2010 IEEE-RAS International Conference on Humanoid Robots
Nashville, TN, USA, December 6-8, 2010

Design and Optimization of a Fuzzy-Neural Hybrid Controller for an Artificial Muscle Robotic Arm Using Genetic Algorithms

Erdem Erdemir, *Member, IEEE,* Mehmed Özkan, *Member, IEEE,* Kazuhiko Kawamura, *Fellow, IEEE,* D. Mitchell Wilkes, *Member, IEEE,* Murat Fırat and Ali Polat

***Abstract*—Humanoids are increasingly used in the service sectors around the world to work with, or assist humans. However current humanoid designs place limitations on direct engagement with the human in terms of safety and usability. In this paper, we present an approach for the control of hybrid, high-speed and safe human-robot interaction systems with highly non-linear dynamic behavior. The proposed approach comprises the three soft computing techniques, namely back propagation neural network, fuzzy and genetic algorithms. This open-loop controller was applied to a Bridgestone Hybrid Robot Arm (BHRA). BHRA has three electric motors and four artificial muscles, arranged in an agonist/antagonist, and opposing pair configuration, that drive the five-degrees of freedom of the robot arm. The behaviors of the artificial muscles are observed under the effects of the links driven by the electric motors and it is shown that the proposed biologically-plausible controller could produce more accurate trajectories at higher speeds when compared to conventional PID and stand alone or combined versions of Neural Network and Fuzzy controllers.**

I. Introduction

Humanoid robots are expected to perform a variety of service tasks in increasingly challenging environments. One of the problems in present service robots lies in the mechanical design of robot platforms. Recently an alternative design strategy called "musculoskeletal robot design strategy" has been proposed as a next-generation of safe and energy-efficient humanoid robot design [1]. This paper is to explore a mechanically and biologically inspired musculoskeletal model of human-like arm which can be implemented in a humanoid robotic platform [2].

Artificial muscles (Rubbertuators, Rubber-Actuators or Pneumatic Muscles) are used to provide a relatively safe human-robot interaction for manipulator, automation and robotic tasks. Lightweight, high power/weight ratio and compliant nature make artificial muscles advantageous over classical motor actuators especially in human-robot coexisting environments. On the other hand, it is very difficult to control artificial muscle systems because of the high compressibility of air, poor damping ability, the strong nonlinearity, the time lag of valve operation, the change in the shape and the temperature of artificial muscles with use, and hysteresis [3][4][5][6]. Accordingly, a number of control strategies addressing some of these difficulties have been introduced in the last two decades. Control techniques based on PID control [7], adaptive control [8], nonlinear optimal predictive control [9], neural network control [6], fuzzy-PID control [10], sliding mode control [3] and frequency modeling and control [11] could offer solutions to the control of highly nonlinear pneumatic systems. One of the novelties of this paper is to develop a bumpless open-loop controller for artificial muscles that work together with motor-driven links at very high speeds and challenging dynamic environments.

The proposed approach has its roots in the biological functioning of the human brain, specifically the neural activity of the primary motor cortex which encodes the magnitude, the direction of movement and force in the physical space on a single-neuron level. Neural activity in the primary motor cortex encodes movement direction and the forces on a single-neuron level [12]. For instance, individual cells in the motor cortex possess directional preference, that is, while some cells fire most during a movement to a particular direction, others fire most during movement towards another direction [13]. Hence, the frequency of discharge is a sinusoidal function of the movement direction [14], i.e. the response of each cell is represented as a direction vector, which points to the preferred direction for the neuron. Moreover, it is possible to reconstruct the muscle tensions, which includes the information of the magnitude and direction of force, from the neural activity of 105 neurons in the primary motor cortex related to the arm as did [12].

In this research, simple neural networks are used to encode movement direction and the forces on a single-neuron level. A fuzzy-neural hybrid controller for high-speed hybrid robot arm BHRA, shown in Figure 1, was developed to work with Humans. The neural network size is limited to 3 nodes and one hidden layer, and a simple fuzzy algorithm with minimal linguistic variables and minimal number of rules was employed. The goal is to combine these two soft computing approaches to achieve a biologically-inspired, fast accurate and safe control of BHRA. The

Erdem Erdemir, Kazuhiko Kawamura and D. Mitchell Wilkes are with the Department of Electrical Engineering and Computer Science, Vanderbilt University, Nashville, TN 37235, USA
(Emails: kawamura@eecsmail.vuse.vanderbilt.edu , mitch.wilkes @vanderbilt.edu, erdem.erdemir@vanderbilt.edu).

Mehmed Ozkan is with the Institute of Biomedical Engineering, Bogazici University, 34342 Bebek, Istanbul, Turkey (Email: mehmed@boun.edu.tr).

Murat Fırat is with the Department of Mathematics and Computer Science Eindhoven University of Technology (Email: m.firat@tue.nl)

Ali Polat is with the Department of Systems and Control Engineering, Bogazici University (Email:aliplt@yahoo.com)

classical motors maintain a high accuracy and power for the arm; on the other hand, the artificial muscles support the complaint nature of the arm.

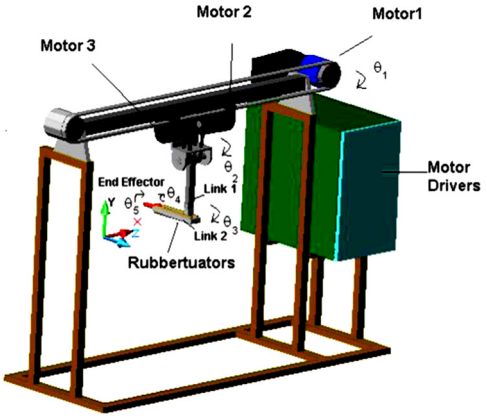

Figure 1: Bridgestone Hybrid Robot Arm (BHRA). The first joint (θ_1) is a linear actuator and powered by Motor1, which moves the arm in X-direction. The second joint (θ_2) is a rotary actuator and powered by Motor2, which moves the Link 1 directly. The third joint (θ_3) is a rotary actuator and powered by Motor3, which moves the Link 2 using a chain. The fourth joint (θ_4) is a rotary artificial muscles (rubbertuators) driven joint and moves the end-effector (in red) in Y-Z plane. The fifth joint (θ_5) is a rotary artificial muscle driven joint and rotates the end-effector around Z-axis.

As discussed in [15], artificial neural networks (ANNs) offer good performance in dealing with inputs with various ranges. While ANN learns the behavior of highly nonlinear systems, fuzzy logic techniques deal with issues on a higher level, noise tolerance, and higher expressive power. However, since fuzzy systems do not have much learning capability, it is difficult to find and tune the fuzzy rules and membership function. The internal layers of a neural network are always opaque to the user, sometimes called "black box", and the mapping rules in the trained network are difficult to understand. Furthermore the convergence of learning, in neural networks, is usually very slow and not guaranteed. A neural network with more hidden layers and more nodes was tried to output adequate pressure signals to artificial muscles. However, the learning performance and the control performance were worse due to its local minimums, especially close to the limits of the workspace. In this study, to benefit from the merits and overcome the demerits of neural networks and fuzzy rules, the two methods are merged and a cost function is optimized by adjusting the linguistic variables of the fuzzy logic through the use of genetic algorithms.

II. The Fuzzy-Neural Hybrid Control Structure

In the fuzzy-neural hybrid control structure of BHRA, the physical model of the robot dynamics is modeled with a multi-layer back propagation neural network system. The nonlinear joint trajectories are divided into segments in 4-degree increments as shown in Figure 2, and a dedicated neural network is trained for each segment. The input output relation of a neural network at this stage is not totally "black-box", but rather a physical model defines the input structure for the desired output. Another fact, that has to be taken into account, is the hysteresis properties of the artificial muscles. In order to overcome hysteresis two neural networks are used for forward and backward direction of the motion. The discrete consecutive neural networks are combined by using a fuzzy algorithm to achieve an accurate but smooth control of BHRA. The rules of the fuzzy sets are refined by genetic algorithm.

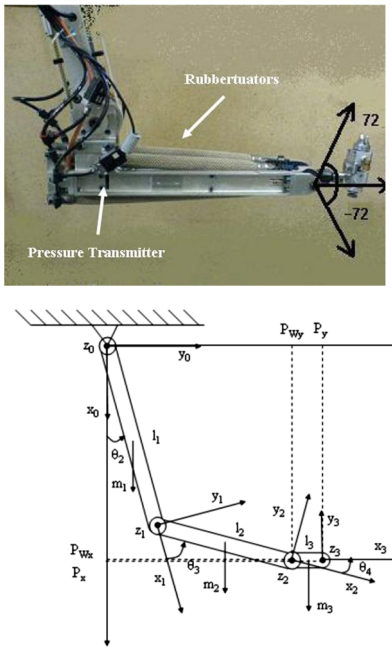

Figure 2: The reach angle and the coordinate frame attachment to the three-link planar manipulator

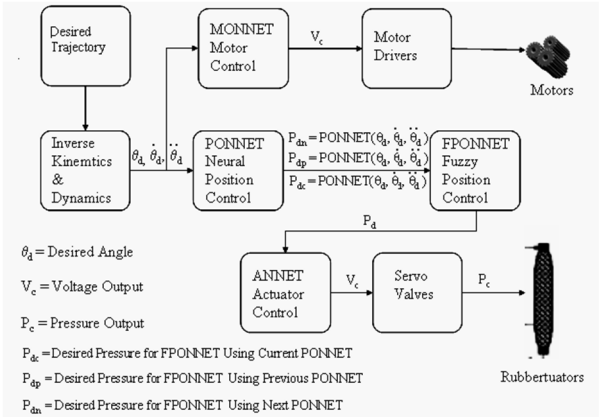

Figure 3: BHRA Open Loop Control Structure

The BHRA control loop comprises two trained neural network, PONNET (Position Control Neural Network) and ANNET (Actuator Control Neural Network), as [6]. In addition there is the third Neural Network layer, FPONNET (Fuzzy Position Control Neural Network) for each artificial muscle and a fourth one is the MONNET (Motor Control Neural Network) needed for each motor of the hybrid arm, shown in Figure 3.

The desired trajectory is the input of the control system. The inverse kinematics and dynamics compute the desired

angular positions, velocities and accelerations for each joint. A PONNET receives the results of the inverse kinematics for its joint and produces a pair of pressure values corresponding to the desired degree of the joint. The output of the neural network, belonging to the previous degree-segment is also computed and the results of two PONNETs are combined by FPONNET to compute the desired pressure by combining the two pressure values. The ANNET is used to compensate the delay by modeling a second order delay function. The output of the ANNET is the desired control voltage reference for the artificial muscles' regulators. The MONNET is a relatively simpler neural network system to control the motors, which takes the desired trajectory and produces the velocity references as voltage outputs.

A. The ANNET

To overcome the delayed actuator response, mostly because of the air compressibility and rubber elasticity, a neural network called ANNET was used with PONNET for the compensation of the delay characteristics [6].

As described in [6], the artificial muscle-air regulator characteristic as a simple exponential delay function is described as:

$$P_m = AV_c(1 - e^{(-t/T)}) \qquad (1)$$

where A is a scaling constant, P_m is the artificial muscle pressure, V_c is the air regulators control voltage and T is the time constant of the artificial muscle-air regulator pair. It can be shown that P_m becomes equal to the target pressure, P_d, with no delay, if the control input to the air regulator is:

$$V_c = \frac{T}{A}\frac{dP_d(t)}{dt} + \frac{P_d(t)}{A} \qquad (2)$$

based on Equation 2, the input nodes of the ANNET neural network are simply two: one for the desired pressure signal and one for the rate of change of the desired pressure signal. A bias term is also included for the completeness of the neural network. Therefore, the simple form of the neural network will be the same as in [6]:

$$net_o = w_1\frac{dP_{d(t)}}{dt} + w_2P_d(t) + w_3 \qquad (3)$$

where w_1,w_2 and w_3 are the weights of ANNET. The output error is defined as:

$$E = \frac{1}{2}\sum(P_d - P_m)^2 \qquad (4)$$

where P_m is the measured pressure of the artificial muscle. Thus, ANNET is aimed to learn the transfer function of the artificial muscle-air regulator pair that is modeled with a first order delay [6]. The mean square error of the system with a very good trained (epoch size of 40000 and 3 neurons) neural network is 0.0506.

In order to improve the ANNET performance described in previous work [6], a simple second order delay function is used in this study as:

$$P_m = AV_c(1 - e^{(-t/T)} - te^{(-t/T)}) \qquad (5)$$

The delay function will be:

$$V_c(t) = \frac{T^2}{A}\frac{d^2P_d(t)}{dt^2} + \frac{T}{A}\frac{dP_d(t)}{dt} + \frac{P_d(t)}{A} \qquad (6)$$

The corresponding form of the neural network will therefore become:

$$net_o = w_1\frac{d^2P_d(t)}{dt^2} + w_2\frac{dP_d(t)}{dt} + w_3P_d(t) + w_4 \qquad (7)$$

ANNET is aimed to learn the transfer function of the artificial muscle-regulator pair that is modeled with a second order delay. The mean square error of the system with a properly trained (epoch size of 40000 and 3 neurons) neural network reduces to 0.0136.

B. The PONNET

In order to construct the position control neural network (PONNET) based on the physical model as proposed, we need to analyze the Language-Euler formulation of the BHRA. However, none of the dynamics parameters involved in a Lagrange-Euler expression will be computed or solved for. These expressions will be used to choose the correct input vectors and to define the correct neural network architecture of the PONNET layer.

The Lagrange dynamic formulation provides a means of deriving the equations of the motion from a scalar function called the "Lagrangian", which is defined as the difference between the kinetic and potential energy of a mechanical system. Lagrangian difference is formulated as:

$$L = K - P \qquad (8)$$

where K and P are, respectively, the total kinetic energy and the total potential energy of the system. The Lagrange's equations of the system are obtained by using Equation 9.

$$\frac{d}{dt}\frac{\partial L}{\partial \dot{q}} - \frac{\partial L}{\partial q} = \tau_i \qquad (9)$$

All the joints are revolute, then the required torque, T_i, of joint i, is expressed as in terms of joint variables, q_i. The torques acting on the end-effector is calculated by using Equation 10, as:

$$\tau_3 = m_3\left(\frac{1}{3}L_3^2 + \frac{1}{2}L_3L_2C_4 + \frac{1}{2}L_1L_3C_{34}\right)\ddot{\theta}_2 + m_3\left(\frac{1}{3}L_3^2 + \frac{1}{2}L_2L_3C_4\right)\ddot{\theta}_3 + \frac{1}{3}m_3L_3^2\ddot{\theta}_4$$
$$+\left(\frac{1}{2}m_3L_2L_3S_4 + \frac{1}{2}m_3L_1L_2S_{34}\right)\dot{\theta}_2^2 + \left(m_3L_2L_3S_4\right)\dot{\theta}_2\dot{\theta}_3 + \left(\frac{1}{2}m_3L_2L_3S_4\right)\dot{\theta}_3^2 + \frac{1}{2}m_3gL_3S_{123}$$

$$\tau_4 = I\ddot{\theta}_5 \qquad (10)$$

where, $C_x = Cos\theta_x$, $S_x = Sin\theta_x$, $C_{xyz} = Cos(\theta_x+\theta_y+\theta_z)$ *and* $S_{xyz} = Sin(\theta_x+\theta_y+\theta_z)$, m_3 is the point mass of the end effector, and L_1, L_2, L_3 are the length of the links, g for the gravity, q_4 and q_5 are the approach and orientation angles of the end-effector with respect to the fixed robot coordinate frame. The above equations are different from [6] because the links' movement is along the horizontal direction in [6] where, in

our system the trajectories are along the vertical direction against the gravity.

The effective point masses and effective link lengths, which were involved in Equation 10 and very hard to found by using very accurate measurement techniques or complex system identification tools, are left to PONNET. However, these parameters are trajectory independent and instead of finding the exact values of these parameters, PONNET can generalize from a representative set of training trajectories.

In order to relate these torques with the artificial muscle torques acted on the end-effector, the artificial muscle pressure for the two joints can be found by Equation 11 [4].

$$P_i = P_{oi} \pm \left[\frac{\tau_i}{2r(\beta_i - \alpha_i \varepsilon_i} - \frac{\gamma_i}{(\beta_i - \alpha_i \varepsilon_i)} \right] \qquad (11)$$

where β, γ and α are artificial muscle specific constants that we also do not wish to compute. Here, i, is the artificial muscle index, P_o the equilibrium pressure, ε is the elongation and "±" is positive for an agonist and negative for an antagonist artificial muscle. For the time being if the joint is assumed to be fixed in position, making ε a constant, the trajectory variables can be separated. It is a paradox to assume ε a constant for a robot that is designed for motion, but the paradox is resolved by dividing the motion space of each artificial muscle into several small segments as depicted in Figure 4 and assuming ε constant only for that small segment, only. With this assumption, combining Equations 10 and 11, a weighted sum of the non-linear functions of trajectory variables can be expressed any of the four artificial muscle pressures as:

$$\begin{aligned} Net_o &= w_1\ddot{\theta}_2 + w_2 C_4 \ddot{\theta}_2 + w_3 C_{34}\ddot{\theta}_2 + w_4\ddot{\theta}_3 \\ &+ w_5 C_4 \ddot{\theta}_3 + w_6\ddot{\theta}_4 + w_7 S_4 \dot{\theta}_2^2 + w_8 S_{34}\dot{\theta}_2^2 \\ &+ w_9 S_4 \dot{\theta}_2\dot{\theta}_3 + w_{10} S_4 \dot{\theta}_3^2 + w_{11} S_{234} + w_{12}\ddot{\theta}_5 \end{aligned} \qquad (12)$$

The weights represent system parameters, such as pulley radius, r, initial pressure P_o joint position ε, the artificial muscle parameters (β, γ and α), the robot parameters (m_3, L_1, L_2, L_3 and g). Furthermore, the artificial muscle pressure is always positive and bounded thus enabling us to use the sigmoid function for error propagation. Therefore we can assign a single layer backpropagation neural network for each artificial muscle using Equation 12, which we call PONNET. There is a dedicated PONNET for each artificial muscle and for each 4° segment of the joint motion as in [6]. Considering the maximum and minimum reach limits of the workspace as -72° and 72°, for each artificial muscle 36 small neural networks are trained (Totally 4x36=144 PONNET). As mentioned in [2][3][4][5] the hysteretic characteristics of artificial muscles cause different torques when moving forward or backwards. In order to overcome this problem, we use two different neural networks for the same angle segment, making the total number of neural networks per artificial muscle 288 PONNETs. However, only one is active at a given time depending on the desired joint position(θ_d) and moving direction as shown in Figure 4.

C. The Estimated Error Function of PONNET

In order to understand the characteristics of the error caused by the discontinuities between discrete PONNETs, again the collected output data is given as input (P_{1c}, P_{2c}, P_{3c}, P_{4c} and joint variables) to the built PONNETs.

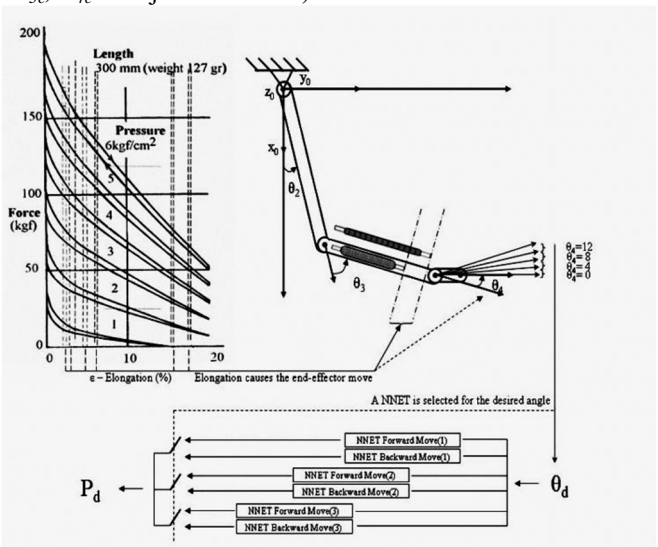

Figure 4: Division of the workspace of the end-effector into small segments such as in [6]

The resulting error is:

$$E = P_c - P_{sPONNET} \qquad (13)$$

where PsPONNET is the simulated pressure output of the PONNETs. As it was discussed in the previous section, we have 72 PONNETs for each artificial muscle. Instead of finding the estimated error function for each 4° segments and for each artificial muscles, all the segments are reduced to one 0°- 4° segments, which will show the general error characteristics of the error distribution, shown in Figure 5.

$$\theta_r = \theta_c - \text{integer\_part\_of}(\theta_d / 4) \qquad (14)$$

where θ_c is the collected angle data and θ_r is the reduced angle. Table 1 shows the results of this process.

$$\hat{E} = f(E(\theta_r)) \qquad (15)$$

The mean error value is found for each reduced angle value and the estimated error function is constructed by fitting a second order polynomial curve to the error distribution, show in Figure 5.

Table 1. All the collected angles are reduced to a reduced angle, which is between 0°- 4°

Collected Angle	Reduced Angle [Angle - int(Angle/4)]	Error (bar)
4	4	0.017
7.2	3.2	0.015
13.5	1.5	0.012
2	2	0.008
32	0	0.018
63	3	0.013

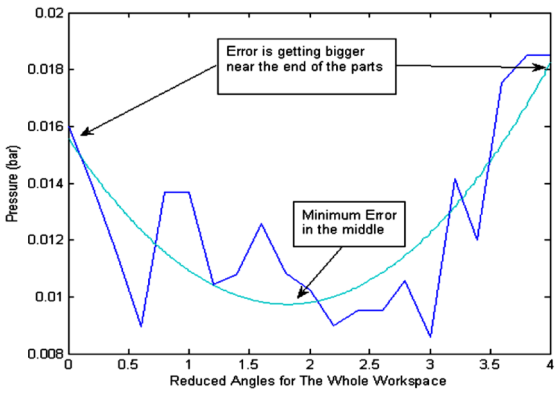

Figure 5. The Generalized Estimated Error Function

D. The Fuzzy-Neural Hybrid System Approach for Robot Arm Controlling

Fuzzy logic approach that was introduced in 1965 by Zadeh [16] is an efficient way to map input spaces in to output spaces of complex systems [17]. The proposed study is a combination of fuzzy algorithm with the neural network pressure outputs and models the elongation nonlinearity to give a final pressure value for controlling the robot arm. There are basically two linguistic variables: estimated error, $\hat{E}$ (unit in pressure) and convex combination parameter, α. The antecedent variable, estimated error is obtained from the error distribution of artificial neural network system outputs. Convex combination parameter, α, is the consequent linguistic variable of the fuzzy algorithm. Both the error and combination parameter have five normal type triangular membership functions that are namely very low, low, medium, high, and very high. The membership functions are basically specified with two parameters, mean of symmetric triangular, x_i, and the width of its sides, μ_i, where $i=1,2,3,4,5$ as shown in Figure 6.

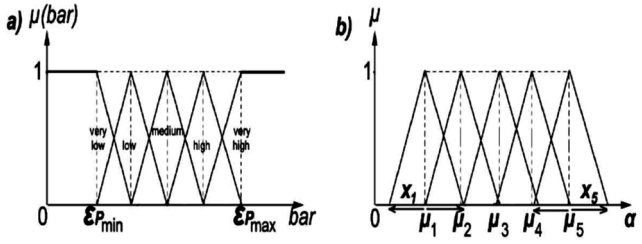

Figure 6. The fuzzy method linguistic variables a) error b) convex combination parameter

The final pressure is calculated as a convex combination of two consecutive neural network pressure outputs as follows:

$$P_f = \alpha P_{no1} + (1-\alpha) P_{no2} \qquad (16)$$

where P_f is the final pressure, P_{no1} and P_{no2} are consecutive neural network pressure outputs.

The rules are adjusted such that minimum error is manipulated in a way that the pressure output of the PONNET is used with maximum contribution to the final pressure output, i.e. α becomes 1 and when the error is maximum, the contribution of the consecutive neural network outputs are forced to be equal, this means α becomes 0.5.

Since there is only one variable in the system there is no need to use a t-norm in the inference. The fuzzy membership function value of the antecedent is directly becomes the fuzzy number of the consequent variable. Center of gravity is used as defuzzification method in the fuzzy algorithm.

E. The GA Optimization

In the previous section, the fuzzy membership function variables are chosen by experimenting. In order to find more appropriate membership function variables, a basic GA optimization is used because it can be applied to solve a variety of optimization problems that are not well suited for standard optimization algorithms, including problems in which the objective function is discontinuous, non-differentiable, stochastic, or highly nonlinear. In addition GA can be used for both constrained and unconstrained optimization problems [18].

The objective or the performance criteria is to minimize the total root mean squared error of the desired pressures and the pressure output of the fuzzy-neural hybrid system. The training data for PONNET is also used for GA optimization. As mentioned before the main goal is to find an off line control structure, so a set of training data is used during the construction of the control system.

The inputs to the GA are the mean and the width of the membership functions shown in Figure 6 and given in Equation 16, thus for each of five membership functions, there are two variables (the center and width of the membership functions) and consequently the number of variables for GA is 10.

We use the root mean square error as the fitness function of the GA. RMSE is computed from the difference of the output of the fuzzy-neural hybrid system and the actual pressure values. The optimization parameters are x_1, x_2, x_3, x_4, x_5, μ_1, μ_2, μ_3, μ_4, μ_5. The population size is 20, the crossover fraction is 0.8, and the migration fraction is 0.2. The floating point representation is selected as the gnome structure for each individual. The genetic algorithm then creates a population of solutions and applies genetic operators such as mutation and crossover to evolve the solutions in order to find the best one.

III. EXPERIMENTAL RESULTS AND CONCLUSIONS

In order to show how accurate the new system is in comparison to the previous one, several trajectories were given to the manipulator to follow using two methods: neural network [6] and fuzzy-neural hybrid. The control objective was to keep the end-effector always perpendicular to the X-Y plane of the operational space as shown in Figure 1. It is assumed that the task is to paint the desired trajectory on the X-Y plane using a pen, which has to be always

perpendicular to the surface. Thus, Joint 4 will move up and down, while Joint 5 will not rotate in order to keep the pen perpendicular to the wall. The trajectory tracking performances, which are the root mean square of the errors (RMSE), between the end end-effector's desired trajectory angle and θ_4 in each step, are calculated for the neural network control method cited in [6] and the new fuzzy-neural hybrid control method in the operational space and in the joint space, and the outputs are shown in Table 2.

Table 2: The RMSE Performance Analysis of Both Systems on Various Trajectories (The RMSEs , shown in the table, are the performance of the controller for sine waves with frequencies of 0.1Hz, 0.3Hz, 0.5Hz, 0.8Hz, ramp and step trajectories)

Trajectory	Neural Network	Fuzzy-Neural Hybrid	Change (%)
Sine,f=0.1Hz	2.2836	1.7699	22.50%
Sine,f=0.3Hz	2.4624	1.9286	21.68%
Sine,f=0.5Hz	2.9403	2.2518	23.42%
Sine,f=0.8Hz	5.7398	4.0077	30.18%
Ramp	1.9413	1.7791	8.36%
Step	2.3789	2.4221	-1.81%

The results show that outputs of the fuzzy-neural hybrid system had less time delay, more smooth and fit to the desired motion trajectory better than the only-neural network method. Using a second-order model instead of first-order caused the outputs to decrease the time delay and using a fuzzy-neural hybrid system caused the system outputs to be more smooth and accurate. The RMSEs of both systems were nearly the same for the step inputs, because the fuzzy system is constructed only for the discontinuities between consecutive small neural networks, not for the high discontinuities in the desired trajectories.

In conclusion, the original system defined in [6] was improved by introducing an efficient second-order delay function and constructing a fuzzy-neural hybrid control system by combining the discrete neural networks without chattering. The new controller encoded the physical quantities such as the magnitude, direction of movement and their derivatives [12][13][14] to meet the primary concepts of the human brain functioning for the arm movement. This model is not only more biologically-plausible than the previous model [6] but also is more accurate and efficient.

IV. REFERENCES

[1] F. Iida and Y. Minekawa, J. Rummel and A. Seyfarth, "Toward a human-like biped robot with compliant legs," *Robotics & Autonomous Systems*, vol.57, pp.1139-1144, 2009

[2] K. Kawamura, R.A. Peters, D. M. Wilkes, W.A. Alford, and T. E. Rogers, "ISAC: Foundations in Human-Humanoid Interaction," *IEEE Intelligent Systems & their applications,* vol.5, no.5, pp. 38-45, 2000.

[3] Lilly, J. and Yang, L., "Sliding mode tracking for pneumatic muscle actuators in opposing pair configuration," *IEEE Trans. on Control Systems Technology.* vol.13, pp.550-558, 2005.

[4] Schroder, J., Erol, D., Kawamura, K., and Dillman, R., "Dynamic pneumatic actuator model for a model-based torque controller," *IEEE Int'l Sym. on Computational Intelligence in Robotics and Automation.* vol.1, pp.342-347, 2003

[5] B. Ulutas, E. Erdemir and K. Kawamura, "Application of a Hybrid Controller with Non-Contact Impedance to a Humanoid Robot," *10th Int'l Workshop on Variable Structure Systems*, 2008.

[6] Ozkan, M., Inoue, K., Negishia, K., and Yamanaka, T., "Defining a neural network controller structure for a rubbertuator robot," *Neural Networks*, vol.13, no.4, pp.533-544, 2000.

[7] N. Tsagarakis, D. G. Caldwell, and G. A. Medrano-Cerda, "A 7 DOF pneumatic muscle actuator (pMA) powered exoskeleton," *IEEE Int'l Workshop on Robot and Human Interaction*, pp. 327-333, 1999.

[8] Lilly, J., "Adaptive tracking for pneumatic muscle actuators in bicep and tricep configurations." *IEEE Trans. Neural Systems &Rehabilitation Engineering*, vol.11, no.3, pp.333-339, 2003.

[9] T. Nagaoka, Y. Konishi, and H. Ishigaki, "Nonlinear optimal predictive control of rubber artificial muscle," *In Proc. SPIE Int. Soc. Opt. Eng*.,vol.25,pp.54-61, 1995.

[10] Chan, S., J.Lilly, Repperger, D., and Berlin, J. ,"Fuzzy PD+I learning control for a pneumatic muscle," *In Proc. IEEE Int. Conf. Fuzzy Systems*, pp.278-283, 2003.

[11] Thongchai, S., Goldfarb, M., Sarkar, N., and Kawamura, K.," A frequency modeling method of rubbertuators for control application in an ima framework," *In American Control Conference*, vol. .2, pp.1710- 1714, 2003.

[12] Choi, K., Hirose, H., Sakurai, Y., Iijima, T., and Koike, Y. ," Prediction of Arm Trajectory from the Neural Activities of the Primary Motor Cortex Using a Modular Artificial Neural Network Model," *Neural Networks*,22: 1214-1223, 2009.

[13] Bear, M.F., Connors, B.W. & Paradiso, M.A. Neuroscience: Exploring the brain. Williams&Wilkins, 2007.

[14] Georgopoulos, A. P., Kettner, R. E., and Schwartz, A. B. ,"Primate motor cortex and free arm movements to visual targets in three-dimensional space. II. Coding of the direction of movement by a neuronal population", *Neuroscience Journal*, 8: 2928-2937, 1988.

[15] Lin, C. and Lee, C. Neural fuzzy systems : a neurofuzzy synergism to intelligent systems. Prentice Hall, NJ, 1st edition., 1996.

[16] Zadeh, L. Fuzzy sets. In Information and Control. vol.8, pp.338-353, 1965.

[17] Tsourveloudis, N. and Phillis, Y., "Fuzzy assessment of machine flexibility," *In IEEE Trans. on Engineering Management*, vol.45, pp.78-87, 1998.

[18] S. Benson, M. Krishnan, L. McInnes, J. Nieplocha and J. Sarich, "Using the GA and TAO Toolkits for Solving Large-Scale Optimization Problems on Parallel Computers", *Trans. on Mathematical Software*,33,2007.