SELECTION OF COMPUTATIONAL INTELLIGENCE METHODS DESIGNED FOR APPLICATION IN PNEUMATIC MUSCLE ACTUATOR

MARIA TOTHOVA, MONIKA TROJANOVA

Technical University of Kosice, Faculty of Manufacturing Technologies with a seat in Presov, Slovak Republic

DOI : 10.17973/MMSJ.2017_12_201748

e-mail : maria.tothova@tuke.sk

Pneumatic muscle actuators have the highest power/weight ratio of any actuator and therefore it has a broad application prospect in soft robotics. This paper presents a two degree of freedom (2-DOF) pneumatic muscle actuator that consists of four pneumatic artificial muscles (PAMs) and two rotation joins. However, this pneumatic muscle actuator has highly nonlinear and hysteretic properties (among muscle force, muscle displacement and pressure in the muscle), which lead to difficulty in accurate position control. Pneumatic muscle actuator usually necessitate the use various nonlinear techniques for control in order to improve their performance. The paper contains selection of computational intelligence methods designed for application in pneumatic muscle actuator. There are described and compared MLP neural network and RBF neural network.

KEYWORDS

pneumatic muscle actuator, artificial muscle, antagonistic connection, intelligence method, neural network

1 INTRODUCTION

Pneumatic artificial muscles (PAMs) due to their unique properties are used for constructors such as actuator in the proposals to various types of machineries and equipment. The use have found for example in the automotive industry, the manufacture of consumer electronics and medicine [Zidek 2012]. An interesting application is their use in so-called exoskeleton [Chang 2010], [Fodor 2010], i.e. external skeleton that can serve handicapped people or soldiers. When used in industry with the assumption of the minimal contact manipulators with humans [Hopen 2005], it is possible to use the maximum stiffness of the antagonistic connection of PAMs under all operating conditions. In the domain of humanoid robots and manipulators with the need for fine attachment, the lower stiffness (higher flexibility), respectively control of stiffness/flexibility will be used [Hosovsky 2016a].

Muscles working medium is a gas, usually modified compressed air. Gas is not inherently the preferred source of energy. Change its properties (state variables) during the working process is a source of many problems in precise positioning [Oliver-Salazar 2017]. Ideally, the behavior is described by the equation [Kundu 2004], [Riccardi 2012] of an ideal gas:

$$p = \rho \cdot R \cdot T\,, \tag{1}$$

where p is the pressure in the muscle, ρ is the density and T is the thermodynamic temperature gas. All these state variables are the time dependencies. Even individual gas constant R is not really constant. Its value depends on the current gas humidity.

It is therefore evident that the control of pneumatic systems may not be a trivial matter. Mostly classical conventional control systems with controllers with fixed parameters may not always have adequate quality control process. In this case, regulators are used, which react quickly enough to the change the properties of the controlled system [Bukovsky 2012]. Therefore the computational intelligence methods are may be used. PID parameters are tuned online using RBF neural network for a 1-DOF manipulator actuated in [Zhao 2015]. A nonlinear PID controller for 2 axes manipulator based on PAM in [Thanh 2006] combines the conventional PID controller and the neural network.

2 PNEUMATIC MUSCLE ACTUATOR

PAM is contractile device that acts by force only in one direction (its length is increased or decreased). In order to have a two direction actuated revolute joint, two PAMs have to be used (an antagonistic connection).

An approximate mathematical model that describes muscle characteristics (muscle contraction, muscle force and pressure in the muscle) can be found in [Pitel 2014], [Tothova 2013]. Under the assumption of zero wall thickness of the muscle, the force exerted by the artificial muscle is given by

$$F = p \cdot l_0 \cdot f_{t0}\left(\varepsilon, l_0 / r\right), \tag{2}$$

where p is the pressure, l_0 is the muscle length (or the maximum length), r is the radius of the muscle, ε is the muscle contraction, f_{t0} is the dimensionless nonlinear function that depends on muscle contraction and the time parameter l_0/r (called the slenderness) [Van Damme 2008].

Two degree of freedom (2-DOF) manipulator actuated by pneumatic muscle actuator with two rotary axis is shown in Fig. 1. The main part of the actuator is formed by four artificial muscles. Every muscle in a pair of muscles (muscles 1 and 2 or muscles 3 and 4) is connected in antagonistic connection through the chain gear. At the end of the actuator an arm with screwed weight is attached [Hosovsky 2016b]. Then the position of the actuator arm (rotation angle φ of actuator) is determined by equilibrium of forces in the first and second pair of muscles through the chain to the roller according to different pressures in each muscle. Inflation and deflation of the muscles is controlled by four ON/OFF twin-solenoid valves (not shown). The compressed air is supplied into the muscles in a form of pressure impulses.

From the knowledge of the kinematic parameters (i.e. point locations of the joins) and actuator parameters, it can be determined the torque characteristics of both joints. Using the nonlinear relation of muscle force, pressure in the muscle and muscle contraction, the torque generated by a artificial muscle can be written as [Van Damme 2008]

$$\tau = p \cdot f(\gamma), \tag{3}$$

where $\gamma = \varphi_1$ for muscles 1 and 2 and $\gamma = \varphi_2$ for muscles 3 and 4. The total torque (in both joints) of pneumatic muscle actuator can be [Van Damme 2008]:

$$\tau = \begin{bmatrix} p_1 \cdot f_1(\varphi_1) + p_2 \cdot f_2(\varphi_1) \\ p_3 \cdot f_3(\varphi_2) + p_4 \cdot f_4(\varphi_2) \end{bmatrix}, \quad (4)$$

where p_i (i = 1,…,4) is the air pressure in the muscle i and f_i is the torque function associated with that muscle.

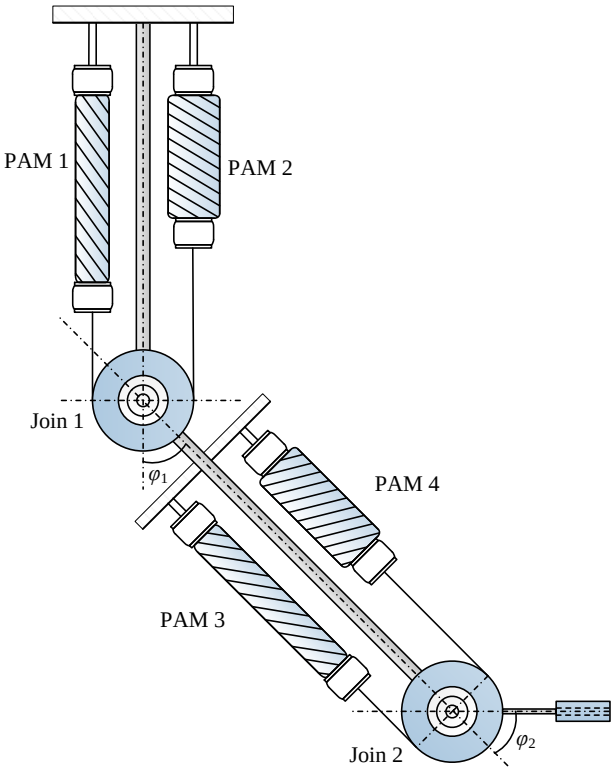

Figure 1. Working principle of 2-DOF pneumatic muscle actuator

3 ARTIFICIAL NEURAL NETWORKS FOR STATIC MODELS

Artificial neural network (ANN) has been motivated by the structures in the brain of human, because they are proper for processing, learning and adaptation of difficult information. Neural network (NN) is a form of artificial intelligence and it can be characterized as a network where all occurring functions are of the same type. Neural network is made of elements called neurons. The general model of the neural network consists of functions that are linked by weights. The network consists of input neurons on the input layer, of output neuron on the output layer and of units on hidden layers. It is not necessary for the neural network to have the hidden layers. Their number is more dependent on the complexity of the problem studied. The input layer receives impulses (signals) from the environment as well as from other neurons. Hidden layers are tasked to process the inputs. The output layer is used to complete statement of the output neuron (in numerical form). Fast implementation of neural networks is ensured by their special characteristics, which provides many advantages. For such characteristics are considered for example that the units are highly parallel and strongly connected, units are resistant to failure and network learns from data [Abraham 2005], [Jain 1996].

Type of neural networks:

- Single-layer Perceptron Networks,
- Multilayer Perceptron Network (MLP),
- Radial Basis Function Network (RBF).

Single-layer perceptron networks can only classify the input objects and assign them into the classes. They represent the neurons with jump activation functions, which are grouped into a single layer. They classify only linearly separable problems.

3.1 Multilayer Perceptron Network (MLP)

MLP is considered to be the most famous architecture of neural networks, and therefore it is widely used. It is a feedforward network with at least one hidden layer. The feedforward attribute means that the direction of data flow is from input to output. The network is learning by the back-propagation algorithm. Algorithm considers a mistake, which will be promoted in the opposite direction; it means from output to input. The basic construction unit of the neural network is a neuron. The neuron of the hidden layer of the multi-layer feedforward network is shown in Fig. 2 [Nelles 2001], [Haykin 2001].

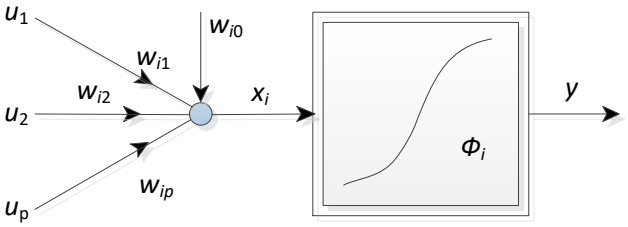

Figure 2. A perceptron of a multilayer perceptron network

Such an individual neuron is called the perceptron. Perceptron summarizes the input u_1 - u_p that assigns weights of connection w_{i1} - w_{ip} before summarization. Each neuron has a certain sensitivity w_{i0}. X_i expresses the internal activity of the neuron. Inputs are transformed to the output y by using activation function Φi. Typical activation functions for MLP neuron are defined on intervals <0;1> for sigmoid [Nelles 2001]:

$$\text{logistic}(x) = \frac{1}{1+\exp(-x)}, \quad (5)$$

or <-1;1> for hyperbolic tangent:

$$\tanh(x) = \frac{1-\exp(-2x)}{1+\exp(-2x)}. \quad (6)$$

The parameters that appear in the MLP networks are of two types. Linear parameter shows the weight of the output layer. They define the functions of the amplitudes and working point. Nonlinear parameter shows the weight of the hidden layer and determines the position, direction and slope of the basic functions. MLP network is obtained by linking several perceptron neurons, which are used in parallel and they are connected to neuron in the output layer. The function of the MLP network is to approximate every function of non-linear character. Basic diagram of MLP network is shown in Fig. 3 [Nelles 2001].

As for the neuron itself, so for the scheme MLP network and mathematical model as well is valid that u_j are the inputs, w_{ij} are weights on the hidden layer, w_i are weights of output neuron, Φi is activation function and y is the output. For MLP network can be determined the total number of network parameters. This is dependent on the number of neurons in the hidden layer and the number of inputs. The mathematical formulation for this kind of network is shown as [Nelles 2001]:

$$y = \sum_{i=0}^{M} w_i \phi_i \left(\sum_{j=0}^{P} w_{ij} u_j \right). \tag{7}$$

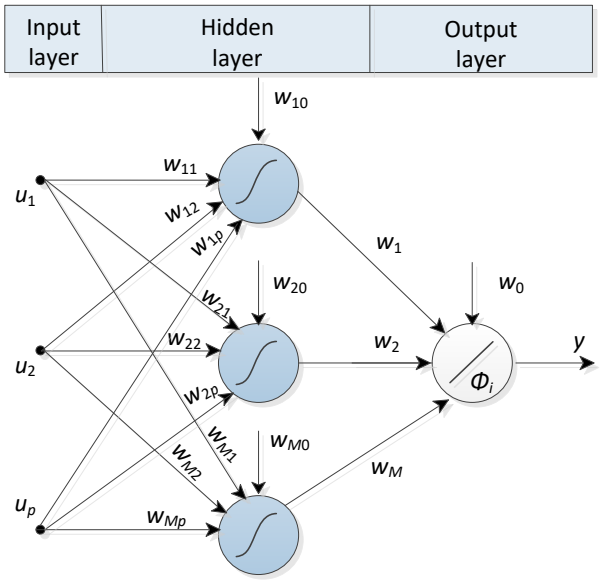

Figure 3. A multilayer perceptron network

3.2 *Radial Basis Function Network (RBF)*

RBF network is a type of neural network, which is using the approximation of the radial basis functions. Compared to the MLP neural network this type of the network function does not affect the result in global scale, only in local. RBF is in contrast with MLP network. RBF is feedforward network, but only with one hidden layer. RBF network can handle even more complex modelling, where MLP networks for similar tasks require several hidden layers. In Fig. 4 can see a neuron of RBF network. Its basic activity can be divided into two tasks. The first task is to formulate scalar distance x_i between the input vector $\underline{u}$ and the central vector $\underline{c}_i$. Second task is transformation of scalar distance to the resulting linear function y using non-linear activation function Φi [Nelles 2001], [Haykin 2001].

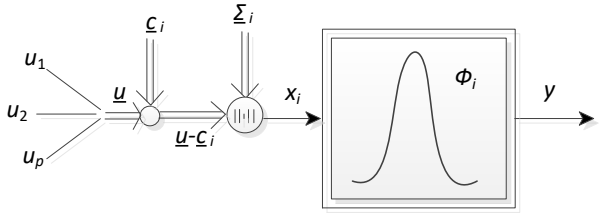

Figure 4. A neuron of a radial basis function network

The radial activation functions are considered more mathematical functions, but the mostly used are in particular Gaussian function (also the activation function for the neuron in Fig. 4 is the Gaussian function [Haykin 2001]), Cauchy function and inverse function MultiCriteria:

$$\phi(x) = \exp\left(-x^2\right), \tag{8}$$

$$\phi(x) = \frac{1}{1+x}, \tag{9}$$

$$\phi(x) = \frac{1}{\sqrt{1+x^2}}. \tag{10}$$

The scheme of RBF neural network is shown in Fig. 5. Every perceptron of each layer carries the basis function. In radial basis function network there are three types of parameters. Weights w_i to output layer are linear parameters that are determining the significance of basis function and output value. Centers $\underline{c}_i$ are the parameters of the hidden layer, which are reflecting the position basis function. Another parameter of the hidden layer is the standard deviation, which determines the rotation and height of the basis function. In the given scheme it is presented as $\underline{\Sigma}_i$ (norm matrix). Both of these parameters, $\underline{c}_i$ and $\underline{\Sigma}_i$ are non-linear [Nelles 2001].

The mathematical formulation of the RBF neural network for Gaussian activation function is

$$y = \sum_{i=0}^{M} w_i \phi_i \left(\left\| \underline{u} - \underline{c}_i \right\|_{\underline{\Sigma}_i} \right). \tag{11}$$

The total numbers of parameters depend on the number of input parameters and their flexibility, on the number of output weights and the number of coordinates of the center [Nelles 2001].

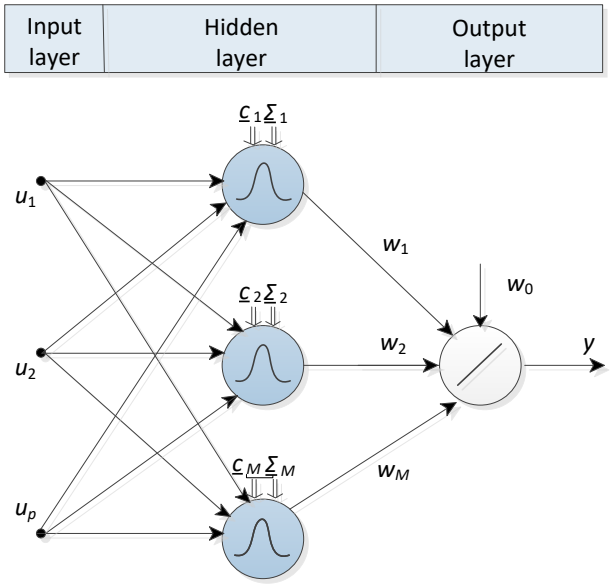

Figure 5. A radial basis function network

3.3 *Comparing of the properties of MLP and RBF networks*

In previous subsections were approached two types of the neural networks (MLP and RBF). These networks differ in many attributes. Some basic properties are shown and compared in Tab 1. The network MLP has a lot of very favourable characteristics compared to the RBF network, but RBF network gives many positives too. It provides the ability to a local approximation and the training speed for local optimalization is favourable [Nelles 2001].

Moreover, the networks vary in the number of hidden layers. RBF network has just one hidden layer, while MLP network may have one or more hidden layers (the number depends on the complexity of the problem solved). MLP network uses one type of neuron, whereas the RBF network uses different type neuron for hidden layer and for the output (which varies not only in type but also in function). Hidden and output layer for the MLP network is usually non-linear, while the hidden layer of RBF network is non-linear, but the output layer is linear. The output layer of the MLP network is only rarely linear. MLP network

uses for learning phase supervised learning, RBF network unsupervised learning. Fundamental difference is the extent of approximation. MLP approximates in global, while RBF only locally. There is also a difference in the activation function between these two networks. The activation function of the MLP determines the scalar product of the input and respective weight. The activation function of RBF defines the distance between the input and the central value of the function [Kvasnicka 1997].

Table 1. Some properties of MLP and RBF networks and their possible effect

Properties	RBF	MLP
Locality	very favourable	undesirable
Accuracy	neutral	very favourable
Smoothness	neutral	very favourable
Sensitivity to noise	favourable	very favourable
Training speed	favourable*/ very undesirable**	very undesirable
Evaluation speed	neutral	favourable

(* for linear optimalization, ** for non-linear optimalization)

4 NEURAL NETWORK TOOLBOX

Neural network toolbox in MATLAB environment provides algorithms, functions and apps to create, train, visualize and simulate neural networks [MathWorks 2017]. Neural network diagram showing in Fig. 6 will be used for design of the neural networks.

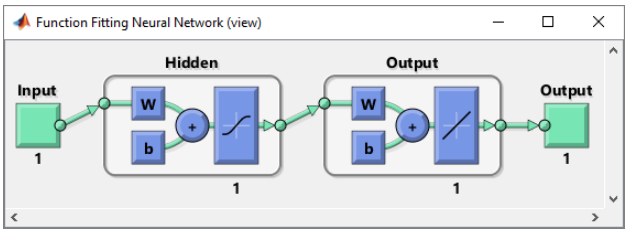

Figure 6. Neural network diagram

Simple equations are created for the structure of the neural network MLP and RBF in MATLAB. These equations provide knowledges about inputs necessary for the compilation of the basic structure.

For MLP neural network structure applies:

$$net = newff(I,O,NHL,AFHL,AFOL,MT,ML,E), \quad (12)$$

where *newff* is command for feedforward neural network, *I* is inputs, *O* is outputs, *NHL* is number of hidden layer, *AFHL* is an activation function for hidden layers (often used 'logsig' or 'tansig'), *AFOL* is an activation function for output layer ('purelin'), *MT* is method of training ('traingdx' or 'trainlm'), *ML* is method of learning ('learngdm') and *E* is error function ('mse') [Foltin 2008].

For RBF neural network structure applies:

$$net = newrb(I,O,EEL,CS,MN,NN), \quad (13)$$

where *newrb* is the command for radial basis neural network, *I* is inputs, *O* is outputs, *EEL* is an ended error of learning (often used '0'), *CS* is the coefficient of the stretching *RB* functions (often used '1'), *MN* is the maximal number of neurons and *NN* is the number of neurons between showing.

Three neural network architectures can be introduced in neural network toolbox for prediction and control in two steps (system identification and control design) when using neural networks [MathWorks 2017]:

- Model Predictive Control - the controller uses a neural network model to predict future plant responses and an optimization algorithm is used to select the control input that optimizes future performance,
- NARMA-L2 (or Feedback Linearization) Control - the controller is simply a rearrangement of the system model,
- Model Reference Control - the controller is a neural network that is trained to control a system so that it follows a reference model (the neural network system model is used to assist in the controller training).

The blocks for three neural network architectures in MATLAB environment are shown in Fig. 7 and their advantages and disadvantages for neural network control are compared in Table 2.

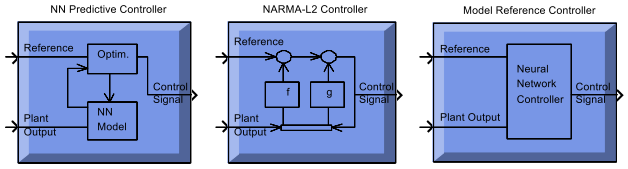

Figure 7. Blocks for neural network architectures

Table 2. Some advantages and disadvantages of neural network architectures

	Advantages	Disadvantages
Model Predictive Control	an optimization algorithm computes the control signals; is trained offline, in batch form	the controller requires a amount of online computation
NARMA-L2 Control	the controller requires the least computation; is trained offline, in batch form	the system must either be in companion form, or be capable of approximation by a companion form model
Model Reference Control	the online computation is minimal; model reference control applies to a larger class of system	a separate neural network controller must be trained offline; the controller training is computationally expensive

5 CONCLUSIONS

2-DOF pneumatic muscle actuator with two rotation joins (showing in Fig. 1) consisting of four artificial muscles in antagonistic connection was designed and realized in order to use in manufacturing technologies and non-productive sphere. Most restrictions on the use of this pneumatic muscle actuator are to achieve the positioning accuracy which is within ±1 mm. This positioning accuracy does not allow the use these

actuators in manipulators with required high precision of the manipulation. Improving the positioning accuracy can be achieved by application of advanced control algorithms for example using methods of computational intelligence.

This paper presents and compares two neural networks: MLP and RBF. Multilayer perceptron network (MLP) usually comprises a continuous non-linear activation function and they are able to approximate any nonlinear transformation. Radial basis function network (RBF) contains three layers of neurons and the hidden layer realizes a nonlinear transformation.

Since in the further research, the practical comparing of the various computational intelligence methods for application in the model of pneumatic muscle actuator in MATLAB environment is expected.

ACKNOWLEDGMENTS

The research work is supported by grant VEGA 1/0822/16 *„Research of Intelligent Manipulator Based on Pneumatic Artificial Muscles with Three Degrees of Freedom"*.

REFERENCES

[Abraham 2005] Abraham, A. Artificial neural networks. Handbook of measuring system design, June 2010, Vol.2, pp 901-908. ISBN 9780470021439

[Bukovsky 2012] Bukovsky, I. Nonconventional Neural Architectures and their Advantages for Technical Applications. Faculty of Mechanical Engineering. Czech republic, Praque: Czech Technical University in Praque, 2012, ISBN 978-80-01-05122-1

[Chang 2010] Chang, M. K. An adaptive self-organizing fuzzy sliding mode controller for a 2-DOF rehabilitation robot actuated by pneumatic muscle actuators. Control Engineering Practice, January 2010, Vol.18, Issue 1, pp 13-22

[Foltin 2008] Foltin, M. and Körösi, L. Matlab (6): neural networks toolbox. AT&P journal, September 2008, Vol.2008, No.9, pp 93-94. ISSN 1336-233X

[Fodor 2010] Fodor, M., et al. New trends in application of artificial muscles for automation devices used in non-productive sector. Manufacturing Engineering, 2010, Vol.9, Issue 4, pp 78-80

[Haykin 2001] Haykin, S. Neural networks: a comprehensive foundation. India, Delhi: Pearson Education, 2001, p. 823. ISBN 81-7808-300-0

[Hopen 2005] Hopen, J. M. and Hosovsky, A. The servo robustification of the industrial robot. Annals of DAAAM & Proceedings, January 2005, pp 161-162

[Hosovsky 2016a] Hosovsky, A. et al. Dynamic characterization and simulation of two-link soft robot arm with pneumatic muscles. Mechanism and Machine Theory, April 2016, Vol.103, pp 98-116

[Hosovsky 2016b] Hosovsky, A., Pitel, J. and Zidek, K. Analysis of hysteretic behavior of two-DOF soft robotic arm. MM Science Journal, September 2016, Vol.18, Issue 1, pp 13-22

[Jain 1996] Jain, A. K. et al. Artificial neural networks: a tutorial. Computer, March 1996, Vol.29, No.3, pp 31-44. ISSN 0018-9162

[Kundu 2004] Kundu, P. and Cohen, I. Fluid mechanics (third edition), Elsevier Academic Press, London, 2004, p. 759. ISBN 9780080470238

[Kvasnicka 1997] Kvasnička, V. et al. Introduction to the theory of neural networks. Slovakia, Bratislava: IRIS, 1997, p. 237. ISBN 9-788-088-77830-1

[MathWorks 2017] MathWorks. Design Neural Network Predictive Controller in Simulink. 2017 [online]. February 2017 [date of citing]. https://www.mathworks.com/help/nnet/ug/design-neural-network-predictive-controller-in-simulink.html

[Nelles 2001] Nelles, O. Nonlinear system identification: from classical approaches to neural networks and fuzzy models. Germany, Berlin: Springer, 2001, p. 786. ISBN 3-540-67369-5

[Oliver-Salazar 2017] Oliver-Salazar, M.A.et al. Characterization of pneumatic muscle and their use for the position control of a mechatronic finger. Mechatronics, 2017, Vol.42, pp 25-40

[Pitel 2014] Pitel, J. and Tothova, M. Dynamics of pneumatic muscle actuator: measurement and modeling. In: Proceedings of the 15th International Carpathian Control Conference (ICCC 2014), May 28-30, 2014, Danvers: IEEE, pp 432-436

[Riccardi 2012] Riccardi, L. et al. A precise positioning actuator based on feedback-controlled magnetic shape memory alloys. Mechatronics, August 2012, Vol.22, Issue 5, pp 568-576

[Thanh 2006] Thanh, T. D. C. and Ahn, K. K. Nonlinear PID control to improve the control performance of 2 axes pneumatic artificial muscle manipulator using neural network. Mechatronics, November 2006, Vol.16, Issue 9, pp 577-587

[Tothova 2013] Tothova, M. and Pitel, J. Simulation of actuator dynamics based on geometric model of pneumatic artificial muscle. In: Proceedings of the 11th International Symposium on Intelligent Systems and Informatics (SISY 2013), September 26-28, 2013, Subotica: IEEE, pp 233-237

[Van Damme 2008] Van Damme, M.,Vanderborght, B.and Beyl, P. et al. Sliding mode control of a "soft" 2-DOF planar pneumatic manipulator. International Applied Mechanics, October 2008, Vol.44, Issue 10, pp 1191-1199

[Zhao 2015] Zhao, J., Zhong, J. and Jizhuang, F. Position control of a pneumatic muscle actuator using RBF neural network tuned PID controller. Mathematical Problems in Engineering, March 2015, Vol.2015, pp 1-16

[Zidek 2012] Zidek, K., et al. Rehabilitation device construction based on artificial muscle actuators. In: Proceedings of the 9th IASTED International Conference: Biomedical Engineering (BioMed 2012), February 15-17, 2012, Innsbruck: IEEE, pp 855-861

CONTACTS:

Ing. Maria Tothova, PhD.
Ing. Monika Trojanova
Technical University of Kosice
Faculty of Manufacturing Technologies
Department of Mathematics, Informatics and Cybernetics
Bayerova 1, 080 01 Presov, Slovak Republic
+421 055 6026 420, maria.tothova@tuke.sk,
monika.trojanova@tuke.sk, http://www.fvt.tuke.sk