Sensors and Actuators A 331 (2021) 112835

Contents lists available at ScienceDirect

Sensors and Actuators A: Physical

journal homepage: www.elsevier.com/locate/sna

Numerical optimization based control design for a ferromagnetic shape memory alloy actuator

J. Jugo, J. Feuchtwanger, J. Corres

Electricity and Electronics Department, University of the Basque Country, UPV/EHU, Campus Leioa, 48940 Leioa (Bizkaia), Spain

ARTICLE INFO

Article history:
Received 11 January 2021
Received in revised form 10 May 2021
Accepted 12 May 2021
Available online 21 May 2021

Keywords:
FSMA
Actuator
Energy-saving control
Optimized tuning
Machine learning based modeling

ABSTRACT

Ferromagnetic shape memory alloys (FSMA) based actuators are able to get very good precision, allowing, indeed, under nanometer positioning applications. To get those precisions, a good controller is needed, and different strategies have been developed. Experimental tests with a FSMA actuator designed by the research group have proven that the use of a controller operating in a "set-and-forget"mode and following an event-based control scheme allows for a position with a given precision and transient behavior to be achieved, with a reduced number of control actions and therefore a reduced energy consumption. However, the tuning of all control parameter can be hard, especially when taking into account the nonlinear characteristics of the actuator. It can be remarked that the NiMnGa alloy used in the actuator is highly hysteric, and that the control action is pulsed. In this work, a numerical optimization based design methodology is proposed, making easier the control design procedure. The tuning procedure use a model, that for the particular actuator considered in this work, is obtained using machine learning tools, in particular the Tensorflow/Keras framework. An application example shows the good results obtained, including experimental result.

1. Introduction

The characteristics of ferromagnetic shape memory alloys (FSMAs), allowing large strains in response to magnetic fields, [1–5], have been exploited last years for the design of actuators that can achieve nanometric resolutions, using suitable control strategies, [6], and different actuator prototypes can be found in the literature [7–12].

Different strategies have been proposed for the control of the strain output obtained with those materials, with accounting for the hysteresis being fundamental. The positioning control for a FSMA actuator have been considered using linear feedback controllers and adaptive controllers, taking into account the hysteresis [13,14]. In [15,16], the use of observer-based inverse hysteresis strategy is studied. The use of a simple proportional-integral-derivative (PID) control strategy applied to an FSMA actuator allows accuracies from 5 to 20 nm, [6,8,17,18].

On the other hand, control strategies for this class of actuators reducing the energy consumption have been also studied. Hubert et al. [19] discuss the design of efficient FSMA based actuators based in a "Push-Pull" configuration. In [20], a FSMA based "push-push" actuator shows a reduction of energy losses of up to 60 %. More recently, a "set-and-forget" submicrometric actuation is proposed in [21,22] obtaining low power consumption and the combination of this pulsed-mode actuator with different event-based control schemes allows an efficiency increment, reducing the number of control actions, without a significant performance reduction, [23].

Those works show that the combination of FSMA actuators and advanced control techniques lead to large strains in responses with high precision. However, the tuning of the controllers to obtain this advantage can be difficult when taking into account the complexity of the overall actuator dynamics. For this reason, the use of numerical optimization techniques can be a valuable help. For instance, a neural network self-tuning control has been proposed recently for a piezoelectric actuator, [24]

In this paper, a numerical optimization based control design strategy is proposed and applied to the FSMA actuator presented in [23], where the control scheme is based on the event-based control strategy. The numerical procedure uses a Montecarlo technique, specifically a random walk algorithm, for the controller's parameter tuning. This tuning process includes a model of the overall actuator, which is calculated using machine learning techniques, in particular, based on the Tensorflow/Keras open source platform, [25,26]. The model considers the overall behavior of the actuator, including the effect of the power system which leads to a non saturation effect

E-mail address: josu.jugo@ehu.eus (J. Jugo).

https://doi.org/10.1016/j.sna.2021.112835

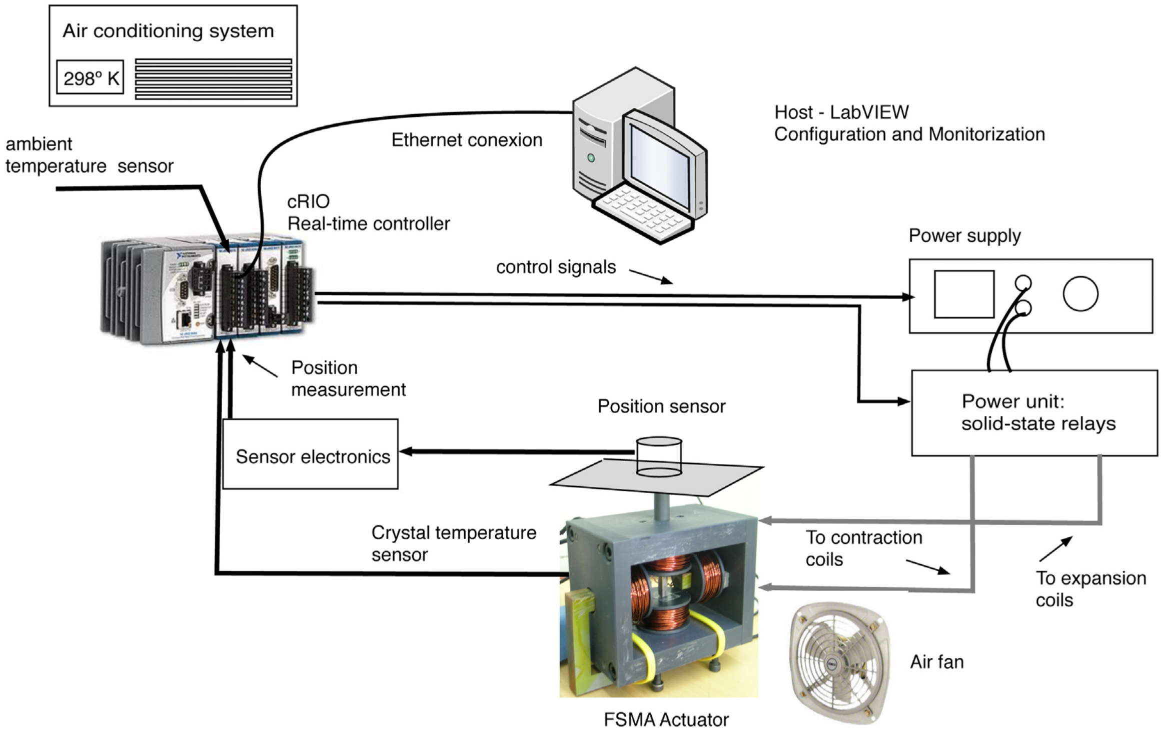


Fig. 1. Scheme of the experimental setup used for actuator testing, [31].

in the characteristic hysteresis of FSMA based actuators discussed in this work. This characteristics of the actuator, makes difficult the use of other modeling techniques proposed in the literature for this kind of materials as the Preisach or the Prandtl–Ishlinskii model, among others, [27,28]. The use of neural networks for the hysteresis modeling applied to Magnetic Shape Memory Alloy actuator has been also proposed in [29] and [30], with a Takagi–Sugeno fuzzy neural network based model technique and a NARMAX model based on diagonal recurrent neural network, respectively, obtaining very good results.

This work is organized as follows: in the next section, the actuator, its principal characteristics and the control are briefly described, emphasizing the modeling and control design difficulties. In the second section, the numerical optimization based control design procedure is proposed, describing the main steps and the proposed random walk algorithm. In the third section, given the need for a model of the actuator, a machine learning model is described. In the fourth section, the methodology proposed is applied to the FSMA actuator and the experimental results are presented. Finally, conclusions make up the last section.

2. Actuator description

The actuator used for the experiments in this work has been used in previous publications [23,31], and a more detailed description can be found in them. A description of the experimental setup can be observed in Fig. 1. It basically consist of a single crystal of Ni–Mn–Ga, with a 5 mm sided square base, and 23 mm long. The actuation field is generated by a set of orthogonal coil pairs through which a capacitor bank is discharged. The coil pair coaxial with the length causes the crystal to contract, and the other set causes the crystal to elongate, leading to a non symmetric deformation curve for contraction and expansion, [23].

The actuator includes a 30 mm in diameter fan to blow air directly into the FSMA crystal to minimize the temperature effect in the system behavior, [31].

This actuator exhibits a large hysteresis when the direction of actuation is changed, and the actuator's response, the deformation/voltage relation, is not symmetric, meaning that the voltages needed to obtain a given deformation in contraction are lower than those needed for the absolute deformation in expansion.

The controller used in this work, shown in Fig. 2 and described in detail in [23,31], was designed taking into account the particular characteristics both of the actuator and the FSMA. One important characteristic of this actuator is that it works in the "set-and-forget" operation mode, because once the desired stroke is achieved the actuator maintains it without the need for further energy supply and, then, the control actions are stopped when the desired position is reached within the required accuracy and is only activated when a reference change is introduced or appears a temperature effect, [31]. It should be noted that the actuator described in this paper is not intended for fast actuation like other commercially available FSMA actuator that can work at frequencies higher than 60 Hz, but rather for applications where low power consumption is essential and only small, sporadic position corrections are required.

The control scheme is based in an event-based control (EBC) algorithm, schematized in Fig. 2, for reducing the control actions and the time constants of the circuit were limited to set a conservative interval minimum of 500 ms between actuations, [22]. This actuation time limits the actuator's time response.

The event-based sampling law, which fires the simultaneous sampling of the system output and the control signals [32–34], is defined as a function of the change in the computed control signal $e_u = u_k - u_n$, that is, the difference between the last applied control action (u_k) and the current calculated value (u_n) according to the sampling law formulated as

$$u_{k+1} = \begin{cases} |u_n| & \textit{if } (u_n > 0)\,\&\left(|e_u| > \sigma_{u,C}\,|u_n| + \varepsilon_{u,C}\right) \\ -|u_n| & \textit{if } (u_n \leq 0)\,\&\left(|e_u| > \sigma_{u,E}\,|u_n| + \varepsilon_{u,E}\right) \\ u_k & \textit{otherwise : eventnottriggered} \end{cases} \tag{1}$$

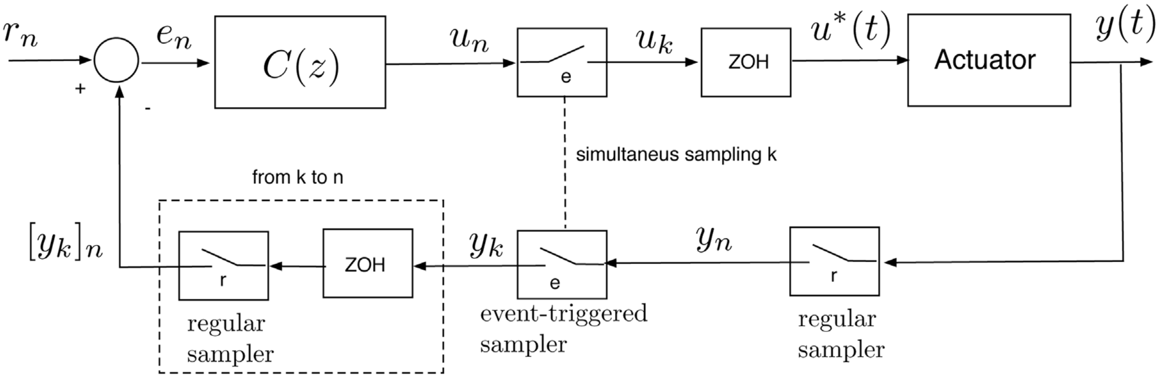


Fig. 2. Scheme of the implemented synchronous discrete event-triggered control system.

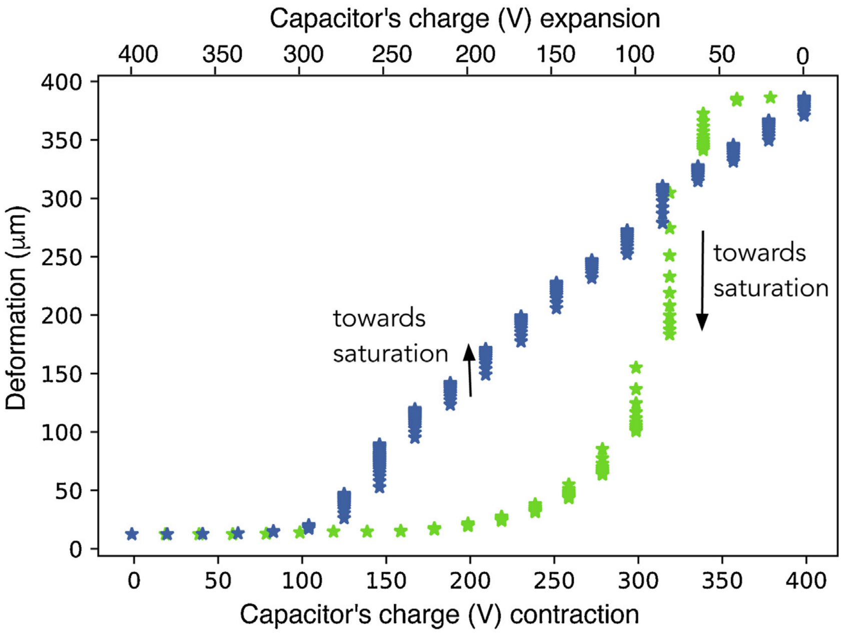


Fig. 3. Voltage dependency of the actuator deformation in contraction (blue line) and expansion mode (green line). Increasing deformations are shown for repeating voltage values until saturation.

where $\varepsilon_{u,C/E}$ and $\sigma_{u,C/E}$ are tunable parameters, chosen by the designer. C/E subscripts make reference to the contraction or expansion mode, respectively. In this form, the sampling law takes into account the asymmetric nature of the FSMA actuator. The use of different values $\varepsilon_{u,C}$, $\varepsilon_{u,E}$, $\sigma_{u,C}$ and $\sigma_{u,E}$ is a refinement of the control scheme used in [23,31], which leads to a more complex control design.

The controller implemented $C(z)$ is defined by a non symmetric Proportional-Integral (PI) algorithm:

$$u_n = \begin{cases} K_{i,C}e_{n-1} + K_{p,C}e_n & \text{if } (u_n > 0) \\ K_{i,E}e_{n-1} + K_{p,E}e_n & \text{if } (u_n \leq 0) \end{cases} \quad (2)$$

Again, the control algorithm includes different control gains for contraction and expansion in order to take into account and compensate the inherent non symmetry and the slope differences of the actuator, increasing the control design complexity.

In previous works [23,31], the value of the controller gains have been obtained using model based simulations, using a simple actuator model, and tuned to the final values experimentally. But, in general, the design process is time consuming and can be difficult to reach a good design, depending on the particular working zone and particular FSMA crystal used, since each crystal presents different behavior, which in addition changes with time.

Finally, although the resolution achievable with the actuator is higher, in the following experiments the specified allowable error has been set to 0.5 µm. Then, in the “set-and-forget” mode, when the error is below the threshold error $e_{\min} = 0.5$ µm, the control action is stopped.

2.1. Non saturation effect

Usually, it is assumed that, once a given deformation is obtained in a particular direction, FSMA based actuators need a higher amplitude magnetic field to increase the deformation in the same direction. Excitations lower than the previous one do not change the deformation in the actuator.

This was the assumption made for this actuator in previous works, [22,23,31]. However, depending on the particular crystal used, this may not be valid. Fig. 3 shows the actuator deformation as a function of the applied voltage. As can be observed, repeating a voltage value in expansion or contraction mode the deformation increases until a saturation value is reached. An explanation of this behavior is that, for this actuator, the maximum magnetic field for a particular voltage is not constant over time but pulsed, and, then, the magnetic field is not maintained the necessary time to position all the material twins with one single pulse. Other actuator configurations use a constant magnetic field, and don't present this behavior, [17].

This actuator characteristics can be very relevant and must be taken into account in a control design process for getting accurate results. Consequently, an actuator model should consider this behavior, complicating the use of Preisach or Prandtl–Ishlinskii like modeling techniques. To the best of the authors' knowledge, there is no hysteresis modeling technique considering explicitly the non-saturation effect described in this section and, then, a rework of other existing hysteresis modeling techniques would be necessary to be applied accurately in the case of the actuator used in this work.

It is must be remarked that this non-saturation effect is derived from the power system and the generation of the voltage pulse. In order to reduce this effect, the pulse duration must be controlled, requiring the redesign of the power system. In this work, the actuator's controller should improve the overall system behavior, effected by the non-saturation.

3. Numerical optimization based design process

The basic procedure of the proposed optimization process of the controller's parameters can be summarized as is shown in Fig. 4. There are three main phases: 1. obtaining of an actuator model, 2. an optimization process of the controller's parameters based on simulations using the model and 3. a final tuning using real actuator in the optimization process.

A good model is fundamental for obtaining an applicable control parameter set from the optimization process. This is a crucial step in the overall process and it is discussed in the next section.

The numerical optimization algorithm used is also very important for achieving the convergence of the control parameters towards a good actuator performance. The list of possible techniques is very long, including gradient descent based algorithms [35], genetic algorithms [36] or Monte Carlo techniques [37]. The actuator behavior, including the EBC controller, has a complex and nonlinear dynamics which makes impractical the use of gradient descent techniques (various attempts have lead to not satisfactory results). The use of random optimization algorithms seems more practical in this case. For this reason, a Monte Carlo random walk technique has been defined, [38]. This algorithm, detailed below, has been compared with a real-coded BLX-α crossover genetic algorithm [39], obtaining better results with lower function cost needing a lower number of iterations (see Section 5).

The proposed optimization algorithm can be summarized in the next steps:

1 In the first step, the actuator model and EBC control algorithm are set with initial control parameters values. The set of parameters $P = \{P_0, P_1, \ldots\}$ are the gains of the event-based sampling law (1) and the PI controller (2). Initially, they are manually tuned and denoted $P[0]$.

2 A cost function J must be defined. This cost function determines the control objectives and is an important choice of the designer. In this case, the cost function is defined as follows:

$$J = K_e|r-y| + K_u|u| + K_n n \tag{3}$$

where $r-y$ is the position error, u is the control signal value (in this case, a voltage) and, finally, n is the number of control actions. This function cost must be minimized constrained to the dynamics of the actuator and to a set of valid values for the parameters to tune $P_i \in \Lambda_i$. The procedure is very flexible for including new constrains.

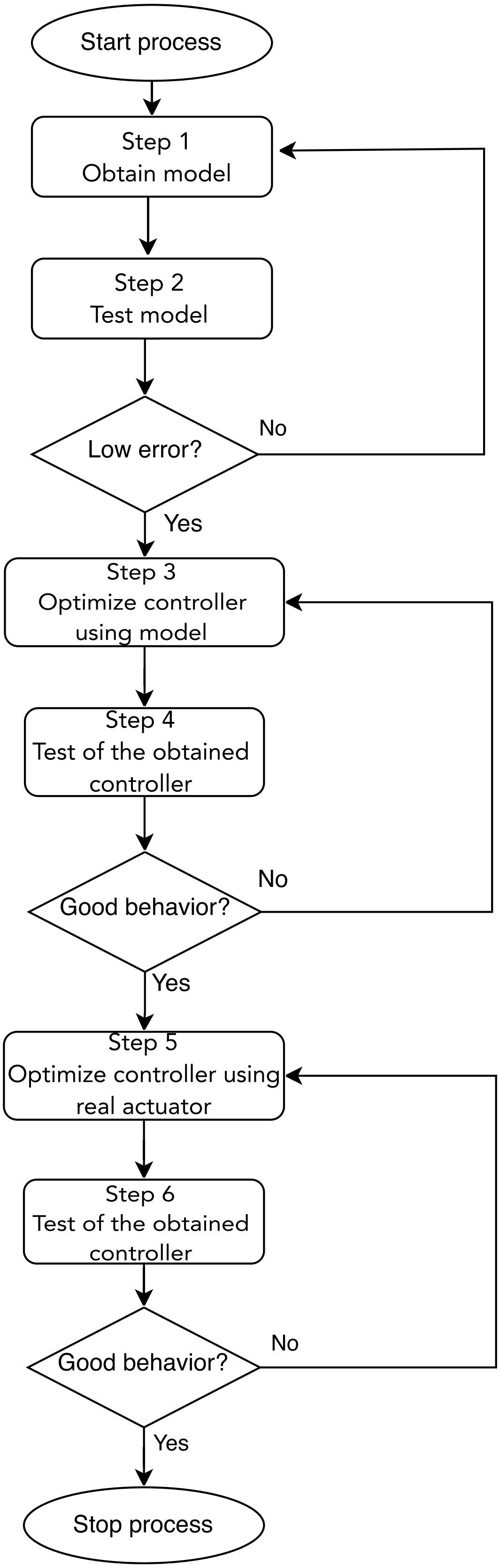


Fig. 4. Main steps to follow in the optimization process.

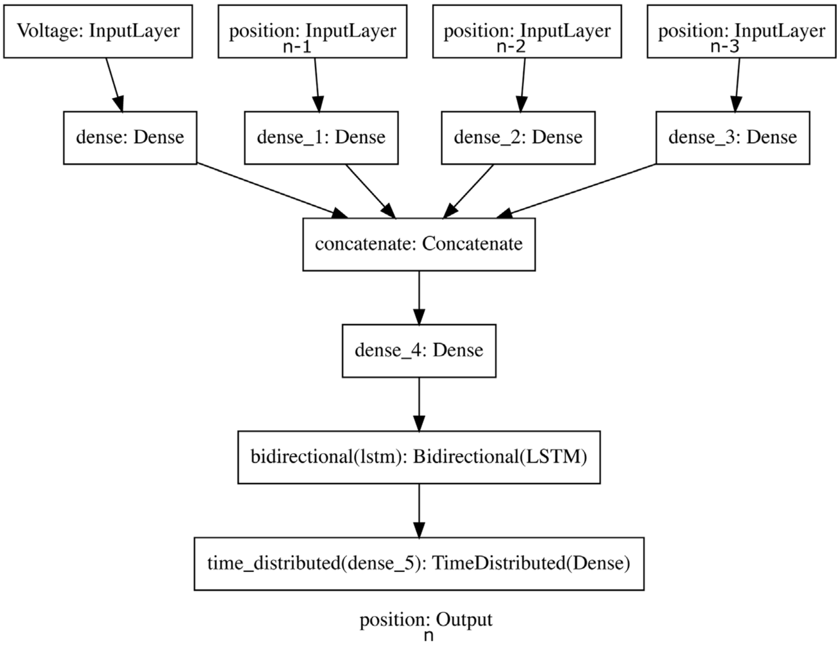


Fig. 5. Keras model structure used for modeling the actuator. Three inputs depend on the previous outputs of the model.

3 A control objective for the system is set around an operation point, using changing position references (for instance, a train of pulses of different values representing desired actuator positions as reference). Once the simulation setup is defined and executed, the initial cost function $J[0]$ with the manually tuned control parameters is computed as first reference.

4 Next, the iterative process starts. In each iteration, new control parameters P_i^r are selected through a random search:

$$P_i^r = P_i[n-1] + \alpha_i, P_i[n-1] \bigwedge P_i^r \in \Lambda_i \tag{4}$$

being all the parameters within the set of constraints Λ_i, which is defined with the knowledge of the designer about the actuator behavior. In each iteration, a random value α_i is added to the actual optimal control parameters $P_i[n-1]$ (for $i = 1 \ldots n_p$, being n_p the number of control parameters) to obtain the new ones to use in the next simulation step, as is defined in (4). So, the randomly chosen parameters in each optimization loop are around the last optimized control solution.

5 One the simulation step is done and the corresponding cost function J_r computed, if the obtained cost function is lower that the previous optimal one $J_r < J[n-1]$, the control parameters P_i^r as well as the current value of the cost function are saved as reference:

$$P_i[n] = P_i^r$$
$$J[n] = J_r$$

Otherwise, the reference values are unchanged.

6 The process is repeated going back to the previous step 4 where the control parameters are selected through the random search around the last saved control and sampling law parameters. The optimization process can be concluded when the cost function is not more minimized after a number of iterations. The number of simulations to carry out can be very high (thousands of iterations can be necessary), but this stage can be finished in several hours. In any case, it must remarked that the optimization process is suboptimal.

The definition of the parameters α_i is done to assure that the new simulation parameters P_i^r are in a bounded region around the last reference values $P_i[n-1]$.

The simulation based optimization process continues until the cost function reaches a (local) minimum. Once this cost function is stalled in a minimum, the resulting control values are tested, obtaining the closed loop time response to observe if can be considered good enough. If the result is satisfactory, the final stage of the procedure is applied, repeating the previous steps but substituting the model based simulations with experimental tests. If the model is accurate, the control parameters obtained in the model based optimization phase are near enough from optimal parameters for the real actuator and this final stage could require only several tens of iterations for the final convergence.

For some applications, it can be considered that the proposed optimization procedure could be applied directly to the real plant, making not necessary the first simulations based phase. However, the large number of iterations normally required could be very time consuming and deteriorate the actuator, making the procedure, in consequence, impractical.

4. Actuator model

The most relevant characteristic to consider in the modeling process of the actuator is the hysteritic response of the NiMnGa alloy used as the active element of the actuator. Different modeling techniques are proposed in the literature to deal with this kind of materials as the Preisach model or the Prandtl–Ishlinskii model, among others, [27,28].

However, this actuator has a characteristic which is no usual in other works: its pulse nature which leads to the non saturation behavior described in Section 2.1. For this reason, the model considered in this work is based in machine learning techniques, for dealing with all the complex details at once. In particular, the Tensorflow/Keras library has been chosen to define the model, [25,26]. The structure of the model selected can be observed in Fig. 5. This model structure has been chosen taking into account that it must consider the actuator's memory effect, including the non-saturation described in Section 2.1. Several structures has been

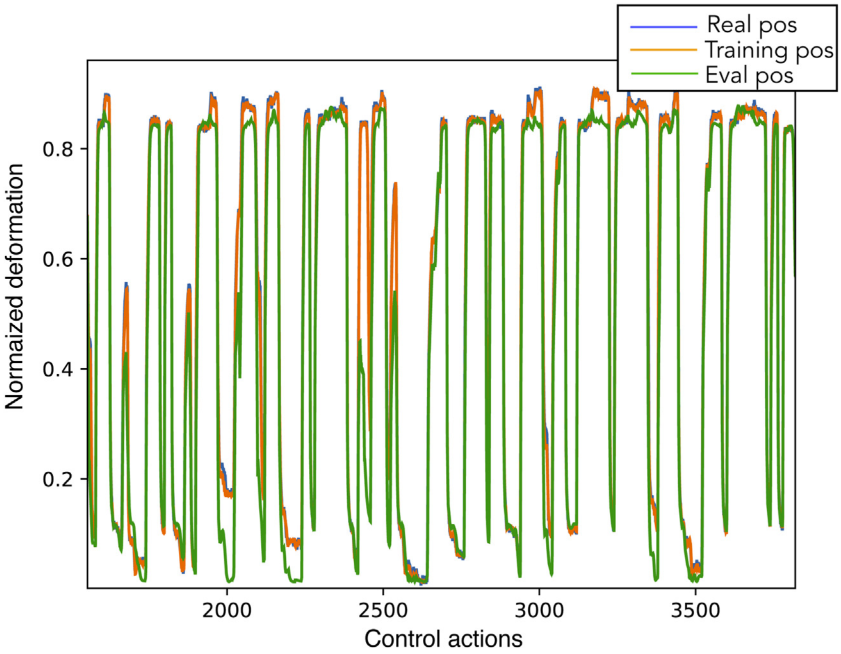


Fig. 6. Example of model evaluation showing normalized position using random input voltages: real actuator position (blue line), model output using real position in the inputs (orange line), model prediction (green line). The orange line shows the result using the training data and the green line the actual model prediction.

Table 1
Example of the result of the training phase, with a particular set of data. The error percentile is the average of all results for the training and prediction phases.

Samples	Epoch	Loss	Mae	% Error training	% Error prediction
14988	1000	$1.0829e-04$	0.0073	1.42%	5.13%

tested by using recursive layers as LSTM or GRU in order to have internal feedback, but the best model response comparing with the real actuator have been obtained feeding some previous outputs as model inputs, with an initial layer with parallel dense layers for each input and an intermediate LSTM layer for including some internal feedback. A final layer is used to generate the one dimensional output. The number of previous outputs to consider as inputs and the particular size of each layer has been selected by mean of a try and error process. So, the model has four inputs: the voltage and the three previous outputs of the model. Most model layers are dense type (dimension, 1000), but a bidirectional LSTM layer (dimension, 200) is included for adding some feedback connections. As stated before, the final time distributed dense layer gives the model output and, then, its dimension is 1. The model is completed with the RMSprop optimizer, the mean squared error loss function and, finally, the mean absolute error performance metric.

The most remarkable feature of the model is that it includes as input not only the voltage, the actual input of the actuator, but also three previous positions. This fact results in a more accurate model but with an additional complication. The data used for training the model are real positions obtained under experimental tests, but the inputs used in the prediction phase are model outputs. This means that the error in each prediction is feedback to the model, making the prediction error higher than expected from training phase. However, the resulting model with those inputs gives better predictions than other configurations.

The modeling process is the usual using Keras, [26]:

- First, the training phase uses data experimentally obtained, measuring the actuator position output for different random values that cover the entire input voltage range $V \in (-400, 400)$. Different training data are obtained in sets of more than 12000 point (V_i, P_i). The total number of datasets depends on the needed accuracy for the model. Part of those datasets are used in the training process.
- In the second step, the model is evaluated with the rest of datasets in order to validate the model.

The partial result of a validation process, comparing the real actuator position and the predicted one can be observed in Fig. 6. In this evaluation, a random set of 13,000 voltages $V \in (-400, 400)$ was used and the total prediction error was 5.4%. The figure shows the comparison of the measured values of the actuator deformation and the model outputs. The model outputs are obtained following two different procedures. The first ones are obtained applying the inputs as in the training process, that is, using the real actuator previous positions as model inputs. In the second case, the output values are obtained feeding the model inputs with the outputs obtained from the model, getting a more real evaluation of the model effectiveness. In Fig. 6, the deformations are normalized to the range [0,1], being 0 the minimum deformation and 1 the maximum deformation obtained for the particular actuators's crystal. This normalized deformation is useful for comparing the results when using different FSMA crystals in the actuator.

The Table 1 gives information about the result in the training and prediction phase. The mean error of the model prediction is around 5%, but in a particular case can be higher that the 10%. In any case, the model response can be considered sufficiently good to be used in the control design process.

5. Experimental results

In this section, the results applying the numerical optimization based control design process described in Section 3 using the modeling procedure described in Section 4 are shown.

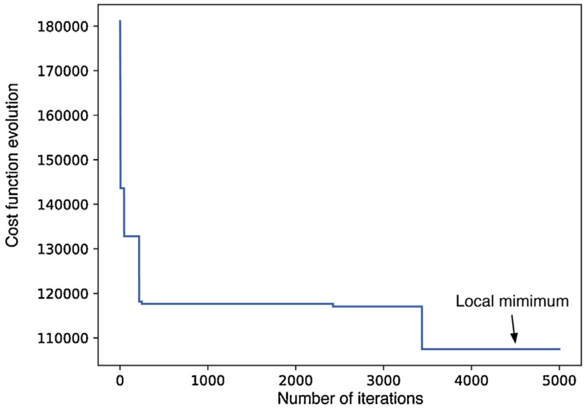


Fig. 7. Minimization of the cost function in a numerical optimization process, leading to a local minimum.

One important consideration is that the actuator crystal will change its behavior with time, resulting in a changing actuator's dynamics. This is due to the occurrence of micro fractures in the FSMA crystal structure, limiting the twins movement and, in consequence, the deformation range of the crystal. So, an actuator model can be inaccurate after a certain time of actuator use. For this reason, it can be convenient to rebuild the actuator model regularly, depending in the particular crystal response. In the case of the actuator considered in this work, a new model has been computed every 6 months.

The complete procedure followed, including the modeling phase and the numerical optimization steps, is the next:

1 Actuator's experimental data collection using random voltage inputs. The experimental data should cover all the working range of the actuator. Different data sample lengths have been tested and the data sample length 12,000-15,000 gives good balance between results and computational effort, for the problem discussed in this paper.
2 Keras framework based model training, using the experimental data and following the process of Section 4.
3 Model evaluation to assure that the model precision has an affordable error (5–15%).
4 Numerical optimization based control design using simulations, described in Section 3, around an operation point. A typical optimization process needs a couple of hours of simulation (around 5000 simulation iterations).
5 Experimental test using the obtained control parameters.
6 Final tuning, using the same procedure with the real actuator, using as the starting point the parameters obtained in step 5. This phase requires few experiments (around 20), which can be done in less than three hours.

Those steps can be automated, requiring for completing all the process a couple of days. The most time demanding step is first one, and, then, if a new optimization process for a different operating point (using the same model) is needed, the process is sensibly faster (around 3 h). Note that the optimization process is sub optimal and in different optimization processes the result obtained will be different, but qualitatively similar.

Following the proposed procedure, setting as position reference a pulse train going from 40 to 100 μm and the cost function gains $K_e = 20$, $K_u = 0.1$, $K_n = 2$, the obtained (sub)optimal control parameters were $K_{p,E} = 2.075$, $K_{i,E} = 22.165$, $K_{p,C} = 1.8$ and $K_{i,C} = 266.78$, and the sampling law parameters were $\varepsilon_{u,C} = 0.66$, $\varepsilon_{u,E} = 0.43$, with $\sigma_{u,C/E} = 0$. Fig. 7 shows the evolution of the function cost in this numerical optimization trial, which leads to a local minimum (function cost value 107486). In addition, Fig. 8 shows the evolution of the control parameters in the process.

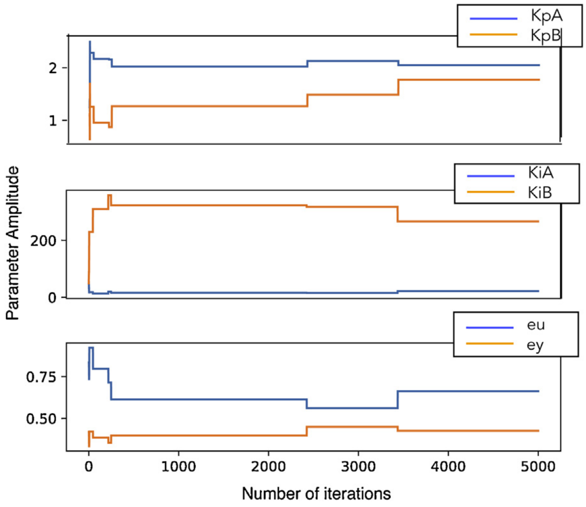


Fig. 8. Evolution of the control parameters in a numerical optimization process.

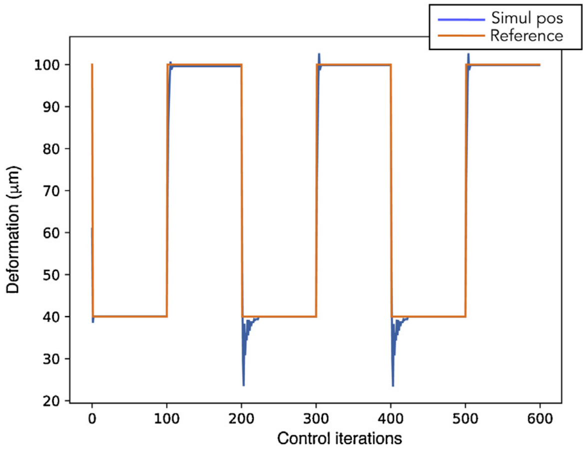


Fig. 9. Actuator position (deformation, blue line) following the desired reference (orange line) using the optimized control sampling law parameters obtained in the first optimization phase.

Fig. 9 shows the result of the first phase of the simulation process, with respect to the control iterations. The figure shows that the model in the contraction mode is smooth enough, but it is not the case of the expansion mode.

In order to value the algorithm performance, a BLX-α crossover genetic algorithm has been implemented to compare the results with the proposed optimization method. From Fig. 7, the local minimum obtained using the random walk method is 107,486 after a total number of 5000 iterations. On the other hand, the better results obtained using the BLX-α crossover genetic algorithm with a initial population size of 3000 genes has been 144,477, needing a total of 18,000 simulations. This one has been the better result of different trails, using different initial population sizes (1000, 2000, 3000, 5000) and different α values..

After this initial optimization process, the obtained parameters were applied as the starting point in the final optimization process using the real actuator. In this final step, the obtained (sub)optimal control parameters were $K_{p,E} = 0.6572$, $K_{i,E} = 197.71$, $K_{p,C} = 1.16435$ and $K_{i,C} = 351.167$, and the sampling law parameters were $\varepsilon_{u,E} = 0.49$, $\varepsilon_{u,C} = 0.8672$, with $\sigma_{u,C/E} = 0$. Figs. 10 and 11 show the resulting actuator behavior. As can be observed in Fig. 10, the actuator position follows the reference with a good transient, being the

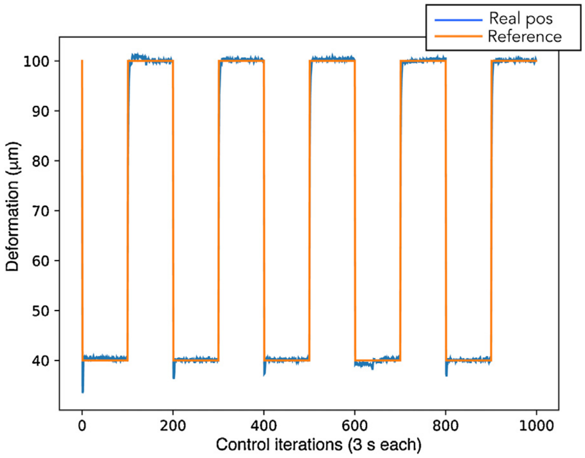


Fig. 10. Actuator position (deformation, blue line) following the desired reference (orange line) using the optimized control sampling law parameters.

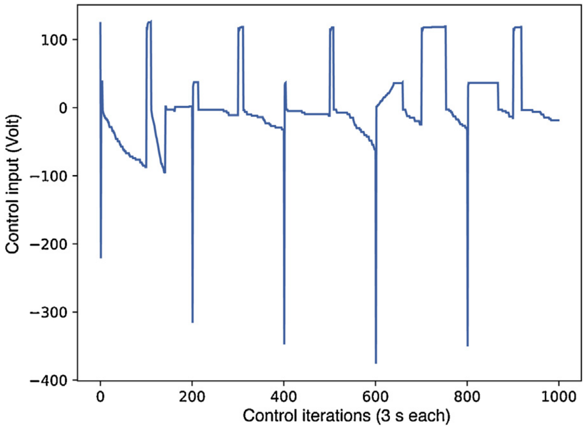


Fig. 11. Control input (voltage) calculated using the optimized control and sampling law parameters.

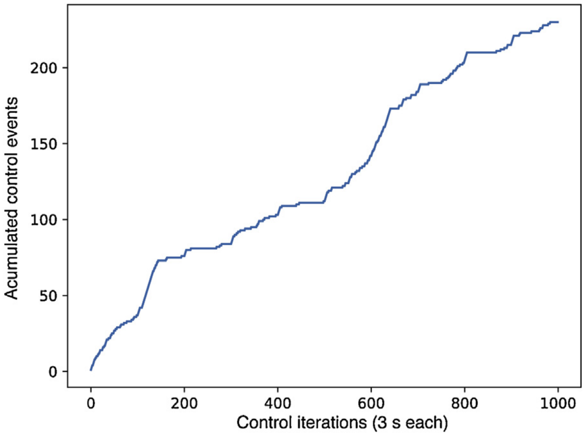


Fig. 12. Evolution of the accumulated events fired by the event sampling law with respect to the control iterations. The final relation is 230 events from 1000 total control iterations (a 77% reduction).

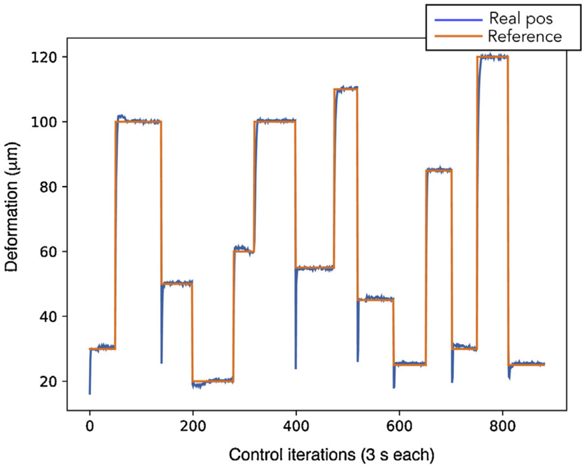


Fig. 13. Actuator position (deformation, blue line) following the desired reference (orange line) using the optimized control sampling law parameters, using random position references.

position error in the allowed range ($< 0.5\,\mu$m). Fig. 11 shows the generated control signal, that is, the voltage value to apply as the actuator input. As can be observed, the behavior of the real plant is more smooth that the model response, but, in any case, the optimization process using this model gives a valid optimization stating point.

Both Figs. 10 and 11 show the evolution of the actuator position with respect to the control iterations. In this work, the minimum time interval between control events is around 3 s, distributed between the time required by the power system to fire the control signal ($\sim$ 500 ms) and a sleep time introduced by the controller to facilitate the heat dissipation and to reduce the temperature effect over the FSMA crystal (2.5 s). However, the EBC controller used does not fire a control action each control iteration. In fact, the number of control actions is one of the elements considered in the cost function to minimize. Fig. 12 shows the evolution of the accumulation of fired control events over time with respect to the total control iterations. The final relation is 230 events from 1000 total control iterations (a 77% reduction), showing an important reduction of the needed energy and, leading to an improvement of the actuator life.

Additionally, Fig. 13 shows the actuator experimental deformation using random references, different from the values used in the optimization process, and with variable pulse lengths. The results show that the actuator behavior is still valid, but presenting initial overshot in the expansion mode, which are rapidly recovered. This result could be expected, since the actuator is highly nonlinear in this mode, making very exigent its control and this behavior is similar to the result obtained in the simulation based optimization phase.

The crystal used in this experiment was aged and, in consequence, the resulting dynamics was somewhat deteriorated, making the manual design process more challenging and time consuming. With the proposed methodology, the design presents a very good result in transient response, precision and number of control actions, following a highly automated procedure. The resulting behavior is very good, indeed comparing with the results presented in previous works applying manually tuned control algorithms, [23].

6. Conclusions

A FSMA actuator, as other actuators based on smart materials, could need an advanced controller to get all the advantages which can be obtained from the material's characteristics as a good positioning precision. The controller design can be a challenging task and the use of a numerical optimization procedure can be helpful.

This is the case of the event based controller implemented in this work for positioning control applications using a FSMA-based actuator. The parameter tuning of the controller is exigent for getting high positioning accuracy and, also, energy savings.

In this work, a random walk algorithm is proposed as the central element of the numerical optimization of the controller parameters, which can be separated in an initial simulation based stage, using an actuator model and a final experimental step, less numerical taxing.

The optimization process is based on an actuator model in a first phase, which should be sufficiently accurate for getting good results. In this case, the model is obtained using a machine learning technique, using Tensorflow/Keras, to overcome all the dynamic characteristics of the system. The hysteresis of the material and the non-saturation behavior due to the pulsed actuation difficulties the use of other possibilities as Preisach or Prandtl–Ishlinskii like modeling.

The experimental results show that the procedure is valid for obtaining accurate results for a demanding design process. This good result is specially satisfactory considering the nonlinear characteristics of the actuator and EBC controller tuning.

It's must be remarked that the proposed procedure is applied in this work using a machine learning based model and an EBC controller, but it can be applied to other actuators with different control algorithms and modeling techniques.

Declaration of Competing Interest

The authors report no declarations of interest.

Acknowledgements

The authors are grateful to the Spanish MINECO and the University of the Basque Country (UPV/EHU) for the partial support of this work through the projects DPI2017-82373-R and GIU18/196, respectively.

References

[1] C.P. Henry, D. Bono, J. Feuchtwanger, S.M. Allen, R.C. O'Handley, AC field-induced actuation of single crystal Ni-Mn-Ga, App. Phys. Lett. 91 (2002) 7810–7811.

[2] K. Ullakko, J.K. Huang, V.V. Kokorin, R.C. O'Handley, Magnetically controlled shape memory effect in. Ni2MnGa intermetallics, Scr. Mater. 36 (1997) 1133–1138.

[3] V.V. Kokorin, V.V. Martynov, Sequential formation of martensitic phases during uniaxial loading of single crystals of alloy Ni2MnGa, Phys. Met. Metall. 72 (3) (1991) 106–113.

[4] V.V. Kokorin, V.V. Martynov, V.A. Chernenko, Phase transitions in Ni3MnGa under compression, Sov. Phys. Solid State 3 (4) (1991) 708–709.

[5] M.A. Marioni, S.M. Allen, R.C. O'Handley, Nonuniform twin boundary motion in Ni-Mn-Ga single crystals, Appl. Phys. Lett. 84 (20) (2004) 4071–4073.

[6] J. Feuchtwanger, E. Asua, A. García-Arribas, V. Etxebarria, J.M. Barandiaran, Ferromagnetic shape memory alloys for positioning with nanometric resolution, App. Phys. Lett. 95 (5) (2009) 054102.

[7] J. Tellinen, I. Suorsa, A. Jääskelä inen, I. Aaltio, K. Ullakko, Basic properties of magnetic shape memory actuators, in: 8th International Conference on Actuator 2002 (2002).

[8] AdaptaMat Inc. http://www.adaptamat.com/products/actuators/, (2013).

[9] Z. Qingxin, Z. Hongmei, W. Fengxiang, The Mathematical Model of A Novel Linear Actuator and its Control Strategy, Proc. 26th Chinese Control Conference (2007) 165–168.

[10] Y. Ganor, D. Shilo, N.R. Zarrouati, D. James, Ferromagnetic shape memory flapper, Sens. Actuators A-Phys. 150 (2009) 277–279.

[11] M. Taya, T. Wada, M. Kusaka, R.C.C. Lee, Smart structures and materials 2003: industrial and commercial applications of smart structures technologies, Eric H, Anderson Editor. 15 (2003) 6–164.

[12] H. Tan, M.H. Elahinia, A nonlinear model for ferromagnetic shape memory alloy actuators, J. Commun. Nonl. Sci. Numer. Simul. 13 (9) (2008) 1917–1928.

[13] L. Riccardi, D. Naso, B. Turchiano, H. Janocha, Design of linear feedback controllers for dynamic systems with hysteresis, IEEE Trans. Control Syst. Technol. 22 (4) (2014) 1268–1280.

[14] L. Riccardi, D. Naso, B. Turchiano, H. Janocha, Adaptive control of positioning systems with hysteresis based on magnetic shape memory alloys, IEEE Trans. Control Syst. Technol. 21 (6) (2013) 2011–2013.

[15] M. Ruderman, T. Bertram, Observer-based inverse hysteresis control of prototypical magnetic shape memory (MSM) actuator, in: Proc. 2011 IEEE Int. Conf. on Mechatronics, Istanbul, Turkey, IEEE, 2011, pp. 627–631.

[16] M. Ruderman, B. Torsten, Control of magnetic shape memory actuators using observer-based inverse hysteresis approach, IEEE Trans. Control Syst. Technol. 22 (3) (2014) 1181–1189.

[17] E. Asua, J. Feuchtwanger, A. García-Arribas, V. Etxebarria, I. Orue, J.M. Barandiaran, Ferromagnetic shape memory alloy actuator for micro- and nano-positioning, Sens. Lett. 7 (3) (2009) 1–3.

[18] A. Sadeghzadeh, E. Asua, J. Feuchtwanger, V. Etxebarria, A. García-Arribas, Ferromagnetic shape memory alloy actuator enabled for nanometric position control using hysteresis compensation, Sens. Actuators A-Phys. 182 (2012) 122–129.

[19] A. Hubert, N. Calchand, Y. Le Gorrec, J. Gauthier, Magnetic Shape Memory Alloys as smart materials for micro-positioning devices, Adv. Electromagnet. 1 (2) (2012) 75–84.

[20] L. Riccardi, B. Holz, H.H. Janocha, Exploiting hysteresis in position control: the magnetic shape memory push-push actuator, Conference on Innovative Small Drives and Micro-Motor Systems, IEEE (2013) 63–68.

[21] E. Asua, A. Sadeghadeh, J. Feuchtwanger, V. Etxebarria, A. García-Arribas, Design of a non FSMA-Based Actuator for nanopositioning applications, in: Int. Conf. Smart Materials and Nanotechnology in Engineering SMN2011, SPIE (8409) (2011) 840906–840911.

[22] E. Asua, A. García-Arribas, V. Etxebarria, J. Feuchtwanger, Pulsed-mode operation and performance of a ferromagnetic shape memory alloy actuator, Smart Mater. Struc. 23 (2) (2014) 025023.

[23] E. Asua, J. Jugo, M. Eguiraun, A. García-Arribas, Feuchtwanger, V. Etxebarria, Energy-saving control for a ferromagnetic shape memory alloy based actuator, Sens. Actuators A: Phys. 249 (2016) 112–121.

[24] W. Li, C. Zhang, W. Gao, M. Zhou, Neural network self-tuning control for a piezoelectric actuator, Sensors 20 (12) (2020) 3342.

[25] Abadi M., Agarwal A., Barham P., Brevdo E., Chen Z., Citro C., Corrado G.S., Davis A., Dean J., Devin M., Ghemawat S., Goodfellow I., Harp A., Irving G., Isard M., Jozefowicz R., Jia Y., Kaiser L., Kudlur M., Levenberg J., Mané D., Schuster M., Monga R., Moore S., Murray D., Olah C., Shlens J., Steiner B., Sutskever I., Talwar K., Tucker P., Vanhoucke V., Vasudevan V., Viégas F., Vinyals O., Warden P., Wattenberg M., Wicke M., Yu Y., and Zheng X., TensorFlow: Large-scale machine learning on heterogeneous systems, (2015). Software available from tensorflow.org.

[26] F. Chollet and others, Keras, GitHub, (2015). https://github.com/fchollet/keras.

[27] K. Kuhnen, Modeling, Identification and Compensation of Complex Hysteretic: A Modified Prandtl - Ishlinskii Approach, European Journal of Control. 9 (4) (2013) 407–418.

[28] V. Hassani, T. Tjahjowidodo, T.N. Do, A survey on hysteresis modeling, identification and control, Mech. Syst. Signal. Process. 49 (2014) 209–233.

[29] C. Zhang, Y. Yu, Y. Wang, M. Zhou, Takagi–Sugeno fuzzy neural network hysteresis modeling for magnetic shape memory alloy actuator based on modified bacteria foraging algorithm, Int. J. Fuzzy Syst. 22 (4) (2020) 1314–1329, http://dx.doi.org/10.1007/s40815-020-00826-9.

[30] Y. Yu, C. Zhang, M. Zhou, NARMAX model-based hysteresis modeling of magnetic shape memory alloy actuators, IEEE Trans. Nanotechnol. 19 (2020) 1–4.

[31] J. Jugo, J. Feuchtwanger, J. Corres, V. Etxebarria, Analysis of temperature effects in high accuracy ferromagnetic shape memory alloy actuators, Sens. Actuators A: Phys. 271 (2018) 174–181.

[32] D. Lehmann, Event-based State-feedback Control, Logos Verlag Berlin GmbH, 2011.

[33] M.C.F. Donkers, W.P.M.H. Heemels, Output-based event-triggered control with guaranteed-gain and improved and decentralized event-triggering, IEEE Trans. Automatic Control. 57 (6) (2012) 1362–1376.

[34] J. Jugo, M. Eguiraun, Stability analysis and control design of a class of event based control systems, Proceedings of UKACC International Conference on Control, UKACC (2012) 252–258, A.I. Tyatyushkin, Numerical optimization methods for controlled systems with parameters. Comput. Math. Math. Phys. 57 (2017) 1592-1606.

[35] A.I. Tyatyushkin, Numerical optimization methods for controlled systems with parameters, Comput. Math. Math. Phys. 57 (2017) 1592–1606.

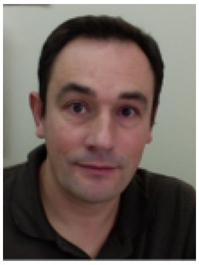
[36] D. Whitley, A genetic algorithm tutorial, Stat. Comput. 4 (1994) 65–85.

[37] D.P. Kroese, R.Y. Rubinstein, Monte Carlo Methods, Wiley Interdiscip. Rev. Comput. Stat. 4 (2012) 48–58.

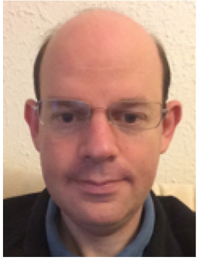
[38] J. Olondriz, J. Jugo, I. Elorza, A. Pujana-Arrese, S. Quesada-Alonso, A feedback control loop optimisation methodology for floating offshore wind turbines, Energies 12 (18) (2019) 3490.

[39] L.J. Eshelman, J.D. Schaffer, Real-coded genetic algorithm and interval-schemata, Found. Genetic Algorithms. 2 (1993) 187–202.

Biographies



Josu Jugo IEEE member, obtained his Ph.D. degree in Control Engineering in 1997 at the University of the Basque Country UPV/EHU. Since 2002, he is an assistant professor of Systems Engineering and Automatic Control in the UPV/EHU and member of the research group on Applied Control GAUDEE. His research work has been focussed to the control of mechatronics systems (active magnetic bearings, wind generators, machine tool, advanced actuators) and other complex systems, as RF power amplifiers, being coauthor of journal and conference papers in such areas. Last years, he has been involved in the development of science instrumentation and control systems.



J. Feuchtwanger graduated 2000 from the National Autonomous University of Mexico with a degree in Chemical Engineering. In 2006 he graduated from the Massachusetts Institute of Technology with a Ph.D. in Materials Science, his thesis topic was Ferromagnetic Shape Memory Alloy composites. Currently he is a research scientist at the University of the Basque Country.

Javier Corres obtained a bachelors in science in Applied Physics (specializing electronics) in 2004 from the University of Cantabria. He obtained the Diploma of Advanced Studies in Materials Science and Technology in 2010 and a Masters in Engineering Physics, oriented to control systems in 2013. Currently, he is a doctoral student in the University of the Basque Country and he is working on the analysis and control of FSMA based actuators.