24th ABCM International Congress of Mechanical Engineering
December 3-8, 2017, Curitiba, PR, Brazil

COBEM-2017-1528
CONTROL OF A BRUSHLESS DC MOTOR BY A FRACTIONAL ORDER PID CONTROLLER WITH A RADIAL BASIS FUNCTION NEURAL NETWORK TUNE METHOD

Suzana Nature Silveira Nunes
Felipe Lopes Romano
Antonio Augusto de Torres Maia
Universidade Federal de Minas Gerais, Department of Mechanical Engineering, Av. Pres. Antônio Carlos, 6627 - Pampulha, Belo Horizonte - MG, 31270-901
suzana.nature.nunes@gmail.com, felipelromano@outlook.com, aamaia@ufmg.br

Abstract. *There is no doubt that automatic control is critical to the modern manufacturing industry. The most commonly used automatic controller in the industry is the Proportional Integral Derivative (PID) controller. The design of ever better controllers is essential to obtain more efficient processes, and the use of fractional controllers may be the next step in the evolution of control systems. Fractional order calculation allows integration and differentiation with arbitrary order of operation. A FOPID adds two extra parameters to the equation, which makes it more appropriate for the system to be more complex to tune. To solve the complex tune issue, several optimization methods have been implemented, and the use of Radial Basis Neural Network (RBFN) seems promising. An application of a fractional order PID controller fitted with the Radial Basis Function Network is presented and studied in this article. This work aims to design a FOPID to control a DC motor. Comparisons are made with an integer PID controller and a FOPID tuned using a Genetic Algorithm (GA) algorithm.*

Keywords: *Fractional control, Dc motor control, RBFN, Controllers Tuning.*

1. INTRODUCTION

In recent decades the control theory has contributed substantially to the technological advances of several areas. Precision engineering and manufacturing processes, for example, have gained prominence and have played an important role in current and future technologies. At the same time, the use of DC motors has grown, and this diversity of applications makes the development of a robust, simple and efficient control solution capable of serving all applications a complex process.

The proportional-integral-derivative controller (PID), the industry's most popular method, is responsible for 97% of existing industrial controllers (Yu, 2006). However, designing and tuning an efficient PID controller can be difficult in practice if multiple objectives are met, such as short transients and high stability (Ang et al, 2005). The PID controller's' parameter settings also have room for improvement, as can be seen from the fact that only 32% of the PID controllers in the industry show satisfactory performance (Yu, 2006). A visit to a process plant will show that a large number of PID controllers are poorly adjusted (Skogestad, 2003).

The use of a FOPID (Fractional Order PID) is an idea with increasing attention in recent years, mainly due to the fact that real physical systems are well characterized by fractional order differential equations (Hwang et al, 2006). From the different FOPID controllers presented in (Cao et al, 2005) (Cervera et al, 2006) (Cao et al, 2006) and (Monje et al, 2004) it is clear that the FOPID controller perform better than Integer PID controllers, both for the control of integer and fractional order systems.

In spite of approaching the models of real systems, the use of FOPID increases its tuning complexity. Thus, the search for an efficient tuning methodology becomes a challenge. And so optimization techniques like (Cao et al, 2005) have been studied. Radial-based Neural Networks (RBF) have an approximation capability that, compared to other neural networks en optimization techniques, provide better approximation properties, such as high accuracy (Baiyu et al, 2010).

Thus, one can see that although there is a large volume of research being developed in this area, there are factors such as the difficulty of developing robust controllers and precise tuning methods that show that this is a topic of great interest for studies and improvements.

The objective of this work is the development of an algorithm and a PID Controller of fractional order and tuning by a Radial Basis Artificial Neural Network.

2. FRACTIONAL ORDER CONTROL

Fractional order control (FOC) is a field of control theory that uses the fractional order integrator as part of the control system design. Podlubny came out with the concept of Fractional order PID Controllers (Podlubny et al., 1997) by demonstrating its superior response when compared to a integer PID Controller in a non-integer plant.

Existing evidence has confirmed that the best fractional-order controller can overcome the best whole-order controller (Podlubny, 1999) for non-integer systems. The literature also addresses the reasons why choose fractional order control even when integer control works comparatively well (Monje et al, 2008). When using non integer order controllers for integer order plants, the FO controllers offer a wider range of possibilities in tuning the gain and phase characteristics than the ones we have by using IO controllers.

The typical form of a fractional order PID controller is the $PI^{\lambda}D^{\mu}$ controller (Podlubny, 1999), with an integrator of order λ and a differentiator of order μ where λ and μ can be any real numbers. The transfer function of this controller can be expressed as

$$G_c(s) = \frac{U(s)}{E(s)} = k_P + k_I \frac{1}{s^{\lambda}} + k_D s^{\mu}, (\lambda, \mu > 0) \quad (1)$$

where $G_c(s)$ is the transfer function of the controller, $E(s)$ is an error, and $U(s)$ is controller's output. The integrator term is $1/s^{\lambda}$.

The $PI^{\lambda}D^{\mu}$ controller, have been researched in time domain by (Podlubny, 1999) and in frequency domain by (Petras, 1999) and in its general form can be expressed in time domain by

$$\begin{aligned} &u(t) = K_p e(t) + T_i D_t^{-\lambda} e(t) + T_d D_t^{\mu} e(t) \\ &(D_t^{(*)} \equiv_0 D_{t(*)}) \end{aligned} \quad (2)$$

where λ and μ are positive real numbers, K_p is the proportional gain, T_i the integration constant and T_d the differentiation constant. It is trivial to see that by taking $\lambda = 1$ and $\mu = 1$, we obtain a classical integer PID controller. We can also have particular cases of $PI^{\lambda}D^{\mu}$ by taking, for instance T_d or T_i equal zero. A block diagram of a FOPID is expressed in Fig 1. Has mentioned above, it is clear that selecting $\lambda = 1$ and $\mu = 1$, we obtain a classical PID controller.

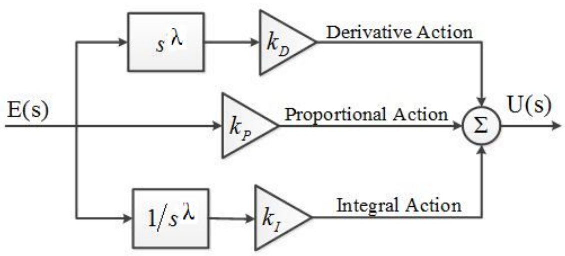

Figure 1. A basic block diagram of a Fractional Order PID Controller

As indicated in (Chen, 2006), for closed-loop control systems, there are four possible situations. Being
1) plant IO (integer order) with IO controller;
2) Plant IO with FO controller (fractional order);
3) FO plant with IO controller and
4) FO plant with FO controller.

For this work we have chosen a FO Plant to be controlled by both an FOPID and a PID.

3. RADIAL BASIS FUNCTION NETWORK

The word network in the term "artificial neural network" refers to the interconnections between neurons in the different layers of each system. An example system has three layers. The first layer has input neurons that send data via synapses to the second layer of neurons and then via more synapses to the third layer of output neurons as it can be seen in Fig 2. More complex systems will have more layers of neurons, some with increased layers of incoming neurons and output neurons. The synapses store parameters called "weights" that manipulate the data in the calculations (Schwenker et al, 2001).

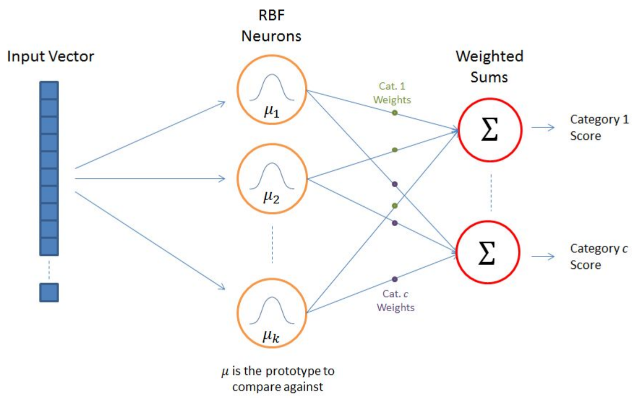

Figure 2 Typical architecture of an RBF Network
Source: (McCormick, 2013).

Starting in the 1940s with the introduction of the concept of neuron models (McCulloch, 1943), the studies and techniques in neural network field achieved vast development and have been applied in many field with high degrees of success. Neural networks differs from other techniques by its highly parallel structure, learning ability, nonlinear function approximation, fault tolerance, and efficient analog VLSI implementation for real-time applications. All of that combined justify the usage of neural networks in nonlinear system identification and control (Hunt et al, 1992).

In many real-world applications, there are plenty nonlinearities, unmodeled dynamics, unmeasurable noise, multi-loop, etc., which pose problems for engineers to implement control strategies.

In designing a $PI^{\lambda}D^{\mu}$ controller, one of the most important problems to be solved is the tuning of parameters, and great efforts were made to deal with it. Cao et al. (2005) provides a method of Genetic Algorithms. While Cervera et al (2006), it applies to the tuning of a $PI^{\lambda}D^{\mu}$ QFT controller. Also a method for as FOPID Controller based on PSO (particle swarm optimization) is proposed by Cao et al. (2006). Given the adaptive nature of the Neural networks, the RBF functions have also been used as methods of tuning $PI^{\lambda}D^{\mu}$ controllers.

A radial basis function network (RBF) is a particular artificial neural network whose activation function are of the radial basis type. This network output is made of a linear combination of radial basis functions of both inputs and neuron bias. Radial-basis function networks have vast application in real world problems such as function approximation, time series prediction, classification, and system control. RBFs were first formulated in by Broomhead and Lowe (1988).

A RBF neural network consists of three layers: an input layer, a hidden layer, and an output layer. Neurons at the hidden layer are activated by a radial basis function. The hidden layer consists of an array of computing units called hidden nodes. Each hidden node contains a center *c* vector that is a parameter vector of the same dimension as the input vector *x*. the Euclidean distance between the center and the network input vector x is defined by $||x(t)-c_j(t)||$.

According to Liu (2013) we can produce the output of hidden layer through a nonlinear activation function $h_j(t)$ as hence:

$$h_j(t) = \exp\left(\frac{\| x(t) - c_j(t) \|^2}{2b_j^2}\right), j = 1,\ldots,m \tag{3}$$

where b_j notes a positive scalar called a width and m notes the number of hidden nodes. The output layer is a linear weighted combination as follows:

$$y_i(t) = \sum_{j=1}^{m} w_{ji} h_j(t), i = 1,\ldots,n \tag{4}$$

where *w* are the output layer weights, n notes the number of outputs, and y notes the network output.

The weight value of RBF is

$$w = [w_1,\ldots,w_m]^T \tag{5}$$

The output of RBF neural network is

$$y(t) = w^T h = w_1 h_1 + w_2 h_2 + \ldots + w_m h_m \tag{6}$$

The control system is shown in Fig. 3.

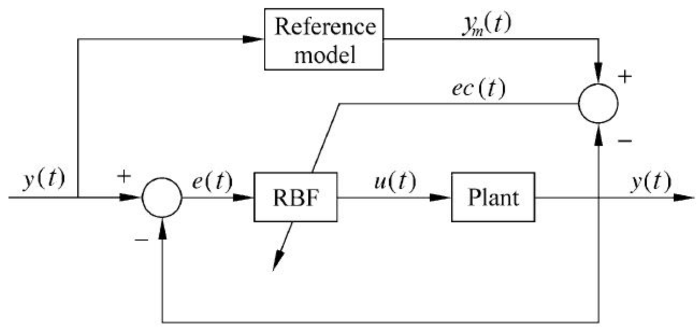

Figure 3. Block diagram of RBF-based model reference adaptive control system

The reference model is $y_m(t)$ then the tracking error is

$$e(t) = y_m(t) - y(t) \tag{7}$$

The criterion on function used here is *u(t)* thus:

$$E(t) = \frac{1}{2} ec(t)^2 \tag{8}$$

The controller is the output of RBF:

$$u(t) = h_1 w_1 + \ldots h_j w_j \ldots + h_m w_m \tag{9}$$

where *m* is the number of neural nets in hidden layer, w_j is weight value of net *j*, h_j is output of Gaussian function.
In RBF, $x=[x_1, \ldots, x_n]^T$ is the input vector, $h\ [h_1, \ldots, h_m]^T$; and h_j is the Gaussian function:

$$hj = \exp\left(-\frac{\|x - cj\|^2}{2b_j^2}\right) \tag{10}$$

Where $i=1,...,n$ and $j=1,...,m$. $b_j>0$, $cj=[c_{j1},...,c_{ji},...,c_{jn}]$, and $b=[b_1,...,b_m]^T$

The weight vector is

$$w = [w_1,..., w_m]^T \tag{11}$$

According to the steepest descent (gradient) method, the learning algorithm is as follows:

$$\Delta wj(t) = -\eta \frac{\partial E(t)}{\partial w} = \eta ec(t) \frac{\partial y(t)}{\partial u(t)} h_j$$
$$w_j(t) = w_j(t-1) + \Delta w_j(t) + \alpha \Delta w_j(t) \tag{12}$$

where η is learning rate, α is momentum factor, η∈[0,1]; and α∈[0,1]:

4. EXPERIMENTAL PROCEDURE

The test bench for data acquisition of this work used a DC motor with brushes, manufacturer Pololu, model 37Dx54L, 12V, 200 RPM, gear reduction box of 50: 1 whose readings were made with a sensor Encoder of quadrature 64 counts per The engine input. The control algorithms were loaded on a BeagleBone Black Rev.C and a VNH2SP30 model H bridge. The language chosen for implementation was Phyton.

4.1 System Identification

The identification of the system's transfer function was the same as (Malek et al, 2013). We applied a step response in the above test bench. The response received by the encoder is saved in a vector file. By means of comparison we try to identify this plant as a time delayed system both as with an integer and a fractional system as expressed in the Eq (13) and Eq(14) respectively.

$$P_{IO} = \frac{K}{Ts \ +1} e^{-Ls} \tag{13}$$

$$P_{FO} = \frac{K}{Ts^{\alpha} +1} e^{-Ls} \tag{14}$$

The next step is to identify the parameters K, T, L an α in these equations. So we take the error signal as

$$E(t) = \sum_{0}^{t_f} \left| y_p(t) - y_m(t) \right| \tag{15}$$

where $y_p(t)$ is the experimental data obtained by applying a step input to the plant, $y_m(t)$ is the step response of the considered models to the input and t_f is the final time for simulation and experimental results.

Applying the Inverse Laplace Transform we finally use the Scilab function fminsearch, that searches for the unconstrained minimum of a given cost function to find the values for parameters K, T, L and alfa. In Fig. 4 we can see the experimental data and both the system identifications response to the step input. It is clear that the fractional order system fits better with the data. So it is easily justifiable to opt for a fractional order PID controller for this system.

Table 1. Estimated Parameters.

	α	K	T	L
Integer Order Plant	1.00	77	0.25	0.01
Fractional Order Plant	0.816	0.80123	15.41	0.01

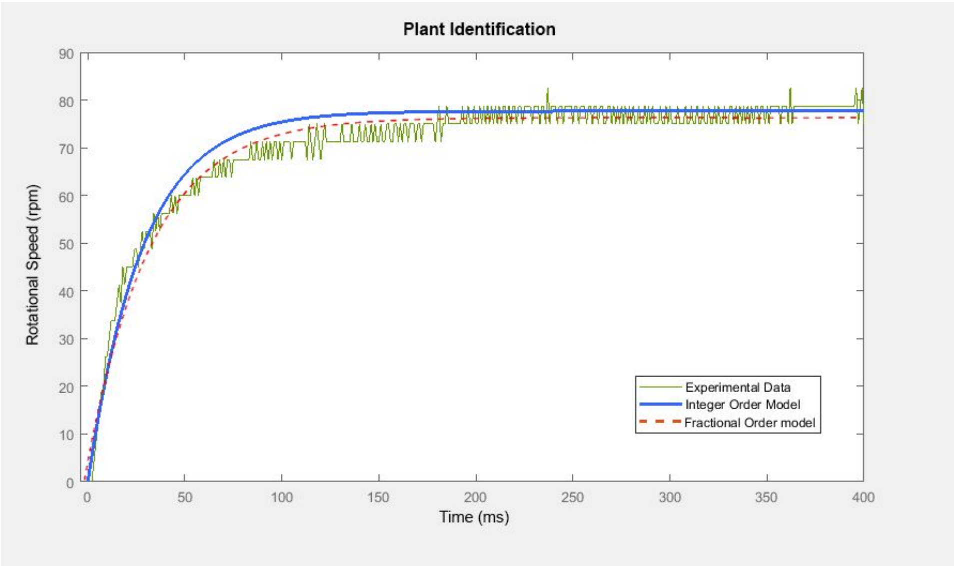

Figure 1. Integer and Fractional Order Models fit to the plants experimental response.

4.2 Controller Design

Considering the equation that bests describes this system is the of of the FO Plant, taking the values from Tab. 1 we have the following transfer functions in the time domain:

$$G(s)_{IO} = \frac{77}{0.25s+1} e^{-0.01s} \tag{16}$$

$$G(s)_{FO} = \frac{0.80123}{15,41s^{0.816}+1} e^{-0.01s} \tag{17}$$

From this values the previously described Radial Basis Function Network Algorithm was used for tuning of the FOPID. The comparison Integer Order PID was tuned both, with the Ziegler-Nichols tuning and the Genetic Algorithm. The comparison can be seen in Fig 5.

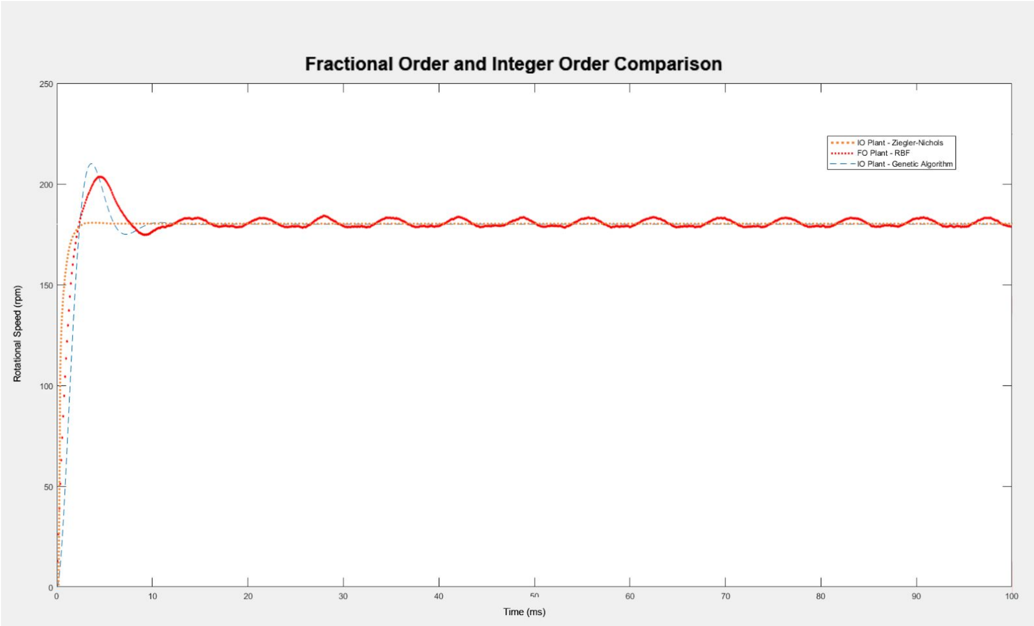

Figure 5.Comparison of output signals in the experimental results.

And as for the FOPID, the tune was made by Ziegler-Nichols using the tuning rules for FOPIDs introduced by Valerio and Costa (Valério et al, 2005) and also Genetic Algorithm.

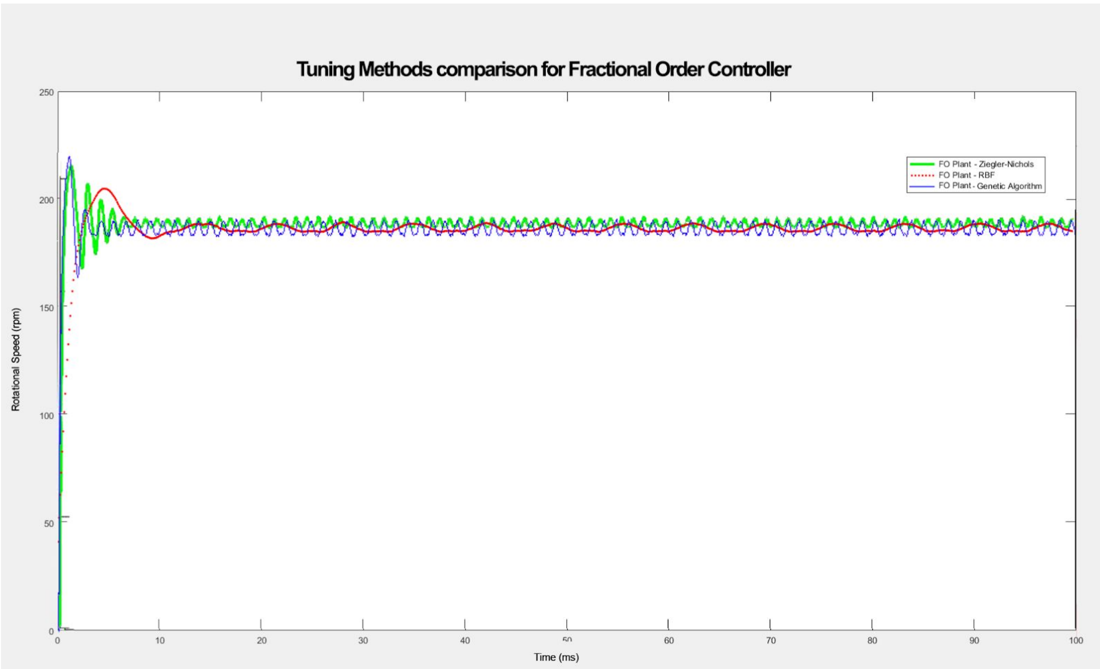

Figure 5.Comparison of output signals in the experimental results using different tuning methods for a Fractional Order Plant.

For the same plant were made measurements with different adjustments of PID controller parameters in order to find the best tuning to be used as reference when comparing with the FOPID controller. For all measurements the setpoint was chosen at 180 rpm because it presented a more stable motor response, according to factory specifications.

It is clear that for the test bench used in this experiment all of the designed controllers present a good response. The Integer Order PID even shows better results in the initial tests.

But on a second test with the use of unbalanced charge we can see undoubtedly that the fractional order controller tuned by a Radial Basis Function have better performance and respond to the disturbances in the system. Figure 8 shows the results both with the best performance PID tune and FOPID tuned by RBF .

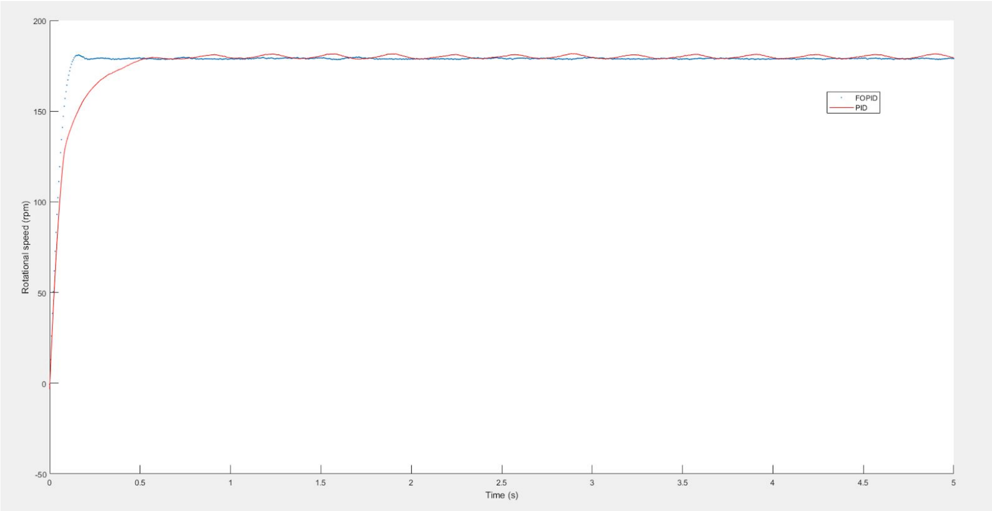

Figure 8. Comparison of output signals in the experimental results with unbalanced charge.

5. CONCLUSIONS

In this paper, a brushless DC motor test bench was identified as an single fractional order pole systems with constant time delay and a an integer order PID and a fractional order proportional integral controllers are designed for it. For a matter of performance and robustness performance the same system, both charge free and with an unbalanced charge, controller is tuned by Ziegler-Nichols IOPID, GA IOPID, Ziegler-Nichols FOPID, GA FOPID, and RBF FOPID. It has been shown that for a charge free system all of the techniques chosen can offer good results, but when presented with disturbances the RBF FOPID demonstrated the best results.
From the this experimental results, it is clear that the Radial Basis Function tuned fractional order controller work more effectively than the designed integer order PID controller. In further tests we will test different learning parameters to the neural network in order to improve the results in both studied situations.

.

6. ACKNOWLEDGEMENTS

The authors acknowledge to FAPEMIG Fundação de Amparo à Pesquisa for for the financial support and for encouraging the science and technology development in Minas Gerais.

7. REFERENCES

Ang, Kiam Heong, Gregory Chong, and Yun Li. "PID control system analysis, design, and technology." IEEE transactions on control systems technology 13.4 (2005): 559-576.

Baiyu, Ou, Lei Song, and Chunlei Chang. "Tuning of fractional PID controllers by using radial basis function neural networks." Control and Automation (ICCA), 2010 8th IEEE International Conference on. IEEE, 2010.

Broomhead, David S., and David Lowe. "Radial basis functions, multi-variable functional interpolation and adaptive networks." No. RSRE-MEMO-4148. Royal Signals and Radar Establishment Malvern (United Kingdom), 1988.

Cao J.Y. , Liang J. , Cao B.G. , "Optimization of fractional order PID controllers based on genetic algorithms", Proceedings of the 4th international conference on machine learning and cybernetics, pp. 5686-5689, 2005

Cao J.Y. , Cao B.G. , "Design of fractional order controllers based on particle swarm optimization", 1st IEEE conference on industrial electronics and applications, pp. 1-6, 2006.

Cervera J., Banos A. , Monje C.A. and Vinagre B.M. , "Tuning of fractional PID controllers by using QFT," IEEE industrial electronics, IECON 2006 32nd annual conference, 2006, pp. 5402-5407

Chen Y., Moore K.L., Vinagre B.M. , and Podlubny I. "Robust pid controller autotuning with a phase shaper. "In Proceedings of The First IFAC Symposium on Fractional Differentiation

and its Applications (FDA04), Bordeaux, France, 2004

Hunt KJ, Sbarbaro D, Zbikowski R, Gawthrop PJ (1992) "Neural networks for control system—a survey." Automatica 28(6):1083–1112

Liu, Jinkun. Radial Basis Function (RBF) neural network control for mechanical systems: design, analysis and Matlab simulation. Springer Science & Business Media, 2013.

Malek, Hadi, Ying Luo, and YangQuan Chen. "Identification and tuning fractional order proportional integral controllers for time delayed systems with a fractional pole." Mechatronics 23.7 (2013): 746-754.

McCormick,C. "Radial Basis Function Network (RBFN) Tutorial". Available in: <http://mccormickml.com/2013/08/15/radial-basis-function-network-rbfn-tutorial/>. 10/05/2017

McCulloch WS, Pitts W (1943) "A logical calculus of the ideas immanent in nervous activity." Bull Math Biophys 5:115–133

Monje C.A., Vinagre B.M., Chen Y.Q. , Feliu V., Lanusse P., Sabatier J., " Proposals for fractional PI λ D μ tuning " in Fractional Differentiation and its Applications, 2004

Monje, C.A., et al. "Tuning and auto-tuning of fractional order controllers for industry applications." Control engineering practice 16.7 (2008): 798-812.

Petras I. "The fractional-order controllers: methods for their synthesis and application", J. of Electrical Engineering, vol. 50, 1999, no. 9-10, pp. 284–288.

Podlubny, I., Dorcak, L., Kostial, I. 1997. "On Fractional Derivatives, Fractional-Order Dynamic Systems and PIλDμ controllers, Proceedings of the 36th Conference on Decision & Control, San Diego, California, USA, December 1997"

Podlubny, I. "Fractional-order systems and PIλDμ-controllers." IEEE Transactions on Automatic Control, vol. 44, no. 1, 1999, pp. 208–214

Schwenker, Friedhelm, Hans A. Kestler, and Günther Palm. "Three learning phases for radial-basis-function networks." Neural networks 14.4 (2001): 439-458.

Skogestad, Sigurd. "Simple analytic rules for model reduction and PID controller tuning." Journal of process control 13.4 (2003): 291-309.

Valério, Duarte, and José Sá da Costa. "Ziegler-Nichols type tuning rules for fractional PID controllers." Proceedings of IDETC 2005 (ASME) (2005): 1431-1440.

Yu, Cheng-Ching. "Autotuning of PID controllers: A relay feedback approach." Springer Science & Business Media, 2006.

8. RESPONSIBILITY NOTICE

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