THE BIAS TEMPERATURE DEPENDENCE ESTIMATION AND COMPENSATION FOR AN ACCELEROMETER BY USE OF THE NEURO-FUZZY TECHNIQUES

Lucian Teodor Grigorie
Department of Avionics, 107 Decebal Blvd.
Faculty of Electrical Engineering, University of Craiova, Craiova, Dolj 200440, Romania
Contact: lgrigore@elth.ucv.ro
Ruxandra Mihaela Botez
Department of Automated Manufacturing Engineering, 1100 Notre Dame West
Ecole de technologie supérieure, Montréal, Québec, Canada, H3C 1K3
Contact: Ruxandra.Botez@etsmtl.ca

Received October 2007, Accepted October 2008
No. 07-CSME-46, E.I.C. Accession 3015

ABSTRACT

In this paper, we describe a new method for improved performance of inertial sensors, with applications in strap-down inertial systems. A new empirical model is proposed for the bias temperature dependence compensation of accelerometers using their input and output data. Experimental testing of the accelerometer is first realized, as data for 2 inputs and 1 output are collected. Based on this data, an empirical model is built using a neuro-fuzzy network, which learns the process behavior and uses a Fuzzy Inference System (FIS) for model realization. The improvement in the reproduction quality of the experimental surface by the neuro-fuzzy model is achieved through the FIS training using a Sugeno learning algorithm with two inputs and one output. Generation and training of the FIS are performed with Matlab functions, the training of which is realized on a high number of epochs, for example, on a number of 10^5 training epochs. It is noticed that the proposed algorithm leads to a 35.5 times reduction in the error due to temperature dependence of the bias.

L'ESTIMATION ET LA COMPENSATION DE LA DÉPENDANCE AVEC LA TEMPÉRATURE DU BIAIS D'UN ACCÉLÉROMÈTRE EN UTILISANT DES TECHNIQUES NEURO-FLOUES

RÉSUMÉ

Dans cet article, nous décrivons une nouvelle méthode pour l'amélioration des performances des capteurs inertiels, avec des applications dans les systèmes inertiels à composantes liés. Un nouvel modèle empirique est proposé pour la compensation de la dépendance avec la température du biais des accéléromètres en utilisant leurs données d'entrées et de sorties. L'essai expérimental de l'accéléromètre est réalisé, car les données pour deux entrées et une sortie sont rassemblés. Basé sur ces données, un modèle empirique est conçu avec un réseau neuro-flou, qui apprend le comportement du processus, et utilise un système flou d'inférence (FIS) pour la réalisation du modèle. L'amélioration de la qualité de reproduction de la surface expérimentale par le modèle neuro-flou est réalisée par l'entrainement du FIS en utilisant un algorithme d'étude de type Sugeno avec deux entrées et une sortie. La génération du FIS et son entrainement sont exécutées avec des fonctions en Matlab. La génération du FIS est réalisée sur un nombre élevé d'époques, par exemple, sur un nombre de 10^5 époques, et il est noté que l'algorithme proposé mène à une réduction de 35.5 fois de l'erreur due à la dépendance avec la température du biais.

1. INTRODUCTION

High precision accelerometers and gyros have biased reduced values, due to the inclusion of compensation devices in their architecture whose laws are dependent on the sensor's working temperature. In fact, the bias is strongly influenced by the temperature of the sensor's category, and for this reason, each sensor contains an auxiliary temperature transducer which allows for the estimation and compensation of this dependency ([1], [2], [3], [4]). In order to estimate the bias variation with temperature, the sensor's producer should conduct a series of tests under various conditions, and should determine the mathematical model characterizing this variation ([5], [6]). This operation is less complicated as it requires the use of a high precision temperature-controlled testing room. Practically, the sensor is tested on a rotating platform at a constant acceleration or angular speed at a constant temperature for which a high number of sensor output readings is acquired. By averaging these values the influence of noise and parasite vibrations induced by the rotating platform on the measurements is eliminated. A parametric investigation is conducted following this procedure for a set of accelerations, or angular speeds, at various temperatures. For temperatures and inputs modified between certain limits, results a measured data set that characterizes the sensor from the point of view of the dependency of its outputs on the temperature.

According to our bibliographical research ([7], [8]), the determination of the error model reduces to an identification problem for a double input – single output system, and further, by use of the Vandermonde matrix method, is reduced to the identification of a double input – single output system using a least squares method. As an example, the measurement errors of a RD2100 gyro, produced by KVH [2], can be represented by an error model of the following form [7]:

$$\varepsilon(\omega,T)=[T^2\ T\ 1]\begin{bmatrix} C_{0,0} & C_{0,1} & C_{0,2} & C_{0,3} \\ C_{1,0} & C_{1,1} & C_{1,2} & C_{1,3} \\ C_{2,0} & C_{2,1} & C_{2,2} & C_{2,3} \end{bmatrix}\begin{bmatrix} \omega^3 \\ \omega^2 \\ \omega \\ 1 \end{bmatrix}, \tag{1}$$

where ε is the bias temperature dependence error, ω is the gyro indicated angular speed, T is the temperature and $C_{i,j}$ are the identified system coefficients.

The paper presents a new method to estimate and compensate the temperature dependence of the bias for accelerometers and gyros based on the numerical values resulting from the sensors experimental testing. In this respect, the fuzzy logic offers remarkable facilities; it allows signal processing by putting up empirical models, which avoid complex mathematic calculus used at present. In addition, the fuzzy models have excellent results on non – linear, multidimensional systems, where there are problems concerning parameters variations or where the signals provided by the sensors are not very accurate ([9] and [10]).

To put up such a model we need a fuzzy set and the original theory of fuzzy logic conceived by Lotfi A. Zadeh. The most serious problem it has to deal with comes from the determination of a complete set of rules and the establishing of the membership functions corresponding to each input. The many attempts to reduce errors and optimize the model are time – consuming and, very often, the results are far from the ones intended in the beginning. A modern design method allows, though, to build up competitive fuzzy models, in a relatively short interval. In short, ANFIS (Adaptiv Neuro-Fuzzy Inference System), the designing technique allows to generate and optimize the set of rules and the parameters of the membership functions by means of using the neural networks ([10]). Moreover, already implemented in Matlab Neuro-Fuzzy software tools, it is relatively easy to use.

2. THE NEW APPROACH

To solve the problem we will try to build a neuro-fuzzy controller for bias modeling. Considering the experimental data, it is possible to arrange an empirical model based on a neuro-fuzzy network. The model can learn the process behavior based on the input-output process data by using a fuzzy inference system (FIS) which should model the data set with two inputs and one output. Following the training, the model may be used for the error value generation corresponding to the sensor's measured parameters (a or ω) and to the temperature transducer parameter (T) followed by its compensation ([9] and [10]).

It is also possible to create a Fuzzy Inference System (FIS) using the Matlab "genfis2" function. This function generates an initial FIS of Sugeno type by decomposition of the operation domain into different regions using the fuzzy subtractive clustering method. For each region, a low order linear model can describe the process local parameters. Thus, the non-linear process is locally linearized around a functioning point by using the Least Squares method. Then, the obtained model is considered valid in the entire region around this point. To limit the operating regions implies the existence of overlapping among these different regions. Their definition is given in a fuzzy manner. Thus, for each model input, several fuzzy sets are associated with their corresponding definitions of their membership functions (mf). By combination of these fuzzy inputs, the input space is divided into fuzzy regions. For each such region, a local linear model is used, while the global model is obtained by defuzzification with the gravity centre method (Sugeno), by which the interpolation of the local models' outputs is done ([9] and [10]).

The Sugeno fuzzy model was proposed by Takagi, Sugeno and Kang to generate the fuzzy rules from a given input-output data set ([11]). For our system (two inputs and one output) a first-order model is considered, and for N rules is given by ([11] and [12]):

$$\begin{aligned}
&\text{Rule } 1\text{: If } x_1 \text{ is } A_1^1 \text{ and } x_2 \text{ is } A_2^1, \text{ then } y^1(x_1,x_2) = b_0^1 + a_1^1 x_1 + a_2^1 x_2,\\
&\qquad\vdots\\
&\text{Rule } i\text{: If } x_1 \text{ is } A_1^i \text{ and } x_2 \text{ is } A_2^i, \text{ then } y^i(x_1,x_2) = b_0^i + a_1^i x_1 + a_2^i x_2,\\
&\qquad\vdots\\
&\text{Rule } N\text{: If } x_1 \text{ is } A_1^N \text{ and } x_2 \text{ is } A_2^N, \text{ then } y^N(x_1,x_2) = b_0^N + a_1^N x_1 + a_2^N x_2,
\end{aligned} \tag{2}$$

where $x_q\,(q=\overline{1,2})$ are individual input variables, $A_q^i\,(i=\overline{1,N})$ are associated individual antecedent fuzzy sets of each input variable, and $y^i\,(i=\overline{1,N})$ is the first-order polynomial function in the consequent. $a_k^i\,(k=\overline{1,2},\, i=\overline{1,N})$ are parameters of the linear function and $b_0^i\,(i=\overline{1,N})$ denotes a scalar offset. The parameters $a_k^i,\, b_0^i\,(k=\overline{1,2},\, i=\overline{1,N})$ are optimized by Least Squares method.

For any input vector, $\boldsymbol{x}=[x_1,\, x_2]^T$, if the singleton fuzzifier, the product fuzzy inference and the centre average defuzzifier are applied, the output of the fuzzy model y is infered as follows (weighted average):

$$y = \left(\sum_{i=1}^{N} w^i(\boldsymbol{x})y^i\right) \Big/ \left(\sum_{i=1}^{N} w^i(\boldsymbol{x})\right), \tag{3}$$

where

$$w^i(\boldsymbol{x}) = A_1^i(x_1) \times A_2^i(x_2). \tag{4}$$

$w^i(\boldsymbol{x})$ represents the degree of fulfillment of the antecedent, that is, the level of firing of the i^{th} rule.

The Matlab "genfis2" function generates the membership functions of the Gaussian type, defined as follows ([10] and [12]):

$$A_q^i(x) = \exp\left\{-0.5\left(\frac{x - c_q^i}{\sigma_q^i}\right)^2\right\}, \tag{5}$$

where c_q^i is the cluster center, and σ_q^i is the dispersion of the cluster.

3. NUMERICAL SIMULATIONS RESULTS

In order to validate the method presented herein, a MEMS accelerometer is used. Its measurement domain is located in the range -10*g* to 10*g*. The measurements were performed at acceleration values ranging from -8*g* to 8*g* and at temperatures ranging from -20°C to 80°C. The accelerometer's measurement errors ε influenced by the temperature T variations for different acceleration a values are presented in Figure 1 ($\varepsilon = f(T)$), Figure 2 (($\varepsilon = f(a)$)) and Figure 3 ($\varepsilon = f(T, a)$), which is a combination of Figures 1 and 2.

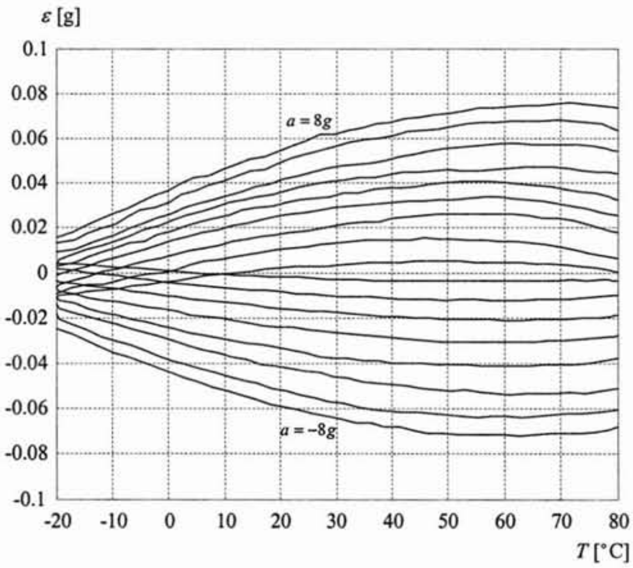


Figure 1 Error variation with the temperature

From the analysis of these three figures, it can be noticed that the bias error ε ranges within the interval limits from -0.08*g* to 0.08*g* for temperature variations of -20°C to 80°C, and for acceleration variations of -8*g* to 8*g*. Moreover, the error increases as the absolute value of the acceleration is augmented at a constant temperature. From the numerical and experimental data, the highest error value obtained for negative accelerations is ε = -0.0724*g* and is reached for a = -8*g* and T = 60°C, while the highest error value for positive accelerations is ε = 0.0759*g*, and is reached for a = 8*g* and for T = 70°C.

The FIS training is obtained with the Matlab "ANFIS" function, which uses a learning algorithm for the identification of the membership functions' parameters of the fuzzy inference system of a Sugeno type with two outputs and one input. We further consider as a starting point the input-output data and the FIS model generated with "genfis2", then the "ANFIS" optimizes the membership functions' parameters for a number of training epochs; this number is set by the user. The optimization is realized for a better process approximation performed by the neuro-fuzzy model by means of a quality parameter present in the training algorithm ([10]).

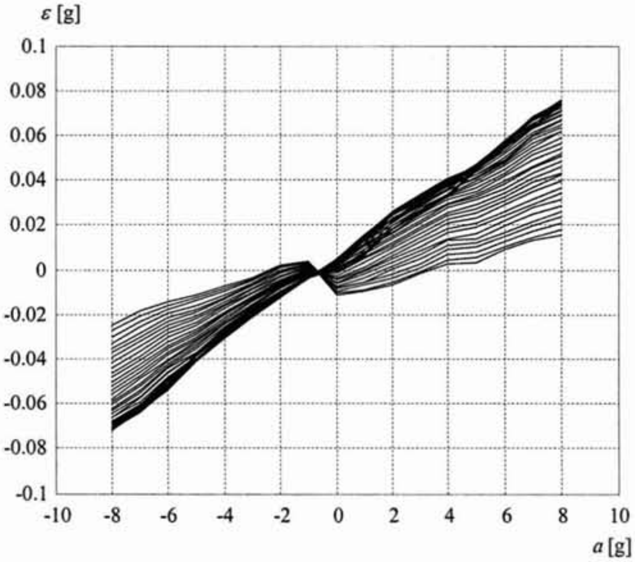


Figure 2 Error variation with the acceleration

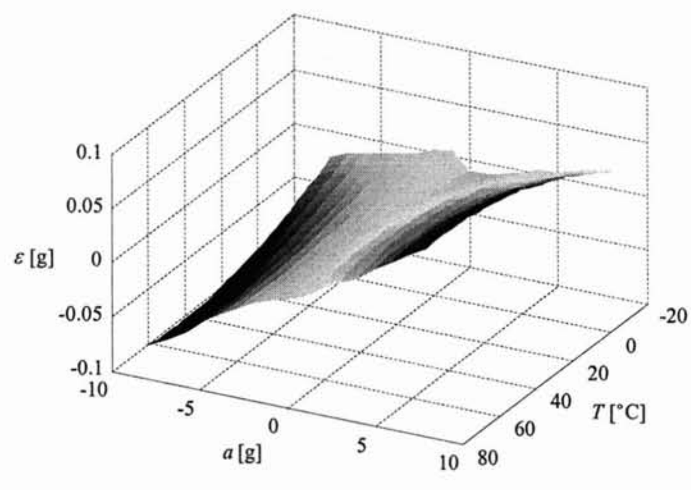


Figure 3 Accelerometer's errors in 3D space

The input-output data corresponding to the accelerometer's tests, shown in Figures 1 to 3, are arranged in a three-column matrix, in which the first two columns contain input data related to the temperature T and the accelerometer's output acceleration a, while the third column contains the output data related to the error ε. Using the Matlab "genfis2" function for the data set, the "BiasFis" Fuzzy Inference System is obtained.

To visualize the "BiasFis" FIS features, use is made of the Matlab "anfisedit" command followed by the FIS importation on the interface level. Visualization of the following characteristics of the untrained FIS is achieved through the use of the interface: Structure (Figure 4), Membership functions mf for the first input (temperature T) (Figure 5), Membership functions mf for the second input (acceleration a) (Figure 6), Defuzzification rules (Figure 7), and Surfaces generated for the model with the two inputs (Figure 8 – The 3D error presentation, Figure 9 – Error dependent on the temperature, Figure 10 - Error dependent on the acceleration).

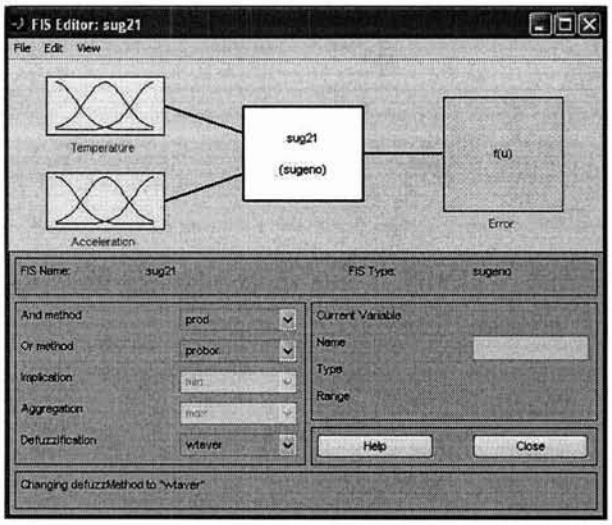


Figure 4 The “BiasFis” FIS structure

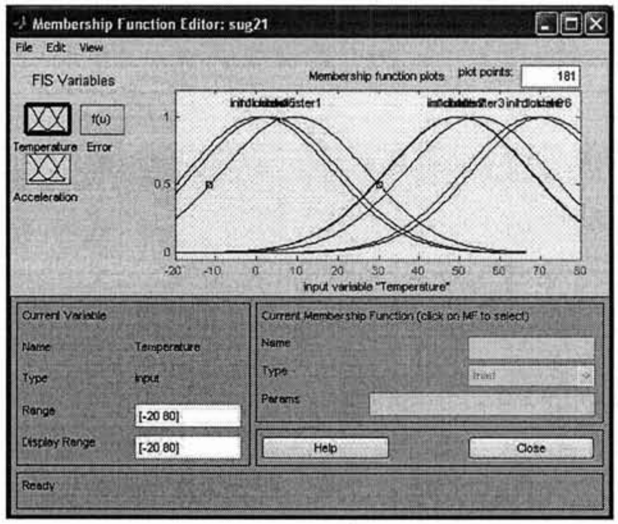


Figure 5 The mf of input 1

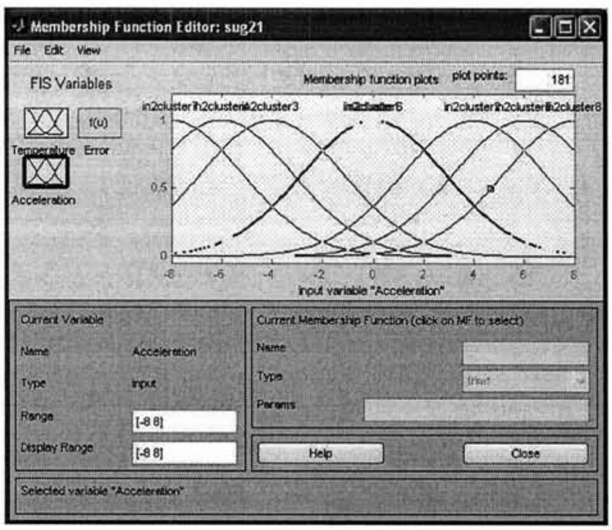


Figure 6 The mf of input 2

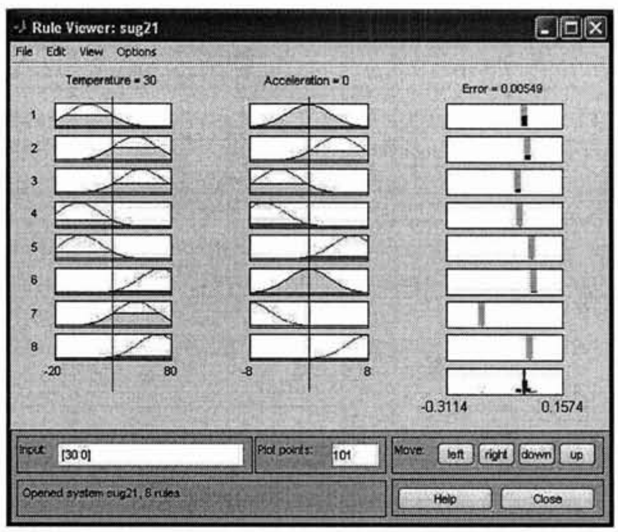


Figure 7 The deffuzification rules

The generated FIS is of Sugeno type, with two inputs and one output; for each of the two inputs, 8 Gaussian type membership functions mf are automatically generated. The output membership functions (mf) result from the use of the Sugeno inference engine based on the following logic: “if (the input 1 is mf k) and (the input 2 is mf k), then (the output is mf k)”, where “mf k” is the k’s membership function for each of the inputs, and respectively for the output. Before the training of the “ErrorFis” FIS, the parameters of the input membership functions mf are presented in Table 1. These parameters are half of the dispersion ($\sigma/2$) and the centre for the membership functions.

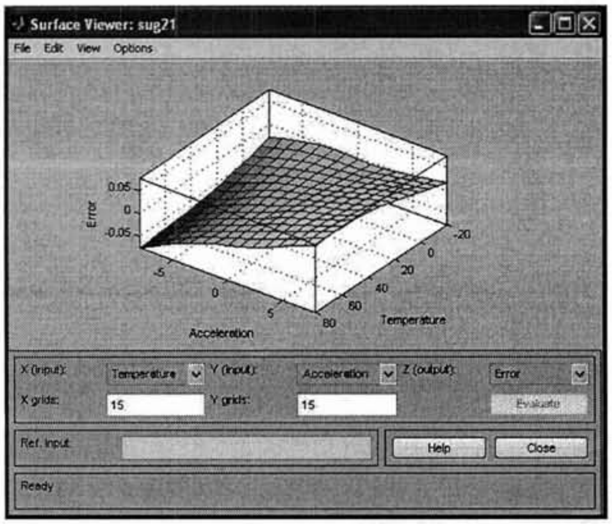


Figure 8 Variation of the error ε with a and T

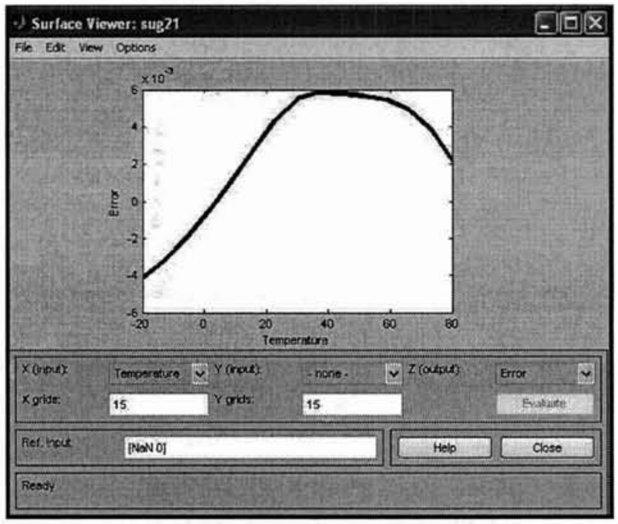


Figure 9 Variation of the error ε with T

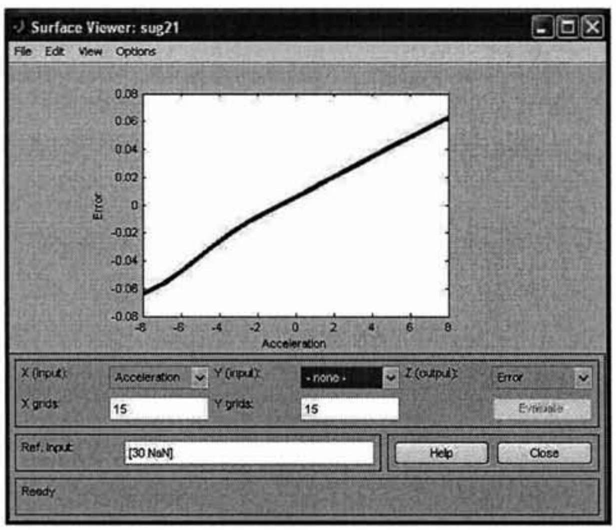


Figure 10 Variation of the error ε with a

By using the "ANFIS" Matlab function, an adaptive neuro-fuzzy type algorithm is used for the identification and modification of the membership function's parameters for the previously generated FIS. Figure 11 displays the deviation between the neuro-fuzzy model and the experimentally obtained data, defining the quality parameter from the training algorithm, for a number of 10, 10^2, 10^4 or 10^5 training epochs. The after training FIS evaluation is performed by use of the "evalfis" command; in Figure 12, the experimental data (represented with solid lines) and the corresponding "BiasFis" model (represented with dotted lines) over 10, 10^2, 10^4 and 10^5 training epochs are shown.

Table 1 The parameters of the inputs membership functions before the FIS training

Temperature [^{o}C]			Acceleration [g]		
	$\sigma/2$	Center		$\sigma/2$	Center
mf1	17.68	9.348	mf1	2.828	0
mf2	17.68	49.57	mf2	2.828	4
mf3	17.68	54.46	mf3	2.828	-4
mf4	17.68	0.92	mf4	2.828	-6
mf5	17.68	2.826	mf5	2.828	6
mf6	17.68	71.03	mf6	2.828	0.02312
mf7	17.68	49.29	mf7	2.828	-8
mf8	17.68	68.86	mf8	2.828	8

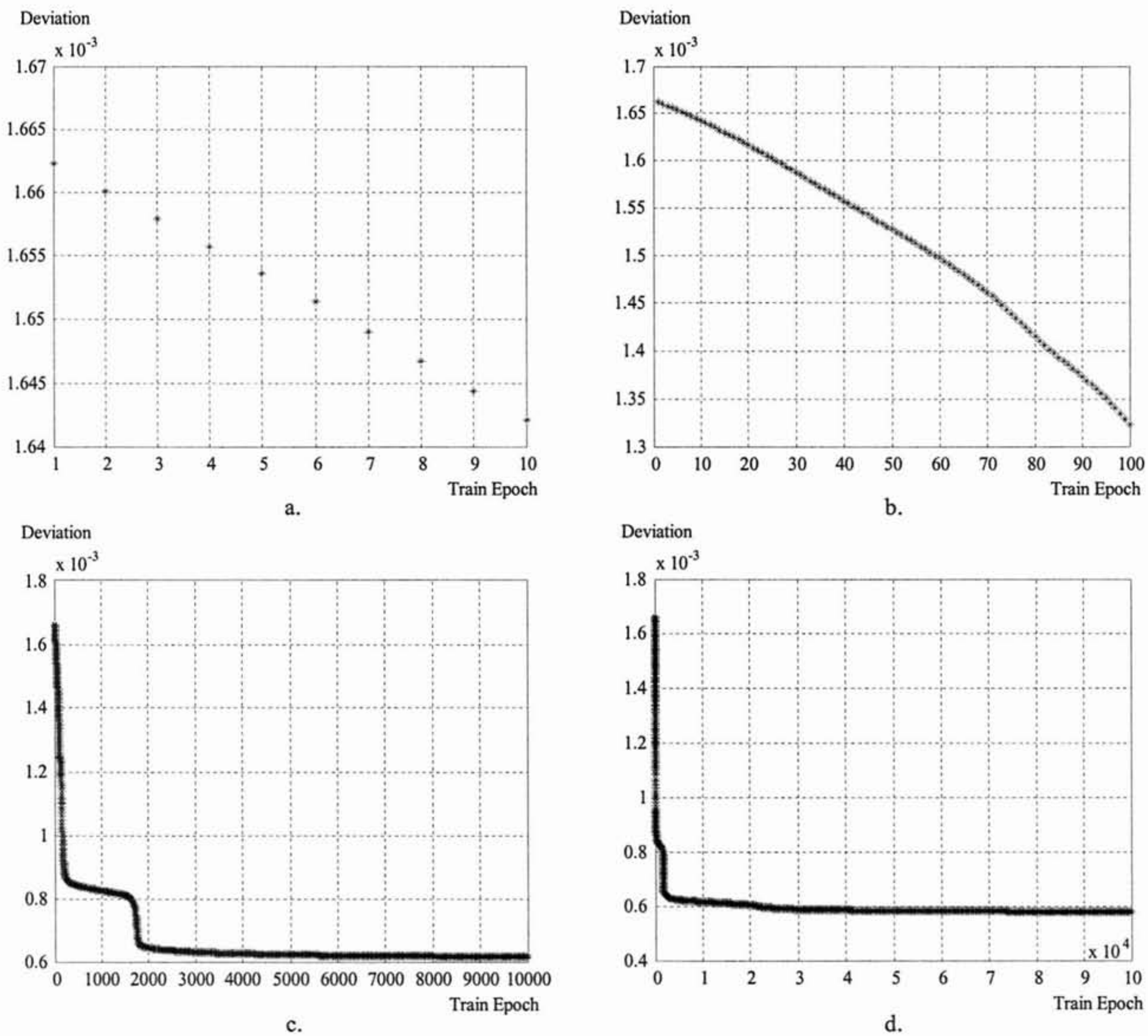


Figure 11 The deviation between the neuro-fuzzy model and the experimental data

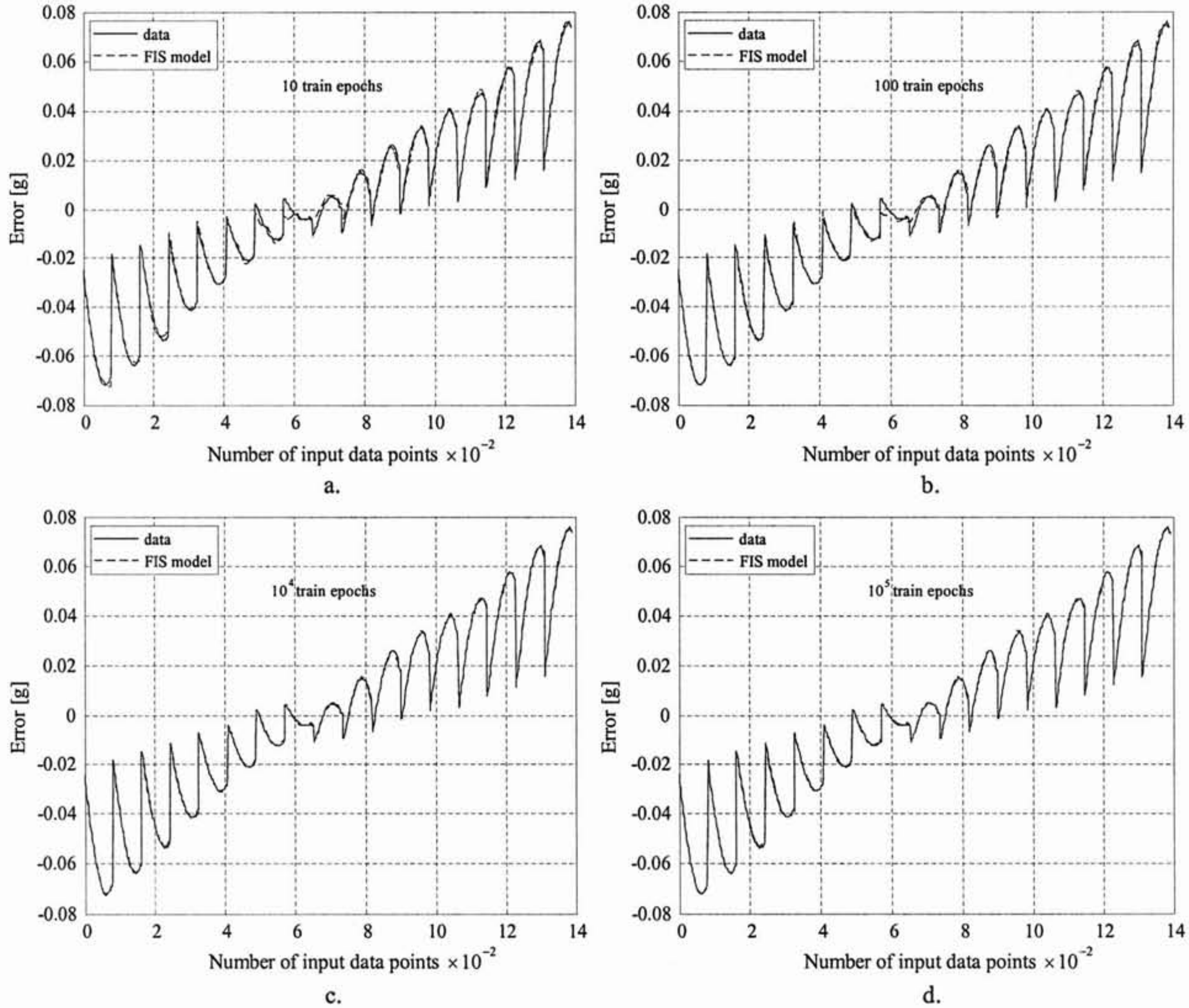


Figure 12 The after training FIS evaluation for different training epochs

Figure 11 shows a rapid decrease in the deviation between the experimental data and the neuro-fuzzy model for the quality parameter within the training algorithm over the first 200 training epochs. This decrease is followed by a slower decrease over the next 1400 epochs, by a quick decrease between the 1600 to 1800 epochs and finally by a very slow decrease from 1800 to 25000 training epochs. From Figure 11, it can also be observed that the FIS model may be trained on 10^5 epochs due to the fact that the deviation has an approximately constant value of $5.8 \cdot 10^{-4}$.

Figure 12 reiterates the same observations as those obtained from Figure 11. Note the overlapping of the FIS model (evaluated for the input data) with the experimental data. This superposition is dependent on the training epochs' number, and is better as the number of training epochs is larger. The model training over more than 10^5 epochs produces a very small deviation below $5.8 \cdot 10^{-4}$. An improved approximation of the real model is achieved with the neuro-fuzzy methods in the case when a higher number of experimental data is used for the error evolution description.

Through visualization of the trained model characteristics, using an "anfisedit" interface, the following figures are obtained: ANFIS model structure (Figure 13), Membership functions of the first input (temperature T) (Figure 14), Membership functions of the second input (acceleration a) (Figure 15), Defuzzification rules (Figure 16) and the Surfaces generated for the model with the two inputs (Figure 17 – The 3D error presentation, Figure 18 – Error dependent on the temperature, Figure 19 - Error dependent on the acceleration). The parameters of the input's membership functions, following the "BiasFis" FIS training over 10^5 epochs, are shown in Table 2.

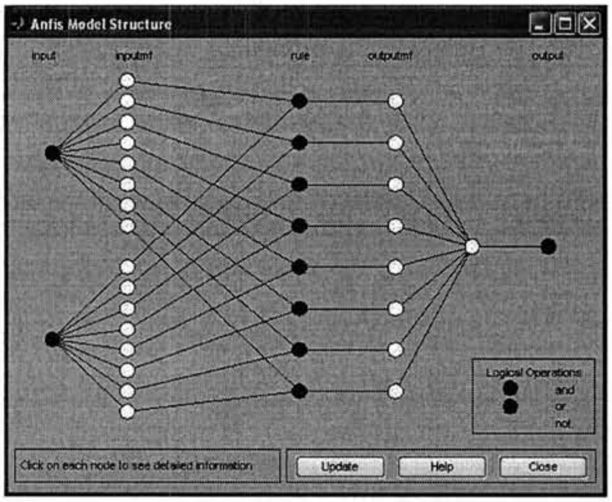


Figure 13 ANFIS model structure

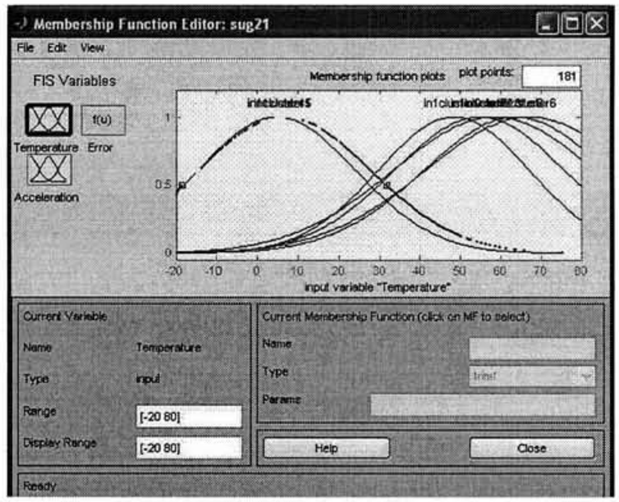


Figure 14 The mf of input 1

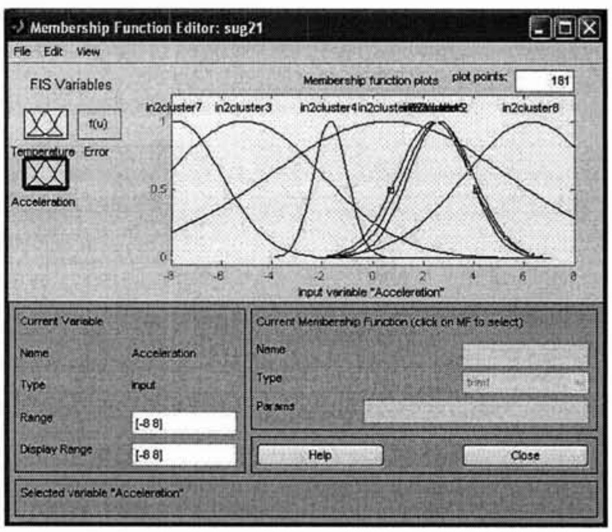


Figure 15 The mf of input 2

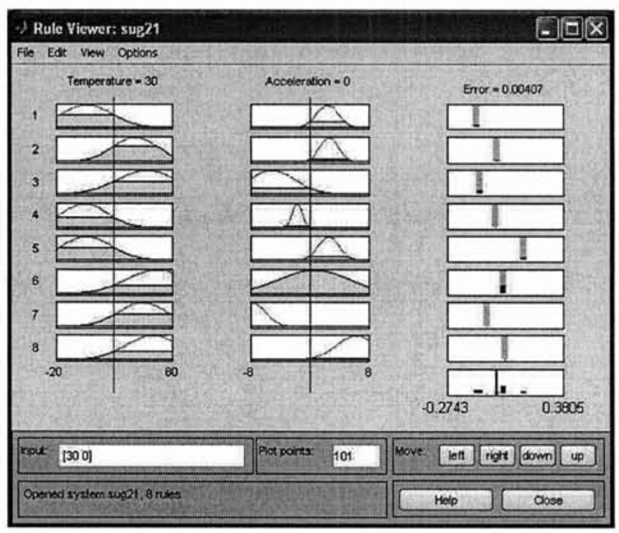


Figure 16 The deffuzification rules

Comparison of the FIS characteristics and the membership functions parameters, before and after the training from Tables 1 and 2, indicates a redistribution of the membership functions in the working domain and a change in their shapes by the dispersion modification. According to the parameter's values from Table 1, the FIS generation with the "genfis2" function has as 1st result the choice of the same dispersion for all membership functions which characterize an input. As 2nd result is the separation of the working space for the respective input, so that the model could be approximated through use of the Least Squares method, with local linear models.

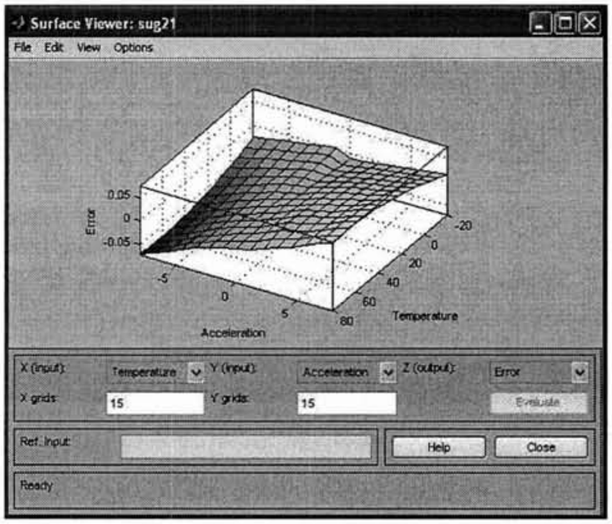


Figure 17 Variation of the error ε with a and T

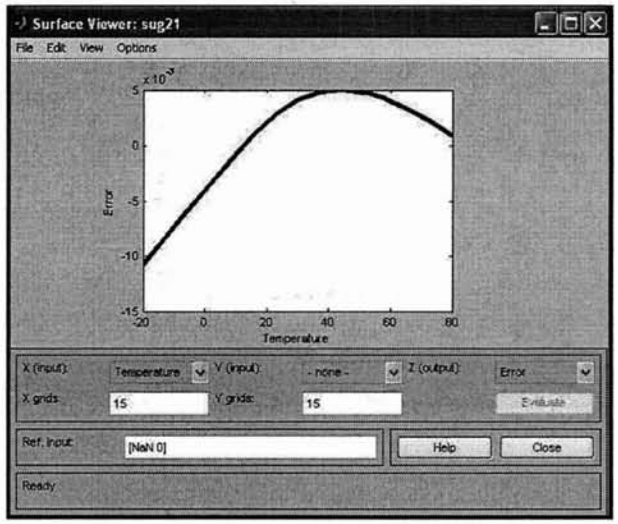


Figure 18 Variation of the error ε with T

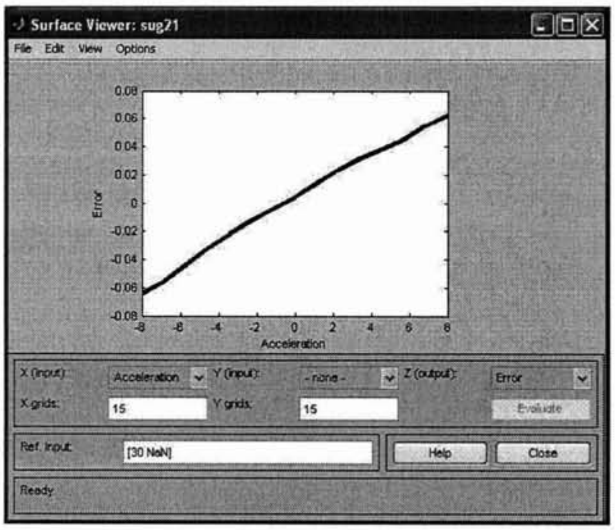


Figure 19 Variation of the error ε with a

The FIS model training with the "ANFIS" function uses a hybrid algorithm for the optimization of membership functions parameters. This algorithm uses a combination of the gradient method along with the Least Squares method ([10]). The results thus obtained are expressed in terms of membership functions parameters values in Table 2. Through training, the error variations curves in Figures 8 to 10 become of the forms in Figures 17 to 19, which tend to reach the shapes of the experimental curves shown in Figures 1 to 3. Subsequently, the error variation surface represented in 3D space in Figure 8 (before the training) and in Figure 17 (after the training) has the same shape as the experimentally obtained error variation surface (Figure 3).

The improvement in the experimental surface reproduction quality by the neuro-fuzzy model is shown from the analysis of the error dependency curves on each separate input parameter, before the training (Figure 9 for $\varepsilon(T)$ and Figure 10 for $\varepsilon(a)$) and after the training (Figure 18 for $\varepsilon(T)$ and Figure 19 for $\varepsilon(a)$).

Thus, the greater similitude of $\varepsilon(T)$ and $\varepsilon(a)$ shapes to the experimental shapes (Figures 1 and 2) is emphasized following the FIS model training during 10^5 epochs.

Table 2 The parameters of the input's membership functions following the FIS training

Temperature [°C]			Acceleration [g]		
	σ/2	Center		σ/2	Center
mf1	21.37	6.771	mf1	1.444	2.415
mf2	18.77	48.41	mf2	1.231	2.636
mf3	25.14	58.46	mf3	3.114	-5.09
mf4	19.45	4.8	mf4	0.6759	-1.675
mf5	21.27	6.497	mf5	1.402	2.604
mf6	25.8	67.1	mf6	4.123	0.873
mf7	21.16	54.82	mf7	1.825	-7.798
mf8	22.8	63.5	mf8	2.654	6.284

Evaluating the FIS for the experimental values of temperature and acceleration used in the modelling, and realizing the error correction, results in the 3D characteristics shown in Figure 20.a, at the same scale as the one in Figure 3. For a better visualization of the error variation domain after the correction, Figure 20.b shows the previous characteristics for a smaller domain on the error axis. The error ε dependence on the temperature and acceleration, after the correction, is shown in Figures 21 and 22. The curves shown in Figures 21 a. and 22 a. are drawn at the same scale as the ones shown in Figures 1 and 2. It is possible to visualize the error variation domain for the two characteristics ($\varepsilon(T)$ and $\varepsilon(a)$), obtained after the compensation, by decreasing the interval on which the error ε is represented (Figure 21.b for $\varepsilon(T)$ and Figure 22.b for $\varepsilon(a)$).

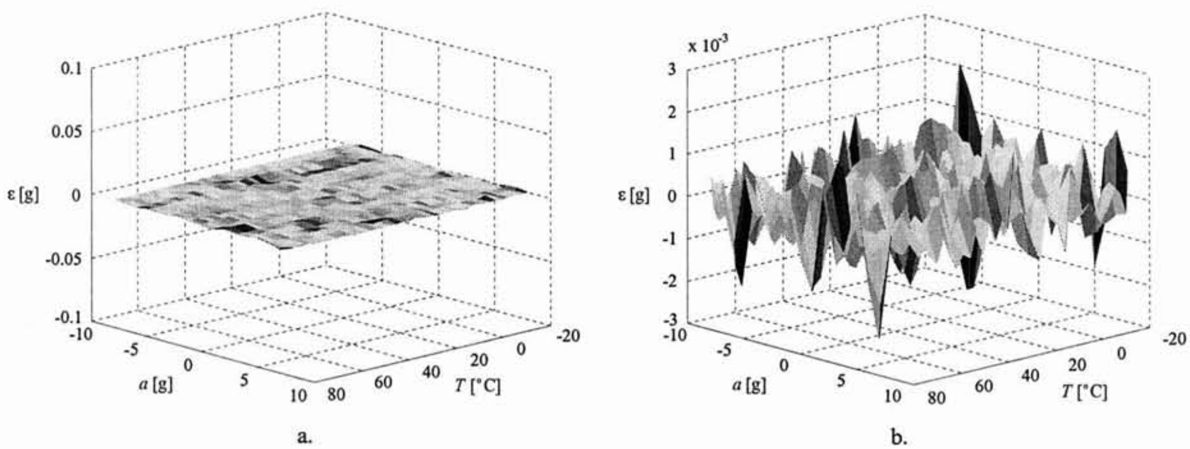


Figure 20 The accelerometer's errors in 3D after the compensation with the neuro-fuzzy model

From the numerical and graphical results obtained with the proposed neuro-fuzzy model simulation (Figures 20 to 22), a considerable decrease in the absolute error value due to the bias dependency with the temperature may be noted. At first glance, the error variation surface resulting from the compensation with the neuro-fuzzy model varies within the limits $-2.2\cdot 10^{-3}g$ to $2.1\cdot 10^{-3}g$. By means of evaluation of the obtained numerical values, we note that after compensation, the maximum error value on the positive axis is $\varepsilon = 0.00204277g$ and is reached for $a = -7g$ and $T = -20$°C, while the maximum error value on the negative axis is $\varepsilon = -0.00214114g$ and is reached for $a = 7g$ and $T = 80^{\circ}$C. The proposed algorithm brings about a decrease in the absolute value of the error, due to the bias depending on the temperature, of approximately 35.5 times (from $0.0759g$ to $0.00214114g$).

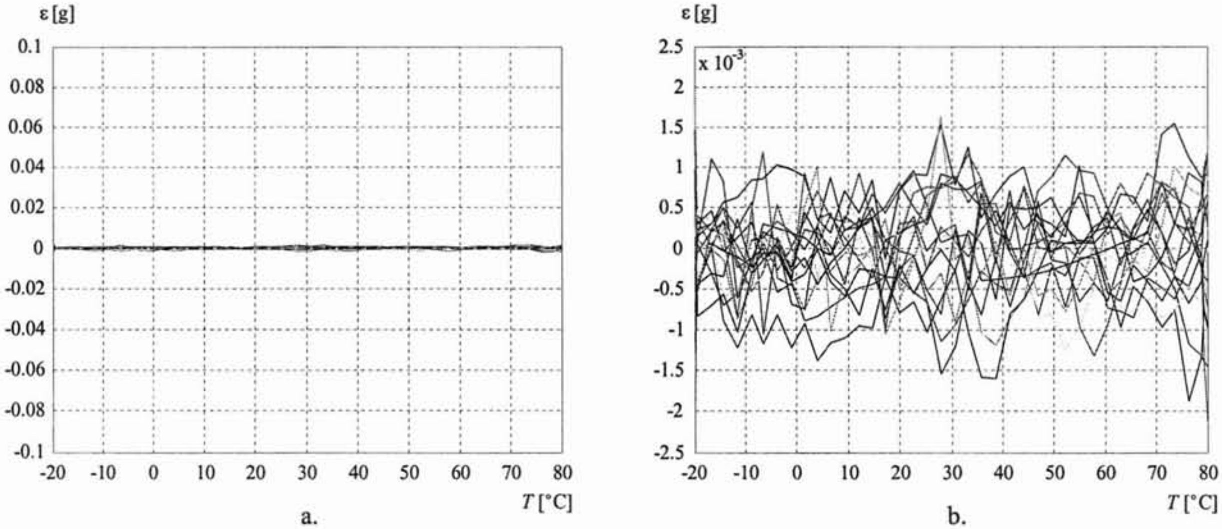


Figure 21 The dependence of the neuro-fuzzy corrected error on the temperature

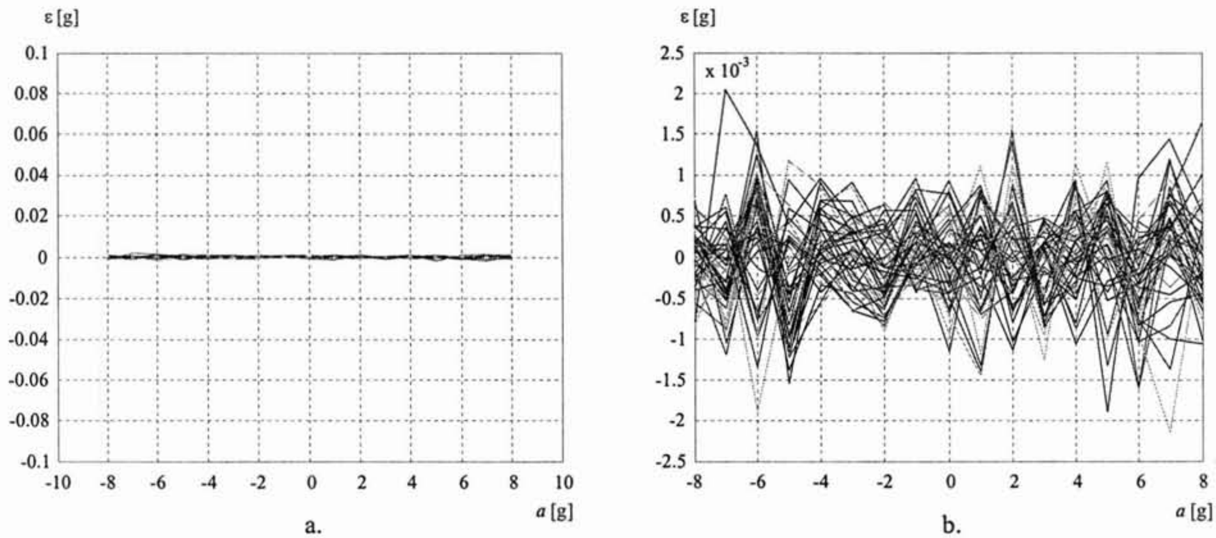


Figure 22 The dependence of the neuro-fuzzy corrected error on the acceleration

If the classical method ([7]) is used and the measurement errors are represented by the next model:

$$\varepsilon(a, T) = [T^2 \; T \; 1] \begin{bmatrix} C_{0,0} & C_{0,1} & C_{0,2} \\ C_{1,0} & C_{1,1} & C_{1,2} \\ C_{2,0} & C_{2,1} & C_{2,2} \end{bmatrix} \begin{bmatrix} a^2 \\ a \\ 1 \end{bmatrix} = [T^2 \; T \; 1] \cdot C \cdot \begin{bmatrix} a^2 \\ a \\ 1 \end{bmatrix}, \tag{2}$$

then the C matrix of the identified system coefficients is the following:

$$C = \begin{bmatrix} 0.00000003771659 & -0.00000107498726 & -0.00000218533022 \\ -0.00000245329695 & 0.00013507497521 & 0.00020861012578 \\ -0.00006327197021 & 0.00483399428181 & 0.00064374215072 \end{bmatrix}. \quad (3)$$

In this case, after the compensation, the maximum error value on the positive axis is $\varepsilon = 0.01018584g$ and is reached for a = -2g and T = 49.5652°C, while the maximum error value on the negative axis is ε = = -0.00699466g and is reached for a = 0g and T = 19.45°C. Therefore, the maximum absolute value of the error after the compensation is $\varepsilon = 0.01018584g$. Better results are obtained if the next model for the measurement errors is used:

$$\varepsilon(a, T) = [T^2 \; T \; 1] \begin{bmatrix} C_{0,0} & C_{0,1} & C_{0,2} & C_{0,3} \\ C_{1,0} & C_{1,1} & C_{1,2} & C_{1,3} \\ C_{2,0} & C_{2,1} & C_{2,2} & C_{2,3} \end{bmatrix} \begin{bmatrix} a^3 \\ a^2 \\ a \\ 1 \end{bmatrix} = [T^2 \; T \; 1] \cdot C \cdot \begin{bmatrix} a^3 \\ a^2 \\ a \\ 1 \end{bmatrix}. \quad (4)$$

In this situation, the C matrix of the identified system coefficients has the following form:

$$C = \begin{bmatrix} 0.00000001265880 & 0.00000004138672 & -0.00000161094446 & -0.00000230686729 \\ -0.00000117760096 & -0.00000260582336 & 0.00018685994707 & 0.00021037861578 \\ 0.00003273960828 & -0.00007062640202 & 0.00332033649480 & 0.00113105630911 \end{bmatrix}, \quad (5)$$

and, after the compensation, the maximum error value on the positive axis is $\varepsilon = 0.00708296g$ and is reached for a = -3g and T = 22.63°C, while the maximum error value on the negative axis is ε = = -0.00722522g and is reached for a = -2g and T = 14.78°C. In this case, the maximum absolute value of the error is $\varepsilon = 0.00722522g$.

After the errors compensation using the model described by equation (4), the variation of the error with the temperature and with the acceleration has the form shown in Figure 23.a, at the same scale as the ones shown in Figures 3 and 20.a. For a better visualization of the error variation domain after the correction, Figure 23.b shows the previous characteristics for a smaller domain on the error axis.

The error ε dependences on the temperature and acceleration, after the correction with the model expressed with equation (4) obtained by using the classical method, are shown in Figures 24 and 25. The curves shown in Figure 24.a are drawn at the same scale as the ones shown in Figures 1 and 21.a, and the curves shown in Figures 25.a are drawn at the same scale as the ones shown in Figures 2 and 22.a. It is also possible to visualize the error variation domain for the two characteristics ($\varepsilon(T)$ and $\varepsilon(a)$), obtained after the compensation with the classical obtained model, by decreasing the interval on which the error ε is represented (Figure 24.b for $\varepsilon(T)$ and Figure 25.b for $\varepsilon(a)$).

From the numerical and graphical results in the classical method case, on can observe that was obtained a decrease in the absolute value of the error, due to the bias depending on the temperature, of approximately 10.5 times (from 0.0759g to 0.00722522g).

Finally, if are compared the results obtained with the neuro-fuzzy proposed algorithm and with the classical method, one can conclude that in first case the absolute values of the errors are reduced of approximately 3.4 times more than in the classical case.

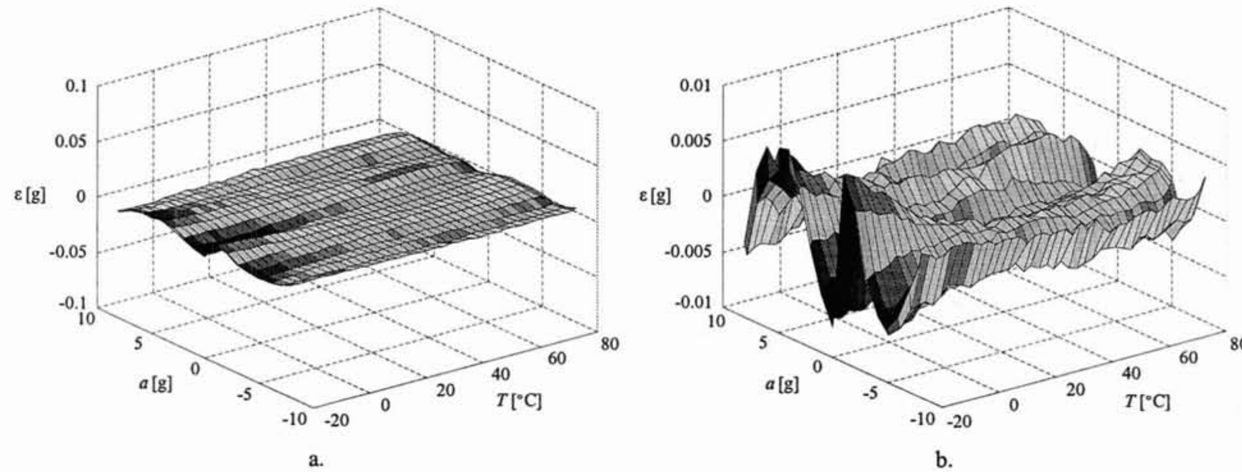


Figure 23 The accelerometer's errors in 3D after the compensation with the classical obtained model

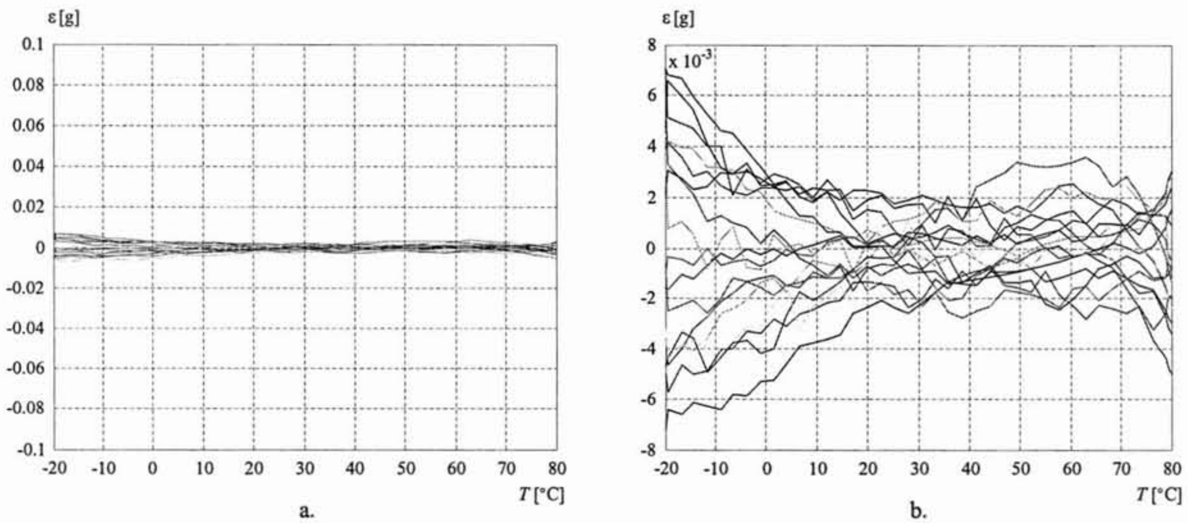
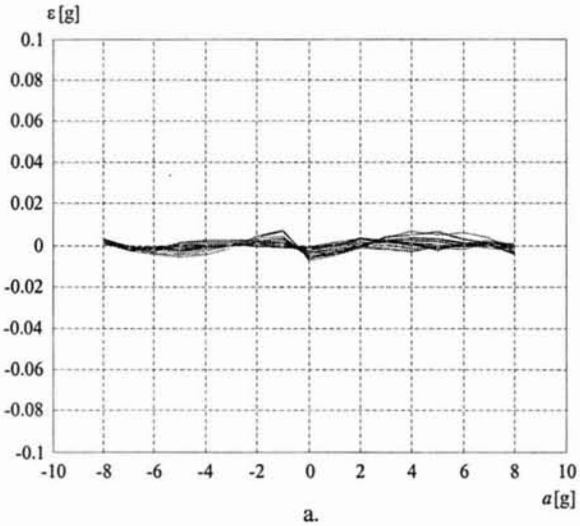


Figure 24 The dependence of the classical corrected error on the temperature

4. CONCLUSIONS

Using the method proposed herein, a new algorithm was conceived for the estimation and compensation of the bias dependence of the temperature for inertial sensors. The validation tests were realized for a MEMS accelerometer with a range of -10*g* to 10*g*. The experimental data were acquired for acceleration values between -8*g* and 8*g*, and for temperature values between -20°C and 80°C. On the basis of these experimental readings, an empirical model was realized based on a neuro-fuzzy network, which learned the process procedure by using a Fuzzy Inference System (FIS). This FIS training was accomplished using the "ANFIS" Matlab function, and the learning algorithm was used for 10, 10^2, 10^4 and 10^5 training epochs.

In Figure 11, a significant decrease in the deviation between the experimental and the neuro-fuzzy model was observed for the quality parameter of the training algorithm for the first 25000 training epochs. The FIS training evaluation for different training epochs, as seen in Figure 12, clearly shows the same types of specifications and conclusions as the ones given in Figure 11; we should emphasize the overlapping of the FIS model (evaluated for the input data) over the experimental data, which is better for higher numbers of training epochs. Following an analysis of these figures, observation was made of the fact that the FIS model was trained over 10^5 training epochs due to the fact that the deviation had an approximately constant value of $5.8 \cdot 10^{-4}$.



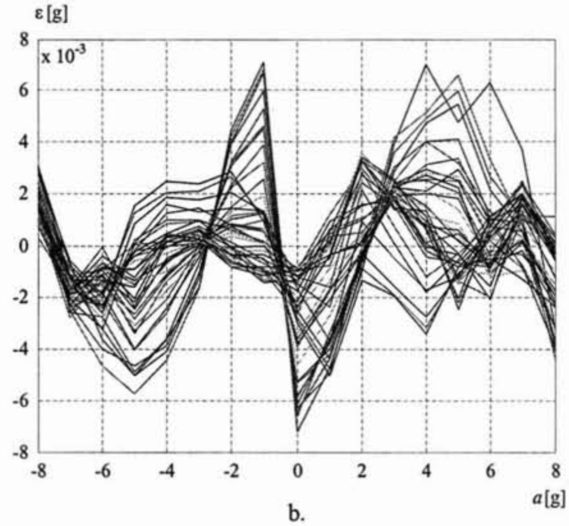


Figure 25 The dependence of the classical corrected error on the acceleration

The improvement of the experimental surface reproduction quality by use of the neuro-fuzzy model was shown from the analysis of the 3D surfaces representing the error dependency from Figures 3 and 17. Figures 18 and 19 emphasize the high similitude for the $\varepsilon(T)$ and $\varepsilon(a)$, and the experimental curves (Figures 1 and 2) after the FIS model training during 10^5 epochs.

The simulation of the proposed neuro-fuzzy model showed a considerable decrease of the absolute error value due to the bias depending on the temperature. By evaluation of the obtained numerical results, it was found that the maximum error value on the positive axis was $\varepsilon = 0.00204277g$ after the compensation, and was reached for $a = -7g$ and $T = -20°C$. The maximum error value on the negative axis was $\varepsilon = -0.00214114g$ and was reached for $a = 7g$ and $T = 80°C$. Note that the presented algorithm caused a decrease in the error absolute value due to the bias depending on the temperature of approximately 35.5 times (from $0.0759g$ to $0.00214114g$).

If a classical method are used (error model described by the formula (4)), after the compensation, the maximum error value on the positive axis is $\varepsilon = 0.00708296g$ and is reached for $a = -3g$ and $T = 22.63°C$, while the maximum error value on the negative axis is $\varepsilon = -0.00722522g$ and is reached for $a = -2g$ and $T = 14.78°C$. In consequence, a decrease in the absolute value of the error, due to the bias depending on the temperature, of approximately 10.5 times (from $0.0759g$ to $0.00722522g$) was obtained with the classical method (3.4 times bigger than in case of the neuro-fuzzy proposed model).

Another advantage of this new model is its rapid generation. The neuro-fuzzy model generation for the error ε is faster, due to the "genfis2" and "ANFIS" functions implemented already in Matlab, and it only assumes the model performance training using the "anfisedit" interface generated with Matlab.

Ease of use in real time is also a characteristic of the model. The only need was the importation of the FIS "BiasFis" on the level of a fuzzy controller in the Matlab/Simulink, followed by the compilation in C code of the controller and by the obtained code written in the navigation processor. Thus, the processor receives the acceleration and temperature data from the accelerometer, and based on the model, the corrected acceleration is generated.

5. REFERENCES

1. Honeywell Avionics webpage <http://www.honeywell.com/sites/aero/technology/avionics.htm>.
2. KVH Industries webpage <http://www.kvh.com>.
3. Northrop Grumman weppage <http://www.northropgrumman.com>.
4. SAAB Technologies webpage <http://products.saab.se>.
5. IEEE Std. 1293-1998 *„IEEE Standard Specification Format Guide and Test Procedure for Linear, Single-Axis, Nongyroscopic Accelerometers"*, Published by IEEE, New York, USA, 16 April, 1999.
6. IEEE Std. 836-1991 *„IEEE Recommended Practice for Precision Centrifuge Testing of Linear Accelerometers"*, Published by IEEE, New York, USA, June 15, 1992.
7. Borenstein, J. *„Experimental evaluation of a fiber optics gyroscope for improving dead-reckoning Accuracy in Mobile Robots"*, 1998 IEEE International Conference on Robotics and Automation, Leuven, Belgium, May 16-21, pp. 3456-3461, 1998.
8. Ojeda, L., Chung, H., Borenstein, J. *„Precision-calibration of Fiber-optics Gyroscopes for Mobile Robot Navigation"*, Proceedings of the 2000 IEEE International Conference on Robotics and Automation, San Francisco, CA, April 24-28 , pp. 2064-2069, 2000.
9. Kosko, B. *„Neural networks and fuzzy systems – A dynamical systems approach to machine intelligence"*, Prentice Hall, New Jersey, 1992.
10. *** Matlab Fuzzy Logic and Neural Network Toolboxes - Help.
11. Mahfouf, M., Linkens, D. A., Kandiah, S. *„Fuzzy Takagi-Sugeno Kang model predictive control for process engineering"*, The Institution of Electrical Engineers. Printed and published by the IEE, Savoy place, London WCPR OBL. UK, 1999.
12. Kung, C.C., Su, J.Y. *„Affine Takagi-Sugeno fuzzy modelling algorithm by fuzzy c-regression models clustering with a novel cluster validity criterion"*, IET Control Theory Appl., 1, (5), pp. 1255-1265, 2007

NOMENCLATURE

A_q^i = associated individual antecedent fuzzy sets of each input variable $(i=\overline{1,N})$

C = matrix of the identified system coefficients

$C_{i,j}$ = identified system coefficients

N = number of the fuzzy rules

T = temperature

a = acceleration of the MEMS sensor

a_k^i = parameters of the linear function in the rules set ($k=\overline{1,2}$, $i=\overline{1,N}$)

b_0^i = scalar offset in the rules set $(i=\overline{1,N})$

c_q^i = cluster center

g = gravitational acceleration

$\boldsymbol{x}$ = input vector

y = output of the fuzzy model

x_q = individual input variables $(q=\overline{1,2})$

y^i = first-order polynomial function in the consequent $(i=\overline{1,N})$

w^i = degree of fulfillment of the antecedent, that is, the level of firing of the i^{th} rule
ε = error due to the bias temperature dependence
ω = angular speed indicated by the gyro
σ = dispersion of the Gaussian membership function
σ_q^i = dispersion of the cluster
FIS = fuzzy inference system
mf = membership functions
mf k = the k's membership function