

Most of the present vehicle navigation instruments rely mainly on the Global Positioning System (GPS) as the primary source of information to provide the vehicles position. GPS is capable of providing precise positioning information to an unlimited number of users any where on the planet. However, GPS can provide this type of information only when there is a direct line of sight to four or more satellite [1]. In other words, the system does not work properly in urban areas due to signal blockage and attenuation that may deteriorate the overall positioning accuracy.

An INS is a self-contained positioning and attitude device that continuously measures three orthogonal linear accelerations and three angular rates. By measuring vehicle acceleration and angular velocity in an inertial frame of reference, integrating it with respect to time and transforming it to the navigational frame, velocity, attitude and position components can be obtained. Sensors used to implement such a system are accelerometers for the measurement of a vehicles linear acceleration (specific force) and gyroscopes for monitoring vehicle rotation (angular velocity) with respect to an inertial frame [2]. Since specific force measurements contain the effect of the earth's gravity field, a gravity model is used to extract vehicle acceleration from the measurements. Because they employ three translational (accelerometers) and three rotational (gyroscopes) sensors, Inertial Measuring Units (IMU) can be used as positioning and attitude monitoring devices [3].

In fact, INS can not operate as a stand-alone navigation system like GPS. Residual bias errors in both the accelerometers and the gyroscopes may deteriorate the long-term positioning accuracy. A comparison between the two navigation systems is illustrated in table (1). In addition, these bias errors are random in nature and need to be modeled using empirical and adaptive model processes. And since there is a lack of researches towards the conceptual intelligent navigator, this paper is devoted to develop an intelligent navigator that consists of daptive Neuro-Fuzzy Inference System (ANFIS) based on Terrestrial SDINS algorithm described in [4]. As each of these limitations contributes to certain amount of positional error accumulation during computational errors, therefore, the proposed new algorithm are expected to reduce the impact of these limitations by reducing the positional error accumulation during navigation phase. In land vehicle and submarine navigation. Regular updates are needed to limit the rapidly growing positional error. It is often possible to considerably improve the accuracy of the SDINS by building a conceptual intelligent navigator to accumulate the navigation knowledge and retrieve it to trim down the SDINS error.

1.1 Objectives and Motivation

This paper aims at introducing a novel method based on Adaptive Neuro-Fuzzy Inference System (ANFIS) to fuse the outputs of IMU and provide accurate positioning and velocity information for the moving vehicle.

In addition, this paper suggests a navigation DataBase to retrieve the navigation knowledge to provide an accurate real-time computing system for the strapdown INS algorithm for vehicular navigation.

In other words, this paper introduces a conceptual intelligent navigator for next generation navigation systems to accumulate the navigation knowledge and retrieving the stored navigation knowledge to be able to provide the real time prediction. Ultimately, the conceptual intelligent navigator is expected to overcome or, at least, reduce the limitations of the conventional based SDINS algorithms.

2. Adaptive Neuro Fuzzy Inference Systems

Ever since the artificial intelligence, considered as a powerfull and applicable tool in engineering modeling, computation, nonlinear function approximation, system identification and estimation theory. The neuro fuzzy models

have the connectionist structure of neural networks combined with flexibility and intuitive learning capabilities of fuzzy system. It has hybrid learning method based on gradient descent and least square estimation [5].

In order for an FIS to be mature and well established so that it can work appropriately in prediction mode, its initial structure and parameters (linear and non-linear) need to be tuned or adapted through a learning process using a sufficient input-output pattern of data. One of the most commonly used learning systems for adapting the linear and non-linear parameters of an FIS, particularly Takagi and Sugeno (TS) type, is the ANFIS. ANFIS is a class of adaptive networks that are functionally equivalent to fuzzy inference systems [6]. Different interpretations for the fuzzy IF-THEN rules result in different mappings of the fuzzy inference engine, also there are different types of fuzzifier and defuzzifier. Several combinations of the fuzzy inference engine, fuzzifier, and defuzzifier may constitute useful fuzzy logic system. If the fuzzy logic system can be represented as a feed forward network, then the idea of back propagation training algorithm can be used to train it. The structure of ANFIS and the main concepts and algorithm adopted during its learning process will be introduced later.

3. Proposed Conceptual Intelligent Navigator based on ANFIS

The proposed conceptual intelligent navigator integrates the data from IMU and mimics the dynamical model of the vehicle to generate navigation knowledge. Thus the latest acquired navigation knowledge can be applied to predict the vehicles velocity and position during IMU errors in real time.

The resulting ANFIS has the structure depicted in figure (1) . It consists of 5 layers. Layer 1 and layer 5 define the input and output spaces respectively. Layer 2 and layer 3 are used to perform the IF part of fuzzy rules. Layer 4 performs the normalization of each node in layer 3. The THEN part of the fuzzy rules is completed in the fifth layer. Detailed descriptions and equations for each layer are given here [7].

Layer 1: Input-variable layer. This is the layer where the input signals first enter the neural network, and each node in layer 1 represents an input linguistic variable.

Layer 2: Each node in layer 2 represents a membership function (MF), which is a Gaussian function of the following form:

$$MF_{ij}(x_j) = \exp\left[-\frac{(x_i - c_{ij})^2}{\sigma_j^{\,2}}\right] \qquad i = 1,2,....r, \quad j = 1,2,...u$$

...(1)

where r is the number of input variables and u is the number of membership functions.

Layer 3: The rule layer associated with the input variables is given by eq.(2). Each node in this layer is a radial basis function (RBF) unit which represents a possible IF- part of the fuzzy rule. The outputs are given by:

$$R_j = \exp\left[-\frac{\sum_{i=1}^{r}(x_i - c_{ij})^2}{\sigma_j^{\,2}}\right] = \exp\left[-\frac{\|X - C_j\|^2}{\sigma_j^{\,2}}\right]$$

...(2)

where

$$X = [x_1, x_2, ..., x_r]^T$$

$$C_j = [c_{1j}, c_{2j}, ..., c_{rj}]^T.$$

Layer 4: This layer consists of normalized nodes. The number of nodes is equal to that of RBF units. The output is given by:

$$N_j = \frac{R_j}{\sum_{j=1}^{u} R_j}$$

...(3)

Layer 5: This is the output layer, which comprises of output nodes, each of which is weighted according to eq. (4). This layer performs defuzzification (weighted average) of the output as follows:

$$y(X) = \sum_{j=1}^{u} w_{2j}.N_j$$

...(4)

The weight is of linear structure and can be expressed as follows:

$$w_{2j} = k_{j0} + k_{j1}x_1 + ...k_{jr}x_r$$

...(5)

where k_{ji} are real-valued parameters.

4. ANFIS Learning using Hybrid Technique

As mentioned earlier, both the premise (non-linear) and consequent (linear) parameters of the FIS should be tuned, utilizing the so-called learning process, to optimally represent the factual mathematical relationship between the input space and output space. Normally, as a first step, an approximate fuzzy model is initiated by the system and then improved through an iterative adaptive learning process.

The training algorithm, namely ANFIS, was developed by [6]. Basically, ANFIS takes the initial fuzzy model and tunes it by means of a hybrid technique combining gradient descent back-propagation and mean least-squares

optimization algorithms figure (2). At each epoch, an error measure, usually defined as the sum of the squared difference between actual and desired output, is reduced. Training stops when either the predefined epoch number or error rate is obtained. The gradient descent algorithm is mainly implemented to tune the non-linear premise parameters while the basic function of the mean least-squares is to optimize or adjust the linear consequent parameters. Table (2) shows the two stages of hybrid learning process of ANFIS.

5. Intelligent Navigator Architecture

The general architecture of ANFIS is shown in figure (1), the input vectors for the ANFIS are the raw accelerations at each current epoch $A_{IMU}(t)$, and angular velocity at current epoch $AV_{IMU}(t)$ while the output of ANFIS was the instant position and velocity at each current epoch for both position and velocity in the three directions for the moving vehicle.

ANFIS and SDINS algorithms receive raw outputs of accelerometers and gyros, respectively as inputs and generate navigation state as outputs as shown in figure (3). Thus, SDINS mechanization illustrated in [4] is replaced by proposed ANFIS.

So, the navigation knowledge can be learnt, stored and accumulated during the availability of the INS signal. On the other hand, during INS signal absence or IMU errors, the latest acquired navigation knowledge can be retrieved from the navigation information DataBase.,

6. Navigation Information DataBase

The second step towards building the intelligent navigator is to store the learnt navigation knowledge provided by SDINS algorithm. As a result, a navigation information DataBase that contains the acquired and learnt navigation knowledge can serve as the "brain" of the intelligent navigator. Therefore, several issues regarding the DataBase are addressed as follows:

- **Content of DataBase:** The DataBase consists of the training samples (input vectors and desired output vectors) and estimated learned parameters during the availability of the IMU signal. Thus, these components can be regarded as the navigation knowledge. In other words, the content of the DataBase varies with

  the structure of different INS algorithm (i.e., according to navigation solutions that will be given by SDINS algorithm).

- **Distributed navigation knowledge storage:** The content of DataBase becomes more complicated with complicated SDINS architectures or solutions. Therefore, considering the efficiency of DataBase maintenance and retrieval, the navigation knowledge learnt by each sub-component should be stored individually in a distributed way, as shown in figure (4).

- **Off-line DataBase maintenance:** The simplest way to reduce the storage requirement

is to remove any redundant training samples that include inputs and their corresponding desired outputs. As for the learned parameters, they should be kept without any change as they are the core component of the navigation knowledge. Using ANFIS architecture as an example, a simple procedure that can be applied prior to navigation (i.e., during alignment) or after navigation before shutting down the system, is given below:

i. Regroup the training samples: Using one of the training inputs (i.e., $A_{IMU}(t)$ or $AV_{IMU}(t)$) as the index; the training inputs can be regrouped to increase the efficiency for maintenance.
ii. Locate redundant navigation knowledge: Although it is difficult to locate a pair of training samples that are exactly the same, searching the most similar pairs of training samples using threshold values then deciding if they are redundant or not is possible.
iii. Remove the redundant navigation knowledge.

It must be mentioned that the implementation of DataBase was done with *MatLab* application. Up to now, the intelligent navigator has been given the ability to generate, and learn navigation knowledge and it also has been given the "space" to store navigation knowledge. However, there is still one thing missing. It requires a way to accumulate the acquired and learnt navigation knowledge and store them for further retrieving or

generalization and this navigation knowledge must be updated during the training procedure when INS signal is available so, a windowing method used to perform this goal as will be described later.

7. Window-Based Learned parameters updating Method

As the learned parameters (x, c, σ) are the core components of the navigation knowledge, the final step towards building the intelligent navigator is to develop a method to accumulate the acquired navigation knowledge by updating the learned parameters whenever the IMU signal is available (i.e., no sensor errors or absence of signal).

In most of their applications, ANFIS are trained using some known training data set (input/desired output) to obtain the optimal values of the learned parameters via off-line training. For any other set of inputs, different from those used in training, the learned parameters can then be applied to provide prediction of the network outputs. It is worth mentioning that ANFIS parameters are frozen after completing the training procedure and no further modification will be made during the prediction process.

In fact, off-line training can work well in case of slowly changing time sequences [8]. In the case of INS navigation applications, it is required to track direction changes and mimic the motion dynamics utilizing the latest available INS data. In other words, the learned parameters should be updated during the navigation process to adapt the network to the latest INS sensor readings whenever the INS signal is available.

To implement such criterion, a window-based learned parameters updating method, which utilizes the learned parameters obtained during the conventional off-line training procedure (or probably from previous navigation missions) is stored in the DataBase and is presented in this paper. This criterion utilizes the latest available navigation information provided by the INS signal window to adapt the stored learned parameters so that they can be applied to mimic the latest motion dynamic. The window-updated learned parameters are stored after each training procedure. They are then used as initial values

for the parameters to be estimated during the next training window or for prediction during INS signal absence. Prior to looking into the details of the window-based learned parameters updating method, several aspects of traditional learned parameters updating methods are given. Traditional methods can be classified as [8]:

1. **Sample-by-sample training**, also known

as on-line or sequential training, that modifies the parameters for each input record after computing the learned parameters updates;

2. **Batch training**, which computes the learned parameters updates for each sample and stores these values (without changing the parameters). At the end of the whole training procedure, all the learned parameters updates are added together and then the parameters are modified with the accumulated learned parameters updates.

From an online operational point of view, the sequential mode of training is preferred over the batch mode since less local storage is required. In addition, the random presentation of the pattern makes it less likely for the standard back-propagation algorithm which is used in ANFIS network to be trapped in a local minimum if the sequential mode of training is utilized. In contrast, the use of batch mode provides a more accurate estimate of the learned parameters, thus giving more accurate estimation of the navigation information learned parameters.

Another major advantage of sequential training over batch training arises if there is a high degree of redundancy in the data.

On the other hand, the sequential training updates the parameters after receiving each record of the input samples. Therefore, it will not be affected by such highly redundant data. However, during batch training, the network can learn more general relationships as it utilizes most of the available training data at the same time instead of sample by sample. Both generalization and training efficiency are very critical for SDINS applications, therefore, developing a special learned parameters updating method that can preserve the generalization ability without losing too much training efficiency is very important.

8. Development of Window-based learned parameters updating method

Although the stored parameters might not be able to provide accurate prediction during all INS absence, it can be applied as the initial parameters at the beginning of a new navigation mission. The INS window signal concept is then applied to introduce new navigation knowledge to modify stored learned parameters during navigation. In fact, this method combines the advantages of both sequential mode and batch mode of training in order to make the training procedure suitable for real-time processes. In addition, the parameters of each window are then updated via batch training mode. In other words, the parameters of each window are updated sequentially. As depicted in figure (5), the procedure of the window-based learned parameters updating method is given below:

i. **Learned parameters initialization:** The initial parameters can be obtained using previously stored parameters that are stored in DataBase or random initialization procedure. In this paper, the initial parameters were obtained using random initialization at the first time when the ANFIS architecture were set up and they are illustrated in table (3). Concequently, the parameters are stored in DataBase after completing one navigation mission and are applied as the initial parameters for the next mission. Accurate initial parameters may significantly reduce training time.

ii. **INS signal reception:** Within the first INS window (i=1), INS(i), the learned parameters are not updated, thus the stored parameters are still the initial parameters P(i-1) (i.e., P(0) in first INS window).

iii. **INS signal reception:** At the next INS window, INS(i+1), the stored parameters, P(i-1), are updated utilizing the presently available INS information (INS(i)). These parameters are stored as P(i) after training is completed. Steps **(ii)** and **(iii)** are repeated until an INS signal blockage is detected.

iv. **INS outage:** As depicted in figure (5), in the case of a INS outage (after INS(i)), P(i-1) is first applied for real-time prediction and then P(i) is then utilized to replace P(i-1) and carry on real-time prediction during

the remaining INS outages.

Since the ANFIS training procedure takes time, updating the learned parameters immediately at the latest available sample of INS signal before outage is difficult. However, the utilization of the proposed method can still provide reasonable prediction accuracy during INS blockage since it provides the latest updated parameters instead of real-time updated parameters for real-time prediction. Therefore, failure in providing real-time updated learned parameters doesn't mean the intelligent navigator is not able to provide real-time prediction. In contrast, it can utilize the latest acquired and learnt navigation knowledge to provide real-time solutions. Combining the latest INS window signals, the stored parameters can be adaptively updated to follow the latest dynamic condition and INS errors thus improving the prediction accuracy during INS outage.

As mentioned previously the 5^{th} layer (output layer) generates velocity and position in the local level frame at the current epoch. Thus, the navigation knowledge can be learnt, stored and accumulated during the availability of the INS signal. On the other hand, during INS signal absence or IMU errors, the latest acquired navigation knowledge can be retrieved from the "brain" (navigation information DataBase) of the intelligent navigator to predict the velocity and position in real time. The proposed navigator consist of two main procedures, the initialization and training procedures so, if the network well initialized and trained a good prediction will be obtained.

During the initializition of ANFIS network, where the parameters x, c, σ are the ANFIS learning parameters computed during the training procedure and they are determine the input/output functionality of the network. The initial values for the ANFIS learning prarmeters are obtained by trail and error and they are stated as in table (3). in contrast, the selection of the number of rules and the learning rate are very important to get fast and accurate training as illustrate below:

i. Selection of the number of rules.

As mentioned previously, the arbitrarily selection of the number of rules (M) is also based on the trial-and-error procedures. The purpose of this illustration to show the relationship between the number of rules and the mean square error (MSE) for the ANFIS network.

The value of M was varied from 5 to 50 in a step of 5. For each value of M, the network is initialized randomly over specified ranges of the parameters x, c, and σ. These ranges will be the same for all values of M.

The results shown in figure (6) were obtained after implementing the six networks of position and velocity components using the same initial values listed in table (3). For each value of M, the number of epochs was 10 and the value of the learning rate was 0.6 in all networks.

It was noticed that, as the number of rules increases as the training process becomes slower; therefore, if the value of the error is decreasing then because of this slow training process, the convergence to a minimum error value will also be slow and the specified number of iterations might be ended without reaching that minimum error value. Hence it can be noticed from figure (6) that, in general, as the number of rules increases as the value of the MSE increases. Even if a high value of M achieves small error value, because of the randomly initialized parameters, it is not recommended to use this value because the training process will be too slow and requires a lot of time, i.e. it is better to look for the value of M that achieves minimum error and with minimum time (small M values).

It can also be seen from figure (6) that the values of M equal to 5, 10, and 15 give minimum error values in all networks; therefore, M=10 was used to implement all the networks in learning process.

ii Selection of the learning rate.

The selection of different values of the learning rate which affect the speed of

convergence during the training process will be presented below.

To illustrate the relationship between the value of the learning rate and the MSE for all networks of position and velocity, the learning rate was varied from 0.1 to 0.9 in a step of 0.1.

For each of these values of the learning rate, the networks were initialized randomly over specified ranges of x, c, and σ. The same initial ranges will be used for all values of the learning rate.

For each network, the same initial values listed in table (3) were used to obtain the results shown in figure (7). For each value of the learning rate, the number of epochs was 10 and the number of rules was 10 in all networks.

It was noticed that when the learning rate was small, the network will adjust its parameters gradually but in this case convergence might be slow, on the other hand, a high learning rate might make drastic changes that are not desirable; therefore, medial values of the learning rate are preferable to choose.

Also, it can be seen from figure (7) that the value of learning rate equal to 0.5 or 0.6 gives small error value in most of the networks; therefore, the value of 0.6 for the learning rate was used to obtain the results from all networks in the training process.

The ANFIS used in this paper to predict the SDINS position and velocity in real time. So after completing the initialzintion procedure the ANFIS network can be trained and according to figure (3) which illustrates the training procedure of the ANFIS, the network output is compared to the SDINS algorithm output figure (8) show the SDINS and ANFIS outputs in inertial-frame for position and velocity and the error between them was shown in figure (9), this error is feed to the ANFIS network, which adjust the network learning parameters in a way to minimize the mean square value of error.

The output is obtained and compared to the target (desired performance) to determine the estimation error. This error is propagated through the network in the backward direction (opposite to the flow of the input data) starting from output layer and is utilized to update the computation of the network parameters. The forward and backward computations are repeated until the optimal value of the learning parameters are achieved, which correspond to certain objective mean square estimation errors, The network parameters are updated according to certain learning rules to minimize the mean square value of the estimation error. the training process continued for 10 epochs to reduce the

MSE value and the relation between the number of epochs and MSE is shown in figure (10).

After the training is completed the network is ready to work in the prediction mode. However the parameters of the networks are modified during the availability of the IMU signals (i.e. the training procedure continues) and network is considered working in the update mode. During IMU errors, the network will use the latest estimation parameters saved in the DataBase to perform the prediction process.

9. Conclusions and Suggestions for Future Work.

The conclusions drawn from the results presented in this paper and future work are:

1. In this paper, an attempt to build a reliable navigation system is made by combining the merits of INS, and ANFIS.
2. The parameters of the intelligent navigator are included in the navigation knowledge. Thus they can be updated without a human expert during navigation whenever newly updated navigation knowledge is acquired.
3. The long procedure of trial-and-error in finding the optimal number of layers in the network, the number of nodes in each layer, and the activation function in each node, which exists in Artificial Neural Network (ANN), has been avoided in this paper by using the proposed ANFIS with its constant structure as described in this paper.
4. The appropriate selection of the initial values of the parameters x, c, and σ has a significant effect on the performance of the

ANFIS network and plays an important role in decreasing the convergence time to the best solution (minimum error value).

5. The selection of M (number of fuzzy rules) is essential in achieving good results. It was noticed that using large number of rules results in slow training and large error values whereas the small M values lead to small error values and fast training performance.

6. The speed of convergence during the training process is remarkably affected by the value of the learning rate. Small values of learning rate cause slow convergence, on the other hand, undesirable results may appear when using large values; therefore, most reasonable performance is achieved with medial values of learning rate.

7. The results presented in this paper strongly indicate the potential of including the intelligent navigator as the core navigation algorithm for the next generation navigation system.

8. Using neural network adaptive wavelets (wavenet) to filter out the noise that exists at the IMU outputs.

9. Finding the appropriate initial values of the parameters x, c, and σ by means of the Genetic Algorithm.

10. Using other types of fuzzy logic system structure that can be built based on different

11. combinations of the fuzzy inference engine, fuzzifier, defuzzifier, and membership function.

12. The proposed conceptual intelligent navigator can be used instead of Kalman filter or others traditional techniques to support INS/GPS integrated system, and to generate more reliable navigation solutions.

10. References.

[1] Shin E-H and El-Sheimy N "Accuracy improvement of low cost INS/GPS for land applications", *Proc. ION National Technical Meeting (San Diego, 28-30 January 2002)* (Fair fax, VA: Institute of Navigation) pp. 146-57, 2002.

[2] David H. and John L., *Strapdown Inertial Navigation Technology*, the Institution of Electrical Engineers, Michael Faraday House, 2004.

[3] Frederick F. Ling, *Modern Inertial Technology Navigation, Guidance, and Control*, Springer-Verlag New York, Inc, 1998.

[4] Salam I., and Samir A., "Development of Six-Degree of Freedom Strapdown Terrestrial INS Algorithm", Journal of Um-salama for Science, Volume2, No.1, pp. 155-164, 2005.

[5] Asadian A., Moshiri B., Khaki A., and Lucas C., "Optimized Data Fusion in an Intelligent Integrated GPS/INS System Using Genetic Algorithm," IEEE Transactions on Engineering, Computing and Technology, ISSN 1305-5315, Vol.5, pp.221-224, April 2005.

[6] Jang, J., "Self-learning fuzzy Controllers based on temporal back propagation", IEEE Transactions on Neural Network, Vol.3, No.5, pp.714-23, 1993.

[7] Er M., Li Z., Cai H., and Chen Q., "Adaptive Noise Cancellation Using Enhanced Dynamic Fuzzy Neural Networks," IEEE Transactions on fuzzy systems, Vol.13, No.3, pp.331-342, June 2005.

[8] Salam I., "Intelligent SINS Navigator Based on ANN", IJCCCE, University of Technology, Vol.6, No.1, 2006.

Table (1): Comparison of INS and GPS Systems.

INS	GPS
Short term position and velocity accuracy	Long term position and velocity accuracy
Accurate attitude information	Noisy attitude information
Decreasing accuracy over time	Uniform accuracy over time
High measurement output	Low measurement output rate
Autonomous	Non-autonomous
No signal outages	Subject to signal outages
Affected by gravity	Not sensitive to gravity

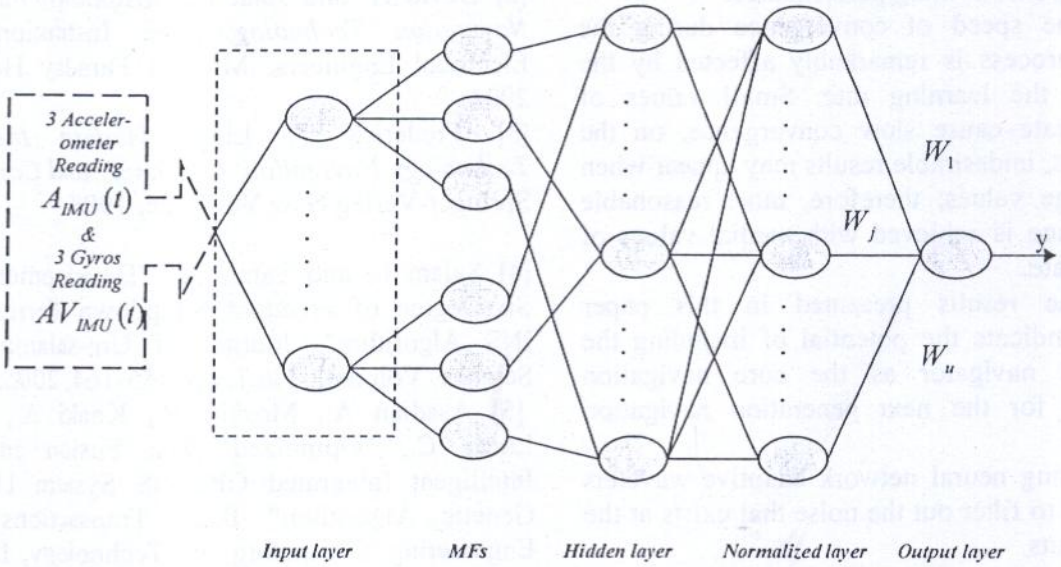

Figure (1): Adaptive Neuro Fuzzy Inference System Structure.

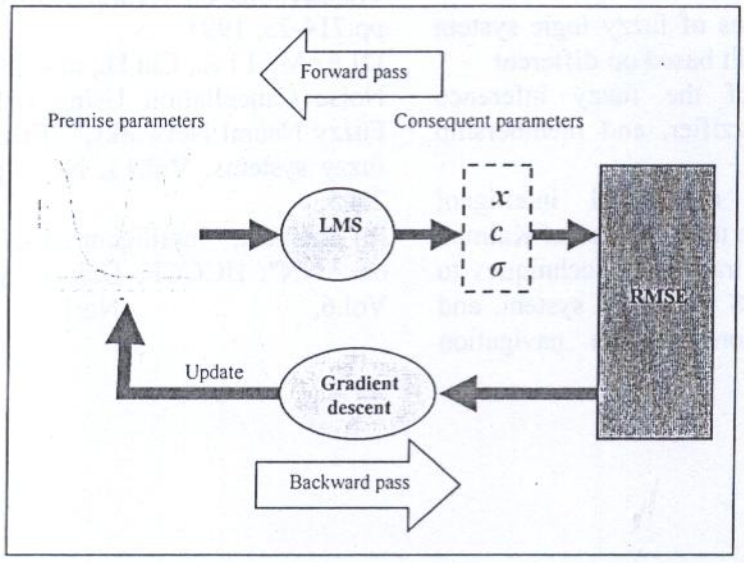

Figure (2): ANFIS learning using hybrid technique.

Table (2): Hybrid learning process of ANFIS.

	Forward Pass	Backward Pass
Premise Parameters	*Fixed*	*Gradient Descent*
Consequent Parameters	*Least-square estimator*	*Fixed*
Signals	*Node outputs*	*Error signals*

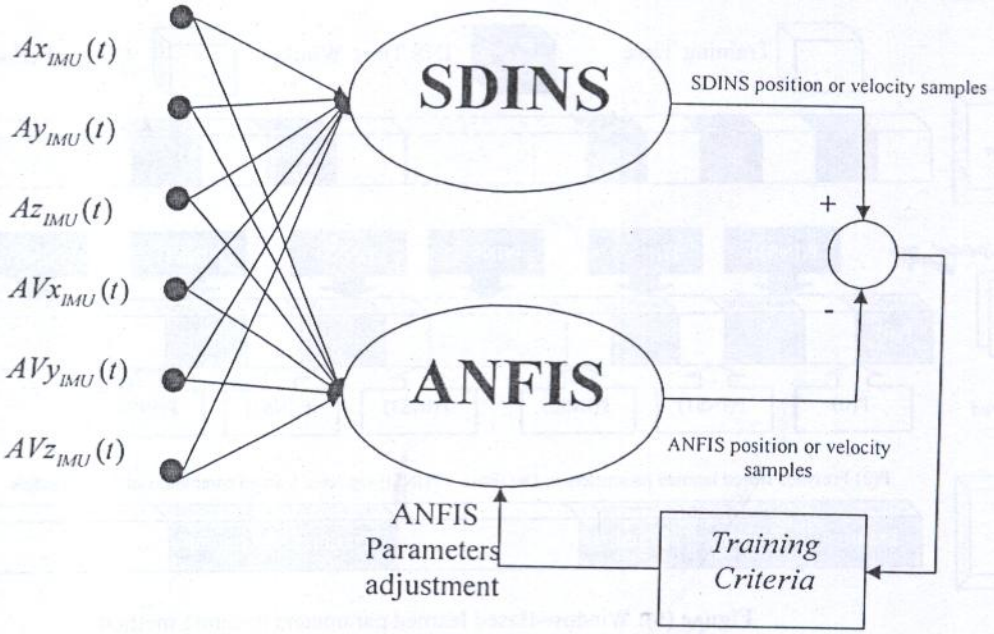

Figure (3): Proposed Conceptual Inelligent Navigator structure.

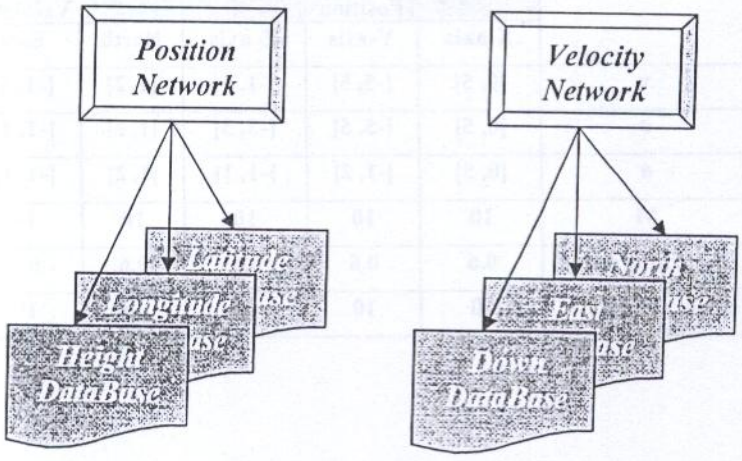

Figure (4): Distributed Navigation DataBase.

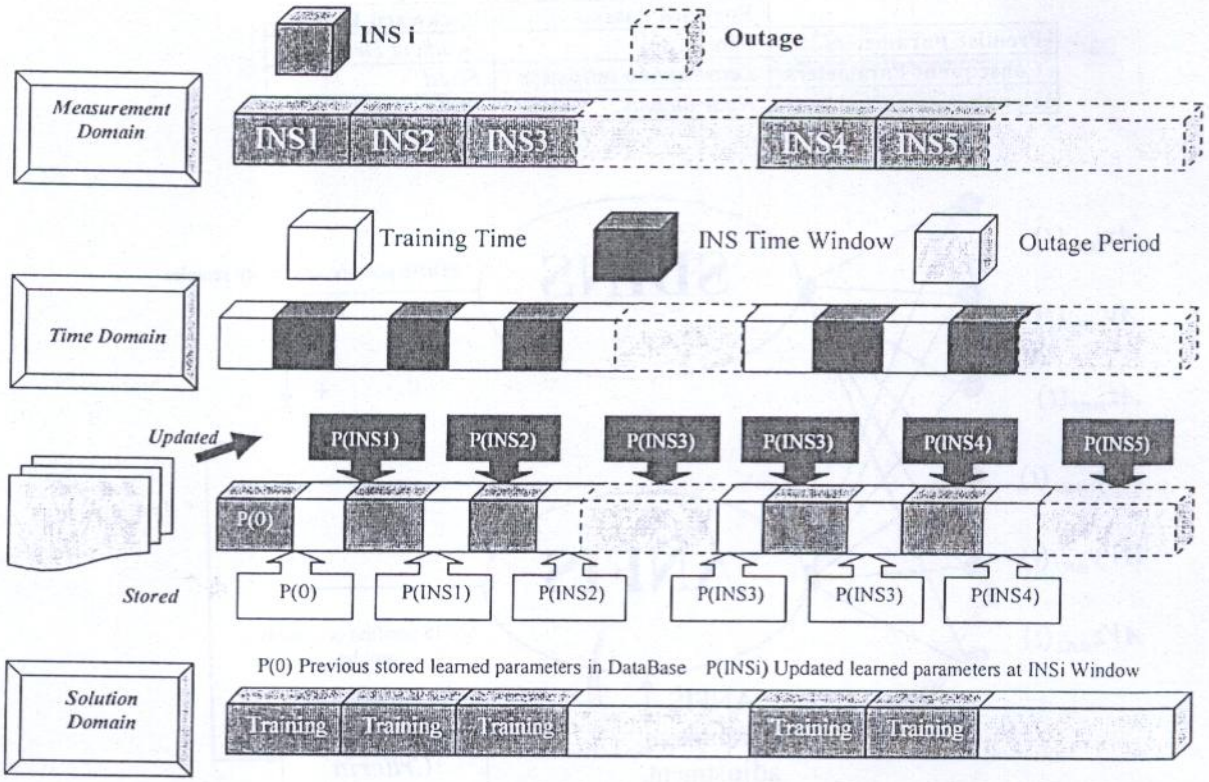

Figure (5): Window-Based learned parameters updating method.

Table (3): Initial values for the ANFIS network for position and velocity.

		Position			Velocity		
		X-axis	Y-axis	Z-axis	North	East	Down
Initial values	x	[0, 5]	[-5, 5]	[-1, 1]	[1, 2]	[-1, 1]	[-2, 1]
	c	[0, 5]	[-5, 5]	[-3, 3]	[1, 2]	[-1, 1]	[-2, 2]
	σ	[0, 5]	[-1, 2]	[-1, 1]	[1, 2]	[-1, 1]	[-2, 2]
	M	10	10	10	10	10	10
	Learning rate	0.6	0.6	0.6	0.6	0.6	0.6
	No. of epoch	10	10	10	10	10	10

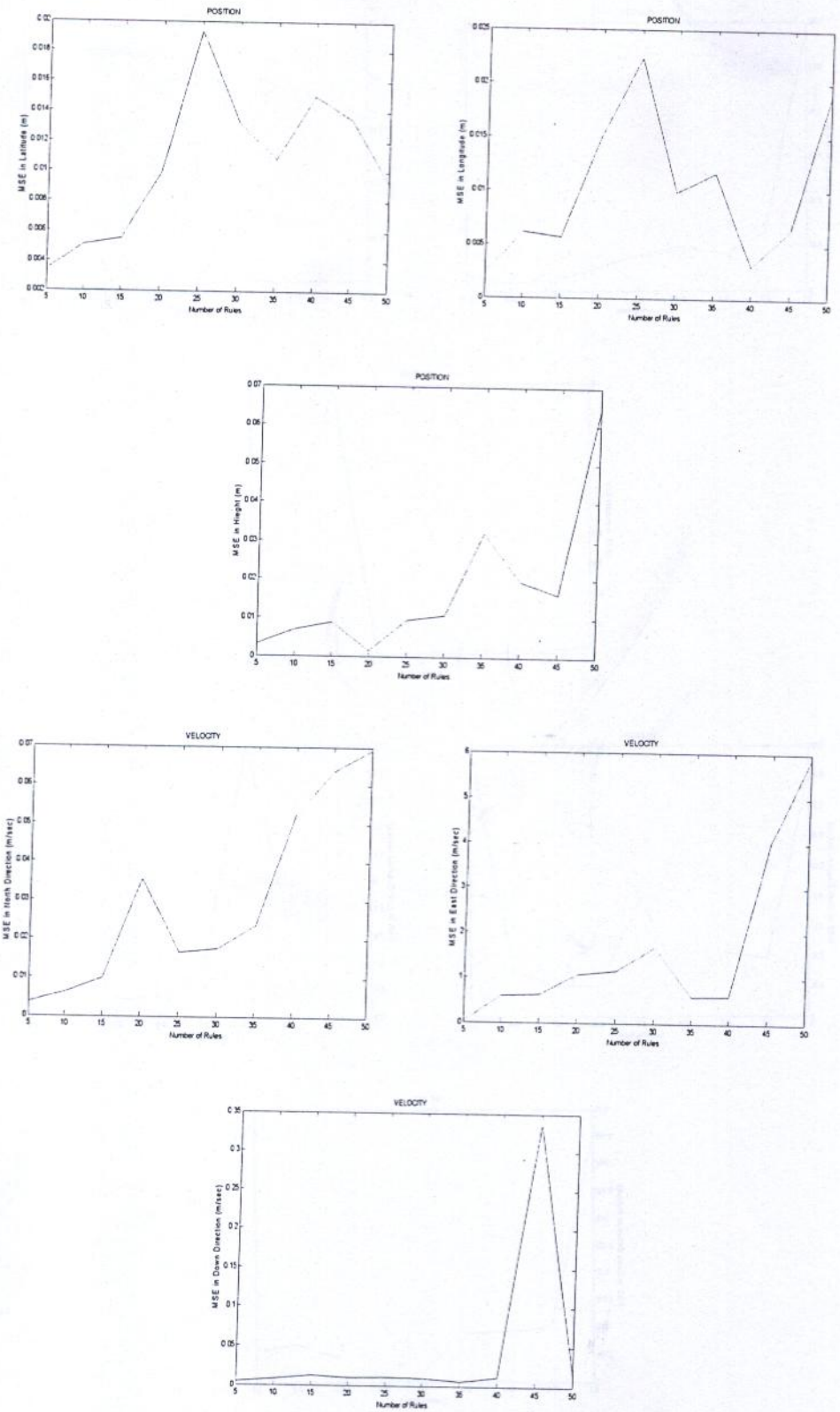

Figure (6): The relation between the number of rules (M) and the MSE for the ANFIS networks of position and velocity components.

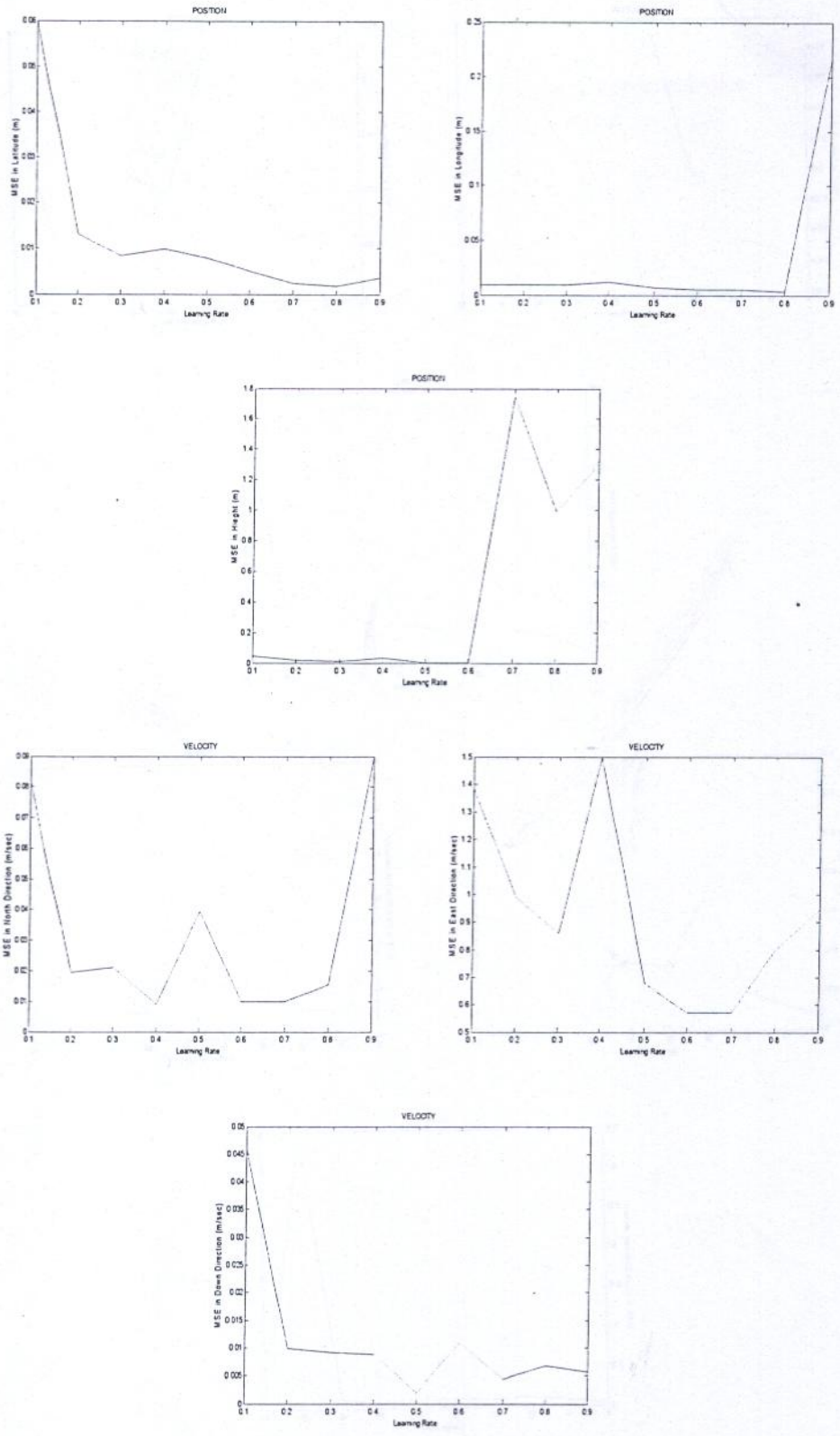

Figure (7): The relation between the value of the learning rate and the MSE for the ANFIS networks of position and velocity components.

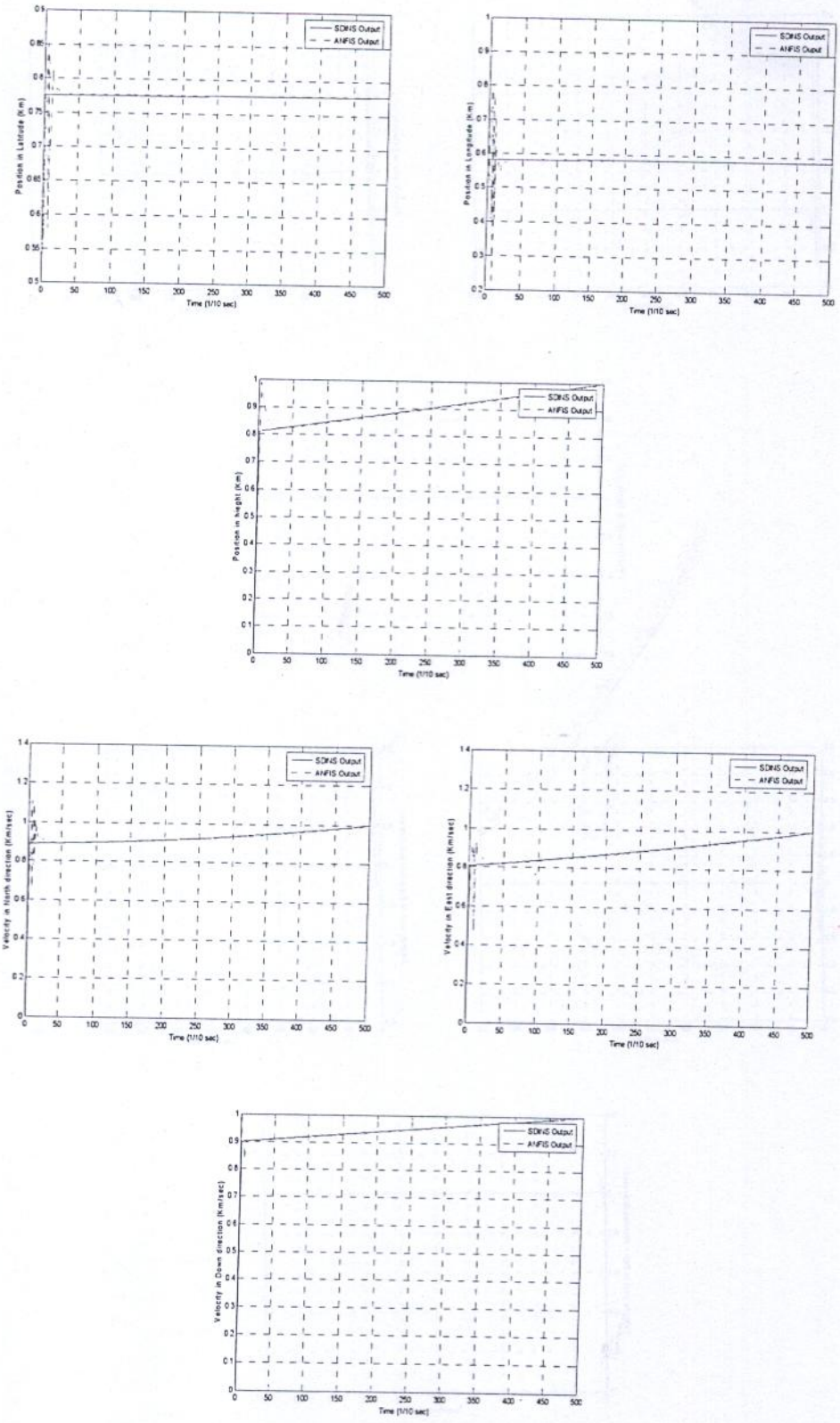

Figure (8): SDINS and ANFIS outputs for position and velocity in inertial-frame for all directions.

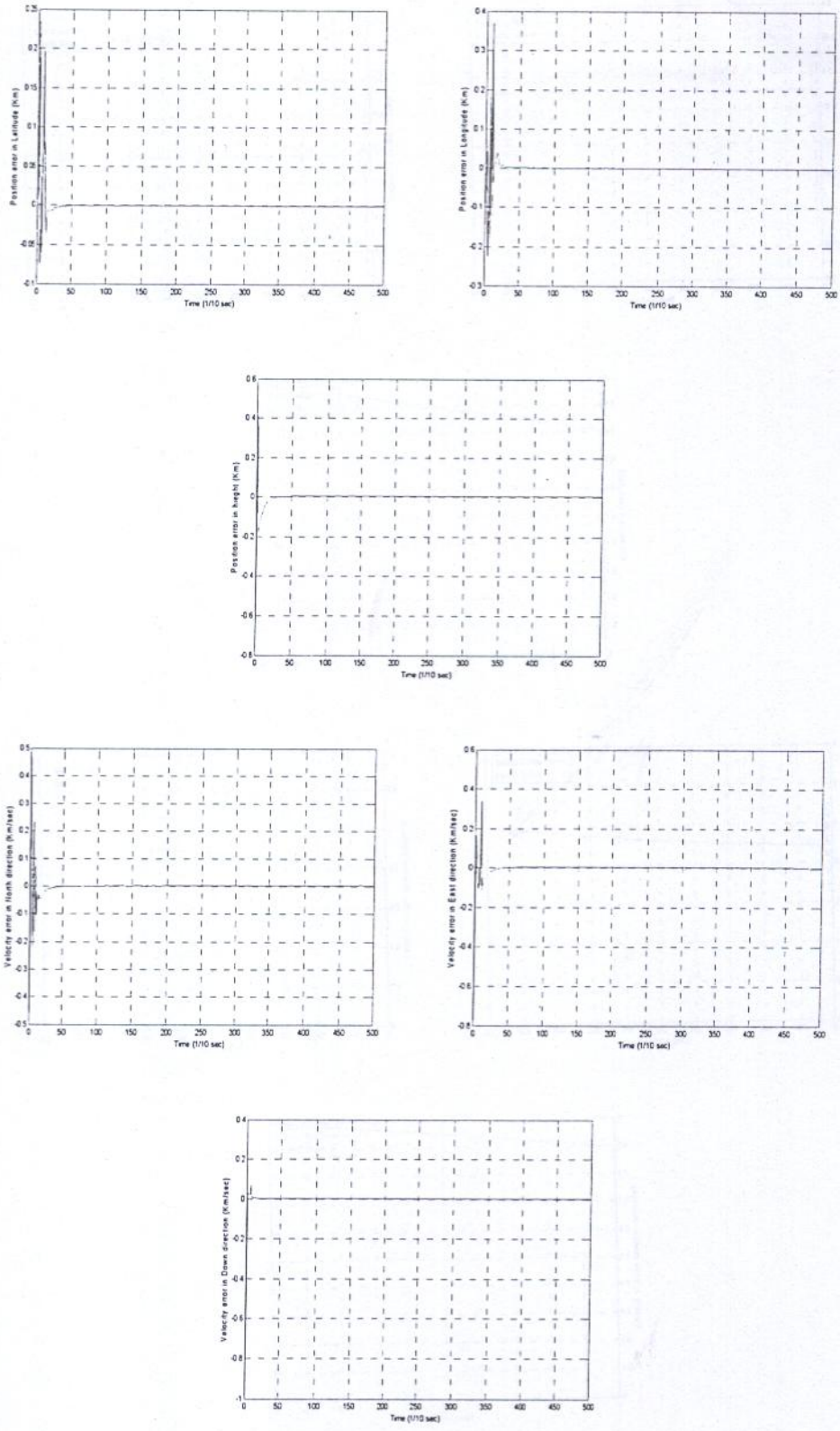

Figure (9): Error between the desired and actual output of the conceptual intelligent navigator along all components for position and velocity.

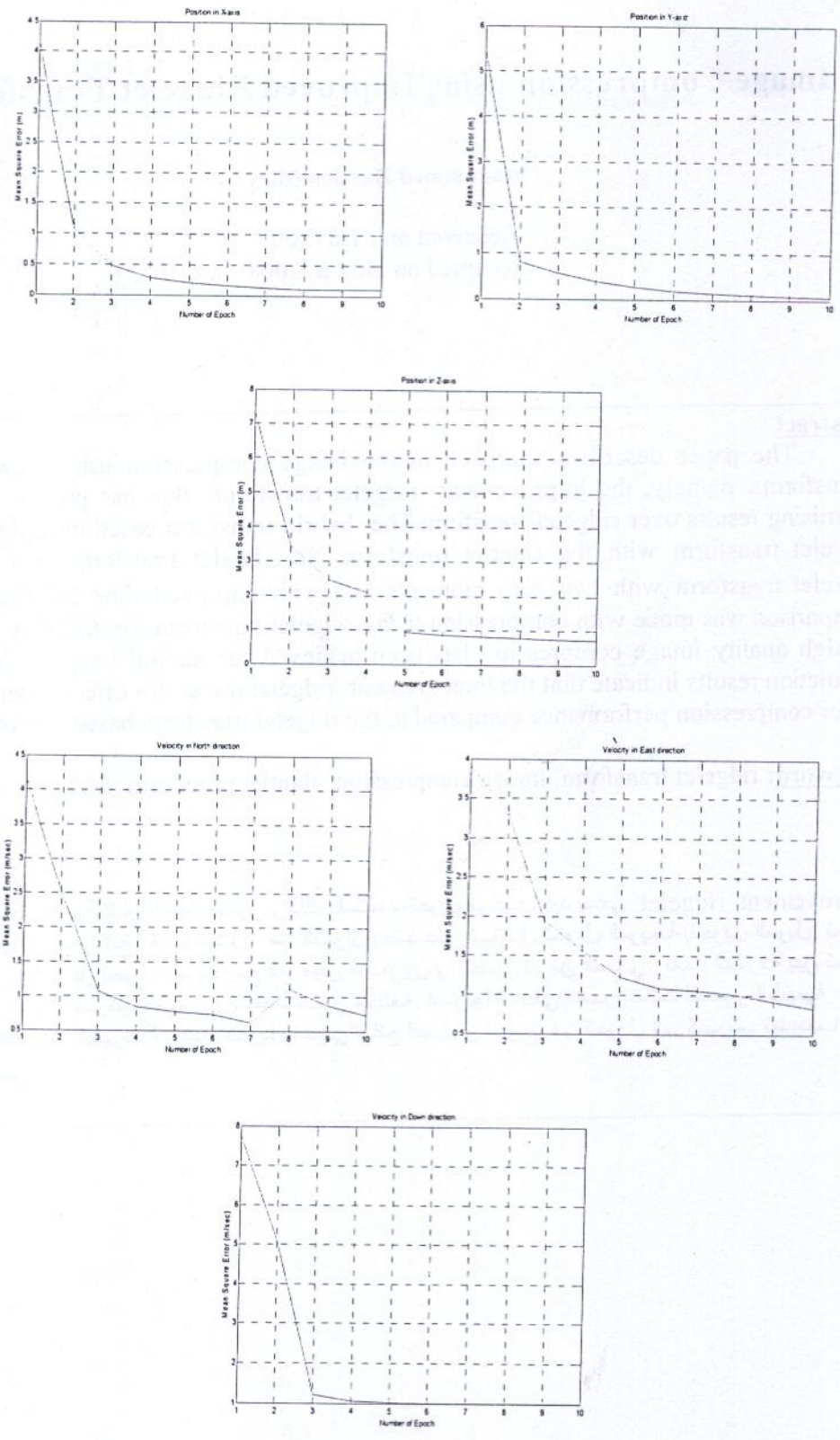

Figure (10): The relation between the number of epochs and the mean square error for the networks of position and velocity.