FACTA UNIVERSITATIS
Series: **Mechanics, Automatic Control and Robotics** Vol.3, N° 15, 2003, pp. 1039 - 1054

APPLICATION OF THE TAKAGI-SUGENO FUZZY CONTROLLER FOR SOLVING THE ROBOTS' INVERSE KINEMATICS PROBLEM

UDC 62-52+681.53+681.58

Vesna Ranković, Ilija Nikolić

Faculty of Mechanical Engineering,
Sestre Janji} 6, 34000 Kragujevac, Serbia and Montenegro, e-mail: inikolic@ptt.yu

Abstract. *For mapping from external to internal coordinates the Takagi-Sugeno controller is used, implemented within the framework of adaptive network, which is similar, by its architecture, to RBF neural network, so it is also called the neuro-fuzzy system. During the learning process, the parameters of the membership functions of the primary fuzzy sets and parameters of the consequences of the Takagi-Sugeno controller were adapted. The results are presented for the straight-line trajectory tracking, with the constant velocity of the gripper and with the triangular velocity profile.*

1. INTRODUCTION

Deficiencies of analytical and numerical methods of solving the inverse kinematics (IK) problem led to searching of the new approaches to mapping from external to internal coordinates.

A large number of papers are devoted to application of the non-recurrent (or feed forward) neural networks for solving the IK. In [6] is used the two-layered neural network (in literature also known as Radial Basis Function – RBF neural network). The dynamic procedure was applied for learning the network. Neurons in the invisible layer have Gaussian activation functions, while the output layer consists of a single neuron whose activation threshold is being adapted. As it is shown, the main deficiency of the RBF networks, trained by learning data sets, which enhance the part of the working space, lies in fact that their hidden layers contain large number of neurons. The RBF neural networks cannot realize mapping from external to internal coordinates in real time for high gripper velocities.

Fuzzy logic was first applied for solving the IK of the planar four-segment redundant robot in paper [4]. This paper was used as a basis for developing the fuzzy logical controller (FLC), presented in paper [5], where the FLC is applied for mapping from the external to internal coordinates for planar two-segment robot and for robots of the $R_zT_zT_y$ and $R_zR_yR_y$

Received April 03, 2002

minimal configuration. The biggest deficiency of such an FLC structure is the long computational time, which is the consequence of the large number of steps during the trajectory generation. In addition, in design of logical fuzzy controller, especially problematic is defining of the rule base and selection of the membership functions of the primary fuzzy sets. Numerous researchers have tried to automatize the modeling process of the fuzzy controller. Because of such investigations the adaptive fuzzy controllers appeared. Adapting mechanism decides which changes should be performed in the fuzzy controller, so it would obtain the desired output, namely it would minimize the error. This mechanism is similar to the training mechanism in neural networks.

Different adaptive systems were developed. The most frequently used adapting mechanism performs the modification in the phase of controlling rules, changes of the weighting functions of the joined fuzzy rules to controlling rules in the rule base, modification of the primary fuzzy sets or the selection of the defuzzyfication methods. Based on available literature, it can be concluded that the different training methods were applied most frequently to the Takagi-Sugeno controller.

In this work is used the Takagi-Sugeno controller implemented within the framework of the adaptive network for solving the IK problem.

In the second section of the paper is presented the basic structure of the controller, and is briefly described the procedure used for training. In the third section are presented results of mapping from external to internal coordinates for the three-segment robot of the $R_zR_yR_y$ minimal configuration. The fourth section contains the concluding remarks.

2. TAKAGI-SUGENO FUZZY CONTROLLER AND THE LEARNING ALGORITHM

The Takagi-Sugeno controller with M inputs and one output is shown in Figure 1. The fuzzy controller has M+1 linguistic variables: M input ones and one output variable. Linguistic variables x_i are A_{1i}, A_{2i}, …, A_{ni}, M. For the system with M inputs and one output, the set of linguistic rules is defined in the form:

R_1: if x_1 is A_{11} and x_2 is A_{12} … and x_M is A_{1M}, then $f_1 = p_{11}x_1 + p_{12}x_2 + \ldots + p_{1M} + c_1$
R_1: if x_1 is A_{11} and x_2 is A_{12} … and x_M is A_{2M}, then $f_2 = p_{21}x_1 + p_{22}x_2 + \ldots + p_{2M} + c_2$
…
R_1: if x_1 is A_{21} and x_2 is A_{22} … and x_M is A_{kM}, then $f_k = p_{k1}x_1 + p_{k2}x_2 + \ldots + p_{kM} + c_k$
…
R_1: if x_1 is A_{n1} and x_2 is A_{n2} … and x_M is A_{nM}, then $f_p = p_{p1}x_1 + p_{p2}x_2 + \ldots + p_{pM} + c_p$

The number of linguistic rules is $p = n^M$.

The output from Takagi-Sugeno controller from Figure 1 is:

$$y = \sum_{i=1}^{p} \overline{u}_i f_i \tag{1}$$

where:

$$\overline{u}_i = \frac{u_i}{\sum_{i=1}^{p} u_i} \tag{2}$$

$$\begin{aligned} u_1 &= \mu_{11}(x_1) * \mu_{12}(x_2) \ldots * \mu_{1M}(x_M) \\ u_2 &= \mu_{11}(x_1) * \mu_{12}(x_2) \ldots * \mu_{2M}(x_M) \\ &\ldots \end{aligned} \tag{3}$$

$$u_k = \mu_{21}(x_1) * \mu_{22}(x_2) \ldots * \mu_{2M}(x_M)$$
$$\ldots$$
$$u_p = \mu_{n1}(x_1) * \mu_{n2}(x_2) \ldots * \mu_{nM}(x_M)$$

* denotes certain *T*-norm.

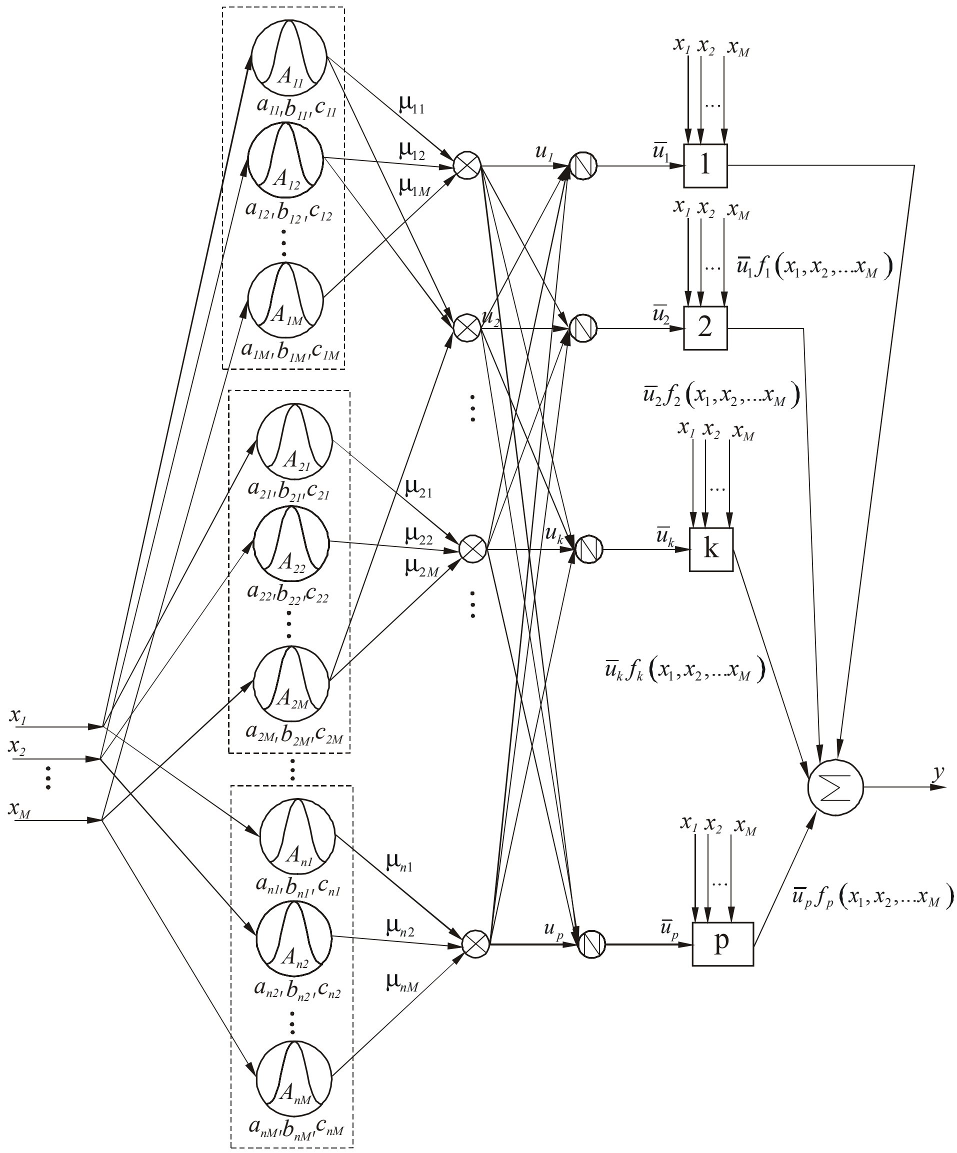

Fig. 1. Takagi-Sugeno controller with M inputs and one output.

If the membership functions are taken in the Gaussian form then:

$$\mu_{ij} = \frac{1}{1 + \left[\left(\frac{x_j - a_{ij}}{c_{ij}} \right)^2 \right]^{b_{ij}}}, \; i = \overline{1,n}, \;\; j = \overline{1,M} \;. \tag{4}$$

Consequences functions of the fuzzy rules are of the form:

$$f_i = \sum_{j=1}^{M} p_{ij} x_j + c_i \tag{5}$$

Substituting (2) into (1) the controller output is obtained as:

$$y = \frac{1}{\sum_{i=1}^{p} u_i} \sum_{i=1}^{p} u_i f_i \tag{6}$$

or, substituting (5) into (6), the output of theTakagi-Sugeno controler is:

$$y = \frac{1}{\sum_{i=1}^{p} u_i} \sum_{i=1}^{p} u_i \left(\sum_{j=1}^{M} p_{ij} x_j + c_i \right). \tag{7}$$

In [2], for adapting parameters (a_{11}, a_{12}, …, a_{1M}, b_{11}, b_{12}, …, b_{1M}, c_{11}, c_{12}, …, c_{1M}) is used the iterative procedure, which is based on method of decreasing gradients. The parameters of the f_i functions (p_{11}, p_{12}, …, p_{1M}, c_1, p_{21}, p_{22}, …, p_{2M}, c_2, …, p_{k1}, p_{k2}, …, p_{kM}, c_k, p_{p1}, p_{p2}, …, p_{pM}, c_p) are being adapted by the least squares method. The total number of parameters for adapting is: $3nM + n^M(M+1)$. The learning method requires set of data for training $P = \{p_1, p_2, …, p_r)$. Each element of the set, $p_k = (x_k, y_{zk})$ is defined by the input vector $x_k = (x_{1k}\; x_{2k} \ldots x_{Mk})$ and desired response y_{zk}.

If the FLC parameters correction is performed after presenting all the samples, then the error sum of squares is:

$$\varepsilon = \sum_{k=1}^{r} \varepsilon_k \tag{8}$$

or:

$$\varepsilon = \sum_{k=1}^{r} (y_{zk} - y_k)^2 \tag{9}$$

where:

$$y_k = \frac{1}{\sum_{i=1}^{p} u_{ik}} \sum_{i=1}^{p} u_{ik} f_{ik} \tag{10}$$

$$f_{ik} = \sum_{j=1}^{M} p_{ij} x_{jk} + c_i \tag{11}$$

$$\begin{aligned} u_{1k} &= \mu_{11}(x_{1k}) * \mu_{12}(x_{2k}) \ldots * \mu_{1M}(x_{Mk}) \\ u_{2k} &= \mu_{11}(x_{1k}) * \mu_{12}(x_{2k}) \ldots * \mu_{2M}(x_{Mk}) \\ &\ldots \end{aligned} \tag{12}$$

$$u_{kk} = \mu_{21}(x_{1k}) * \mu_{22}(x_{2k}) \ldots * \mu_{2M}(x_{Mk})$$
$$\ldots$$
$$u_{pk} = \mu_{n1}(x_{1k}) * \mu_{n2}(x_{2k}) \ldots * \mu_{nM}(x_{Mk})$$

Minimization of the function given with (8) can be realized through the following iterative procedure for parameters adapting:

$$a_{zq}(l+1) = a_{zq}(l) - \eta_a \frac{\partial \varepsilon}{\partial a_{zq}}$$
$$b_{zq}(l+1) = b_{zq}(l) - \eta_b \frac{\partial \varepsilon}{\partial b_{zq}} \quad (13)$$
$$c_{zq}(l+1) = c_{zq}(l) - \eta_z \frac{\partial \varepsilon}{\partial c_{zq}}$$

From equations (13) one obtains:

$$a_{zq}(l+1) = a_{zq}(l) - \eta_a \sum_{k=1}^{r} 2(y_k - y_{zk}) \frac{\partial y_k}{\partial a_{zq}} \quad (14)$$

where:

$$\frac{\partial y_k}{\partial a_{zq}} = \frac{1}{\sum_{i=1}^{p} u_{ik}} \left(\sum_{i=1}^{p} \frac{\partial u_{ik}}{\partial a_{zq}} f_{ik} - y_k \sum_{i=1}^{p} \frac{\partial u_{ik}}{\partial a_{zq}} \right)$$

$$b_{zq}(l+1) = b_{zq}(l) - \eta_b \sum_{k=1}^{r} 2(y_k - y_{zk}) \frac{\partial y_k}{\partial b_{zq}} \quad (15)$$

where:

$$\frac{\partial y_k}{\partial b_{zq}} = \frac{1}{\sum_{i=1}^{p} u_{ik}} \left(\sum_{i=1}^{p} \frac{\partial u_{ik}}{\partial b_{zq}} f_{ik} - y_k \sum_{i=1}^{p} \frac{\partial u_{ik}}{\partial b_{zq}} \right)$$

$$c_{zq}(l+1) = c_{zq}(l) - \eta_c \sum_{k=1}^{r} 2(y_k - y_{zk}) \frac{\partial y_k}{\partial c_{zq}} \quad (16)$$

where:

$$\frac{\partial y_k}{\partial c_{zq}} = \frac{1}{\sum_{i=1}^{p} u_{ik}} \left(\sum_{i=1}^{p} \frac{\partial u_{ik}}{\partial c_{zq}} f_{ik} - y_k \sum_{i=1}^{p} \frac{\partial u_{ik}}{\partial c_{zq}} \right).$$

Parameters of the f_i functions are being determined by the least square method. From equations (1) and (5) one obtains that the response of the fuzzy controller to the k-th element is:

$$y_k = \bar{u}_{1k}(p_{11}x_{1k} + p_{12}x_{2k} + \ldots p_{1M}x_{Mk} + c_1) + \bar{u}_{2k}(p_{21}x_{1k} + p_{22}x_{2k} + \ldots p_{2M}x_{Mk} + c_2) + \ldots$$
$$\bar{u}_{pk}(p_{p1}x_{1k} + p_{p2}x_{2k} + \ldots p_{pM}x_{Mk} + c_p) \quad (17)$$

The aim of learning is that the real response is equal to the desired one. If the adapting is done after presenting of all samples, then:

$$\begin{bmatrix} \bar{u}_{11}x_{11} & \bar{u}_{11}x_{21} & \cdots & \bar{u}_{11}x_{M1} & \bar{u}_{11} & \bar{u}_{21}x_{11} & \bar{u}_{21}x_{21} & \cdots & \bar{u}_{21}x_{M1} & \bar{u}_{21}\cdots \\ \bar{u}_{12}x_{12} & \bar{u}_{12}x_{22} & \cdots & \bar{u}_{12}x_{M2} & \bar{u}_{12} & \bar{u}_{22}x_{12} & \bar{u}_{22}x_{22} & \cdots & \bar{u}_{22}x_{M2} & \bar{u}_{22}\cdots \\ \vdots & \vdots & \cdots & \vdots & \vdots & \vdots & \vdots & \cdots & \vdots & \vdots\cdots \\ \bar{u}_{1r}x_{1r} & \bar{u}_{1r}x_{2r} & \cdots & \bar{u}_{1r}x_{Mr} & \bar{u}_{1r} & \bar{u}_{2r}x_{1r} & \bar{u}_{2r}x_{2r} & \cdots & \bar{u}_{2r}x_{Mr} & \bar{u}_{2r}\cdots \end{bmatrix}$$

$$\begin{matrix} \bar{u}_{p1}x_{11} & \bar{u}_{p1}x_{21} & \cdots & \bar{u}_{p1}x_{M1} & \bar{u}_{p1} \\ \bar{u}_{p2}x_{12} & \bar{u}_{p2}x_{22} & \cdots & \bar{u}_{p2}x_{M1} & \bar{u}_{p2} \\ \vdots & \vdots & \cdots & \vdots & \vdots \\ \bar{u}_{pr}x_{1r} & \bar{u}_{pr}x_{2r} & \cdots & \bar{u}_{pr}x_{M1} & \bar{u}_{pr} \end{matrix}\Bigg] \begin{Bmatrix} p_{11} \\ p_{12} \\ \vdots \\ p_{1M} \\ c_1 \\ p_{21} \\ p_{22} \\ \vdots \\ p_{2M} \\ c_2 \\ \vdots \\ p_{p1} \\ p_{p2} \\ \vdots \\ p_{pM} \\ c_p \end{Bmatrix} = \begin{Bmatrix} y_{z1} \\ y_{z2} \\ \vdots \\ y_{zr} \end{Bmatrix} \qquad (18)$$

Equation (18) is presented in the form:

$$\boldsymbol{U}_x \boldsymbol{P}_x = \boldsymbol{Y}_z \qquad (19)$$

From (19), by the least squares method, the parameters vector $\boldsymbol{P}_{\mathrm{x}}$ is being determined:

$$\boldsymbol{P}_x = (\boldsymbol{U}_x^T \boldsymbol{U}_x)^{-1} \boldsymbol{U}_x^T \boldsymbol{Y}_z \qquad (20)$$

The structure of the fuzzy system from Figure 1 is similar to the structure of the RBF network. In [3] is given the detailed comparison of these two systems. It was concluded that there exist the functional equivalency if:

- The number of invisible units of the RBF network is equal to number of linguistic rules in fuzzy system
- For the fuzzy system of Figure 1 holds $f_i = const$
- The Gaussian functions are selected as the membership primary fuzzy sets for the system of Figure 1
- For determination of the T-norm in fuzzy systems, one uses the algebraic product
- The activation threshold of the output neuron for the RBF network is zero.

Since the structure of the fuzzy system of Figure 1 is similar to the structure of the RBF neural network, and taking into account that there exist functional equivalency, the adaptive fuzzy system is frequently called the neuro-fuzzy system.

3. APPLICATION OF THE TAKAGI-SUGENO FUZZY CONTROLLER TO SOLVING THE PROBLEM OF ROBOTS' INVERSE KINEMATICS

In paper [5] the FLC is applied to solving the problem of inverse kinematics. In all controllers, the fuzzy reasoning of the first type was applied, where the Mamdani minimum rule is applied as a function in the implication phase. The biggest deficiency of such FLC structures is long computational time. Especially problematic is definition of the base rule and selection of the membership functions of the primary fuzzy sets.

In this paper to solving of the IK is applied the Takagi-Sugeno controller shown in Figure 1. Simulations were done for the robot shown in Figure 2 (the $R_zR_yR_y$ configuration). The robot segments' lengths are: $l_1 = 1.3$ m, $l_2 = l_3 = 1$ m. It is taken that there are constraints in joints ($\theta_1 = \theta_2 = \theta_3 = [-\pi/2, \pi/2]$). The input and the output variables of the FLC for solving the IK are shown in Figure 3.

Simulations were performed with the program package MATLAB, by using the fuzzy toolbox.

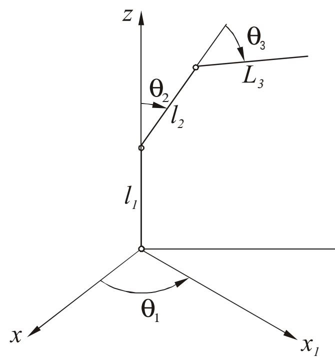

Fig. 2. The RRR robot of minimal configuration

Fig. 3. Illustration of the input and the output variables of the Takagi-Sugeno FLCs for solving the IK of the $R_zR_yR_y$ robot.

In the first example, the gripper was moving along the straight-line trajectory, from the point A(1 –0.4 2.7) to point B(1.36 0.34 2.2), with the triangular velocity profile. The time taken for performing the motion is $t_{\max} = 2$s. In this example by the data set for training of the FLCs is enhanced only one straight-line trajectory.

The learning speeds η_a, η_b and η_c were taken according to [4] in the form:

$$\eta_a = \eta_b = \eta_c = \frac{k}{\sqrt{\left(\frac{\partial \varepsilon}{\partial a_{zq}} + \frac{\partial \varepsilon}{\partial b_{zq}} + \frac{\partial \varepsilon}{\partial c_{zq}}\right)^2}}$$

where k is the step. The step size is variable, and so does the convergence rate. According to [2], if k is small, the convergence is slow, and several iteraqtions would be necessary, to achieve satisfactory accuracy. If k is big, the convergence would, at the beginning of the learning process, be fast, but the algorithm will oscillate.

The step k changes according to the following rules:

- If the error is decreasing in the four consecutive epochs, k increases for 10 %.
- If the error in one epoch decreases, then in the next one k decreases for 10 %.

In the first example for the FLC 1 is $k = 0.001$, and for FLC 2 and FLC 3 is $k = 0.01$.

The fuzzy partitioning of the input variables of the FLC is realized by selection of 2 primary fuzzy sets. Selected are the Gaussian membership functions, since the best results are achieved with them.

The total number of the fuzzy rules is 2^2. In the learning process the membership functions parameters of the primary fuzzy sets (12) were adapted as well as parameters of the Takagi-Sugeno controller (12). In Figure 4 are presented the membership functions of the primary fuzzy sets of variables x and y before learning, and in Figure 5 are presented the membership functions after learning.

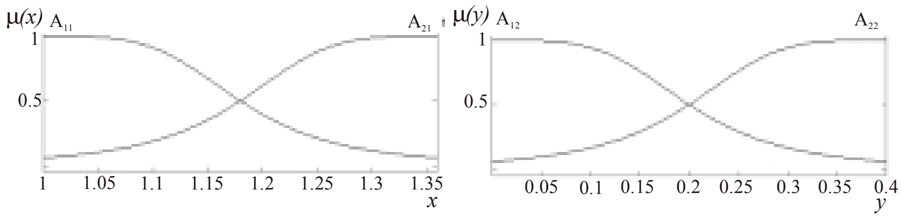

Fig. 4. Membership functions of variables x and y before learning.

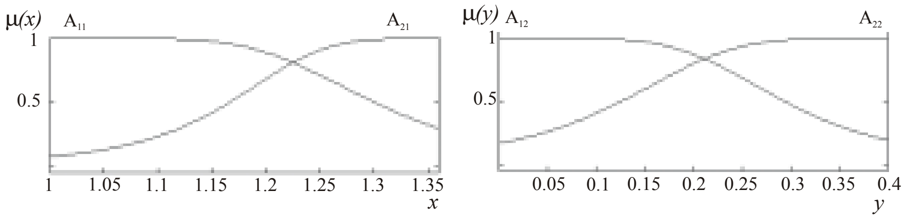

Fig. 5. Membership functions of variables x and y after learning the FLC 1.

In Figure 6 is presented the variation of error, given by expression (8), as a function of the number of epochs.

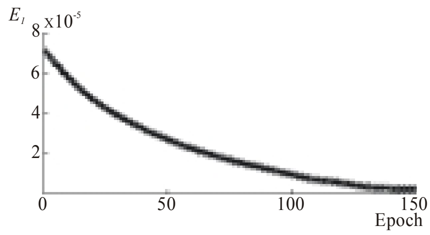

Fig. 6. Error variation during the learning process of the FLC 1.

Parameters of the fi functions, $i = 1,4$, are given in Table 1.

In FLC 2 to each input variable were assigned two membership functions. The controller has total of 4 rules. The total number of parameters, which are being adapted is 24, out of which 12 are the premise parameters and 12 are the consequences parameters. In Figure 7 are shown the membership functions of the primary fuzzy sets of variables x_i and z before learning, and in Figure 8 are shown the membership functions after learning.

The variation of error with the number of epochs is shown in Figure 9.

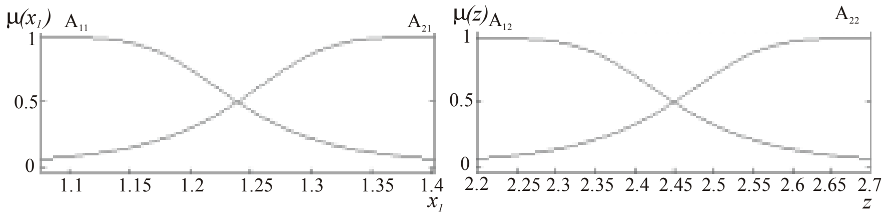

Fig. 7. Membership functions of variables x_i and z before learning.

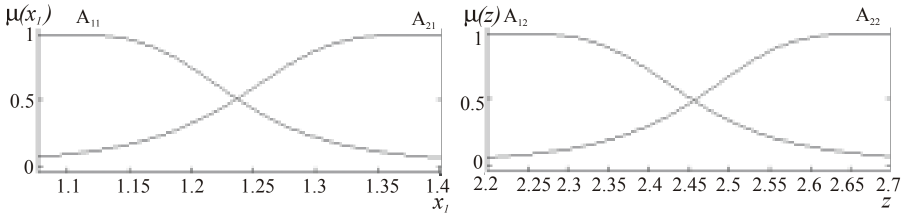

Fig. 8. Membership functions of variables x_i and z after learning the FLC 2.

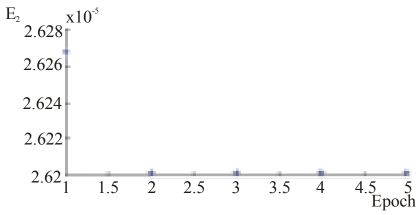

Fig. 9. Error variation during the learning process of the FLC 2.

Parameters of the *fi* functions, i = 1,4, for FLC 2 are given in Table 1. FLC 3 has 4 rules (two membership functions were assigned to each input). The membership functions of the primary fuzzy sets of variables x_i and z, prior to the adaptation process, are shown in Figure 7, while the figure 10 shows them after the learning process. In Figure 11 is illustrated the variation of error, given with expression (8) as a function of the number of epochs.

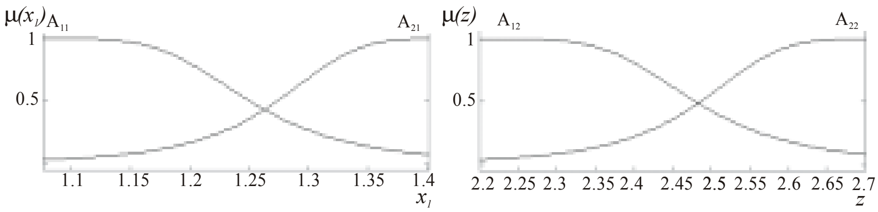

Fig. 10. Membership functions of variables x_i and z after learning the FLC 3.

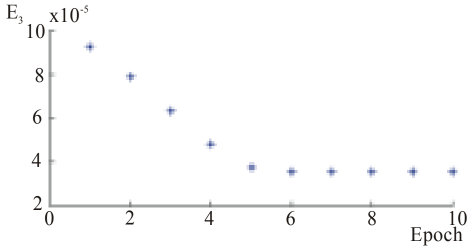

Fig. 11. Error variation during the learning process of the FLC 3.

Parameters of the *fi* functions, i = 1,4, for FLC 3 are given in Table 1. In Figure 12 is given the variation of the interior coordinates during executing the working task, and in Figure 13 is given the variation of the total error of tracking the straight-line trajectory.

Table 1. Parameters of consequences of the Takagi-Sugeno controllers

	FLC 1	FLC 2	FLC 3
p_{11}	-0.04188	0.4295	0.6089
p_{21}	0.9598	-0.2858	0.2626
c_1	0.04491	0.3602	-0.1068
p_{21}	-0.2086	1.349	-1.596
p_{22}	0.9193	0.5364	-1.766
c_2	0.2336	-2.765	7.49
p_{31}	-0.0335	0.7996	-0.2134
p_{32}	0.6508	-0.3319	0.7663
c_3	0.04044	0.03785	-0.1752
p_{41}	0.1331	-0.6414	0.4164
p_{42}	0.4679	0.6582	-0.4741
c_4	-0.0975	-0.3231	0.3245

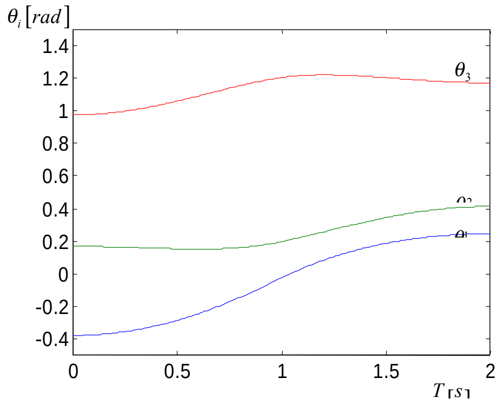

Fig. 12. Variation of the internal coordinates along the trajectory

From Figure 13 can be seen that the maximum total error of the straight-line trajectory tracking, in the first example is less than $5.5 \cdot 10^{-2}$ mm. The simulation results show that the application of the neuro-fuzzy controllers, with the membership functions, shown in Figures 5, 8 and 10, and with parameters of the consequences functions given in Table 1, give the satisfactory results. The FLCs from this example have smaller number of linguistic rules than the number of the invisible units of the RBF networks, generated for solving the same problem, shown in [6].

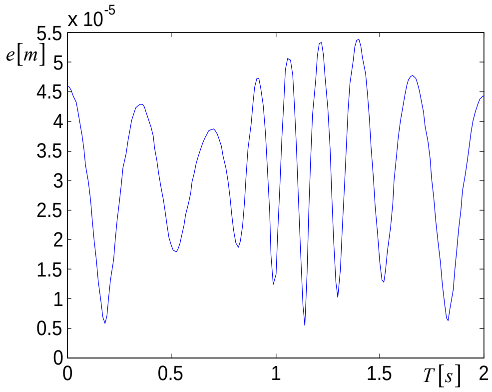

Fig. 13. The total tracking error

In the second example the gripper was moving along the straight-line trajectory, form the point A(1.366 0 2.666) to point B(1.6 0.3 2.3), with the speed equal to v = 0.01 *m/s*, and the training data set enhanced the part of the working area x = [1; 1.7], y = [0; 0.5] and z = [2; 2,8]. The training set predicted that the trajectories can be realized both over the positive or negative values of the internal coordinate θ_3. The FLC 1 determines the internal coordinate θ_1, the FLC 2 θ_2 for positive values of θ_3, and FLC 3 generates the coordinate θ_2 for negative values of θ_3, and FLC 4 θ_3. Selection of the FLC, which would determine the coordinate θ_2, is realized based on the robot's initial position.

In FLC 1 to each input variable are assigned three membership functions. The controller has total of 9 rules. The total number of parameters, which are being adapted is 45, out of which 18 are the premise parameters, and 27 are the consequences parameters.

Fuzzy partitioning of the input variables FLC 2, FLC 3 and FLC 4 is realized by selection of 5 primary fuzzy sets for each variable. The membership functions are selected as Gaussian. The total number of the fuzzy rules of each controller is 5^2. In the learning process, the parameters of the membership functions of the primary fuzzy sets were adapted (30) as well as the parameters of the consequences of the Takagi-Sugeno controller (75). In Figures 14 and 15 are shown the membership functions of the primary fuzzy sets of variables x and y, for FLC 1, and variables x and z1 for FLC 2, FLC 3 and FLC 4, prior to learning. In Figures 16 to 19 are presented the membership functions after learning.

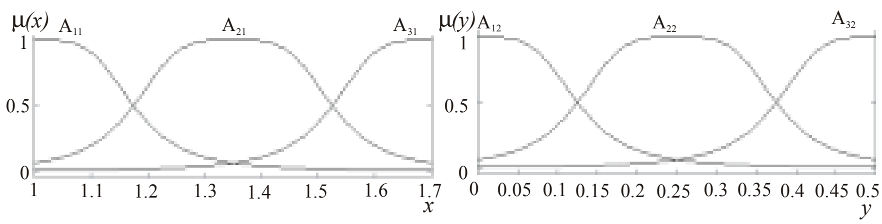

Fig. 14. Membership functions of variables x and y before learning.

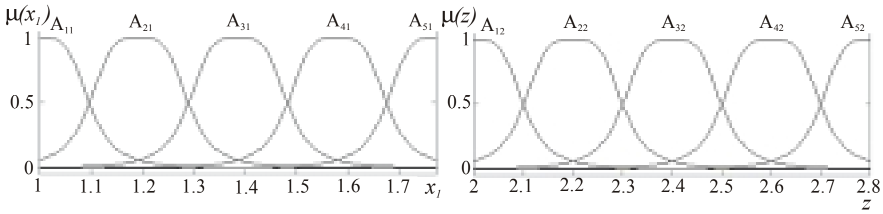

Fig. 15. Membership functions of variables x and z_i before learning.

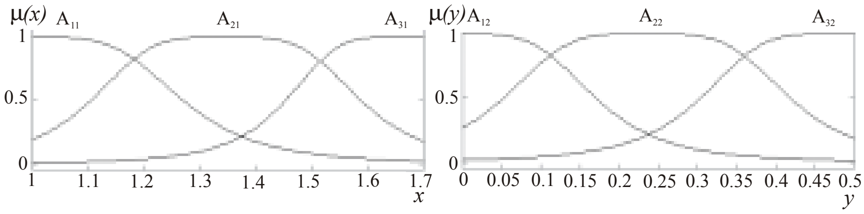

Figure 16. Membership functions of variables x and y after learning the FLC 1.

In Figures 20 to 23 are shown error variations as a function of number of epochs for all four networks.

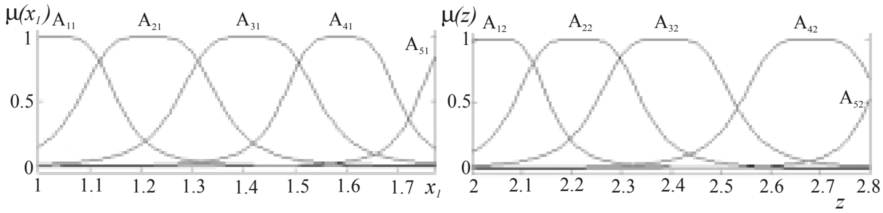

Fig. 17. Membership functions of variables x and z_i after learning the FLC 2.

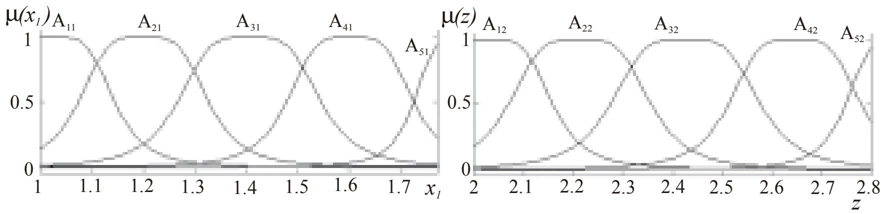

Fig. 18. Membership functions of variables x and z_i after learning the FLC 3.

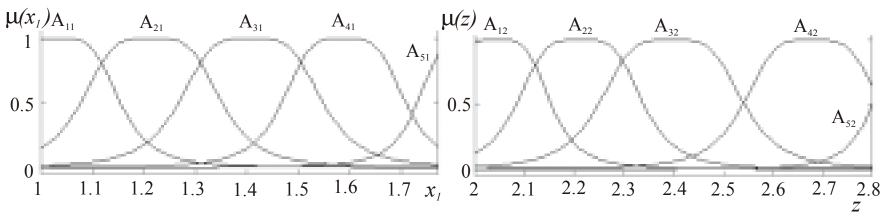

Fig. 19. Membership functions of variables x and z_i after learning the FLC 4.

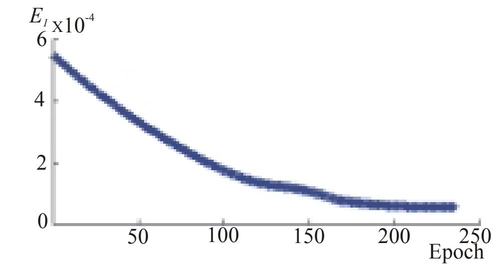

Fig. 20. Error variation during the learning process of the FLC 1.

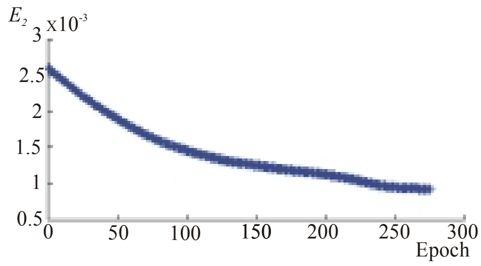

Fig. 21. Error variation during the learning process of the FLC 2.

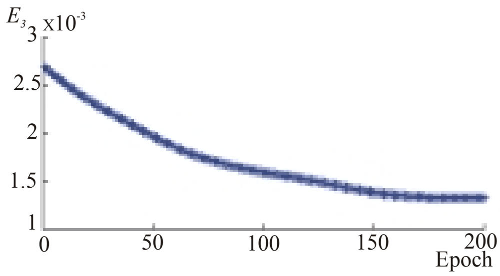

Fig. 22. Error variation during the learning process of the FLC 3.

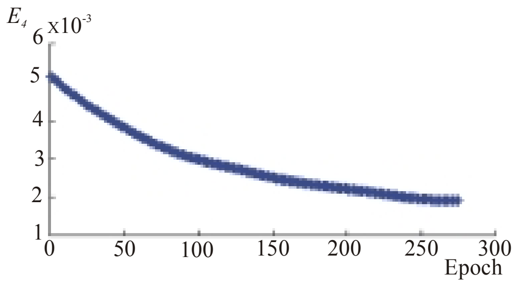

Fig. 23. Error variation during the learning process of the FLC 4.

Parameters of functions *fi*, for FLC 1, FLC 2, FLC 3 and FLC 4 are given in Table 2.

In Figure 24 is presented variation of the internal coordinates during the task execution, and in Figure 25 is given the total error variation of the straight-line trajectory tracking.

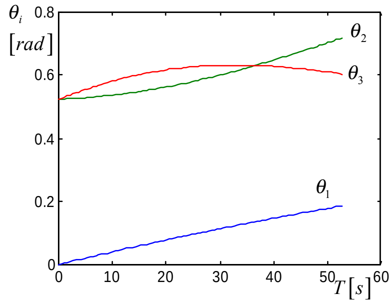

Fig. 24. Variation of the internal coordinates along the trajectory

Table 2. Consequences parameters

	FLC 1	FLC 2	FLC 3	FLC 4
p_{11} p_{21} c_1	0.05958 0.8742 -0.07642	1.096 -0.04593 -0.9365	0.8264 1.564 0.6234	-0.1666 -0.5935 2.979
p_{21} p_{22} c_2	-0.2636 0.7485 0.3164	1.335 0.07134 -1.456	-0.2381 0.8175 -0.1638	-1.668 -0.9591 5.381
p_{31} p_{32} c_3	-0.4423 0.7228 0.552	0.9635 0.1823 -1.401	-0.04555 -0.4413 2.498	-0.8691 -1.144 5.088
p_{41} p_{42} c_4	0.03968 0.656 -0.06219	0.4822 0.2814 -1.222	-0.05995 -0.8446 3.489	0.2794 -1.276 4.39
⋮	⋮	⋮	⋮	⋮
p_{231} p_{232} c_{23}	- - -	2.569 -0.2808 -2.536	-0.5526 0.1969 1.025	-4.865 1.752 3.674
p_{241} p_{242} c_{24}	- - -	3.875 5.577 2.182	-3.093 -4.732 -1.78	-7.614 -11.09 -4.322
p_{251} p_{252} c_{25}	- - -	0.1062 0.1394 0.05523	-0.09013 -0.1048 -0.04015	-0.1473 -0.1618 -0.06809

Maximum total error of the trajectory tracking in the second example is less than 1.1 mm.

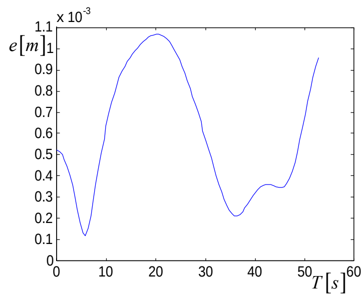

Fig. 25. The total tracking error

4. CONCLUSION

Results of both simulations, presented in this paper, show that the application of the neuro-fuzzy system to solving IK gives satisfactory results. Advantage of application of the Takagi-Sugeno controllers, implemented within the adaptive network, related to solving IK by application od the FLCs proposed in [4], lies in the fact that the step between inter-points on the straight-line trajectory, does not affect the accuracy of tracking.

The tracking error is smaller for the case of solving the IK by RBF neural networks [1]. However, the main deficiency of networks trained by data sets for learning, which enhance the part of the working space, is the fact that their hidden layers contain large number of neurons. The FLCs from the second example have smaller number of linguistic rules than the number of the invisible units of the RBF networks generated for solving the same problem. For solving the IK during the straight-line motion along the path from point A(1.366 0 2.666) to point B(1.6 0.3 2.3) with the velocity v = 0.01 m/s, the number of neurons of the hidden layer of the first RBF network is 20, while the number of rules FLC 1 is 9. The RBF 2 and RBF 3 have 186 and 174 neurons, respectively, and FLC 2 and FLC 3 have 25 rules, each. Simulated FLCs have larger number of parameters, which are being adapted. Weights of connections between invisible units and outputs, in RBF networks are constant values, while the consequences functions for Takagi-Sugeno controllers are linear functions of inputs into the controller.

In realization of fast trajectories, advantage for solving IK should be given to Takagi-Sugeno controllers with respect to the RBF neural networks [6]. Due to large number of neurons, RBF neural networks cannot realize mapping from the external to internal coordinates in real time for high gripper velocities.

If it is necessary to realize trajectory with an error less than 1 mm, then the data set for training should enhance only trajectories, which the gripper tracks during the task execution.

REFERENCES

1. Godjevac, J., (1997) "A method of the design of neuro-fuzzy controllers; an application in robot learning," Doctorial dissertation, EPFL, Lausanne, Switzerland.
2. Jang, S. R., (1993) "ANFIS: Adaptive-Network-Based Fuzzy Inference System," *IEEE Trans. on Systems, Man and Cybernetics,* vol. 23, no. 3, pp. 665-685.
3. Jang, S. R., C. T. Sun, (1993) "Functional Equivalence between Radial basis Function Networks and Fuzzy Inference Systems," *IEEE Trans. on Neural Networks,* vol. 4, no. 1, pp. 156-159.
4. Nedungadi, A., (1995) "Application of Fuzzy Logic to Solve the Robot Inverse Kinematic Problem," in Proc. of *Fourth World Conference Robotics Research*, Pittsburgh, PA, USA, pp. 13-1 - 13-14.
5. Nikolić, I., M. Janković, (1998) "Fuzzy Controller for Solving the Inverse Kinematics of Industrial Robots", Proceedings of the *XXVII ETRAN Conference*, Vrnjačka Banja, Volume 3, pp. 297-300.
6. Ranković, V., I., Nikolić, (2001) "Solving of the Robots' Inverse Kinematics by Two-Layer Neural Network", Proceedings of *XLV ETRAN Conference*, Bukovička Banja, Volume 3, pp. 127-130.

PRIMENA TAKAGI-SUGENO FUZZY KONTROLERA ZA RE[AVANJE PROBLEMA INVERZNE KINEMATIKE ROBOTA

Vesna Rankovi}, Ilija Nikoli}

Za preslikavanje iz spoljašnjih u unutrašnje koordinate je korišćen Takagi-Sugeno kontroler, implementiran u adaptivnu mre`u, koji je po arhitekturi sličan RBF neuronskoj mre`i, pa se takodje naziva neuro fuzzy sistemom. Tokom procesa u~enja su adaptirani parametri funkcija pripadnosti primarnih fuzzy skupova i parametri posledica Takagi-Sugeno kontrolera. Prikazani rezultati se odnose na pra}enje pravolinijske trajektorije sa konstantnom brzinom hvataljke i trougaonim profilom brzine.