25th ABCM International Congress of Mechanical Engineering
October 20-25, 2019, Uberlândia, MG, Brazil

COBEM2019-1080
CONVENTIONAL FUZZY AND TYPE-2 FUZZY LOGIC CONTROLLER TO DC MOTOR POSITION CONTROL TUNNED BY PARTICLE SWARM OPTIMIZATION

Mário Sérgio Freitas Ferreira Cavalcante
Mayranne Émylle Medeiros Furtunato
Willians Ribeiro Mendes
Fábio Meneghetti Ugulino de Araújo
Universidade Federal do Rio Grande do Norte
marioffcavalcante@gmail.com
mayranneemf@live.com
willians.mendes@pdl.ifmt.edu.br
meneghet@dca.ufrn.br

Abstract. *This paper proposes a method of dynamic adjustment of parameters of conventional fuzzy and fuzzy type-2 controllers using the particle swarm algorithm, or particle swarm optimization (PSO), for the control problem of a servo motor. Thus, two control techniques based on conventional fuzzy and fuzzy type-2 are presented, so a comparison is made between them in order to identify the best alternative for the control problem. The results obtained and analyzed using the metrics provided by the controller evaluation indices as ITEA, IEA and goodhart index.*

Keywords: *Position Control, Fuzzy Controller, Type fuzzy, Controller Optimization*

1. INTRODUCTION

Servo motors are highly used in industrial application and automation systems. These motors can be found in different devices such as automatic doors, route control, lenses in cameras. Although these motors are widely used and different controllers have used to increase efficiency, they are still influenced by non-linear parameters that affect their efficiency (Kandel *et al.*, 2000; Miller III *et al.*, 1990).

Nonlinear torque meters affect the servo motor operation in different ways. Saturation limits the position of the output of reaching the desired angular position, increases the overshoot in the system which can lead to instability (Jang, 2007). A servo motor has two main points that can be controlled, its angular velocity and its angular position (Rios-Gutierrez and Makableh, 2011).

The strategy enable by fuzzy logic allows to deal with uncertainties and imprecision of real linear systems. This logic was proposed by Zadeh (1965) as a new type of logic, known as Fuzzy logic. The fuzzy logic introduced the concept of dealing with vague aspects of information (Sandri and Correa, 1999), instead of limiting the element of a set to a fully contained (true) or not contained (false), this new logic allows to deal with information' uncertainty in a discourse universe, which allows the analysis element to be "partially" true or "partially" false in a given area of information.

Although conventional fuzzy logic allows a good representation for uncertainties, the words meaning has them on uncertainty, and it can not be well represent by this logic. Then, Zadeh (1975) proposed a modification in fuzzy sets. In this new technique, the membership functions are surrounded by a footprint of uncertainty - FOU which position and even the shapes of membership functions are not well defined.

Applications using fuzzy logic for the control of non-linear system can be seen in the petrochemical industries (Liao *et al.*, 2008), energetic (Bonabeau *et al.*, 1999), aeronautics (Berenji *et al.*, 2001). However, this techniques is not restrict to control area, several areas such as: signal processing, communication, expert system, medical, psychology (Erdinc *et al.*, 2009).

One of the main problems of fuzzy logic in general is its a lack of methodology to choose number, shape and tunning of membership functions as well as relationship between the inputs and outputs. Thus, this paper intents to investigate the optimization technique' utilization for tunning of both proposed fuzzy controllers and a traditional PI controller. Therefore, a particle swarm algorithm was used to define the position and quantity of membership function as well as finding the best values for the traditional controller.

2. EXPERIMENTAL PROCEDURES

A conventional fuzzy Takagi-Sugeno-Kang (TSK) was used to solve the position control problem, using the minimum operations for calculation of the intersection of the sets. The controller has two input and a single output. The inputs are: error and error derivative, the output was variation of control signal. The inputs have three membership functions.

Sugeno Fuzzy Controllers resembles PID controllers when the inputs uses the error information. For Fuzzy-PI controllers, the fuzzy controllers inputs must be: error and error-derivative, and the input must be control signal variance. The output is calculated by:

$$du = K_1 * e(t) + K_2 * \frac{de(t)}{dt} \tag{1}$$

Where k_i is first Sugeno constant, $\frac{de(t)}{dt}$ is the erro derivative (or variation), k_2 is the second Sugeno output constant and du is the control signal variation. Once, the output is the control signal variation, it is necessary to integrate de output. Matematically:

$$\begin{aligned} \int du &= k_1 * \int e(t) + K_2 * \int \frac{de(t)}{dt} \\ u &= k_1 * \int e(t) + K_2 * e(t) \end{aligned} \tag{2}$$

Where k_1 is the first Sugeno constant that assembles K_i constant on PID controllers and K_2 is the second output constant that assembles K_p constant on PID controllers. In this paper, a PI controller is compared with Conventional Sugeno PI-Fuzzy controller and Intervalar Type-2 PI-Fuzzy Controller.

2.1 Conventional Fuzzy Controller

Computer programs make strict decisions by hard rules and hard sets, based only in two values: 0, case false, or 1, case true. Such binary sets are known as crisp sets. Fuzzy sets provides a infinity number of possibilities, which allows values of "half-true" or partially-true, as well as, parcially-false.

Fuzzy logic provides a method to translate verbal expressions, imprecise and uncertain, commons in human knowledge in numerical values that are interpretable by a computer. Zadeh (1965) proposed fuzzy logic as an option to treat inaccurate information present in human language, once expressions such as: ”almost", ”a little", ”too much" are not part of the domain of classic logic sets because they can not be interpreted as ”all or nothing”(1 or 0).

This type of logic allows to express approximate representations of elements membership in sets. Typically, fuzzy set are represented in membership functions. A membership function is a numerical or mathematical function that assigns fuzzy membership values to discrete values of a variable in its discourse universe (Shaw and Simões, 1999).

The result after a statistical analysis with fifty tests, the membership functions generated by the optimization is present in the Fig. 1a for error (first input) and in Fig. 1b for error derivative (second input).

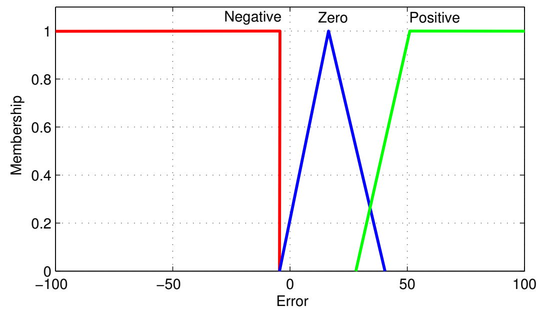
In TSK conventional fuzzy, the output is calculated as a weighted averaged of rules implications and output functions, as shown in Eq.3.

$$y = \frac{\sum_{i=1}^{n} w_i \cdot z_i}{w_i} \tag{3}$$

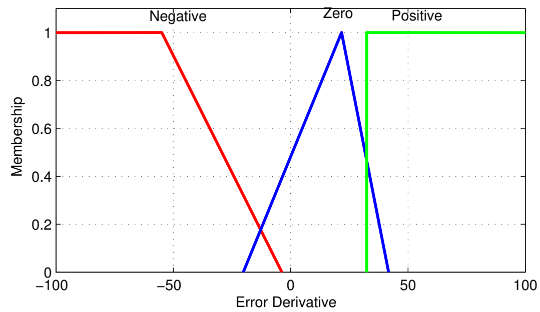
where, w_i is the rule implication of i rule ; z_i is the output function of i rule, y is the output of fuzzy logic. The rules used are shown in Tab. 1.

Table 1: Rules for conventional fuzzy controller

	Error Derivative		
Erro	Negative	Zero	Positive
Negative	S2	S1	S3
Zero	S2	S1	S1
Positive	S3	S1	S3



(a) Error



(b) Error Derivative

Figure 1: Conventional Fuzzy Membership Functions

Where S1, S2 and S3 are Sugeno output functions presents on Tab. 2.

Table 2: Sugeno output functions for conventional fuzzy controller

Name	Error Multiplier	Derivative Multiplier	Constant
S1	14,40	4,19	0,08
S2	81,68	51,21	0,01
S3	8,53	0,73	0,02

2.2 Type-2 Fuzzy Controller

Type-2 fuzzy logic was proposed by Zadeh in 1975 as an extension of conventional fuzzy logic. Its emergence is due to insufficiency of the conventional fuzzy locus dealing with uncertainties related to the very meaning of the words.

Fuzzy sets are, in general, one of the main aspects of fuzzy logic itself. In fuzzy type-2 logic, the degree of membership of an element in a set is not restricted to only one value, as is the case with conventional logic, but rather a conventional membership function (of type 1) (Karnik and Mendel, 1998).

It is conventional to represent type-2 fuzzy logic membership functions with a footprint of uncertainty (FOU). Mendel and John (2002) describe FOU as a "union of primary memberships", where the primary membership are the subsets of unit interval. However, Mendel emphasizes "FOU can let us easily visualize a T2 FS in two-dimensions" (Mo *et al.*, 2014). Figure 2 represents a type=2 fuzzy set.

In this paper, the result after a statistical analysis with fifty tests, the membership functions generated by the optimization is present in the Fig. 3a for error (first input) and in Fig. 3b for error derivative (second input).

In Type-2 TSK fuzzy model, the output is calculated based on weighted averaged of rules implication and output functions, shown in Eq. 4.

$$Y(x) = [y_l, y_r] = \int_{z_l^1 \in z_r^1} \dots \int_{z_M^1 \in z_r^M} \int_{f^1 \in [\underline{f}^1, \bar{f}^1]} \dots \times \int_{f^1 \in [\underline{f}^M, \bar{f}^M} \frac{1}{\sum_{i=1}^{M} f^i z^i / \sum_{i=1}^{M} f^i} \tag{4}$$

Where z is the Sugeno output functions, $\bar{f}$ is the upper membership function and $\underline{f}$ is the lower membership function.

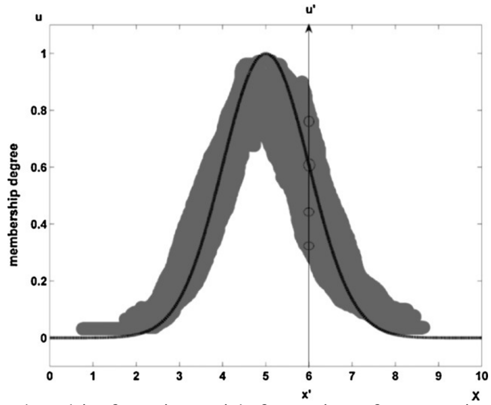
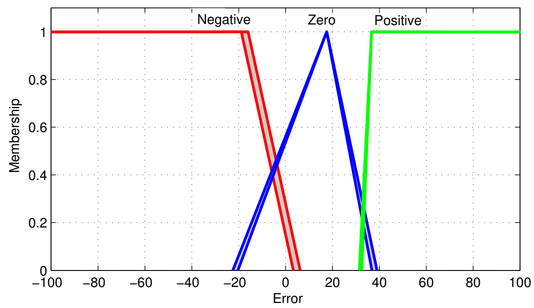
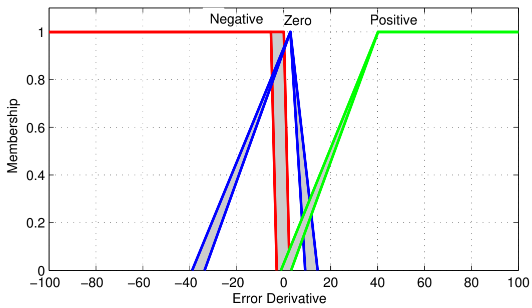


Figure 2: Type-2 membership function with footprint of uncertain Kandel *et al.* (2000).



(a) Error



(b) Error Derivative

Figure 3: Type-2 Fuzzy Membership Functions

This interval set is determined by two end points, y_l and y_r, which corresponds to the centroid of the fuzzy type-2 interval sets, you can obtain:

$$y_l = \frac{\sum_{i=1}^{M} f_l^i z_l^i}{\sum_{i=1}^{M} f_l^i} \tag{5}$$

$$y_r = \frac{\sum_{i=1}^{M} f_r^i z_r^i}{\sum_{i=1}^{M} f_r^i} \tag{6}$$

The y_l and y_r values define a interval output of type-2 fuzzy logic. The output valor is the averange valor between those values. Equation 7

$$y = \frac{y_r + y_l}{2} \tag{7}$$

The rules are shown in Tab. 3.

Where S1, S2, S3 are Sugeno output functions which values are on Tab. 4.

Table 3: Rules for conventional fuzzy controller

	Error Derivative		
Erro	Negative	Zero	Positive
Negative	S2	S1	S3
Zero	S2	S1	S1
Positive	S3	S1	S3

Table 4: Sugeno output functions for type-2 fuzzy controller

Name	Error Multiplier	Derivative Multiplier	Constant
S1	9,05	4,01	0,02
S2	4,91	5,08	0,01
S3	22,3	4,41	0,01

2.3 Particle Swarm Optimization

Meta-heuristics optimization algorithm are becoming more popular in engineering applications because a number of factors: rely on rather simple concepts (some are bio-inspired); they are easy to implement; can bypass local optima problem; can be utilized in a wide rage of problems covering different disciplines (Mirjalili and Lewis, 2016).

Particle Swarm Optimization (PSO) algorithm is a nature-based algorithm inspired by the social behavior of bird flocking and schoals of fishes, it was proposed by Kennedy and Eberhart (1995).

It use a number of particle (candidate solutions) which fly around in the discourse universe trying to find the best solution to the problem. The particles move through space respecting a series of mathematical equations and their position and velocity are updated by this equations which consider both best global solution and best personal solution.

In other words, the particle consider their own best solutions as well as the best solutions the swarm has obtained so far. Each particle in PSO should consider the current positions, the current velocity, the distance to *pbest* (personal best solution) and the distance to *gbest* (global best solution). Mathematically modelled as Eqs. 8 - 9.

$$v_i(k+1) = w \cdot v_i(k) + c_1 \cdot (p_{best_i} - x_i(k)) + c_2 \cdot (g_{best_i} - x_i(k)) \tag{8}$$

$$x_i(k+1) = x_i(k) + v_i(k+1) \tag{9}$$

Where $v_i(k)$ is the velocity of particle i at iteration k, w is weighting function (or constant), $x_i(k)$ is current position of the particle i at k iteration, p_{best_i} is the position of the best position of the best solution found by the particle i, g_best is the position of the best solution ever found by the swarm algorithm (Hatti, 2018).

Figure 4 presents a flowchart of PSO algorithm.

On control problem, which particle represents a specific controller, with own membership functions, own tunning, rules and Sugeno output functions. The Number of membership functions, position and openning were defined by the Particle Swarm Optimization (PSO) algorithm .

3. RESULT

The execution time of the optimization algorithm for the conventional fuzzy controller was 157.31 seconds, for type-2 fuzzy of 157.84, which indicates that for that case, the processing between the two similar logics, that is, the addition of the FOU did not present a great computational expense. For the PI controller the processing time for the optimization was 45.89 seconds. PSO parameters are shown on Tab. 5.

Table 5: PSO parameters

Number of Particles	20 Particles
Initial Velocity	0
Personal Factor - r_{1i}	0,8
Social Factor - r_{2i}	1,2
Inertia Factor	0,9 to 0,4
Stop Criteria	Particle Stagnation
	100 interactions

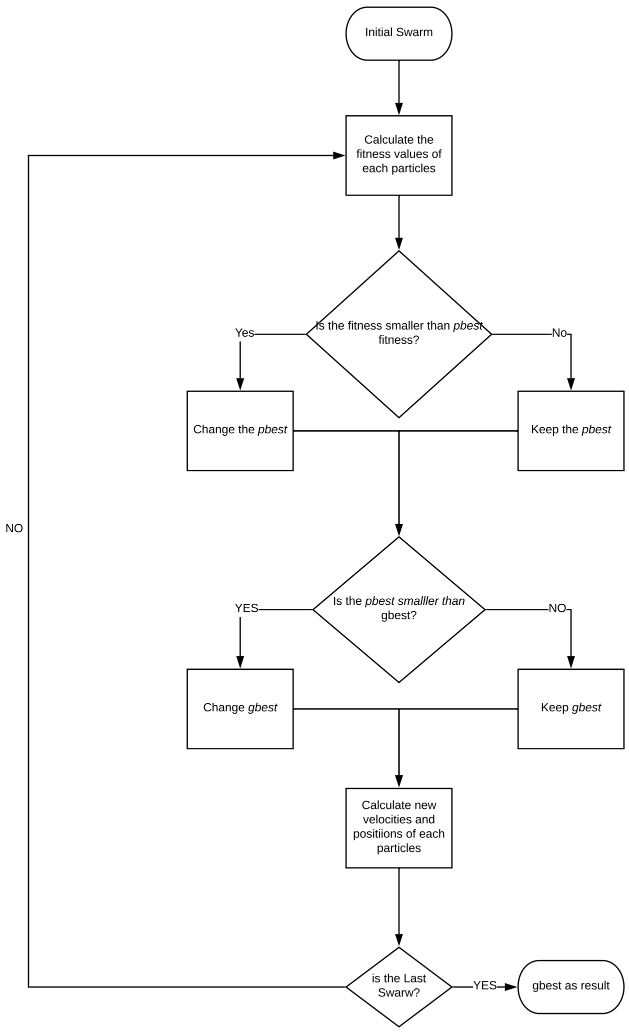
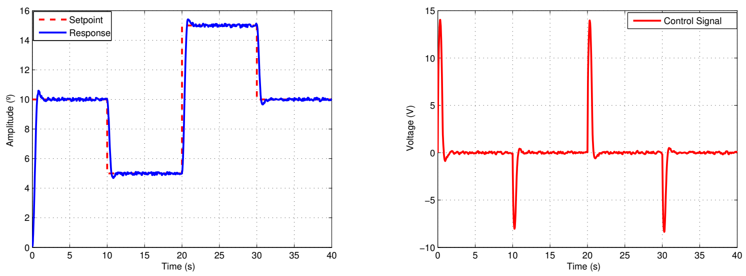


Figure 4: PSO Algorithm

3.1 Conventional Fuzzy Controller

Figure 5a represents the response to the conventional fuzzy controller optimized with PSO controller. The system response presents small overshoots for all steps applied and a small steady state error correction time. The oscillation present in the system response is relative to noise. The control signal shown in Fig.5b shows behavior with high voltage peaks when there is a setpoint change.

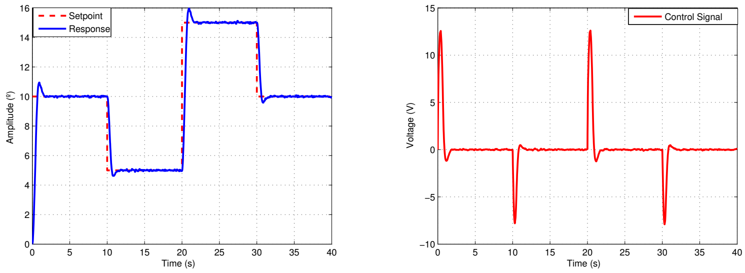


(a) Response (b) Control Signal

Figure 5: Conventional Fuzzy Controller

3.2 Type-2 Fuzzy Controller

Figure 6a represents the response to the type-2 fuzzy controller optimized with PSO controller. Note that the system response presents small overshoots for all steps applied in the reference and a small steady state error correction time. The oscillation present in the system response is relative to noise. The control signal present in Fig. 6b shows a behavior with high peaks when changing setpoints.

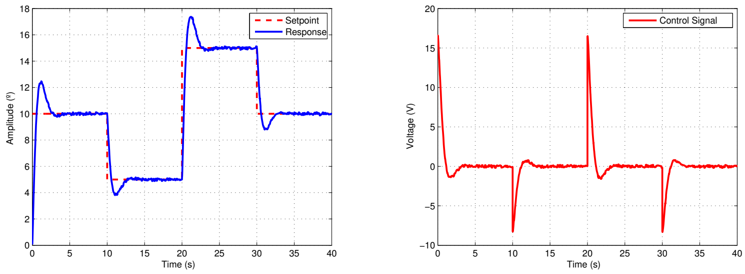


(a) System Response (b) Control Signal

Figure 6: Type-2 Fuzzy Controller

3.3 PI controller

The PI Controller generates response shown on Fig. 7a to this system. This controller was tunned by the PSO algorithm, $K_p = 2.29$ and $K_i = 1.47$. High overshoot and correction time to steady state error are the system behavior to the control signal, even with the overshoot, the system remains slow. The control signal, on Fig. 7b, shows voltage peaks when there are setpoint changes, but this peaks do not reaches saturation values.



(a) System Response (b) Control Signal

Figure 7: PI Controller

3.4 Results Analysis

Based on the behavior presented by the system it is possible to calculate the performance indexes of the controllers obtained through the PSO algorithm. The indices are shown in Tab. 6.

Table 6: Controllers Evaluation Index

	IEA	ITEA	IG
Conventional Fuzzy	$4,50 \times 10^{-1}$	$3,04 \times 10^{-1}$	$2,36 \times 10^{0}$
Type-2 Fuzzy	$4,34 \times 10^{-1}$	$2,20 \times 10^{-1}$	$1,80 \times 10^{0}$
PI Controller	$6,06 \times 10^{-1}$	$5,60 \times 10^{-1}$	$5,15 \times 10^{0}$

The conventional and fuzzy type-2 fuzzy controllers optimized by the PSO algorithm obtained better results than the PI controller. Among the fuzzy controllers, due to lower stabilization and rise time, the type-2 fuzzy controller obtained better values of performance indices. In the setpoint changes, the PI controller displays a large overshoot, reaching 20 % of the applied pulse value. The control signal also reaches large peaks when there is a change of reference.

For the Goodhart index, it is observed that the type-2 fuzzy controller has a better result than the conventional fuzzy controller, because of the larger weighting of the control signal. Conventional fuzzy presents larger peaks than the fuzzy type-2 controller. The PI controller has the highest Goodhart index among the controllers, due to the presence of large voltage peaks when there is a change of setpoint.

4. CONCLUSION

The work presents a strategy of tuning conventional and type-2 fuzzy using particle swarm optimization. The algorithm proved to be effective in finding parameters of complex controllers such as FT2. The FT2 controller presented better response than the conventional one, but it has more computational cost. This controller allows to better treat the plant nonlinearity and the knowledge impression.

The optimization of the type-2 fuzzy controller provided a gain of about 32.79 % in controller quality when examined by the evaluation function, corresponding to a significant gain for the plant. The addition of the FOU presents a significant gain in the performance of the optimized controller.

5. REFERENCES

Berenji, H.R., Saraf, S., Chang, P.W. and Swanson, S.R., 2001. "Pitch control of the space shuttle training aircraft". *IEEE Transactions on Control Systems Technology*, Vol. 9, No. 3, pp. 542–551.

Bonabeau, E., Marco, D.d.R.D.F., Dorigo, M., Théraulaz, G., Theraulaz, G. *et al.*, 1999. *Swarm intelligence: from natural to artificial systems*. 1. Oxford university press.

Erdinc, O., Vural, B. and Uzunoglu, M., 2009. "A wavelet-fuzzy logic based energy management strategy for a fuel cell/battery/ultra-capacitor hybrid vehicular power system". *Journal of Power sources*, Vol. 194, No. 1, pp. 369–380.

Hatti, M., 2018. *Artificial Intelligence in Renewable Energetic Systems: Smart Sustainable Energy Systems*, Vol. 35. Springer.

Jang, J.O., 2007. "Neural network saturation compensation for dc motor systems". *IEEE Transactions on Industrial Electronics*, Vol. 54, No. 3, pp. 1763–1767.

Kandel, E.R., Schwartz, J.H., Jessell, T.M., of Biochemistry, D., Jessell, M.B.T., Siegelbaum, S. and Hudspeth, A., 2000. *Principles of neural science*, Vol. 4. McGraw-hill New York.

Karnik, N.N. and Mendel, J.M., 1998. "Introduction to type-2 fuzzy logic systems". In *1998 IEEE International Conference on Fuzzy Systems Proceedings. IEEE World Congress on Computational Intelligence (Cat. No. 98CH36228)*. IEEE, Vol. 2, pp. 915–920.

Kennedy, J. and Eberhart, R., 1995. "Particle swarm optimization". In *Proceedings of ICNN'95 - International Conference on Neural Networks*. Vol. 4, pp. 1942–1948 vol.4. doi:10.1109/ICNN.1995.488968.

Liao, R., Chan, C., Hromek, J., Huang, G.H. and He, L., 2008. "Fuzzy logic control for a petroleum separation process". *Engineering Applications of Artificial Intelligence*, Vol. 21, No. 6, pp. 835–845.

Mendel, J.M. and John, R.I.B., 2002. "Type-2 fuzzy sets made simple". *IEEE Transactions on Fuzzy Systems*, Vol. 10, No. 2, pp. 117–127. ISSN 1063-6706. doi:10.1109/91.995115.

Miller III, W.T., Sutton, R.S. and Werbos, P.J., 1990. *Neural Networks for Control*. Neural Networks for Control. Cambridge, MA: MIT Press.

Mirjalili, S. and Lewis, A., 2016. "The whale optimization algorithm". *Advances in Engineering Software*, Vol. 95, pp. 51 – 67. ISSN 0965-9978. doi:https://doi.org/10.1016/j.advengsoft.2016.01.008. URL `http://www.sciencedirect.com/science/article/pii/S0965997816300163`.

Mo, H., Wang, F.Y., Zhou, M., Li, R. and Xiao, Z., 2014. "Footprint of uncertainty for type-2 fuzzy sets". *Information Sciences*, Vol. 272, pp. 96–110.

Rios-Gutierrez, F. and Makableh, Y.F., 2011. "Efficient position control of dc servomotor using backpropagation neural network". In *2011 Seventh International Conference on Natural Computation*. IEEE, Vol. 2, pp. 653–657.

Sandri, S. and Correa, C., 1999. "Lógica nebulosa". *Instituto Tecnológico da Aeronáutica–ITA, V Escola de Redes Neurais, pp. C073-c090, São José dos Campos*.

Shaw, I.S. and Simões, M.G., 1999. "Controle e modelagem fuzzy".

Zadeh, L.A., 1965. "Fuzzy sets". *Information and control*, Vol. 8, No. 3, pp. 338–353.

6. RESPONSIBILITY NOTICE

The authors are the only responsible for the printed material included in this paper.