NEURO-FUZZY MODELS AND ALGORITHMS FOR IMPROVING THE DYNAMIC CHARACTERISTICS OF MIMO CONTROL SYSTEMS FOR A MANIPULATOR ROBOT

A. V. Smirnov* and Yu. A. Bykovtsev**

*,**MIREA — Russian Technological University, Moscow, Russia

*✉ alexander-smirnov-avs@yandex.ru, **✉ bykovcev@mirea.ru

Abstract. The theory of Multi-Input Multi-Output (MIMO) control systems is an effective approach when dealing with manipulator robots (manipulators) and other objects with a state vector characterized by the mutual influence of components. In the case of a manipulator, the torques that determine static and dynamic loads on the shafts of the actuator subsystem depend nonlinearly on the values of generalized variables. At the same time, the dynamic model describing this dependence often includes several uncertainties related to a specific task being executed, operational environment conditions, robot design, and changes in system parameters during operation. In this regard, it is topical to apply knowledge processing technologies that expand the operational capabilities of control systems under uncertainty. In particular, approximation problems of nonlinear functions of several variables (which include intelligent MIMO control of manipulators) are effectively solved by systems based on neuro-fuzzy approaches; among the latter, the Adaptive-Network-Based Fuzzy Inference System (ANFIS) technology has become the most widespread. This paper considers a modified ANFIS structure for approximating the solution of the manipulator's inverse dynamics problem as well as its software and algorithmic implementation. Experimental studies are conducted to compare the proposed modification with an approximator based on a set of conventional ANFIS structures and the corresponding control systems. As shown, the proposed approach significantly speeds up the fuzzy inference procedure and, consequently, improves the tracking accuracy of a reference trajectory by a MIMO control system.

Keywords: manipulator robots, intelligent control, neuro-fuzzy control, ANFIS.

INTRODUCTION

The complexity of implementing Multi-Input Multi-Output (MIMO) control systems for manipulator robots (further called manipulators for brevity) is due to the significant nonlinearity of mutual influences between the degrees of freedom, the parametric uncertainty of the dynamic model, and the unpredictability of the operational environment and exogenous disturbances [1–3]. These aspects restrict the application of classical control methods. At the same time, intelligent approaches expand their capabilities under significant uncertainty [4]. In particular, a manipulator can be effectively regulated by a *neuro-fuzzy* [5–7] controller. Note that a neuro-fuzzy structure is a fuzzy inference system with parameters tuned by common neural network training methods (least squares, backpropagation, etc.).

This approach is generally reduced to approximating *the solution of the inverse dynamics problem* (IDP) of a manipulator by a *neuro-fuzzy structure* and implementing a control algorithm based on the resulting structure. For example, within a neuro-fuzzy knowledge processing technology, manipulator control was implemented by direct dynamic compensation [8–10], feedback linearization [11, 12], sliding modes [13–15], and backstepping [16]. Judging by the cited research, this class of intelligent controllers ensures high-performance control under uncertainty.

Among neuro-fuzzy approaches, *Adaptive Network-Based Fuzzy Inference System* (*ANFIS*) *structures* [7] have become widespread recently. An analysis of their application in control systems [17–24] al-

lows concluding on the effectiveness of the ANFIS technology when approximating multivariable functions. Furthermore, according to the results of [25], the ANFIS-based implementation of a MIMO control system for manipulators ensures high dynamic accuracy during trajectory tracking. At the same time, note that hardware and software support for this class of neuro-fuzzy structures has received little attention: the above works were devoted to a general description of the control algorithms rather than to the practical issues of their implementation.

Below, we propose a *modification of the ANFIS architecture* to approximate the manipulator's IDP under hardware constraints imposed on the control system. Also, we describe the software implementation of the proposed structure and provide the results of *hardware-in-the-loop* (*HIL*) simulation of the control system. (As an example, a system is designed by *the feedback linearization method*, although systems of other types can be obtained using a trained ANFIS structure.)

1. CONTROLLED OBJECT AND CONTROL SYSTEM

In this paper, the controlled object is a manipulator. Its mathematical model represents a complex MIMO system characterized by torques and forces of various physical natures: gravitational, Coriolis, centrifugal, and inertial. Their combined description in the form of a matrix differential equation, known as the dynamics problem, is used to design most high-precision manipulator control systems.

The solution of the manipulator's inverse dynamics problem can be written as a matrix equation of the form [2]

$$\boldsymbol{\tau} = D(\mathbf{q})\ddot{\mathbf{q}} + H_1(\mathbf{q})\dot{\mathbf{q}}_{\mathbf{ij}} + H_2(\mathbf{q})\dot{\mathbf{q}}_{\mathbf{ii}} + C(\mathbf{q}), \qquad (1)$$

where $\boldsymbol{\tau}$ denotes the vector of torques generated by the electric actuators; $\mathbf{q}$ and $\ddot{\mathbf{q}}$ are the vectors of generalized coordinates and accelerations, respectively; $\dot{\mathbf{q}}_{\mathbf{ij}}$ and $\dot{\mathbf{q}}_{\mathbf{ii}}$ are the vectors containing the cross-products of the distinct and identical generalized velocities, respectively; $D(\mathbf{q})$ is the matrix relating the generalized accelerations to the components of the dynamic torques; $H_1(\mathbf{q})$ and $H_2(\mathbf{q})$ are the matrices relating the products of the generalized velocities to the Coriolis torques and the centrifugal torques, respectively; and finally, $C(\mathbf{q})$ is the vector of gravitational torques. The vectors $\boldsymbol{\tau}$, $\mathbf{q}$, $\ddot{\mathbf{q}}$, $\dot{\mathbf{q}}_{\mathbf{ii}}$, and $C(\mathbf{q})$ have dimensions $n_{\mathrm{DOF}} \times 1$, where n_{DOF} is the number of the manipulator's degrees of freedom); the vector $\dot{\mathbf{q}}_{\mathbf{ij}}$ has dimensions $n_{\mathrm{DOF}}(n_{\mathrm{DOF}} - 1)/2 \times 1$; and finally, the matrices $D(\mathbf{q})$ and $H_2(\mathbf{q})$ have dimensions $n_{\mathrm{DOF}} \times n_{\mathrm{DOF}}$ whereas the matrix $H_1(\mathbf{q})$ dimensions $n_{\mathrm{DOF}} \times n_{\mathrm{DOF}}(n_{\mathrm{DOF}} - 1)/2$.

We emphasize the significant nonlinearity of equation (1): the components of the matrices $D(\mathbf{q})$, $H_1(\mathbf{q})$, and $H_2(\mathbf{q})$, as well as those of the vector $C(\mathbf{q})$, depend in a complex manner on the values of the generalized coordinates $\mathbf{q}$. This aspect considerably complicates the control system design. Feedback linearization [26] is an effective approach for such controlled objects. Consider its application to build a manipulator's control law [26].

Assuming the control action is a torque vector (in N·m), we introduce the control vector $\mathbf{u}$ (equal to $\boldsymbol{\tau}$) into equation (1):

$$\mathbf{u} = D(\mathbf{q})\ddot{\mathbf{q}} + H_1(\mathbf{q})\dot{\mathbf{q}}_{\mathbf{ij}} + H_2(\mathbf{q})\dot{\mathbf{q}}_{\mathbf{ii}} + C(\mathbf{q}).$$

Note that the generalized acceleration vector $\ddot{\mathbf{q}}$ is linearly related to the control vector $\mathbf{u}$:

$$\ddot{\mathbf{q}} = D^{-1}(\mathbf{q})\left(\mathbf{u} - H_1(\mathbf{q})\dot{\mathbf{q}}_{\mathbf{ij}} - H_2(\mathbf{q})\dot{\mathbf{q}}_{\mathbf{ii}} - C(\mathbf{q})\right).$$

Then the control vector for the linearized system takes the form

$$\mathbf{v} = \ddot{\mathbf{q}}.$$

Let the control system error be defined as

$$\mathbf{e} = \mathbf{g} - \mathbf{q},$$

where $\mathbf{g}$ denotes the vector of reference signals. Since the system has order 2, the error satisfies the condition

$$\ddot{\mathbf{e}} + \gamma\left(\dot{\mathbf{e}} + \mathbf{e}\right) = \mathbf{0},$$

where γ is a positive coefficient. Therefore, the control vector for the linearized system can be written as

$$\mathbf{v} = \ddot{\mathbf{q}} = \ddot{\mathbf{g}} + \gamma\left(\dot{\mathbf{e}} + \mathbf{e}\right).$$

By substituting $\mathbf{v}$ and $\mathbf{u}$ into equation (1), we obtain the control law

$$\begin{aligned}\mathbf{u} &= D(\mathbf{q})\left(\ddot{\mathbf{g}} + \gamma\left(\dot{\mathbf{e}} + \mathbf{e}\right)\right) \\ &+ H_1(\mathbf{q})\dot{\mathbf{q}}_{\mathbf{ij}} + H_2(\mathbf{q})\dot{\mathbf{q}}_{\mathbf{ii}} + C(\mathbf{q}).\end{aligned} \qquad (2)$$

2. FUZZY INFERENCE ARCHITECTURE

One approach to constructing an ANFIS approximator for the manipulator's IDP was discussed in [25]; see the structural diagram in Fig. 1. It is based on the following principles:

- The components of equation (1) (corresponding to different physical moments for a given degree of freedom) are approximated using separate ANFIS structures.
- The generalized coordinates $\mathbf{q}$ and accelerations $\ddot{\mathbf{q}}$, as well as the products of the generalized velocities $\dot{\mathbf{q}}_{\mathbf{ij}}$ and $\dot{\mathbf{q}}_{\mathbf{ii}}$, are selected as input variables.

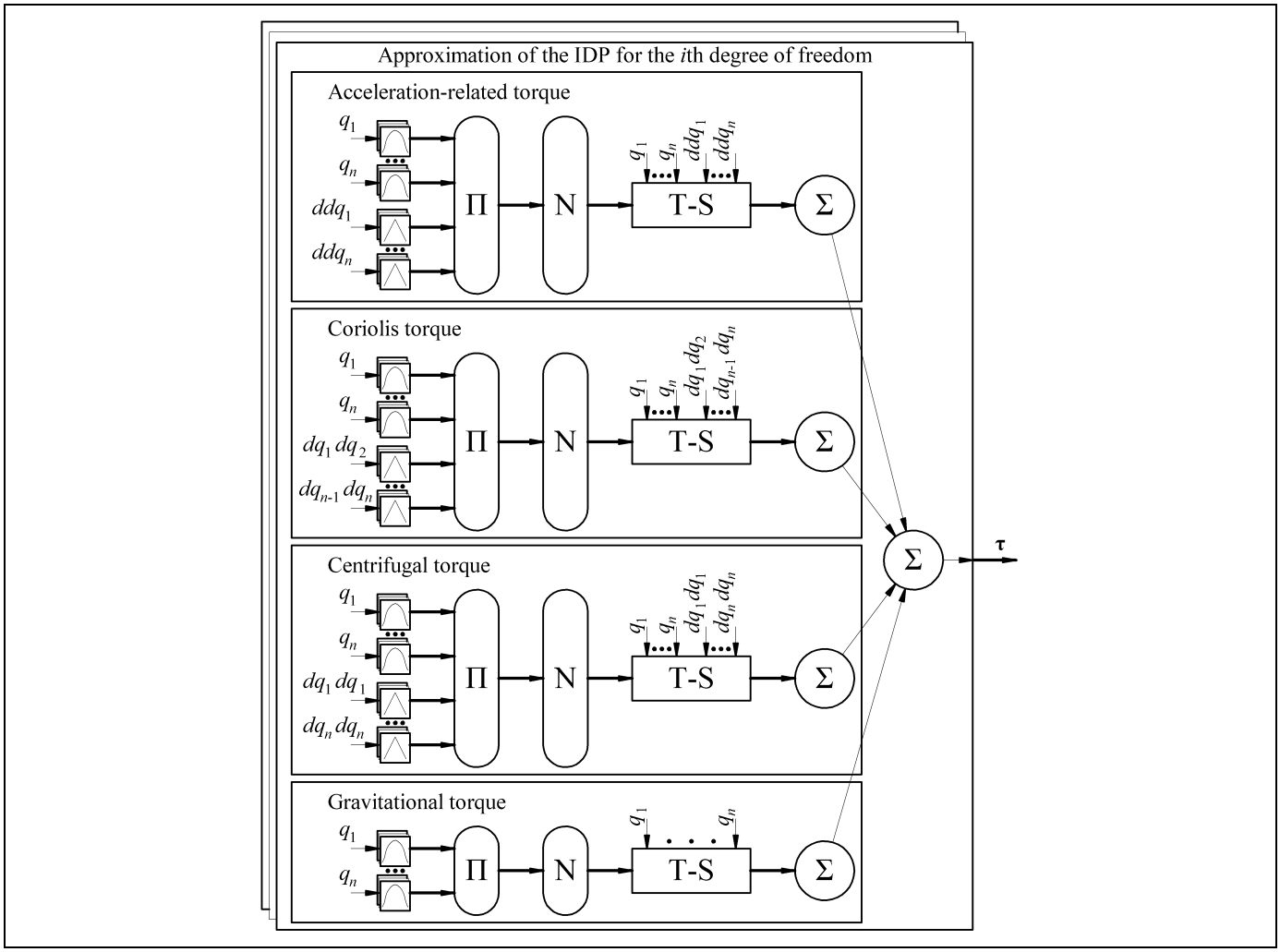

Fig. 1. The structural diagram of the approximator based on a set of conventional ANFISs.

• The generalized coordinates **q**, which influence the control torque vector **τ** nonlinearly, are fuzzified using Gaussian membership functions; and the generalized accelerations $\ddot{\mathbf{q}}$ and the products of the generalized velocities $\dot{\mathbf{q}}_{ij}$ and $\dot{\mathbf{q}}_{ii}$, which influence the vector **τ** linearly, are fuzzified using triangular membership functions.

• At most one acceleration $\ddot{\mathbf{q}}$ or product of the velocities $\dot{\mathbf{q}}_{ij}$ and $\dot{\mathbf{q}}_{ii}$ is included in each fuzzy rule.

Although the proposed approximator architecture has demonstrated its effectiveness [25], we note its redundancy: the same variables are fuzzified, and the identical fuzzy rules are applied, multiple times by different ANFIS structures. For instance, a manipulator with n_{DOF} degrees of freedom needs

$$n_{MF} = n_{\text{DOF}} \times \left(n_{\text{DOF}} \times n^1_{MF\_G} \right) + n^1_{MF\_T} \times n_{\text{DOF}} \times \left(n_{DOF} + \sum_{i=2}^{n_{\text{DOF}}} (i-1) + (n_{\text{DOF}} - 1) \right) \quad (3)$$

membership functions and

$$n_R = n_{\text{DOF}} \times \left({n^1_{MF\_G}}^{n} \times n^1_{MF\_T} \times \left(n_{\text{DOF}} + \sum_{i=2}^{n_{\text{DOF}}} (i-1) + (n_{\text{DOF}} - 1) \right) + {n^1_{MF\_G}}^{n} \right) \quad (4)$$

fuzzy rules, where $n^1_{MF\_G}$ ($n^1_{MF\_T}$) is the number of Gaussian (triangular, respectively) membership functions for a single variable.

Furthermore, this approach involves sequential training for each ANFIS structure. Therefore, to generate a training dataset, it is required either to solve equation (1) explicitly or to identify its components by processing the measurements obtained during system operation. The entire approximator can be trained simultaneously, albeit with a more complex training algorithm.

In this paper, we modify the above approximator by combining ANFIS structures into a single fuzzy inference topology with multiple outputs (Fig. 2). It is based on the following principles:

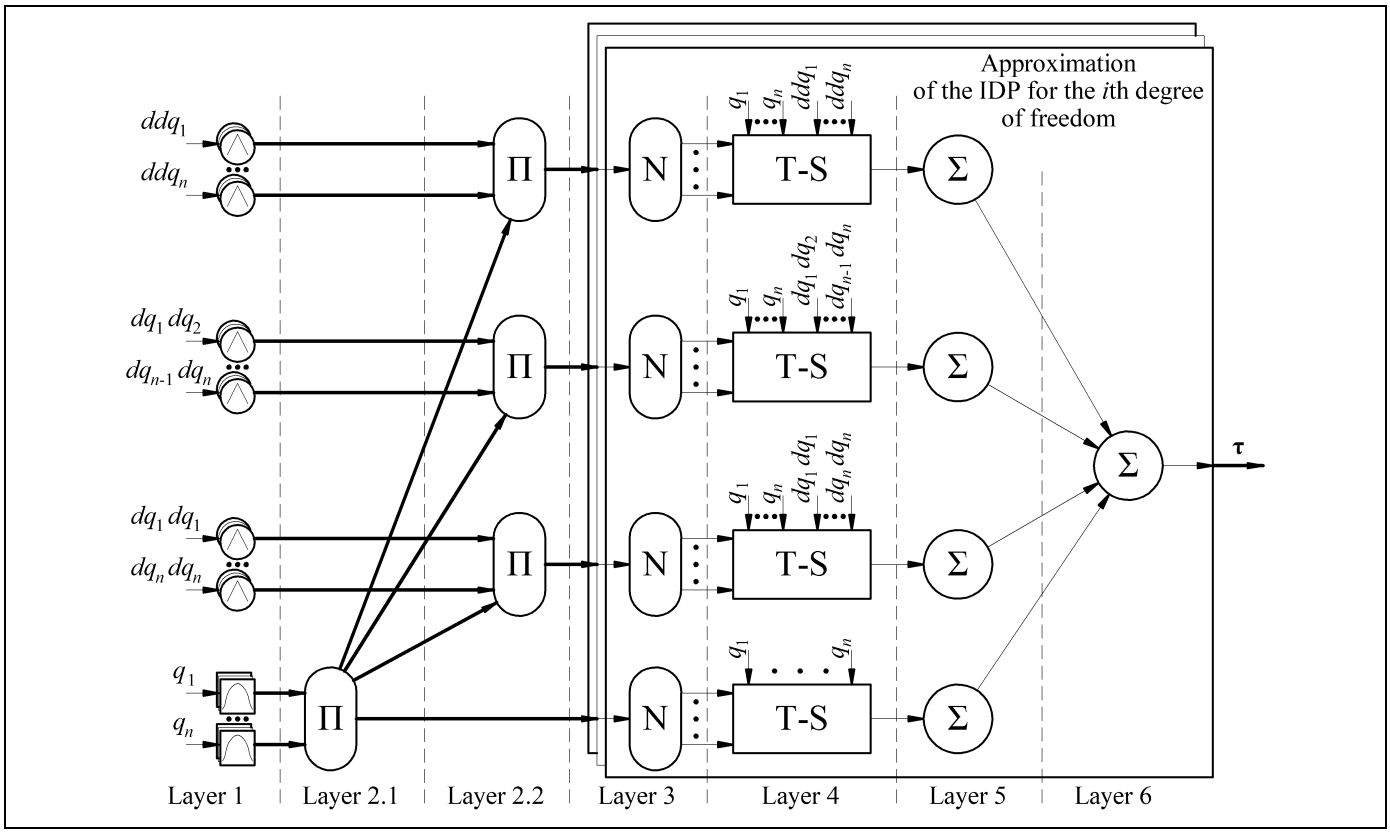

Fig. 2. The structural diagram of the approximator based on the proposed ANFIS modification.

• Fuzzification and processing of fuzzy rules (layers 1 and 2) are the same for the entire ANFIS approximator.

• The normalization of fuzzy rules and partial and overall defuzzifications (layers 3, 4, and 5) are separated with respect to the components of the IDP (1), being identical to the original approach.

• The additional (sixth) layer sums the IDP components according to their degrees of freedom. (A separate component can be derived if necessary.)

• The method for selecting input variables, the principles of fuzzification, and the formation of fuzzy rules are identical to the original approach; however, triangular membership functions are not considered adaptive (their parameters are non-tunable) since fuzzification affects only the variables entering equation (1) linearly.

• Training is executed both on individual components and on the entire structure simultaneously.

The proposed architecture needs

$$n_{MF} = n_{\text{DOF}} \times n^1_{MF\_G} + n^1_{MF\_T} \times \left(n_{\text{DOF}} + \sum_{i=2}^{n_{\text{DOF}}} (i-1) + (n_{\text{DOF}} - 1) \right) \quad (5)$$

membership functions and

$$n_R = {n^1_{MF\_G}}^n \times n^1_{MF\_T} \times \left(n_{\text{DOF}} + \sum_{i=2}^{n_{\text{DOF}}} (i-1) + (n_{\text{DOF}} - 1) \right) + {n^1_{MF\_G}}^n \quad (6)$$

fuzzy rules.

The comparative graphs in Figs. 3 and 4 show how the number of membership functions and fuzzy rules, respectively, depends on the number of degrees of freedom (see formulas (3)–(6)). Based on the above description, we emphasize the following potential advantages of the proposed modification:

• a significant reduction in the computational cost of the fuzzy inference procedure, due to fewer membership functions and fuzzy rules;

• higher training flexibility, with potential opportunities for retraining and self-learning;

• a reduction in the computational cost of the training procedure, due to a significant decrease in the number of variable parameters of the fuzzification layer. However, note that the total number of variable parameters will change slightly since their number in the partial defuzzification layer (layer 4) remains the same.

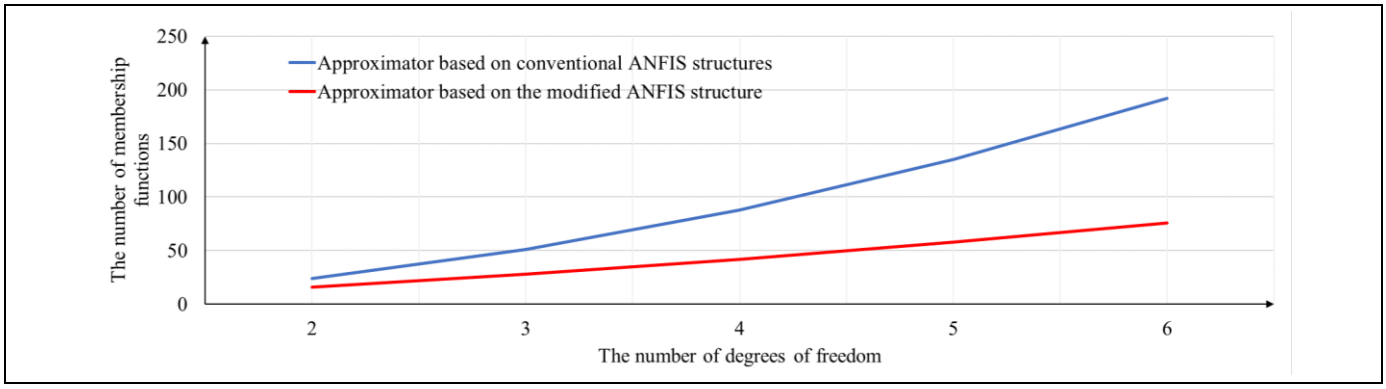

Fig. 3. The number of membership functions depending on the number of degrees of freedom for both approaches.

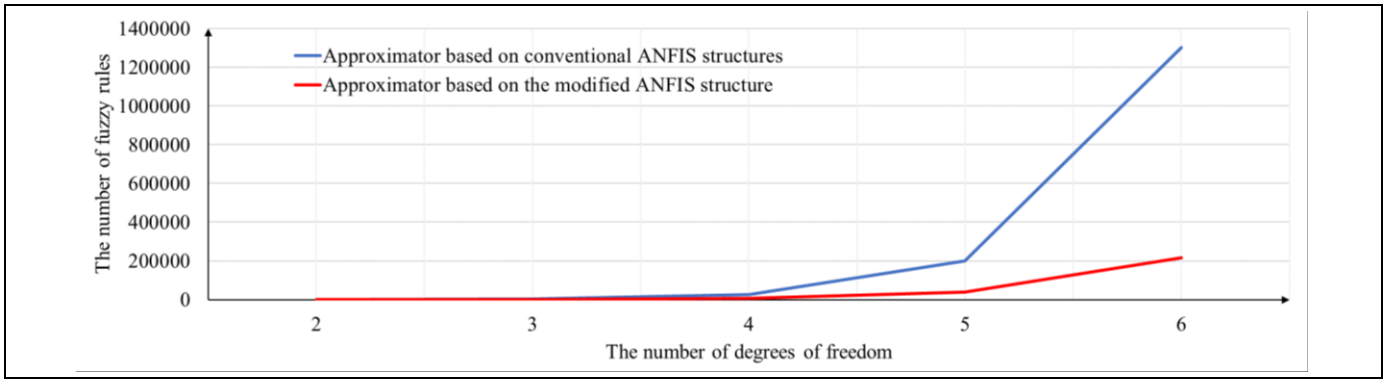

Fig. 4. The number of fuzzy rules depending on the number of degrees of freedom for both approaches.

To compare the described approaches objectively, we identify three key criteria: the speed of inference and training procedures, as well as approximation accuracy. The software implementation of the approximators and experimental studies are required to assess these criteria thoroughly.

3. DEVELOPMENT OF SOFTWARE AND ALGORITHMIC SUPPORT

For experimental studies of the ANFIS approximators above, software in Python was developed. Its key functionality, implemented as a library, concerns fuzzy inference and training procedures. The corresponding algorithms for the modified ANFIS structure are described below.

3.1. Fuzzy Inference Algorithm

The flowchart in Fig. 5 presents the fuzzy inference algorithm of the ANFIS modification. Consider its operation step by step. The vector of input variables (the generalized coordinates $\mathbf{q}$, accelerations $\ddot{\mathbf{q}}$, and the products of the generalized velocities $\dot{\mathbf{q}}_{ij}$ and $\dot{\mathbf{q}}_{ii}$) is supplied to the fuzzification layer (layer 1 in Fig. 2); and the resulting membership grades of the generalized coordinates $\mathbf{q}$ are processed by the first set of fuzzy rules (rules 1, see layer 2.1 in Fig. 2) corresponding to the vector of the gravitational torques $C(\mathbf{q})$ in equation (1). The rules associated with the other components of the solution of the IDP (1) (rules 2, see layer 2.2 in Fig. 2) are applied by multiplying the weights from the previous block (rules 1) by the membership grades of the generalized accelerations $\ddot{\mathbf{q}}$ and the products of the generalized velocities $\dot{\mathbf{q}}_{ij}$ and $\dot{\mathbf{q}}_{ii}$. Thus, this stage of the algorithm yields four vectors of the weights corresponding to all components of the solution of the IDP (1). Two of them, associated with the approximation of $C(\mathbf{q})$ and $D(\mathbf{q})\ddot{\mathbf{q}}$ (1), are immediately normalized (layer 3 in Fig. 2), since the sets of corresponding fuzzy rules are identical for the fuzzy inference of the IDP for all manipulator joints. (Normalization reduces all weights to a common scale, which is needed for the correct functioning of the fuzzy rules.) Next, the following sequence of steps is executed iteratively (in a loop) for each degree of freedom:

1) The necessary subsets of the products of the generalized velocities $\dot{\mathbf{q}}_{ij}$ and $\dot{\mathbf{q}}_{ii}$ and the weights of the fuzzy rules associated with the approximations of $H_1(\mathbf{q})\dot{\mathbf{q}}_{ij}$ and $H_2(\mathbf{q})\dot{\mathbf{q}}_{ii}$ (1) are identified.

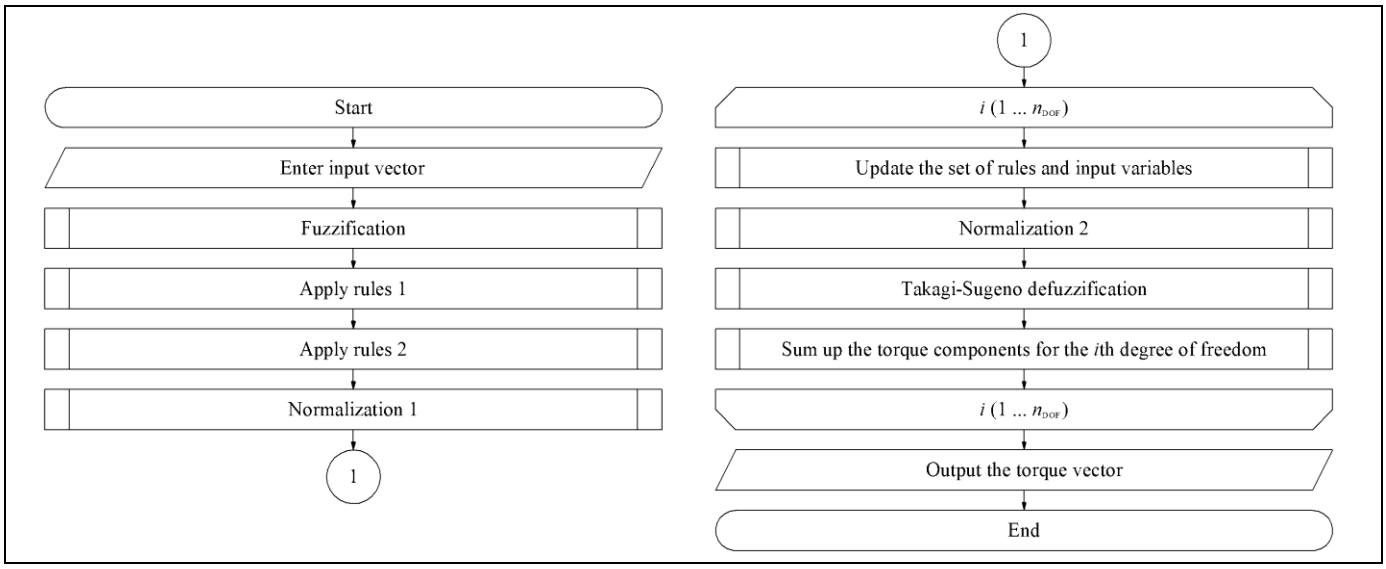

Fig. 5. The flowchart of the fuzzy inference algorithm.

2) The selected subset of the weights is normalized (normalization 2, see layer 3 in Fig. 2).

3) Each component is defuzzified using the Takagi–Sugeno model (layer 4 in Fig. 2).

4) The individual components of the approximated torque are summed (layers 5 and 6 in Fig. 2).

Upon completion of the loop, the algorithm outputs the vector of the manipulator's approximated torques.

3.2. Hybrid Training Algorithm

Similar to conventional ANFIS structures, the proposed modification is trained using the so-called hybrid method; see the original paper on ANFIS [7]. The flowchart in Fig. 6 shows the corresponding algorithm. A given number of epochs (n_{epoch}) is taken as the termination condition. A training dataset matrix with n_{dataset} examples is supplied to the input; it can be formed from the data representing either the complete solution of the IDP (1) for the nth degree of freedom of a manipulator (n = 1,…, n_{DOF}) or its individual components. Next, over a specified number of epochs, the least squares method and backpropagation are applied alternately to update the Takagi–Sugeno polynomial coefficients and to tune the parameters of the membership functions of the generalized coordinates, respectively. Let us briefly describe the mathematics of the training algorithms.

3.2.1. Tuning Defuzzification Parameters by the Least Squares Method

The parametric tuning problem of the Takagi–Sugeno polynomials can be described by a system of linear algebraic equations. We write this system in matrix form:

$$AX = B,$$

where A, X, and B are the system matrix and the vectors of unknowns and output parameters, respectively. When tuning an ANFIS structure (including the proposed modification), A, X, and B become

$$A = \begin{pmatrix} x_1^1 w_1^1 & \dots & x_k^1 w_1^1 & w_1^1 & \dots & x_1^1 w_m^1 & \dots & x_k^1 w_m^1 & w_m^1 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ x_1^p w_1^p & \dots & x_k^p w_1^p & w_1^p & \dots & x_1^p w_m^p & \dots & x_k^p w_m^p & w_m^p \end{pmatrix},$$

$$X = \begin{pmatrix} \alpha_1^1 & \dots & \alpha_{k+1}^1 & \dots & \alpha_1^m & \dots & \alpha_{k+1}^m \end{pmatrix}^{\mathrm{T}},$$

$$B = \begin{pmatrix} \tau^1 & \dots & \tau^p \end{pmatrix}^{\mathrm{T}},$$

where x_j^i is the value of the jth input variable from the ith record of the training dataset; w_j^i is the weight of the jth rule for the ith record of the training dataset; α_j^i is the jth coefficient of the Takagi–Sugeno polynomial for the ith rule; and finally, τ^i is the torque for the ith record of the training dataset.

The implemented least squares procedure is described as follows [7]:

$$\begin{cases} S_{i+1} = S_i - \dfrac{S_i a_{i+1} a_{i+1}^{\mathrm{T}} S_i}{1 + a_{i+1}^{\mathrm{T}} S_i a_{i+1}} \\ X_{i+1} = X_i + S_{i+1} a_{i+1} \left(\tau^i - a_{i+1}^{\mathrm{T}} X_i \right) \\ i = 1, \dots, n_{\text{epoch}}, \end{cases}$$

where a_i denotes the ith row of the state matrix A; n_{epoch} is the number of records in the training dataset; S_i is the covariance matrix for the ith iteration of the algorithm. For the first training epoch,

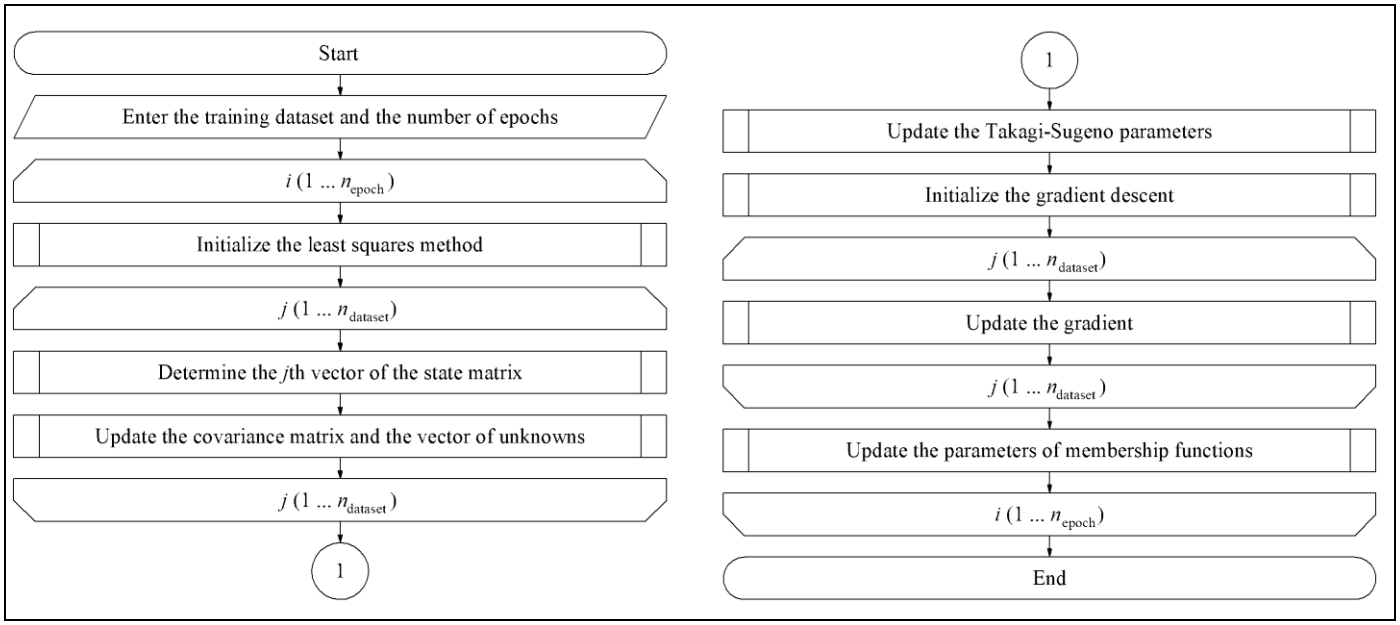

Fig. 6. The flowchart of the hybrid training algorithm.

$$S_1 = \gamma I ,$$

where I stands for an identity matrix of dimensions $(k + 1)m \times (k + 1)m$; γ is a positive number.

3.2.2. Tuning the Fuzzification Parameters by Backpropagation

We also mathematically describe the software implementation of the backpropagation method. To increase the convergence rate, Nesterov's accelerated gradient is used [27]:

$$\begin{cases} \Delta X_i = \gamma \times \Delta X_{i-1} + \eta \nabla_X E_i \left(X - \gamma \times \Delta X_{i-1} \right) \\ X = X - \Delta X_i \\ X = \begin{pmatrix} c_1 & \sigma_1 & \dots & c_{n_{MF\_G}} & \sigma_{n_{MF\_G}} \end{pmatrix}^{\mathrm{T}} \\ \nabla_X E_{n,p} = \begin{pmatrix} \dfrac{\partial E_{n,p}}{\partial c_1} & \dfrac{\partial E_{n,p}}{\partial \sigma_1} & \dots & \dfrac{\partial E_{n,p}}{\partial c_n} & \dfrac{\partial E_{n,p}}{\partial \sigma_n} \end{pmatrix}^{\mathrm{T}} \\ i = 1, \dots, n_{\text{epoch}}, \end{cases}$$

where X is the adaptive parameters of the fuzzification layer (the centers c_j and variances σ_j of the Gaussian membership functions, $j = 1,\dots, n_{MF\_G}$, with n_{MF_G} specifying the number of Gaussian membership functions in the fuzzification layer); ΔX_i is the increment of the vector X for the ith epoch; E_i is the loss function; γ is the accumulation coefficient; and finally, η is the convergence rate coefficient. Note that adaptive parameters are those tuned during the training process.

The partial derivatives of the loss function with respect to the parameters of the fuzzification layer are given by

$$\frac{\partial E_{n,p}}{\partial c_i} = -2e_{n,p} \sum_{\substack{j:O_i^1 \sim O_j^2 \\ v:O_j^4 \sim O_v^5}} \left(\left(f_i - O_v^5 \right) \frac{O_{j,p}^3}{O_{i,p}^1} \right) \frac{x_m - c_i}{\sigma_i^{\,2}} e^{-\frac{\left(x_m - c_i\right)^2}{2\sigma_i^{\,2}}},$$

$$\frac{\partial E_{n,p}}{\partial \sigma_i} = -2e_{n,p} \sum_{\substack{j:O_i^1 \sim O_j^2 \\ v:O_j^4 \sim O_v^5}} \left(\left(f_i - O_v^5 \right) \frac{O_{j,p}^3}{O_{i,p}^1} \right) \frac{\left(x_m - c_i\right)^2}{\sigma_i^{\,3}} e^{-\frac{\left(x_m - c_i\right)^2}{2\sigma_i^{\,2}}},$$

where O_{np}^k denotes the output of the nth node of the kth layer (the output of the fuzzy inference structure) for approximating the pth set of the input variables of the training dataset; x_m is the mth component of the vector X; and finally, $f_i(X)$ is the output of the ith Takagi–Sugeno polynomial.

4. EXPERIMENTAL STUDIES

4.1. Performance Comparison of the Approximators

This section addresses the problem of approximating the manipulator's dynamic model. The studies are intended to compare the accuracy and speed of the fuzzy inference procedure, as well as the training time for the above fuzzy inference approximators (their software and algorithmic implementations). In this context, the speed of the fuzzy inference procedure is critical from the perspective of implementing a control

system, which is essentially a real-time system. Note that this example considers supervised offline training.

Within this stage of experimental studies, all computations were carried out on a Raspberry Pi 5 single-board computer. This device is widely used in robotics applications due to its affordability and high performance (a quad-core Broadcom BCM2712 2.4 GHz CPU).

The solution of the IDP for a three-link PUMA-type manipulator [28] was taken for approximation. The training and test datasets were generated analytically from all combinations of the generalized variables (Table 1). Component-wise training was applied for both structures to approximate the individual components of the IDP (1).

Since the generalized accelerations $\ddot{\mathbf{q}}$ and the products of the generalized velocities $\dot{\mathbf{q}}_{\mathbf{ij}}$ and $\dot{\mathbf{q}}_{\mathbf{ii}}$ are linear variables, they can be effectively fuzzified using two membership functions, as confirmed in [25]. At the same time, a trade-off between the speed of the fuzzy inference procedure and the potentially achievable approximation accuracy is required when choosing the number of membership functions for the fuzzification of the generalized coordinates $\mathbf{q}$. Therefore, we conducted experimental studies by varying the number of membership functions from three to six.

The graphs in Figs. 7 and 8 show the time costs of the fuzzy inference and training procedures to achieve a given accuracy (10% and 5%, respectively) for both approximator structures. The error was computed by

$$e = \sqrt{\frac{1}{n_{\text{dataset}}} \sum_{i=1}^{n_{\text{dataset}}} e_i^2} \; / \; \Delta_{\text{out}},$$

where n_{dataset} specifies the dataset size; e_i is the absolute error for the ith dataset record; and finally, Δ_{out} is the output value range of the training dataset.

Based on the results obtained, we draw the following conclusions regarding the advantages and drawbacks of the proposed modification of the ANFIS approximator:

- Depending on the number of membership functions used for the fuzzification of the generalized coordinates $\mathbf{q}$, the modification provides a performance gain of up to 750%.
- In some cases, the modification requires more training time to achieve the specified accuracy.
- For both structures, the relationship between the training time and the number of membership functions used to fuzzify the generalized coordinates $\mathbf{q}$ is non-monotonic. Thus, the minimum possible number of membership functions is not always optimal in terms of training speed.
- In one case, achieving the specified accuracy with the modified structure required more membership functions to fuzzify the generalized coordinates $\mathbf{q}$.

4.2. HIL Simulation of the Control System

This section presents the results of the second stage of the experimental studies, i.e., an assessment of the tracking accuracy of a reference trajectory by the manipulator's control systems with the control law (2) based on the above approximators. For the fuzzy inference structures of both approximators, the target training accuracy was set to 5%. According to the results of subsection 4.1, the number of membership functions to fuzzify the generalized coordinates $\mathbf{q}$ was set to 4 and 5 for the approximators based on conventional and modified ANFIS structures, respectively (the smallest number ensuring the specified accuracy).

Table 1

Training and test datasets

No.	q_2, °		q_3, °		$\dot{q}_1$, $\dot{q}_2$, and $\dot{q}_3$, rad/s		$\ddot{q}_1$, $\ddot{q}_2$, and $\ddot{q}_3$, rad/s^2	
	Training dataset	Test dataset	Training dataset	Test dataset	Training dataset	Test dataset	Training dataset	Test dataset
1	–225	–205	–45	25	–5	–4	–100	–50
2	–185	–165	–5	15	–2.5	–1	0	50
3	–145	–125	35	55	0	1	100	–
4	–105	–90	75	90	2.5	4	–	–
5	–75	–55	104	125	5	–	–	–
6	–35	–15	145	165	–	–	–	–
7	5	25	185	205	–	–	–	–
8	45	–	225	–	–	–	–	–

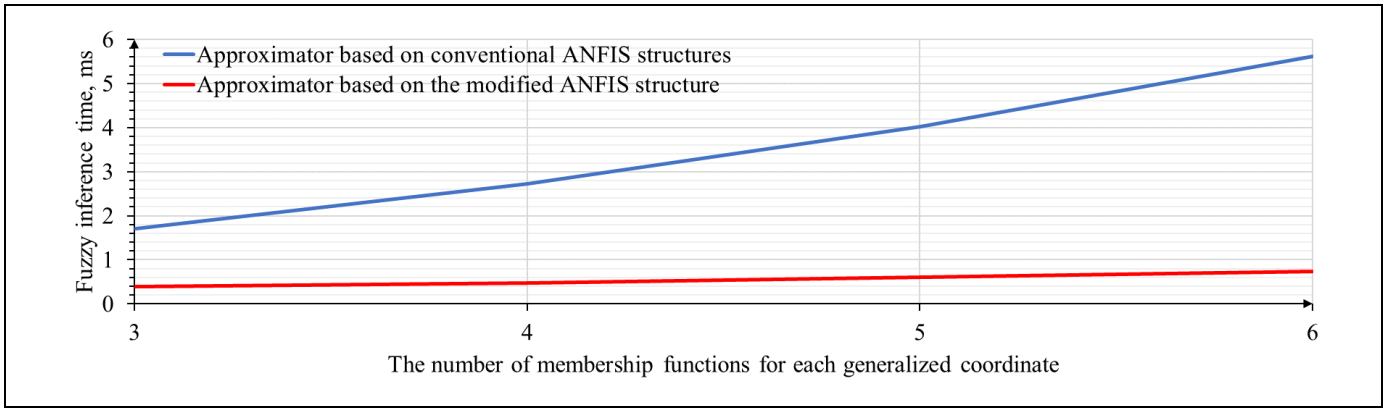

Fig. 7. The fuzzy inference time depending on the number of membership functions used to fuzzify generalized coordinates.

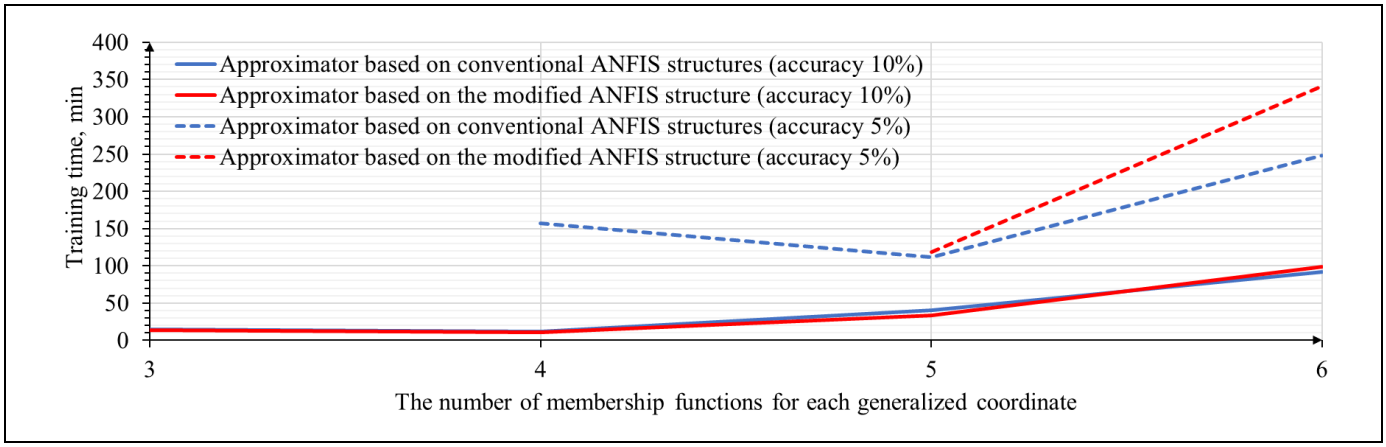

Fig. 8. The training time depending on the number of membership functions used to fuzzify generalized coordinates.

The controlled object was a three-link manipulator with PUMA-type kinematics; its dynamic model was used to generate the training and test data (subsection 4.1). During the studies, the object was simulated on a personal computer in real time. The computations associated with the operation of the neuro-fuzzy controller were executed on a Raspberry Pi 5 single-board computer. A serial interface was used for data exchange. The control system model is demonstrated in Fig. 9. Let us briefly describe the blocks:

– The serial port block sends the current generalized coordinates to the single-board computer and receives control signals.

– The manipulator is modeled as a rigid-body robot with PUMA-type kinematics.

– The error calculation block evaluates the tracking accuracy of a reference trajectory.

– The trajectory generator outputs a polynomial curve in Cartesian coordinates connecting the start and end points of the trajectory and solves the inverse kinematic problem for the current time instant. (A trajectory identical to this curve is stored in the single-board computer software as an array and is used to generate control system setpoints.)

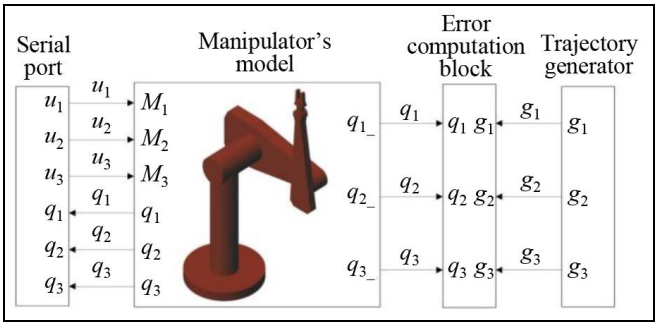

Fig. 9. The control system model.

During the simulation, the specified test trajectory in generalized coordinates (Fig. 10) was traced. Table 2 summarizes the maximum absolute tracking errors for each degree of freedom of the control systems with both types of approximators. According to the simulation results, the control system based on the ANFIS modification improves the tracking accuracy by up to a factor of two.

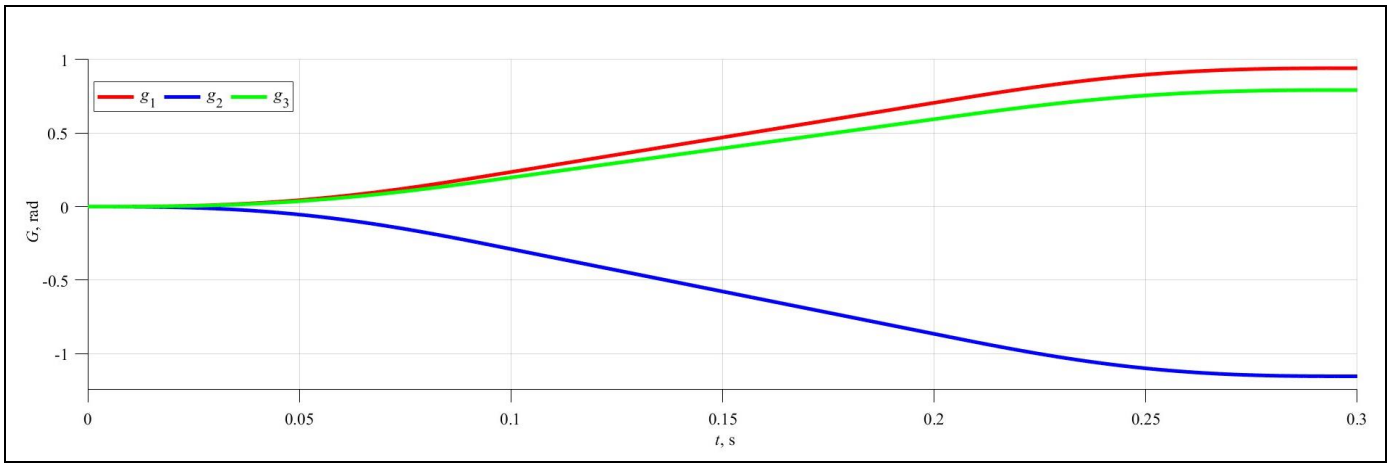

Fig. 10. The test trajectory in generalized coordinates.

Table 2

Simulation results

Approximator	$\|e_1\|_{max}$, rad	$\|e_2\|_{max}$, rad	$\|e_3\|_{max}$, rad
Based on conventional ANFIS structures	0.0071	0.0060	0.0063
Based on the modified ANFIS structure	0.0027	0.0045	0.0049

CONCLUSIONS

In this paper, we have investigated a control system for a multi-link manipulator robot based on a modified ANFIS architecture intended to approximate the robot's dynamic model. The software implementation of the proposed neuro-fuzzy inference system and approximator based on conventional ANFIS architectures has been considered. Experimental studies have been conducted to assess the approximation quality and the speed of the fuzzy inference and learning procedures. According to the results, the proposed modification allows speeding up the fuzzy inference procedure by up to 750% (depending on the number of membership functions used to fuzzify the generalized coordinates). HIL simulations have also been conducted to study trajectory tracking by a MIMO control system with both the proposed approximator and an approximator based on conventional ANFIS structures. The proposed modification ensures up to a twofold increase in tracking accuracy, due to its higher speed and, consequently, lower delay. The above results open up prospects for designing special-purpose microprocessor hardware and software control systems based on the modified ANFIS structure. Owing to higher speed and lower memory requirements, such systems could potentially be built using domestic components. Further development of the approach—the solution of the self-learning problem—opens up prospects for building a high-precision MIMO control system for a manipulator operating under uncertainty.

REFERENCES

1. Yurevich, E.I., *Osnovy robototekhniki* (Foundations of Robotics), 4th ed., St. Petersburg: BKhV-Peterburg, 2020. (In Russian.)
2. Fu, K.S., Gonzales, R.C., and Lee, C.S.G., *Robotics: Control, Sensing, Vision, and Intelligence*, New York: McGraw Hill, 1987.
3. Korendyasev, A.I., Salamandra, B.L., and Tyves, L.I., *Teoreticheskie osnovy robototekhniki* (Theoretical Foundations of Robotics), Moscow: Nauka, 2006. (In Russian.)
4. *Intellektual'nye sistemy avtomaticheskogo upravleniya* (Intelligent Automatic Control Systems), Makarov, I.M. and Lokhin, V.M., Eds., Moscow: Fizmatlit, 2001. (In Russian.)
5. Takagi, H. and Hayashi, I., NN-Driven Fuzzy Reasoning, *International Journal of Approximate Reasoning*, 1991, vol. 5, no. 3, pp. 191–212.
6. Jang, J.S.R., Fuzzy Modeling Using Generalized Neural Networks and Kalman Filter Algorithm, *Proceedings of the 9th National Conference on Artificial Intelligence*, 1991, vol. 2, pp. 762–767.
7. Jang, J.S.R., ANFIS: Adaptive-Network-Based Fuzzy Inference System, *IEEE Transactions on Systems, Man, and Cybernetics*, 1993, vol. 23, no. 3, pp. 665–685.
8. Fahmy, A.A. and Abdel Ghany, A.M., Neuro-fuzzy Inverse Model Control Structure of Robotic Manipulators Utilized for Physiotherapy Applications, *Ain Shams Engineering Journal*, 2013, vol. 4, no. 4, pp. 805–829.
9. Gierlak, P., Muszynska, M., and Zylski, W., Neuro-Fuzzy Control of a Robotic Manipulator, *International Journal of Applied Mechanics and Engineering*, 2014, vol. 19, no. 3, pp. 575–584.
10. Xia, K., Gao, H., and Ding, L., Trajectory Tracking Control of Wheeled Mobile Manipulator Based on Fuzzy Neural Network and Extended Kalman Filtering, *Neural Computing and Applications*, 2016, vol. 30, no. 2, pp. 447–462.

11. Ngo, T., Wang, Y., and Mai, Y.L., Robust Adaptive Neural-Fuzzy Network Tracking Control for Robot Manipulator, *International Journal of Computers, Communications & Control*, 2012, vol. 7, no. 2, pp. 341–452.
12. Mustapha, S., Fayçal, K.M., and Mohammed, S., Sequential Adaptive Fuzzy Inference System Based Intelligent Control of Robot Manipulators, *International Journal of Intelligent Systems and Applications*, 2014, vol. 6, no. 11, pp. 49–56.
13. Piltan, F., Sulaiman, N., and Nasiri, H., Novel Robot Manipulator Adaptive Artificial Control: Design a Novel SISO Adaptive Fuzzy Sliding Algorithm Inverse Dynamic Like Method, *International Journal of Engineering*, 2011, vol. 5, no. 5, pp. 399–418.
14. Wai, R.J. and Muthusamy, R., Fuzzy-Neural-Network Inherited Sliding-Mode Control for Robot Manipulator Including Actuator Dynamics, *IEEE Transactions on Neural Networks and Learning Systems*, 2013, vol. 24, no. 2, pp. 274–287.
15. Xuan, Q.N., Cong, C.N., and Ba, N.N., Robust Adaptive Trajectory Tracking Sliding Mode Control for Industrial Robot Manipulator Using Fuzzy Neural Network, *Journal of Robotics and Control*, 2024, vol. 5, no. 2, pp. 490–499.
16. Mai, T.L. and Wang, Y., Adaptive-Backstepping Force/Motion Control for Mobile-Manipulator Robot Based on Fuzzy CMAC Neural Networks, *Control Theory Tech*., 2014, vol. 12, no. 4, pp. 368–382.
17. Guo, Y. and Mohamed, M.E.A., Speed Control of Direct Current Motor Using ANFIS Based Hybrid P-I-D Configuration Controller, *IEEE Access*, 2020, vol. 8. DOI: 10.1109/ACCESS.2020.3007615
18. Shanmugasundaram, R., Ganesh, C., and Singaravelan, A., High-Performance ANFIS-Based Controller for BLDC Motor Drive, *Proceedings of ICUIS*, Singapore, 2021, pp. 435–449.
19. Ganesan, R., Suresh, S., and Sivaraju, S.S., ANFIS Based Multi-Sector Space Vector PWM Scheme for Sensorless BLDC Motor Drive, *Microprocessors and Microsystems*, 2020, vol. 76, no. 3, art. no. 103091.
20. Mopidevi, S., Kiransai, D., SSSR Sarathbabu, D., et al., Design, Control and Performance Comparison of PI and ANFIS Controllers for BLDC Motor Driven Electric Vehicles, *Measurement: Sensors*, 2024, vol. 31, no. 2/3, art. no. 101001.
21. Intidam, A., El Fadil, H., Housny, H., et al., Development and Experimental Implementation of Optimized PI-ANFIS Controller for Speed Control of a Brushless DC Motor in Fuel Cell Electric Vehicles, *Energies*, 2023, vol. 16, no. 11, art. no. 16114395.
22. Babes, B., Latrèche, S., and Bouafassa, A., A dSPACE-Based Implementation of ANFIS and Predictive Current Control for a Single Phase Boost Power Factor Corrector, *Scientific Reports*, 2024, vol. 14. DOI: https://doi.org/10.1038/s41598-024-63740-2
23. George, M.A., Kamat, D.V., and Kurian, C.P., Electric Vehicle Speed Tracking Control Using an ANFIS-Based Fractional Order PID Controller, *Journal of King Saud University. Engineering Sciences*, 2024, vol. 36, no. 4, pp. 256–264.
24. Bekhiti, B., Al-Sabur, R., and Roudane, M., Intelligent Neuro-Fuzzy Adaptive MIMO Control for a Self-balancing Two Wheeled Autonomous Robot via Recursive Resolution of the Matrix Diophantine Equation, *Discover Robotics*, 2025, vol. 1, no. 11. DOI: https://doi.org/10.1007/s44430-025-00011-3
25. Smirnov, A.V. and Bykovtsev, Y.A., Development of Adaptive Neuro-Fuzzy Inference Systems Technology to Improve the Dynamic Characteristics of Control Systems for Manipulative Robots, *The Electronic Scientific Journal "IT-Standard"*, 2025, no. 4, pp. 38–49. (In Russian.)
26. Kim, D.P., *Teoriya avtomaticheskogo upravleniya. Mnogomernye, nelineinye, optimal'nye i adaptivnye sistemy* (Automatic Control Theory. Multidimensional, Nonlinear, Optimal, and Adaptive Systems), 3rd ed., Moscow: Yurait, 2023. (In Russian.)
27. Nesterov, Yu.E., A Method of Solving a Convex Programming Problem with Convergence Rate $O(\frac{1}{k^2})$, *Sov. Math., Dokl.*, 1983, vol. 27, pp. 372–376.
28. Armstrong, B., Khatib, O., and Burdick, J., The Explicit Dynamic Model and Inertial Parameters of the PUMA 560 Arm, *Proceedings of 1986 IEEE International Conference on Robotics and Automation*, San Francisco, 1986, pp. 510–518.

This paper was recommended for publication by S.A. Krasnova, a member of the Editorial Board.

Received March 6, 2025,
and revised May 17, 2026.
Accepted May 17, 2026.

Author information

Smirnov, Aleksandr Vasil'evich. Student, MIREA — Russian Technological University, Moscow, Russia
✉ alexander-smirnov-avs@yandex.ru
ORCID iD: https://orcid.org/0009-0009-8147-2141

Bykovtsev, Yuri Alekseevich. Cand. Sci. (Eng.), MIREA — Russian Technological University, Moscow, Russia
✉ bykovcev@mirea.ru
ORCID iD: https://orcid.org/0009-0003-6671-5674

Cite this paper

Smirnov, A.V. and Bykovtsev, Yu.A., Neuro-Fuzzy Models and Algorithms for Improving the Dynamic Characteristics of MIMO Control Systems for a Manipulator Robot. *Control Sciences* **4**, 65–75 (2026).

Original Russian Text © Smirnov, A.V., Bykovtsev, Yu.A., 2026, published in *Problemy Upravleniya*, 2026, no. 4, pp. 77–89.

Translated into English by *Alexander Yu. Mazurov*,
Cand. Sci. (Phys.–Math.),
Trapeznikov Institute of Control Sciences,
Russian Academy of Sciences, Moscow, Russia
✉ alexander.mazurov08@gmail.com