New adaptive controller method for SMA hysteresis modelling of a morphing wing

T. L. Grigorie and R. M. Botez
Laboratory of Research in Active Controls,
Avionics and AeroServoElasticity LARCASE
École de Technologie Supérieure,
www.larcase.etsmtl.ca
Montréal, Quebec
Canada

ABSTRACT

A neuro-fuzzy controller method for smart material actuator (SMA) hysteresis modelling is presented, conceived for a morphing wing application. The controller correlates each set of forces and electrical currents that are applied to the smart material actuators with the actuator elongation. The actuator is experimentally tested for four forces, using a variable electrical current. The final controller is obtained through the Matlab/Simulink integration of three independent neuro-fuzzy controllers, designed for the increase and decrease of electrical current, and for null electrical current in the cooling phase of the actuator. This final controller gives a very small error with respect to the experimental values.

NOMENCLATURE

A_q^i	associated individual antecedent fuzzy sets of each input variable
a_k^i	parameters of the linear function ($k = \overline{1,2}$, $i = \overline{1,N}$)
b_0^i	scalar offset ($i = \overline{1,N}$)
C_p	pressure coefficient
C_q^i	cluster centre
F	force
i	electrical current
m	variable for neuro-fuzzy controller selection
N	number of rules
Re	Reynolds number
T	temperature of the smart material actuator
T_{amb}	ambient temperature
V	speed
w^i	degree of fulfillment of the antecedent, i.e., the firing level of the ith rule
$\mathbf{x}$	input vector
x_q	individual input variables ($q = \overline{1,2}$)
y	output of the fuzzy model
y^i	first-order polynomial function in the consequent ($i = \overline{1,N}$)
α	angle-of-attack
σ	dispersion
σ_q^i	dispersion of the cluster
Δt	time
$\Delta\delta$	elongation of the actuator

Paper No. 3410. Manuscript received 26 February 2009, revised 20 August 2009, accepted 2 September 2009.

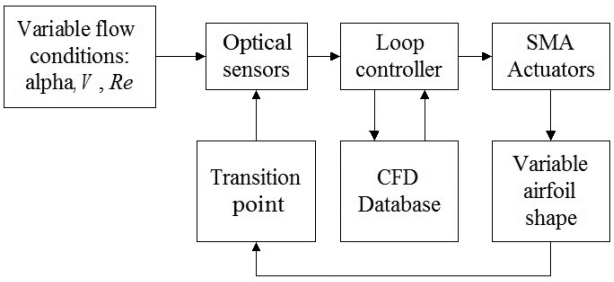

Figure 1. Closed-loop morphing wing system.

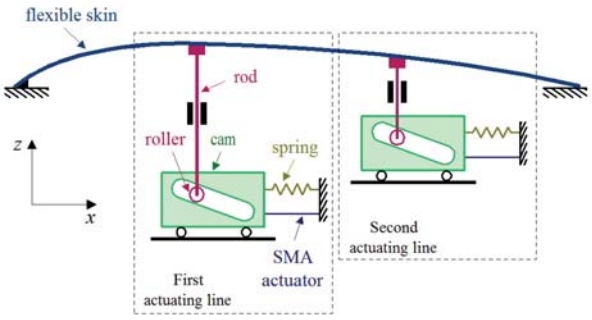

Figure 2. Structure of the actuating system with SMAs.

1.0 INTRODUCTION

The continuous need to reduce fuel consumption and other direct operating costs for new aircraft has spurred significant research efforts in the morphing wing area. These technologies will enable aircraft to be more efficient and operate under a wide range of varying flight conditions. Also, the morphing technologies will be used to improve aircraft performance, expand the flight envelope, replace conventional control surfaces, reduce drag and thus improve range, and reduce vibrations and flutter[1].

Multidisciplinary morphing wing research projects involve extremely complex interactions between controls, aerodynamics, structures, actuator power requirements, sensor integrations and all of the other components[2].

Within this context, our team has included wind tunnel simulations and experimental multidisciplinary studies for a morphing wing equipped with a flexible skin, smart material actuators (SMAs) and optical sensors. The aim of these studies is to move the transition point from the laminar to the turbulent regime closer to the trailing edge by use of a controller in order to obtain a larger laminar flow region. This research project included the following: optical sensor selection and testing for laminar-to-turbulent flow transition validation (by use of XFoil code and Matlab), smart material actuator controller methods, aeroelasticity wing studies using MSC/Nastran, open loop and closed loop transition delay controller design, integration and validation on a wing equipped with SMAs and optical sensors (simulation versus test results).

As shown in Fig. 1[3], a complex system was obtained, which modifies the aerofoil shape in order to optimise it from the perspective of the laminar flow region. For various flight conditions (angles-of-attack α, speeds V and Reynolds numbers Re), the loop controller would receive the aerofoil upper surface C_p distribution, determined from the surface pressure distribution measured by the optical sensors. The C_p distribution is compared with a computational fluid dynamics (CFD) database, which is generated so that for different aerpfoil types, the transition point is given as a function of the C_p distribution. Once a match is found, a transition point is offered to the loop controller by the CFD database; the controller will be able to decide if the aerofoil shape must be adjusted or not. The adjustment of the aerofoil shape is made in real time using the SMA actuators. The loop is closed by the aerofoil shape, which offers the optical sensors another surface pressure distribution.

In order to validate the morphing wing system (numerical simulation versus test results), good numerical models for each of the physical elements in the system must be obtained. The aim of this paper is to offer a good model for SMA actuators, with direct application to our morphing wing system.

Our goal is to build a neuro-fuzzy controller for SMA hysteresis modelling that is focused directly on our application. Other authors have presented SMA modelling using Neural Network methods[4] where the SMA actuator was controlled without using a position sensor, and then the neural network inverse model was used as a feedforward controller. The experimental results demonstrated the effectiveness of the neural network open loop controller for tracking control of the SMA wire actuator. A nonlinear control system for an SMA actuator using neural network based controllers was designed, and the experimental results showed that an SMA resistance control was possible[5]. There are other approaches for controller modelling of SMA actuators, based on neural network and fuzzy logic techniques. The technique presented is a combination of both, and thus it is called the neuro-fuzzy method. The model is empirical, based on the numerical values resulting from the SMA's experimental testing. It profits by the outstanding properties of fuzzy logic, which allows the empirical processing of the signals without mathematical analytical models. Another advantage of fuzzy logic is that it works very well on nonlinear and multi-dimensional systems.

To elaborate the model, a fuzzy rules set and the membership functions (mf) associated with each of the inputs are required[6,7]. Obtaining a good fuzzy model depends on the ability and the experience of the designer to consider the rules and the membership functions of each of the inputs. However, a relatively new design method allows a competitive model to be built by using a combination of fuzzy logic and neural-network techniques. Moreover, this method makes it possible to generate and to optimise the fuzzy rules set and the parameters of the membership functions. Already implemented in Matlab[6,8], the method is easy to use, and gives very good results very quickly.

2.0 SMA EXPERIMENTAL TESTING

The SMA actuating system has the structure shown in Fig. 2. Each system actuating line contains a cam, which moves in translation relative to the structure (on the x-axis in Fig. 2); this cam causes the movement of the related rod on the roller and on the skin (on the z-axis). The recall is in the form of a compression spring. When the SMA is heated, the actuator contracts and the cam moves to the right, resulting in the upwards roller movement and in the upwards skin displacement. On the contrary, SMA cooling results in a movement of the cam to the left, and thus into a downwards skin displacement.

As illustrated in Fig. 3, the SMA model's aim is to achieve actuator elongation ($\Delta\delta$) under the application of a thermo-electro-mechanical load over a certain period of time (Δt). The load's actuator can be operated by varying the temperature (T_{amb}), by injecting an electric current (i) or by application of a force (F). The actuator geometry consists of an SMA wire with constant section area and perimeter.

The SMA testing was performed at T_{amb} = 24°C using the bench test shown in Fig. 4, for four load cases of values 60N, 110N, 160N and 205N. During these tests, electrical current was supplied by use of a zero-increasing-decreasing-zero cycle for each of these four load cases. The following parameters were recorded: the time, the SMA supplying electrical current, the load force, the material temperature and the actuator elongation (which was measured using a linear variable differential transformer LVDT).

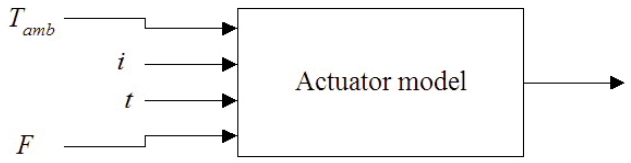

Figure 3. The SMA model input/output structure.

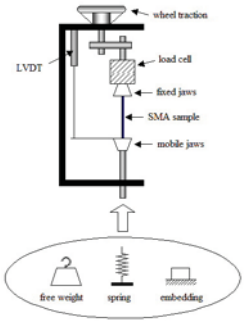

Figure 4. The SMA bench test.

In this study, the hysteresis cycles of elongation as a function of temperature were shown. The ambient temperature, the electrical currents and the forces, experimentally obtained, were applied to the SMA model as inputs, with the elongation and the material temperature considered as outputs. The elongation curves as a function of temperature for different load cases, and the 3D characteristics for the experimental data in terms of temperature, elongation and force are depicted in Fig. 5. A comparison between the elongation curves obtained experimentally and those obtained with the new technique is presented in the next sections.

3.0 THE NEW APPROACH

A neuro-fuzzy network empirical model is devised based on the numerical values from the SMA experimental testing. This model can learn the process behaviour, based on the input-output process data, by using a fuzzy inference system (FIS) to model the experimental data. The experimental elongation-current curves obtained in the four load cases considered here are represented in Fig. 6.

One can observe that each of these curves describes a hysteresis cycle, characterised by three distinct zones: electrical current increase, electrical current decrease and null electrical current in the cooling phase of the actuator. Therefore, three fuzzy inference systems FISs are used to obtain three neuro-fuzzy controllers: one for the current increase, one for the current decrease, and one for the null current (after its decrease).

For the first and for the second controller, the force and the current inputs are used, while for the third controller, the force and the time inputs are required for the SMA to recover its initial temperature (of approximately 24°C). Finally, the three controllers would be integrated into a single controller.

The design of the first two controllers is based on the available experimental data, in which two elongations are used for the same values of forces and currents (see Fig. 6). Due to the experimental data hysteresis, they cannot be represented as algebraic functions. Therefore, it is impossible to use data in the same FIS representation because an interpolation between the two elongation values is obtained for the same values of forces and currents. On the other hand, the null current values following the current decrease phase should not be considered as inputs in the third controller, because they are not suggestive. During this phase, for practical reasons, the actuator temperatures should be used instead, but temperature should

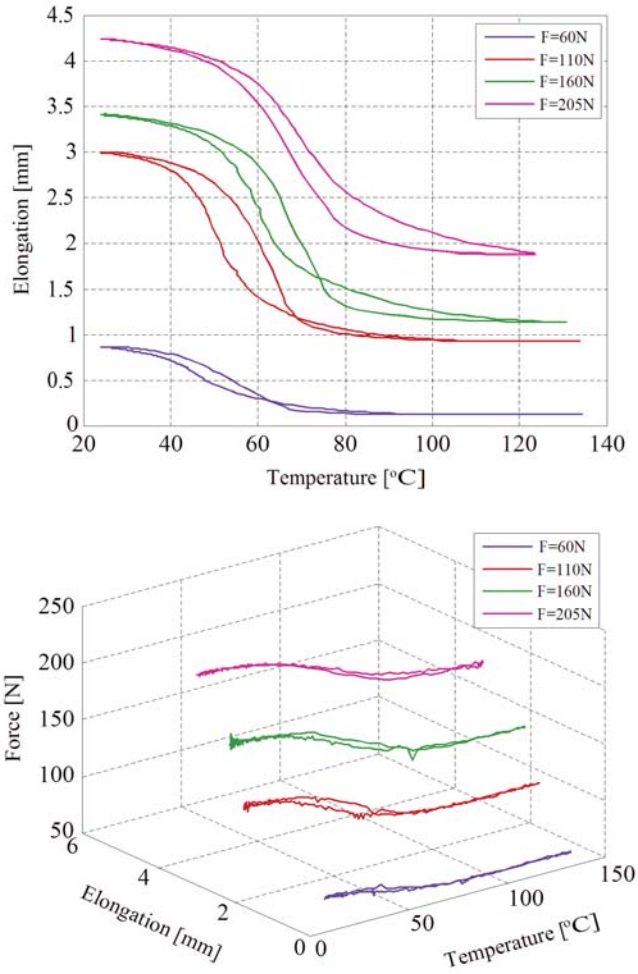

Figure 5. Experimental data evaluation.

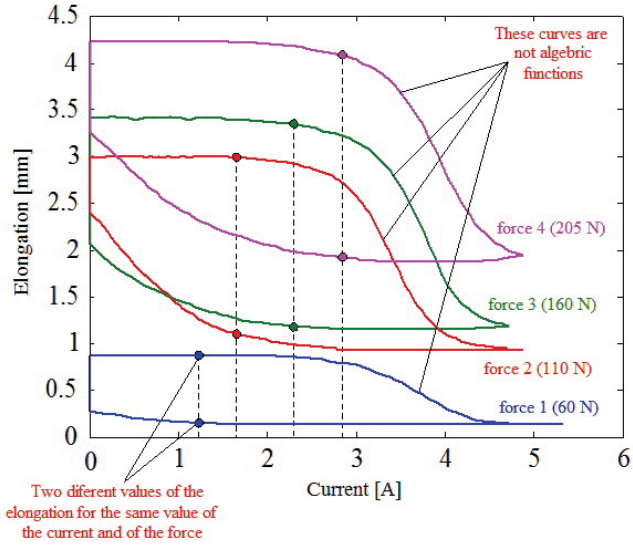

Figure 6. Elongation versus the current values for four forces.

remain one of the model outputs. The time values are very relevant for this phase as they represent actuator thermal inertia measures. Therefore, time is the second input of the third controller, with force as its first input (the time should be counted as starting when the current becomes null).

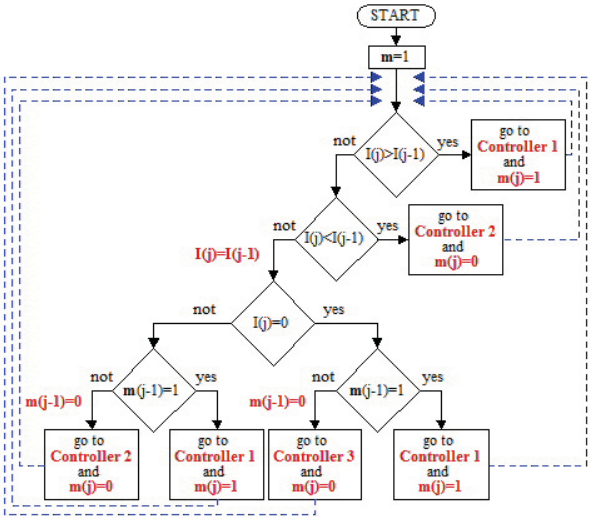

Figure 7. Logical scheme for the three controller's integration.

It is also possible to create fuzzy inference systems (FISs) using the Matlab 'genfis2' function, which generates an initial Sugeno-type FIS by decomposition of the operation domain into different regions using the fuzzy subtractive clustering method. For each region, the local process parameters could be described with a low order linear model. The non-linear process is thereby linearised locally around a functioning point by using the least squares method. The obtained model is then considered valid in the entire region around this point. The limitations of the operating regions imply 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 membership function (*mf*) definitions. By combining these fuzzy inputs, the input space is divided into fuzzy regions. For each such region, a local linear model is used, while a global model is obtained by defuzzification with the gravity centre method (Sugeno), through which the interpolation of the local models' outputs is realised[6, 8].

The Sugeno fuzzy model was proposed by Takagi, Sugeno and Kang to generate fuzzy rules from a given input-output data set[9]. For our system, and for all the three FISs (two inputs and one output), a first-order model was considered; the description of the N rules is given as Refs 9 and 10:

Rule 1: If x_1 is A_1^1 and x_2 is A_2^1, then $y^1(x_1,x_2) = b_0^1 + a_1^1 x_1 + a_2^1 x_2$,

Rule i: If x_1 is A_1^i and x_2 is A_2^i, then $y^i(x_1,x_2) = b_0^i + a_1^1 x_1 + a_2^i x_2$, . . . (1)

Rule N: If x_1 is A_1^N and x_2 is A_2^N, then $y^N(x_1,x_2) = b_0^N + a_1^N x_1 + a_2^N x_2$,

where $x_q (q = \overline{1,2})$ are the individual input variables, $A_q^i (i = \overline{1,N})$ are the 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. In addition, $a_k^i (k = \overline{1,2},\ i = \overline{1,N})$ contains the linear function parameters and $b_0^i (i = \overline{1,N})$ denotes a scalar offset. The parameters a_k^i, $b_0^i (k = \overline{1,2},\ i = \overline{1,N})$ b_0^i $(i = \overline{1,N})$ are optimised by use of the least square method.

For any input vector $\mathbf{x} = [x_1, x_2]^T$, in the case with a singleton fuzzifier, the product fuzzy inference and the centre-average defuzzifier are applied and the fuzzy model output y is inferred as follows (weighted average):

$$y = \left(\sum_{i=1}^{N} w^i(\mathbf{x}) y^i\right) / \left(\sum_{i=1}^{N} w^i(\mathbf{x})\right) \quad \ldots (2)$$

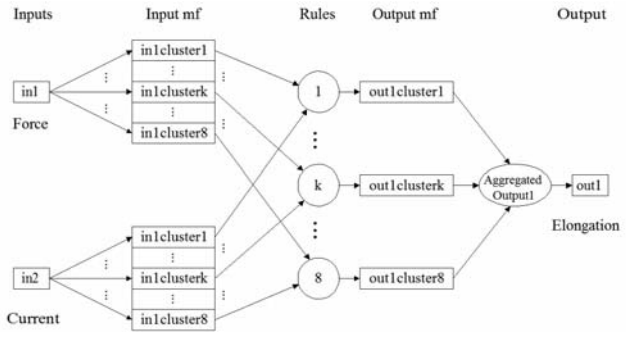

Figure 8. The structure of 'ElongationFis' FIS.

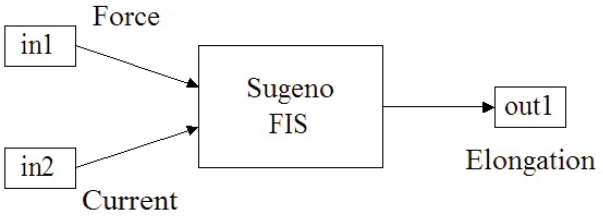

Figure 9. The structure of Controller 1.

$$w^i(\mathbf{x}) = A_1^i(x_1) \times A_2^i(x_2). \quad \ldots (3)$$

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

The Matlab 'genfis2' function generates Gaussian type membership functions, defined as follows[7, (10)]:

$$A_q^i(x) = \exp\left\{-0{\cdot}5\left(\frac{x - c_q^i}{\sigma_q^i}\right)^2\right\} \quad \ldots (4)$$

where c_q^i is the cluster centre and σ_q^i is its dispersion.

The FIS's training is achieved with the Matlab 'ANFIS' function, which uses a learning algorithm for the identification of the membership functions' Sugeno-type FIS parameters with two outputs and one input. As a starting point, the input-output data and the FIS model generated with the 'genfis2' function are considered. Matlab's 'ANFIS' optimises the membership functions' (*mf*) parameters for a given number of training epochs; a number set by the user. This optimisation is realised so that a better process approximation can be performed by the neuro-fuzzy model thanks to a quality parameter in the training algorithm[8]. Following the training, the models may be used for elongation value generation corresponding to the input parameters.

After the three controllers (Controller 1 for increasing current, Controller 2 for decreasing current and Controller 3 for null current) have been obtained, they should be integrated. This scheme is shown in Fig. 7. The decision to use one of the three controllers depends on the current vector type (increasing ($I(j) > I(j-1)$), decreasing ($I(j) < I(j-1)$), constant ($I(j) = I(j-1)$) or zero($I(j) = 0$)) and on the 'm' variable value. This 'm' value determines if a constant current value is part of an increasing vector or part of a decreasing vector. The initial 'm' value is equal to 1 when Controller 1 is used, and is equal to 0 when Controller 2 or Controller 3 is used.

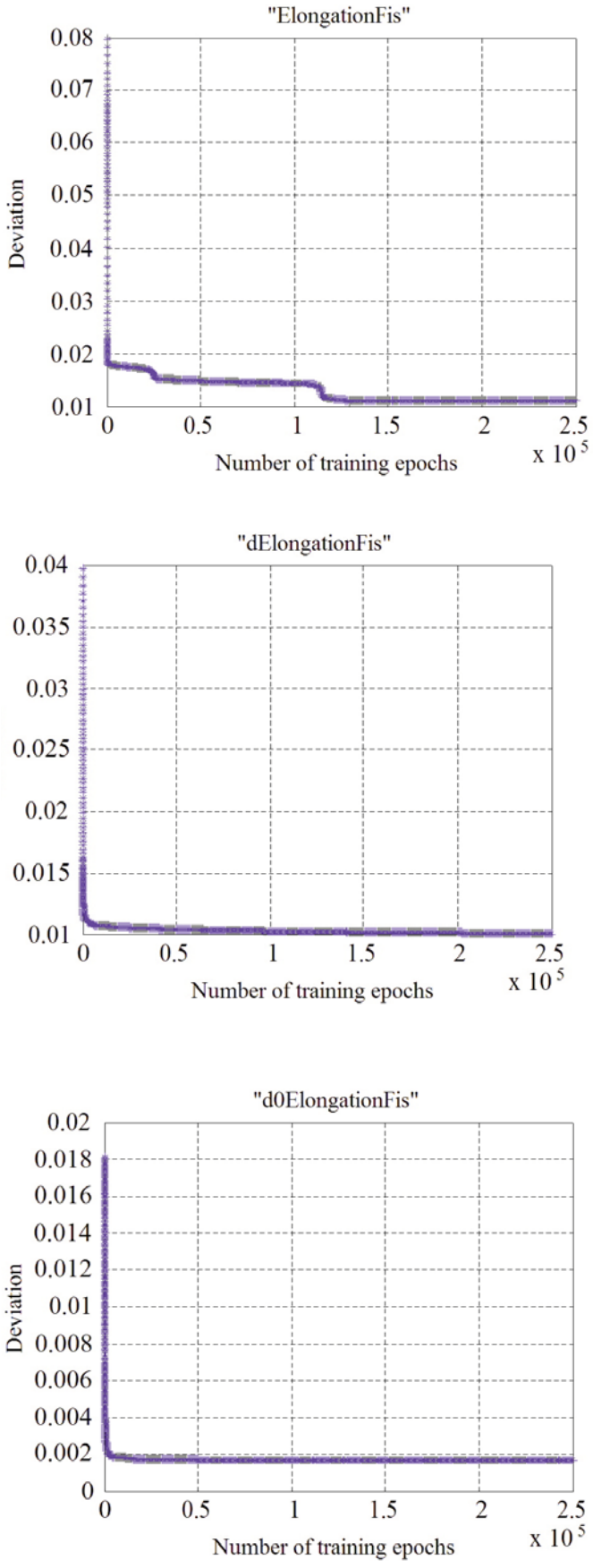

Figure 10. The training error for all three fuzzy inference systems.

4.0 THE CONTROLLER DESIGN AND EVALUATION

The 'genfis2' Matlab function[8] was used to build and train the following three fuzzy inference systems: 'ElongationFis' (for the current increase phase), '*d*ElongationFis' (for the current decrease phase), and '*d*0ElongationFis' (for the null current values obtained after the decrease phase). The three FISs were trained for different training epochs and the best results were obtained for 250,000 epochs.

For the system presented here, a set of eight rules are obtained for 'ElongationFis', seven for 'dElongationFis' and ten for 'd0ElongationFis'. These rules are of the following types: if (in1 is in1cluster'*k*') and (in2 is in 2 cluster'*k*') then (out1 is out1cluster'*k*'). The number of rules is generated automatically by the 'genfis2' Matlab function starting from the experimental data set.

For each of the two FIS's inputs, 8, 7 and 10 Gaussian-type membership functions (*mf*) were generated, respectively; within this set of rules, these *mf* are denoted by: in'*j*' cluster'*k*'; *j* is the input number (*j* = 1 or 2 for all of the three FISs), and k is the number of membership functions (*k* = 1÷8 for 'ElongationFis', *k* = 1÷7 for 'dElongationFis', and *k* = 1÷10 for 'd0ElongationFis'). For example, the 'ElongationFis' FIS has the structure shown in Fig. 8, while its corresponding controller (Controller 1) has the structure shown in Fig. 9.

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 FISs. In Fig. 10, the deviation between the neuro-fuzzy models and the experimentally obtained data is shown for different training epochs, where the quality parameter is defined from the training algorithm. Figure 10 shows a rapid decrease in the deviation between the experimental data and the neuro-fuzzy model's data for the quality parameter within the training algorithm over the first 10^4 training epochs and for all three FISs. This decrease is followed by a slower decrease over subsequent epochs, and finally by constant values of approximately 0·012 for the first FIS, 0·01 for the second, and 0·0018 for the third FIS. From Fig. 10, it can also be observed that the FIS models may be trained on 250,000 epochs, since the deviations have approximately constant values after this number of training periods.

In order to visualise the FIS's features, the Matlab 'anfisedit' command is used[8] followed by the FISs' importation at the interface level. Visualisation of the following FIS characteristics is achieved via the interface: (a) for the 'ElongationFis': the FIS rules, the *mf*s for input 1 (force), the *mf*s for input 2 (current) and the surface — Fig. 11; (b) for the 'dElongationFis': the FIS rules, the *mf*s for the input 1 (force), the mf's for the input 2 (current) and the surface — Fig. 12; and (c) for the 'd0ElongationFis': the FIS rules, the mf's for input 1 (force), the *mf*s for input 2 (time) and the surface — Fig. 13. The left-column figures show the untrained FIS's, and the right-column figures show the trained FIS's results. The parameters of the input's membership functions for each of the three FIS's before and after training are shown in Table 1 and in Table 2, respectively. These parameters are one-half of the dispersion ($\sigma/2$) and the centre of the membership functions.

From Tables 1 and 2, the comparison of the FIS's characteristics and the membership functions' parameters before and after their training indicates that a membership function's redistribution in the working domain and a change in their shapes are affected by the dispersion modification. According to the parameter values given in Table 1, generating FIS's with the 'genfis2' function gives as 1st result the same dispersion choices for all of the membership functions which characterises an input. The working space separation for the respective input is shown as second result, so that its local linear model can be approximated through the use of the least squares method.

The evaluation of the FISs (untrained and trained) is performed using the 'evalfis' Matlab command; in Fig. 14, the experimental

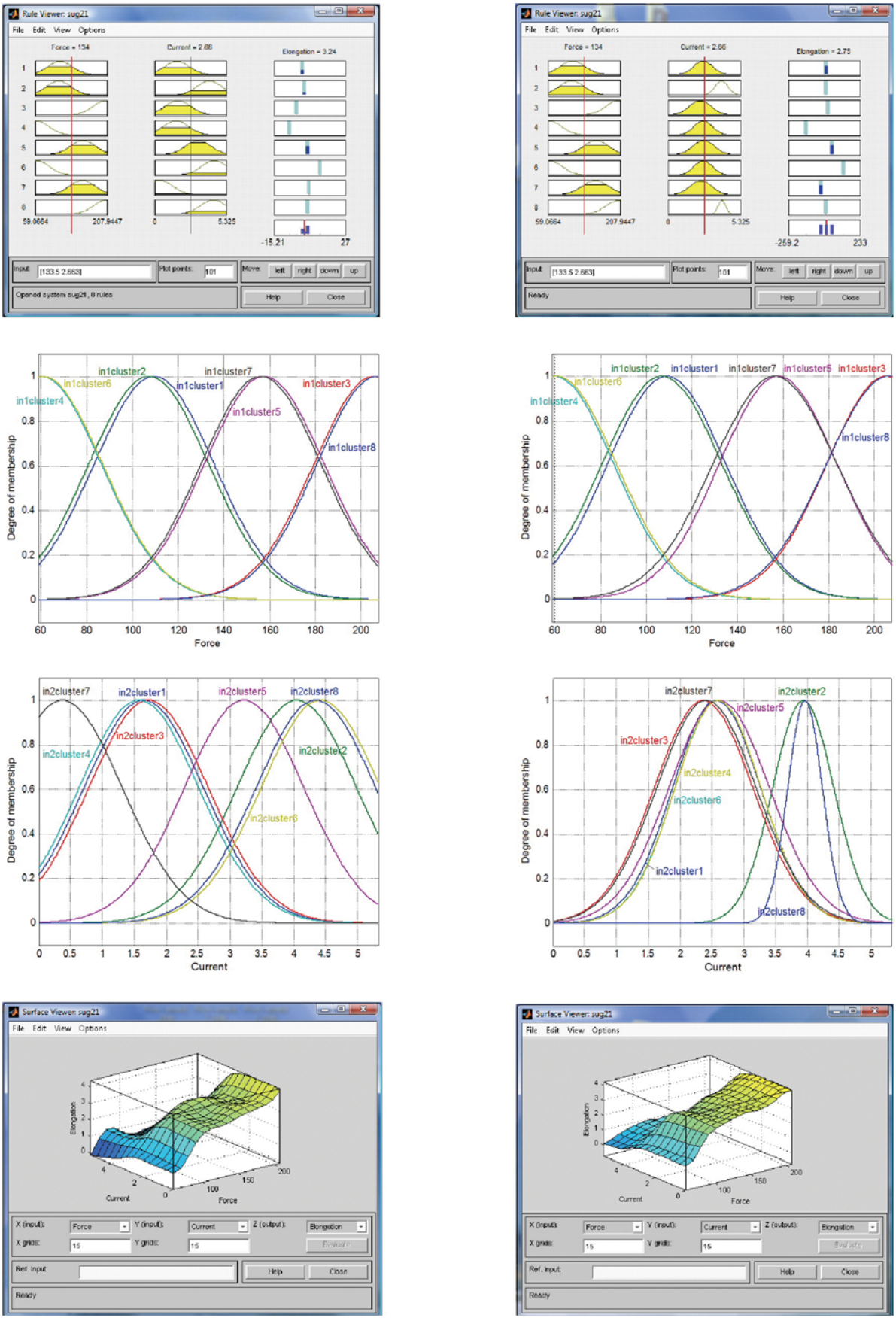

Figure 11. Properties of the 'ElongationFis'.

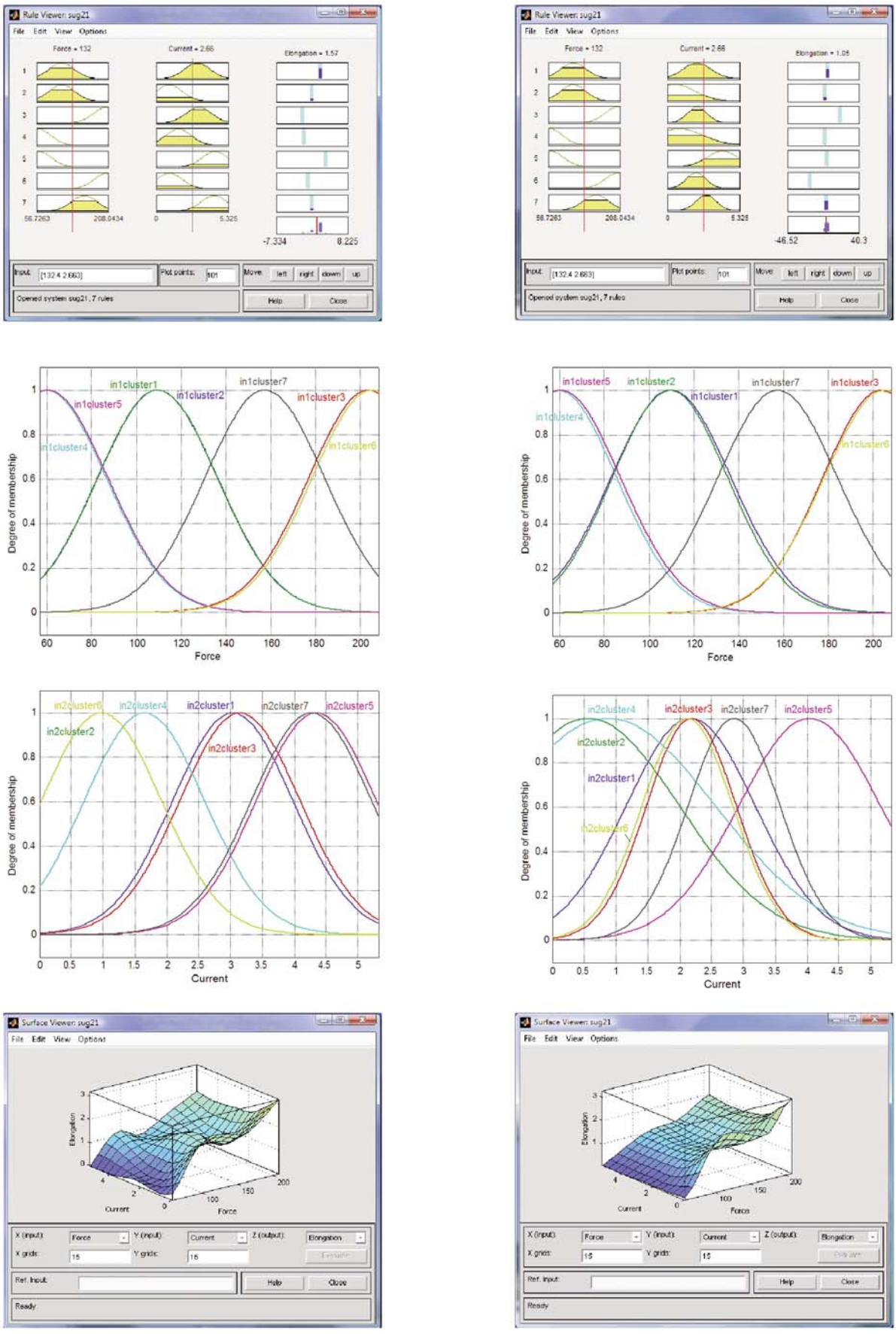

Figure 12. Properties of the '*d*ElongationFis'.

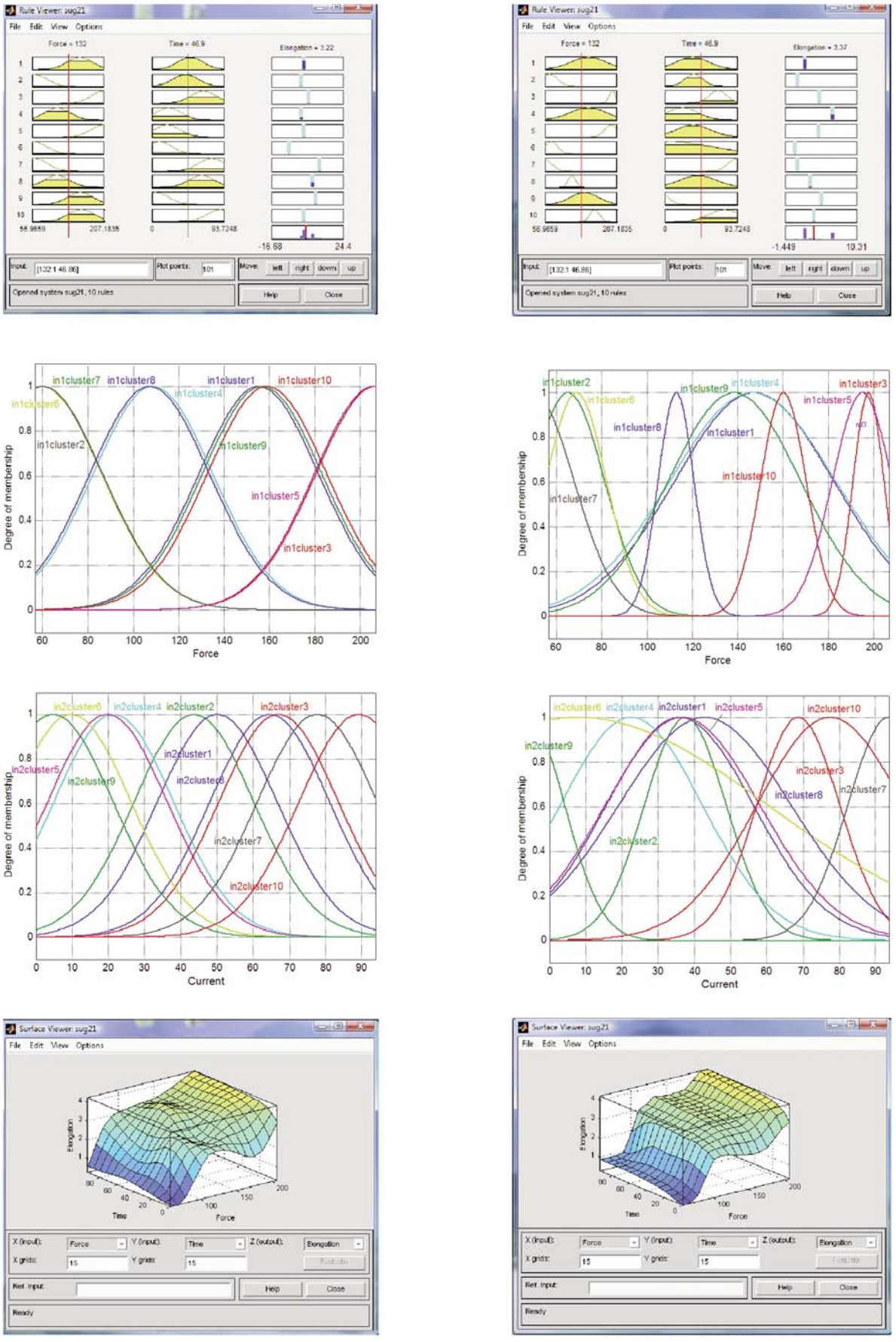

Figure 13. Properties of the '*d*0ElongationFis'.

Table 1
Parameters of the input's FIS membership functions before training

	ElongationFis				dElongationFis				dElongationFis			
	Force [N]		Current [A]		Force [N]		Current [A]		Force [N]		Time [s]	
	σ/2	Centre	σ/2	Centre	σ/2	Centre	σ/2	Centre	σ/2	Centre	σ/2	Centre
*mf*1	26.32	109.5	0·9413	1·65	26·75	109·5	0·9413	3·038	26·55	155·4	16·57	50·11
*mf*2	26·32	107·2	0·9413	4·05	26·75	109·4	0·9413	0·975	26·55	60·07	16·57	43·68
*mf*3	26.32	205.7	0.9413	1.725	26.75	203.7	0·9413	3·15	26·55	207	16·57	67·04
*mf*4	26·32	60·17	0·9413	1·575	26·75	59·86	0·9413	1·65	26·55	108·7	16·57	21·53
*mf*5	26·32	157·7	0·9413	3·225	26·75	60·55	0·9413	4·35	26·55	206·3	16·57	19·81
*mf*6	26.32	60.77	0.9413	4.425	26.75	205.2	0·9413	0·975	26·55	59·66	16·57	9·794
*mf*7	26·32	156·3	0·9413	0·375	26·75	157·3	0·9413	4·275	26·55	60·05	16·57	77·72
*mf*8	26·32	206·9	0·9413	4·35	-	-	-	-	26·55	107·2	16·57	64·59
*mf*9	-	-	-	-	-	-	-	-	26·55	156·9	16·57	4·837
*mf*10	-	-	-	-	-	-	-	-	26·55	159·1	16·57	89·18

Table 2
Parameters of the input's FIS membership functions after training

	ElongationFis				dElongationFis				d0ElongationFis			
	Force [N]		Current [A]		Force [N]		Current [A]		Force [N]		Time [s]	
	σ/2	Centre	σ/2	Centre	σ/2	Centre	σ/2	Centre	σ/2	Centre	σ/2	Centre
*mf*1	26·01	109·1	0·6943	2·577	26·88	109·8	1·013	2·175	35·16	147·3	20·34	35·98
*mf*2	26·41	107	0·4827	3·939	25·9	109·3	1·432	0·565	16·3	65	11·33	37·54
*mf*3	26·2	205	0·7779	2·371	26·31	203·6	0·7056	2·187	6·908	197·8	11·48	68·71
*mf*4	26·1	59·87	0·6785	2·597	26·32	59·35	1·685	0·8776	36·63	146·6	19·68	22·61
*mf*5	25·91	157·9	0·7936	2·618	26·78	60·56	1·113	4·044	13·25	195·1	21·38	36·73
*mf*6	26·33	60·83	0·6774	2·598	27·22	205·2	0·7129	2·123	13·38	68·84	52·35	8·052
*mf*7	27·29	156·5	0·7837	2·417	26·83	157·3	0·723	2·856	18·04	49·84	11·3	93·61
*mf*8	27·4	206·4	0·2733	3·968	-	-	-	-	18·04	49·84	23·84	42·91
*mf*9	-	-	-	-	-	-	-	-	29·28	138·6	10·18	–5·898
*mf*10	-	-	-	-	-	-	-	-	10·11	160·3	20·34	77·51

data and their corresponding FIS models are shown for untrained (left-column figures) and trained (right-column figures) FISs. Figure 14 reveals the same observations as those shown in Fig. 10. Note the overlapping of the FIS models with their experimental elongation data. This superposition is dependent on the training epochs' number, so that it is much better when the training epochs' number is higher. In other words, an improved real model approximation can be achieved with the neuro-fuzzy methods when a higher experimental data number is used.

Representations of the elongations (both those obtained experimentally and from using the three FIS models) expressed as functions of the electrical current for the first two FISs, and as a function of time for the third, are shown in Fig. 15. The curves for all four SMA load cases are represented, before and after the FIS's training (left- and right-column figures). One can easily observe that with training, the FIS's data model fits the experimental data very well, and that the SMA has different thermal constants, depending on the force values.

The mean absolute relative errors between the experimental data and the FIS models for untrained and for trained FISs are given in Table 3.

Starting from the three obtained FISs, three controllers are generated: Controller 1 ('ElongationFis'), Controller 2 ('dElongationFis') and Controller 3 ('d0ElongationFis'). The integration of these three controllers is realised by using the logical scheme shown in Fig. 7; the Matlab/Simulink model is obtained as shown in Fig. 16.

In the Matlab/Simulink model, the second input of Controller 3 (Time) is generated by using an integrator, and starts at the time when Controller 3 is used. Since it is possible that the simulation sample time may be different than the sample time used in the experimental data acquisition process, the 'Gain' block giving their ratio was used; 'Te' is the sample time in the experimental data and "T" is the simulation sample time. The 'C' constant represents the maximum time considered for the actuator to recover its initial temperature when the current becomes null (approximately 24oC).

Evaluating the integrated controller model (see Fig. 16) for all four of the experimental data cases produces the results shown in Fig. 17. These results are represented in the form of elongations versus temperature using the experimental data and the neuro-fuzzy controller model for the SMA hysteresis. The curves show that there is a good overlap between the neuro-fuzzy integrated controller outputs and the experimental data.

The same conclusions are obtained from the 3D characteristics of the experimental data and for the neuro-fuzzy modelled data in terms of temperature, elongation, and force, depicted in Fig. 18.

The mean absolute relative errors of the neuro-fuzzy controller versus the experimental data have the following values: 4·15% and 0·64% for a neuro-fuzzy controller with untrained and trained FISs, respectively.

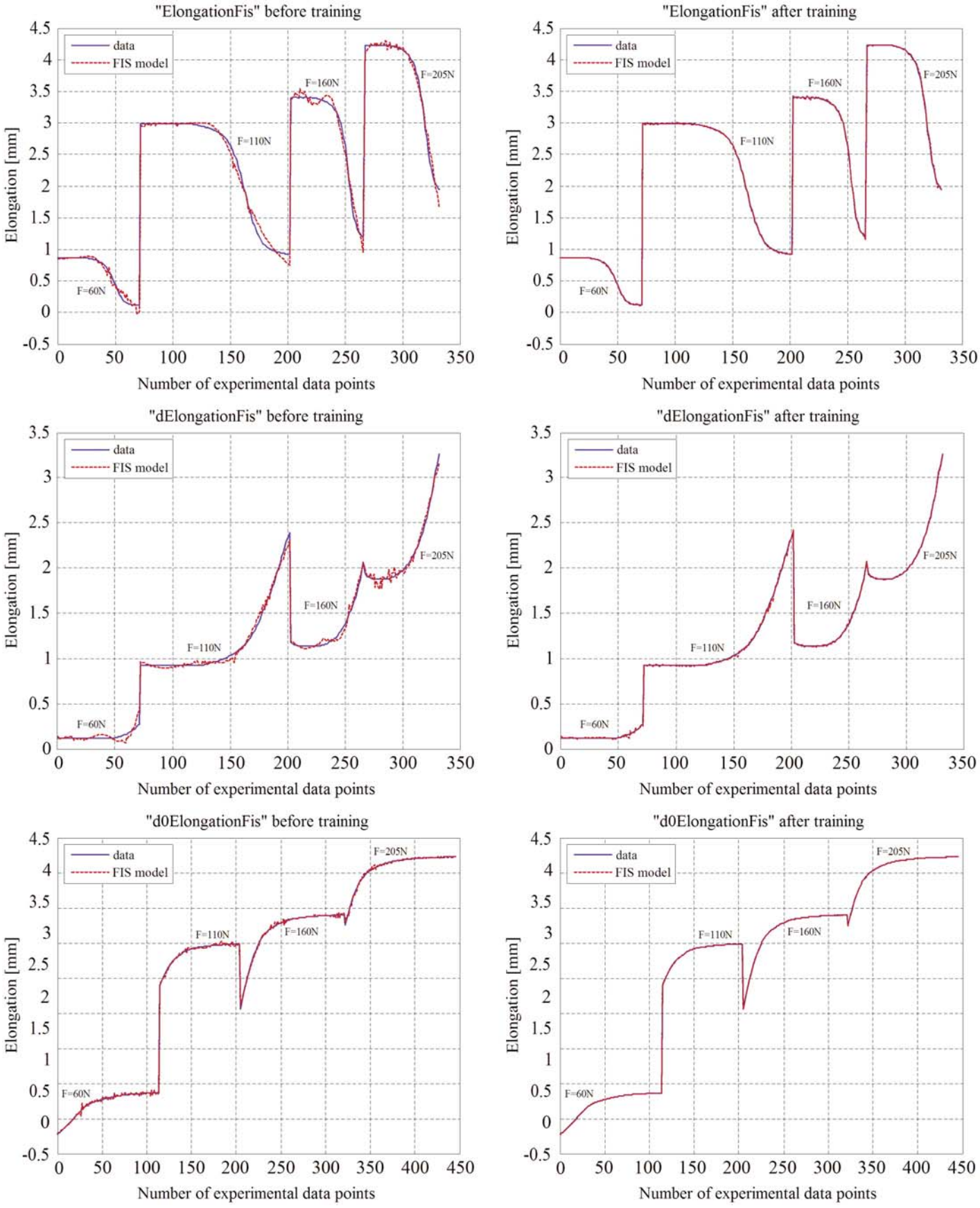

Figure 14. FIS evaluation as a function of the experimental data points number.

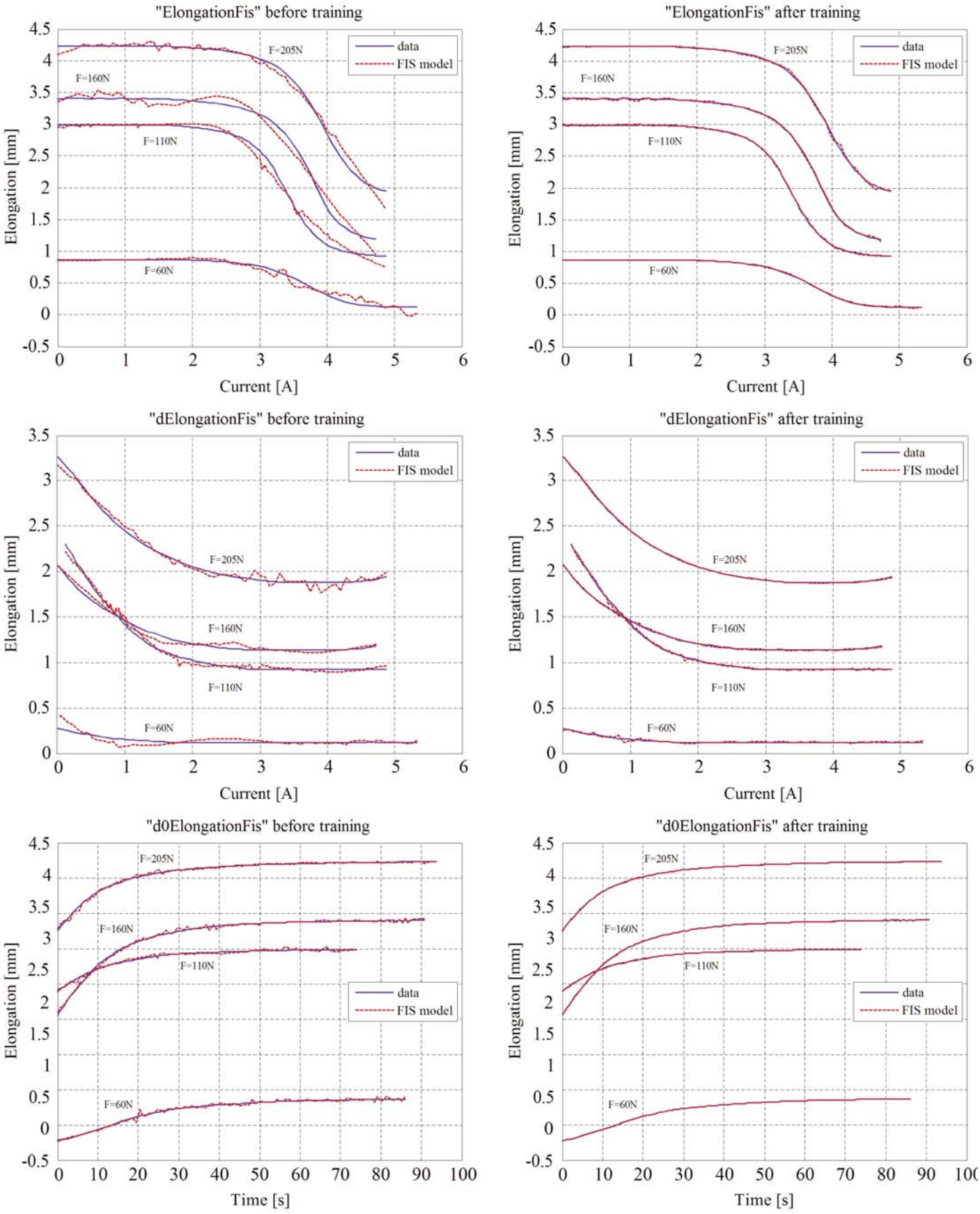

Figure 15. FIS's evaluation as functions of current or time.

Table 3
The mean absolute relative errors between experimental data and FIS's models

FISs	ElongationFis	dElongationFis	d0ElongationFis
Untrained FIS's	5·751968437978%	5·905991709200%	0·817600754857%
Trained FIS's	0·509291574547%	1·345191820144%	6·730977419645e–002%

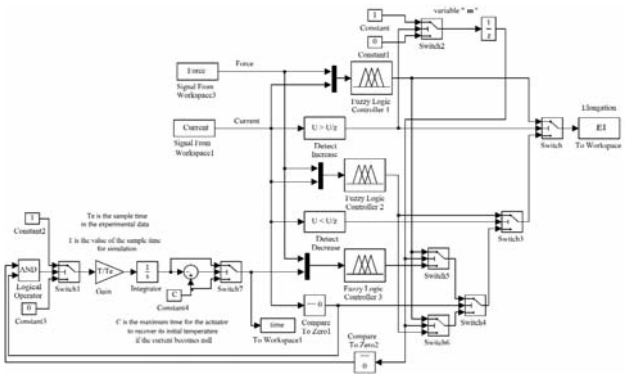

Figure 16. The integration model scheme in Matlab/Simulink.

5.0 CONCLUSIONS

A neuro-fuzzy controller for smart material actuator (SMA) hysteresis modelling was obtained. The direct controller application is a morphing wing system. This controller connects the forces and the electrical currents applied for different time periods to a smart material actuator in the morphing wing system. The controller was built by integrating, using Matlab/Simulink, three independent neuro-fuzzy controllers designed for the increase and the decrease of electrical current and for null electrical current in the actuator cooling phase. During the design of the integrated controller, experimental phase data were used for four smart material actuator load cases. The three fuzzy inference systems (FISs) associated with the independent controllers were trained for 250,000 epochs. The resulting mean absolute relative errors of the neuro-fuzzy controller versus the experimental data are: 4·15% for the neuro-fuzzy controller with untrained FISs and 0·64% for the neuro-fuzzy controller with trained FISs.

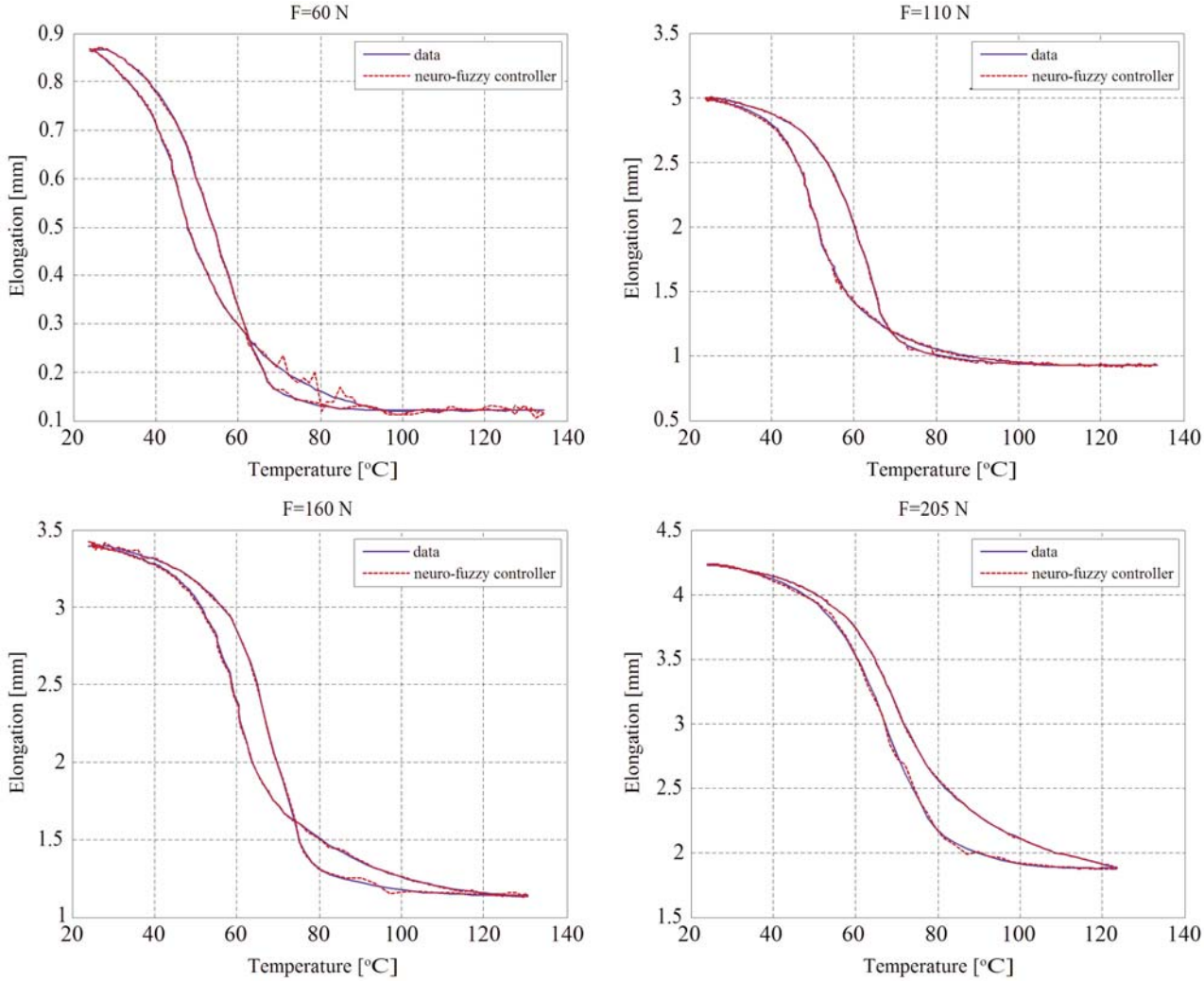

Figure 17. The integrated neuro-fuzzy controller evaluation versus experimental data.

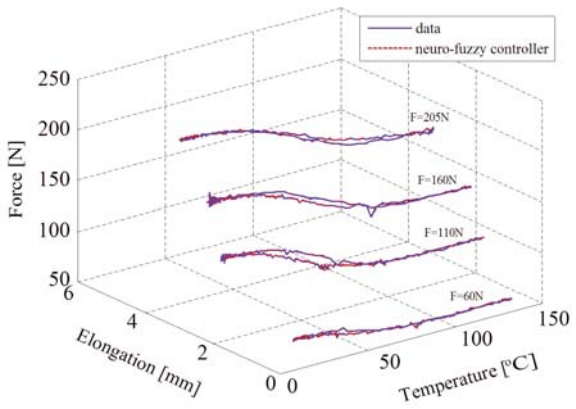

Figure 18. 3D evaluation of the integrated neuro-fuzzy controller versus experimental data.

A main advantage of this new model is its rapid generation due to the 'genfis2' and 'ANFIS' functions implemented already in Matlab. The user then only needs to assess the three FIS's training performances using the 'anfisedit' interface generated with Matlab.

ACKNOWLEDGMENTS

We would like to thank for the received funds which make possible this research to CRIAQ (Consortium of Research in the Aerospatial Industry in Quebec), Thales Avionics, Bombardier Aerospace and the NSERC (National Sciences and Engineering Research Council). Many thanks are dues to Professors Vladimir Brailovski and Patrick Terriault who gave us the opportunity to use their SMA bench test equipment for the realisation of this new controller method. We mainly thank to George Henri Simon for initiating the CRIAQ 7.1 project as well as to Philippe Molaret from Thales Avionics and to Eric Laurendeau from Bombardier Aeronautics for their collaboration on this work.

REFERENCES

1. RODRIGUEZ, A.R. Morphing aircraft technology survey, 2007, Paper AIAA-2007-1258, 45th AIAA Aerospace Sciences Meeting and Exhibition, 8-11 January 2007, Reno, Nevada.
2. MOORHOUSE, D., SANDERS, B., VON SPAKOVSKY, M. and BUTT, J. Benefits and design challenges of adaptive structures for morphing aircraft, *Aeronaut J*, 2006, **110**, (1105), pp 157-162.
3. POPOV, A.V., BOTEZ, R. and LABIB, M. Transition point detection from the surface pressure distribution for controller design. *J Aircr*, 2008, **45**, (1), pp 23-28.
4. SONG, G., CHAUDHRY, V. and BATUR, C. A neural network inverse model for a shape memory alloy wire actuator. *J Intelligent material Systems and Structures*, 2003, **14**, (6), pp 371-377.
5. LEE, H.J., LEE, J.J., KWON, D.S. and YOON, Y.S. Neural network based control of SMA actuator for the active catheter, *J HWRS-ERC*, 2001, pp 1-6.
6. SIVANANDAM, S.N., SUMATHI, S. and DEEPA, S.N. *Introduction to Fuzzy Logic using MATLAB*, 2007, Springer, Berlin Heidelberg.
7. KOSKO, B. *Neural Networks and Fuzzy Systems — A Dynamical Systems Approach to Machine Intelligence*, 1992, Prentice Hall, New Jersey, USA.
8. Matlab fuzzy logic and neural network toolboxes — Help.
9. MAHFOUD, M., LINKENS, D.A. and KANDIAH, S. Fuzzy Takagi-Sugeno Kang model predictive control for process engineering, 1999, 4 pp, IEE, London, UK.
10. KUNG, C.C. and SU, J.Y. Affine Takagi-Sugeno fuzzy modeling algorithm by fuzzy c-regression models clustering with a novel cluster validity criterion, *IET Control Theory and Applications*, 2007, **1**, (5), pp 1255-1265.