Modelling a Bouc-Wen Hysteresis by SOFNN-AGA

Otávio Augusto Rodrigues de Oliveira Freitas *
Samir Angelo Milani Martins *

* GCOM - Grupo de Controle e Modelagem
UFSJ - Universidade Federal de São João del-Rei
Pça. Frei Orlando, 170 - Centro - 36307-352 - São João del-Rei, MG,
Brasil, (e-mail: otavio_augusto_freitas@hotmail.com,
martins@ufsj.edu.br)

Abstract: This paper shows a method to obtain a model for a system with hysteresis in an online way. For this purpose, it is used a Self Organized Fuzzy Neural Network with Adaptive Gradient algorithm (SOFNN-AGA). Additionally, it is proposed to use sign function of velocity as an additional information to the network. A Bouc-Wen model for a magnetic-rheological damper (MRD) was used as a system test. A simple model was easily obtained since the structure and parameters are updated each iteration.

Resumo: Este artigo mostra um método para se obter um modelo para um sistema com histerese de maneira *on-line*. Para tanto, utiliza-se de Redes Neurais Fuzzy Auto-Estruturantes com Algoritmo Gradiente Adaptativo (SOFNN-AGA). Também é proposto o uso da função sinal da velocidade como informação adicional à rede. Um modelo de Bouc-Wen para um amortecedor reológico magnético (MRD) foi usado como teste do sistema. Um modelo simples foi facilmente obtido visto que a estrutura e os parâmetros são atualizados a cada iteração.

Keywords: System Identification; Fuzzy Neural Network; Hysteresis; Nonlinear Systems

Palavras-chaves: Identificação de sistemas; Redes Neurais Fuzzy; Histerese; Sistemas Não lineares

1. INTRODUCTION

Representing systems by mathematical models has been a challenge because sometimes it is not possible to get models through the process's physics. Most systems are nonlinear and although it is possible to represent them during a restrict operating range by a linear system approximation, nonlinear systems can only be fully described by a nonlinear model (Aguirre, 2007).

Hysteresis is a nonlinear quasi-static phenomena found in several mechanics and magnetic systems such as magnetic-rheological dampers (Zamani et al., 2019), electromechanical actuators and sensors (Meng et al., 2019), involving memory effects between input and output (Martins and Aguirre, 2016). Over the years, manifold models were developed to represent systems with hysteresis. Among them, noteworthy models are Bouc-Wen (Wen, 1976), Prandtl-Ishilinsk (Kuhnen, 2003) and Preisach (Preisach, 1935). Approaches based in conventional mathematics (e.g. differential equations) are most widely used to model nonlinear systems.

They are able to reduce complexity and describe features of nonlinear systems such as NARX (Nonlinear Autoregressive Exogenous Model) (Bouaddi et al., 2016; Martins

and Aguirre, 2016) and artificial neural networks (ANN) (Cheng et al., 2015; Laudani et al., 2014).

The ANN has been used to model systems with hysteresis identification (Wang and Chen, 2017; Wang et al., 2018; Zhang et al., 2010) due to the lower complexity. ANN aims to simulate a human neuron and one of the main features is the learning ability using algorithms. Using the characteristics of ANN's Han et al. (2017) proposed a self-organized fuzzy neural network with an adaptive gradient algorithm (SOFNN-AGA), capable to define simultaneously in structure and determine parameters.

SOFNN-AGA has been applied to nonlinear dynamic systems. Despite these applications, none of them have had hysteresis. This work aims a model that can describe a system with hysteresis using the approach proposed by Han et al. (2017). Using the features of SOFNN-AGA, it is possible to obtain a model of a system with hysteresis.

This paper is organized as follow. In the section 2 is described preliminaries concepts required to understand this work. In section 3 are presented the methods that were used, the section 4 shows the numerical results. Concluding remarks and perspectives of future researches are given in section 5.

* The authors thank GCOM, PPGEEL, CAPES, CNPq/INERGE, FAPEMIG and the Federal University of São João del-Rei for their support.

2. PRELIMINARIES

2.1 Self-organizing fuzzy neural network (SOFNN)

The self-organizing fuzzy neural network was introduced by Leng et al. (2004). The method builds a neural network self-organizing which is designed to approximate to a fuzzy algorithm through a ANN, thus creates a hybrid ANN model, using the features of the fuzzy logic and ANN.

The structure of SOFNN has four layers and its mathematical description is given as follow:

(1) Input layer

There are m neurons in this layer, being m the number of input variables. The output of the i -th neuron of this layers is given by

$$u_i(k) = x_i(k); (i = 1, 2, \dots, m) \quad (1)$$

where $u_i(k)$ is the i th output value at time t , and $\mathbf{x}(k) = [x_1(k), x_2(k), \dots, x_i(k), \dots, x_m(k)]$ is the vector of the model inputs at time t .

(2) Radial basis function (RBF) layer

Each neuron of this layer is a RBF function. The output values are

$$\varphi_j(k) = \prod_{i=1}^m e^{-\frac{(u_i(k) - c_{ij}(k))^2}{2\sigma_{ij}^2}} \quad (2)$$

$$= \sum_{i=1}^m -\frac{(u_i(k) - c_{ij}(k))^2}{2\sigma_{ij}^2}$$

where $\varphi_j(k)$ is the output of the j th neuron of this layer at time t , $\mathbf{c}_j(k) = [c_{1j}(k), c_{2j}(k), \dots, c_{ij}(k), \dots, c_{mj}(k)]$ are the center parameters of the RBF in the j th neuron at time t , $\sigma_j(k) = [\sigma_{1j}(k), \sigma_{2j}(k), \dots, \sigma_{ij}(k), \dots, \sigma_{mj}(k)]$ are the width parameters of the of the RBF in the j th neuron at time t , $u_i(k)$ is the i th output value at time t , m is the number of neurons at input layer and P is the number of neurons at RBF layer.

(3) Normalized layer

This layer has the same length of the RBF layer. The output of the normalized layer is given by

$$v_l(k) = \frac{\varphi_l(k)}{\sum_{i=1}^P \varphi_i(k)} \quad (3)$$

where $v_l(k)$ is the output of the l th neuron of this layer at time t , where $\varphi_l(k)$ is the output of the l th neuron of this layer at time t and P is the number of neuron in the layer.

(4) Output layer

The output of this layer is a weighted sum of normalized layer outputs given by:

$$y(k) = \sum_{l=1}^P w_l v_l(k) \quad (4)$$

where w_l is the weight associated with the l th output of normalized layer and $y(k)$ is the network output.

Figure 1. Architecture of SOFNN. Source: Han et al. (2017)

2.2 Potentiality of Fuzzy Rule

This method was proposed by Han et al. (2017). The PFR method gives an index for every neuron. Based on that index, the SOFNN-AGA is able to create a new neuron or prune an irrelevant one.

Before introducing the PFR method, the linear regression model of fuzzy neural networks (FNN) is given bellow considering the past k input-output data

$$\mathbf{y}_d(k) = \mathbf{W}(k)\Phi(k) + \mathbf{e}(k) \quad (5)$$

where $\mathbf{y}_d(k) = [y_d(k - N + 1), y_d(k - N + 2), \dots, y_d(k)]$ is the desired output, $\mathbf{e}(k) = [e(k - N + 1), e(k - N + 2), \dots, e(k)]$ is the network error, $\mathbf{W}(k) = [\mathbf{w}(k - N + 1), \mathbf{w}(k - N + 2), \dots, \mathbf{w}(k)]^T$ is the weight between output and normalized layers and the regressors $\Phi(k) = [\mathbf{v}(k - N + 1), \mathbf{v}(k - N + 2), \dots, \mathbf{v}(k)]$ such that $\mathbf{v}(k) = [v_1(k), v_2(k), \dots, v_P(k)]$ is the output of normalized layer at time k .

The matrix Φ can be orthogonalized by a Gram-Schmidt process and decomposed by

$$\Phi^T(k) = \mathbf{Q}(k)\mathbf{A}(k) \quad (6)$$

Where $\mathbf{Q}(k) = [\mathbf{q}_1(k), \mathbf{q}_2(k), \dots, \mathbf{q}_1(k), \dots, \mathbf{q}_P(k)]$ is a orthogonal matrix and $\mathbf{A}(k)$ is a transformation matrix upper triangular.

Afterwards, the error reduction rate (ERR) Billings (1980) can be computed by

$$err_l(k) = \frac{\mathbf{y}_d(k)\mathbf{q}_l^T(k)}{\mathbf{q}_l^T(k)\mathbf{q}_l(k)\mathbf{y}_d^T(k)\mathbf{y}_d(k)} \quad (7)$$

The PRF is given by

$$PFR_l = \frac{R_l}{\sum_{j=1}^P R_j} \quad (8)$$

where l refers to l th neuron of normalized layer, and

$$R_l(t+1) = \lambda R_l(k) + v(t - \rho + 1)err_l(k), \quad (9)$$

$$l = 1, 2, \dots, P; \rho = k, k-1, \dots, 1.$$

where $0 < \lambda < 1$ is a constant.

The SOFNN-AGA evaluates the PFR in order to select which neurons will be created or pruned. The process is described below.

- (1) Growing phase: $PFR_i > \beta_0$

where $0 < \beta_0 \leq 1$ is the growing threshold and PFR_i is the potentiality of fuzzy rule of i th normalized layer neuron. If this condition is accomplished, a new neuron is created and the parameters of new neurons is designed as:

$$\begin{aligned} \mathbf{c}_{new}(k) &= \frac{1}{2}(\mathbf{c}_i(k) + \mathbf{x}(k)), \\ \sigma_{new}(k) &= \sigma_i(k), \\ \mathbf{w}_{new}(k) &= \mathbf{w}_i(k); \end{aligned} \quad (10)$$

where $\mathbf{c}_{new}(k)$, $\sigma_{new}(k)$, $\mathbf{w}_{new}(k)$ are the center, width and weight parameters respectively of new neuron and $\mathbf{c}_i(k)$, $\sigma_i(k)$, $\mathbf{w}_i(k)$ are the center, width and weight parameters respectively of i th neuron.

- (2) Pruning phase: $PFR_i < \beta_1$

where $0 < \beta_1 \leq 0,01$ is the pruning threshold. If the condition is satisfied, the i th will be pruned and the parameters of the nearest (the definition of nearest neuron is described in Wu et al. (2001)) neuron will be updated as:

$$\begin{aligned} \mathbf{c}_{ia}(k) &= \mathbf{c}_{ib}(k); \\ \sigma_{ia}(k) &= \sigma_{ib}(k); \\ w_{ia}(k) &= \frac{(w_{ib}(k)v_{ib}(k) + w_i(k)v_i(k))}{v_{ib}(k)}; \end{aligned} \quad (11)$$

where $\mathbf{c}_{ib}(k)$, $\sigma_{ib}(k)$, $w_{ib}(k)$ are the center, width and weight parameters of the nearest neuron before pruning respectively, $\mathbf{c}_{ia}(k)$, $\sigma_{ia}(k)$, $w_{ia}(k)$ are the center, width and weight parameters of the nearest neuron after pruning respectively, $w_i(k)$, $v_i(k)$ are the weight and output of normalized layer associated to pruned neuron.

2.3 Adaptive gradient algorithm(AGA)

AGA is the method used for center, length and weight parameters estimation Han et al. (2017). This method proposes an adaptive learning rate in order to minimize a target function described as:

$$E(k) = \frac{1}{2}[y_d(k) - y(k)]^2; \quad (12)$$

then is calculated the adaptive learning rate given as follow:

$$\eta_c(k) = \mu \frac{\|e(k)\|^2}{\|\frac{\delta y(k)}{\delta \mathbf{C}(k)} e(k)\|^2}; \quad (13)$$

$$\eta_\sigma(k) = \mu \frac{\|e(k)\|^2}{\|\frac{\delta y(k)}{\delta \Sigma(k)} e(k)\|^2}; \quad (14)$$

$$\eta_w(k) = \mu \frac{\|e(k)\|^2}{\|\frac{\delta y(k)}{\delta \mathbf{W}(k)} e(k)\|^2}; \quad (15)$$

where $\eta_c, \eta_\sigma, \eta_w$ are the adaptive learning rates, $\mathbf{C}(k) = [\mathbf{c}_1(k), \mathbf{c}_2(k), \dots, \mathbf{c}_P(k)]$, $\Sigma(k) = [\sigma_1(k), \sigma_2(k), \dots, \sigma_P(k)]$, $\mathbf{W}(k) = [w_1(k), w_2(k), \dots, w_P(k)]$ are the parameters of center, width and weight respectively, $e(k) = y_d(k) - y(k)$ and $\mu \in [0, 0.05, 0.01]$ is a constant.

After calculating adaptive learning rates it is possible to determine parameters of center, width and weight between the normalized and output layers of l th neuron as follow:

$$\mathbf{c}_l(t+1) = \mathbf{c}_l(k) + \eta_c(k) \frac{\delta E(k)}{\delta \mathbf{c}_l(k)} \quad (16)$$

$$\sigma_l(t+1) = \sigma_l(k) + \eta_\sigma(k) \frac{\delta E(k)}{\delta \sigma_l(k)} \quad (17)$$

$$w_l(t+1) = w_l(k) + \eta_w(k) \frac{\delta E(k)}{\delta w_l(k)} \quad (18)$$

2.4 Bouc-Wen Model for a Magnetic-rheological Damper (MRD)

The equation (19) describes a Bouc-Wen model for a MRD with hysteresis.

$$\begin{aligned} \mathcal{F} &= c_1 \dot{y} + k_1(x - x_0), \\ \dot{y} &= \frac{1}{c_0 + c_1}[\alpha z + c_0 \dot{x} + k_0(x - y)], \\ \dot{z} &= -\gamma|\dot{x} - \dot{y}|z|z|^{n-1} - \beta(\dot{x} - \dot{y})|z|^n + A(\dot{x} - \dot{y}), \\ \alpha &= \alpha_a + \alpha_b u_{BW}, \\ c_1 &= c_{1a} + c_{1b} u_{BW}, \\ c_0 &= c_{0a} + c_{0b} u_{BW}, \\ \dot{u}_{BW} &= -\eta(u_{BW} - E). \end{aligned} \quad (19)$$

where x is displacement, E is voltage input, z is the variable hysteretic, η viscosity of fluid, c_0 is damping coefficient, k_1 , k_2 , and c_1 are associated with linear solid material, $\mathcal{F}$ is resultant force. The parameter can be found in Spencer et al. (1997). This hysteresis model was used in several works, such as Martins and Aguirre (2016) and Zhu et al. (2019)

3. METHODS

The data used in identification process was acquired simulating the MRD model proposed by Spencer Jr et al. (1997) described in equation (19). The inputs (voltage E and displacement x) was generated by a low-pass filtering White Gaussian Sequences and sinusoidal signals in different frequencies. It was used 1000 samples to train network and 3000 to validate the model.

Using the approach proposed by Martins and Aguirre (2016), it was used as model input the sign function of velocity($\dot{x}$). The equation bellow describes the sign function:

$$\text{sign}(\dot{x}) = \begin{cases} 1, \dot{x} > 0; \\ 0, \dot{x} = 0; \\ -1, \dot{x} < 0; \end{cases} \quad (20)$$

The sign of velocity indicates if displacement instant is loading or unloading. For hysteresis systems, when it has a loading-unloading signal as input, the input-output plane is different for loading and unloading instants, so the information of loading or unloading instant is important due to its characteristics. Using $\text{sign}(\dot{x})$ as a model input, SOFNN-AGA can converge to a model more easily.

The parameters was initialized as:

$$\mathbf{c}_1(1) = \sigma_1(1) = [a_1, \dots, a_P]; \quad w_1(1) = 1; \quad (21)$$

where $\mathbf{c}_1, \sigma_1$ and w_1 are parameters of center, width and weight of SOFNN-AGA, P is number of inputs, and $a_1 = a_2 = \dots = a_P = 1$.

Applied the three input in SOFNN-AGA, the output is calculated. The PFR is calculated and evaluated comparing with growing threshold (β_0) and pruning threshold (β_1). It has been done empirical tests to choose best parameters growing threshold, pruning threshold and the constant λ . Based on the analysis, the algorithm creates or prunes a neuron.

The parameters were estimated using the AGA, this algorithm is able to find parameters that will converge to a good model. As in the PFR the parameter of adaptive learning rate (μ) and initial centers, width and weight was defined empirically based in a great number of test using.

The SOFNN-AGA pseudo-code is described in algorithm 3.

To validate model was done using the RMSE (Root-Mean-Squared error) which is defined as:

$$RMSE = \frac{\sqrt{\sum_{k=1}^N (y_d(k) - y(k))^2}}{\sqrt{\sum_{k=1}^N (y_d(k) - \bar{y}(k))^2}} \quad (22)$$

were $y_d(k)$ is desired output, $y(k)$ is SOFNN-AGA output, $\bar{y}(k)$ is mean of output value.

4. RESULTS

During identification process, parameters were defined $\beta_0, \beta_1, \lambda$ and μ . Values are presented in table 1

Parameter	Value
β_0	0,6
β_1	0,07
λ	0,4854
μ	0,3

Table 1. Chosen parameter to model a MRD

Figures 2 and 4 show the training phases. In the case where the input was Gaussian white noise (figure 2) around 300 samples was required to SOFNN-AGA find parameters and select the structure. The second case

Algorithm 1 Pseudo-code of the SOFNN-AGA to model a MRD

```

1: input displacement( $x$ ), sign( $\dot{x}$ ), voltage( $E$ )
2: initialize initial SOFNN parameters
3:  $c_{11}(1) \leftarrow c_{11}$ 
4:  $\sigma_{11}(1) \leftarrow \sigma_{11}$ 
5:  $w_{11}(1) \leftarrow w_{11}$ 
6: initialize prune and growing parameters
7:  $\beta_0 \leftarrow \beta_0 \in [0; 1]$ 
8:  $\beta_1 \leftarrow \beta_0 \in [0; 0, 1]$ 
9: initialize constants  $\mu, \lambda$ 
10:  $\lambda \leftarrow \lambda \in [0; 1]$ 
11:  $\mu \leftarrow \mu \in [0, 05; 0, 1]$ 
12: for  $k=1:T$  do
13:   for  $j=1:P$  do
14:     calculate RBF layer output
15:     calculate normalized layer output
16:   end for
17:   calculate SOFNN output
18:   for  $i=1:P$  do
19:     calculate ERR
20:     calculate PFR
21:     if  $PFR_i > \beta_0$  then
22:       neuron must be created
23:       initialize parameters of new neuron
24:     end if
25:     if  $PFR_i \leq \beta_1$  then
26:       neuron must be pruned
27:       update parameters of nearest neuron
28:     end if
29:   end for
30:   calculate  $E(k)$ 
31:   calculate adaptive learning rate
   ( $\eta_c(k), \eta_w(k), \eta_\sigma(k)$ )
32:   update SOFNN-AGA parameters
33:   if  $k = 1000$  & Model do not converge then
34:     initialize new initial parameters and repeat process
35:   else
36:     continue with training process
37:   end if
38: end for

```

Figure 2. Training phase for input low-pass filtering White Gaussian Sequences

where the network was trained using as input sinusoidal function (figure 4) SOFNN-AGA required less than 100 samples to find a model. In both cases, the structure selected has 4 neurons in RBF and Normalized layers.

Testes	SOFNN-AGA	NARX polynomial (Martins and Aguirre, 2016)
White Gaussian Noise inputs	0.2259	0.1350
Sinusoidal Function	0.0510	0.1190

Table 2. RMSE values for tests in SOFNN-AGA and NARX polynomial models

Figure 3. Test phase for input low-pass filtering White Gaussian Sequences

Figure 4. Training phase for input $E = 0.5 \sin(\pi t) + 1.6V$ and $x = 0.5 \sin(6\pi t)cm$

Figure 5. Test phase for input $E = 0.5 \sin(\pi t) + 1.6V$ and $x = 0.5 \sin(6\pi t)cm$

Using SOFNN-AGA the process of structure selection becomes simple due to the PFR method. For each sample, the model selects the least important neurons and prune them or creates new neurons if necessary. The parameters

Figure 6. NARX polynomial model found in Martins and Aguirre (2016)

are updated at each iteration. The AGA evaluates error and updates parameters of center, width and weight in order to fit data.

The limitation of this approach is that the network performance depends on constants $\beta_0, \beta_1, \lambda$ and μ chosen. It was necessary empirical tests to find values that worked.

In comparison with the model obtained by Martins and Aguirre (2016) the RMSE of the both models are very similar. The proposed SOFNN-AGA model requires less work in the identification process than the model proposed by NARX polynomials which is more complex. On the other hand, the polynomial NARX model has a larger scope, being able to represent the system in a larger operating range.

5. FINAL REMARKS

This paper presents a way to obtain a model for systems with hysteresis using Self Organizing Fuzzy Neural Network. The Bouc-Wen model for a magnetic-rheological damper was used as test systems. This works also proposes using sign function of velocity as model input, this way SOFNN-AGA has the information if at the moment it is an instant of loading or discharging and thus improves network performance. The results were compared with a NARX model obtained by Martins and Aguirre (2016).

As a proposal for future researches, we intend to create a NARX-SOFNN. The auto recursive resources can give good information related to the dynamic features of the system. This way, the NARX-SOFNN can provide a fine model to represent hysteresis system. Future researches intend to find a method to determine the constants of the system.

REFERENCES

Aguirre, L.A. (2007). *Introdução à Identificação de Sistemas - Técnicas Lineares e Não-Lineares Aplicadas a Sistemas Reais*. Editora da UFMG. 3ª edição.

- Billings, S. (1980). Identification of nonlinear systems—a survey. In *IEE Proceedings D (Control Theory and Applications)*, volume 127, 272–285. IET.
- Bouaddi, S., Ahmed, I., et al. (2016). Modeling and prediction of reflectance loss in csp plants using a non linear autoregressive model with exogenous inputs (narl). In *Renewable and Sustainable Energy Conference (IRSEC), 2016 International*, 706–709. IEEE.
- Cheng, L., Liu, W., Hou, Z.G., Yu, J., and Tan, M. (2015). Neural-network-based nonlinear model predictive control for piezoelectric actuators. *IEEE Transactions on Industrial Electronics*, 62(12), 7717–7727.
- Han, H.G., Lin, Z.L., and Qiao, J.F. (2017). Modeling of nonlinear systems using the self-organizing fuzzy neural network with adaptive gradient algorithm. *Neurocomputing*, 0, 1–13. doi:10.1016/j.neucom.2017.05.065.
- Kuhnen, K. (2003). Modeling, identification and compensation of complex hysteretic nonlinearities: A modified prandtl-ishlinskii approach. *European journal of control*, 9(4), 407–418.
- Laudani, A., Lozito, G.M., and Fulginei, F.R. (2014). Dynamic hysteresis modelling of magnetic materials by using a neural network approach. In *AEIT Annual Conference-From Research to Industry: The Need for a More Effective Technology Transfer (AEIT), 2014*, 1–6. IEEE.
- Leng, G., Prasad, G., and McGinnity, T.M. (2004). An on-line algorithm for creating self-organizing fuzzy neural networks. *Neural Networks*, 17(10), 1477–1493.
- Martins, S.A.M. and Aguirre, L.A. (2016). Sufficient conditions for rate-independent hysteresis in autoregressive identified models. *Mechanical Systems and Signal Processing*, 75, 607–617. doi:10.1016/j.ymssp.2015.12.031.
- Meng, L., Li, F., and Li, H. (2019). Electromechanical hysteresis model and identification for soft dielectric elastomer actuator. In *Electroactive Polymer Actuators and Devices (EAPAD) XXI*, volume 10966, 1096610. International Society for Optics and Photonics.
- Preisach, F. (1935). Über die magnetische nachwirkung. *Zeitschrift für Physik A Hadrons and Nuclei*, 94(5), 277–302.
- Spencer, B.F.J., Dyke, S.J., Sain, M.K., and Carlson, J.D. (1997). Phenomenological Model for Magnetorheological Dampers. *Journal of Engineering Mechanics*, 123(3), 230–238. doi:10.1061/(ASCE)0733-9399(1997)123:3(230).
- Spencer Jr, B., Dyke, S., Sain, M., and Carlson, J. (1997). Phenomenological model for magnetorheological dampers. *Journal of engineering mechanics*, 123(3), 230–238.
- Wang, G. and Chen, G. (2017). Identification of piezoelectric hysteresis by a novel duhem model based neural network. *Sensors and Actuators A: Physical*, 264, 282 – 288. doi:https://doi.org/10.1016/j.sna.2017.07.058.
- Wang, Y., Xu, R., and Zhou, M. (2018). Prandtl-ishlinskii modeling for giant magnetostrictive actuator based on internal time-delay recurrent neural network. *IEEE Transactions on Magnetics*, 1–4. doi:10.1109/TMAG.2018.2827397.
- Wen, Y.K. (1976). Method for random vibration of hysteretic systems. *Journal of the engineering mechanics division*, 102(2), 249–263.
- Wu, S., Er, M.J., and Gao, Y. (2001). A fast approach for automatic generation of fuzzy rules by generalized dynamic fuzzy neural networks. *IEEE Transactions on Fuzzy Systems*, 9(4), 578–594.
- Zamani, A.A., Tavakoli, S., Etedali, S., and Sadeghi, J. (2019). Modeling of a magneto-rheological damper: An improved multi-state-dependent parameter estimation approach. *Journal of Intelligent Material Systems and Structures*, 1045389X19835938.
- Zhang, X., Tan, Y., Su, M., and Xie, Y. (2010). Neural networks based identification and compensation of rate-dependent hysteresis in piezoelectric actuators. *Physica B: Condensed Matter*, 405(12), 2687 – 2693. doi:http://dx.doi.org/10.1016/j.physb.2010.03.050.
- Zhu, H., Li, Y., Shen, W., and Zhu, S. (2019). Mechanical and energy-harvesting model for electromagnetic inertial mass dampers. *Mechanical Systems and Signal Processing*, 120, 203 – 220. doi:https://doi.org/10.1016/j.ymssp.2018.10.023.