Received: October 19, 2025. Revised: November 30, 2025.

International Journal of
Intelligent Engineering & Systems

http://www.inass.org/

Model with Neural Network Component for Adaptive Manipulator Control under Variable Load

Amer Abu-Jassar[1]* **Mohammad Hamdan[2]** **Nowfal Aweisi[3]** **Mahmoud Howaidi[4]**
Vladyslav Yevsieiev[5] **Vyacheslav Lyashenko[6]**

[1]*Department of Computer Science, College of Information Technology, Amman Arab University, Amman, Jordan*
[2]*College of Cyber security, Amman Arab University, Amman, Jordan*
[3]*School of Electrical Engineering and Information Technology, German Jordanian University, Amman, Jordan*
[4]*Department of Electrical and Computer Engineering, Kennesaw State University, Georgia, USA*
[5]*Department of Computer-Integrated Technologies, Automation and Robotics, Kharkiv National University of Radio Electronics, Kharkiv, Ukraine*
[6]*Department of Media Systems and Technology, Kharkiv National University of Radio Electronics, Kharkiv, Ukraine*
* Corresponding author's Email: A.abujassar@aau.edu.jo

Abstract: The method of adaptive control of a manipulator under variable load using a neural-adaptive PID controller, which combines the classical principles of the theory of automatic control with intelligent self-learning algorithms, is considered. The purpose of this study is to develop a mathematical model of the manipulation system and implement a method for adaptive compensation of dynamic disturbances resulting from changes in the mass of the working tool or load. The object of analysis is the nonlinear system "manipulator – regressor-compensation – adaptive PID", for which the dynamics equation, adaptation laws are constructed, and numerical simulation of control processes is carried out. The novelty of such a study lies in the use of a combined approach based on the regressor $Y(q,\dot{q},\ddot{q})\hat{\theta}(t)$ and variable coefficients $K_p(t), K_i(t), K_d(t)$, which are updated in real time using a neural network to compensate for changes in the inertial parameters of the system. During numerical simulation, it was found that when the load changes by 50%, the adaptive PID provides a deviation of no more than 0.04 rad with a settling time of about 3.2 s, and the neural model reduces the tracking error to the level of 0.01 rad. Analysis of the evolution of the coefficients showed stabilization of K_p At about 20.03, which indicates the consistency of the adaptive mechanism. The results obtained confirm the adequacy of the mathematical models and prove the effectiveness of the proposed approach in ensuring stable motion of the manipulator under variable dynamic conditions. The proposed approach can be applied in collaborative robot systems, robotic manipulators, and mechatronic complexes with high requirements for accuracy and adaptability.

Keywords: Adaptive control, Manipulator, Variable load, Neural network, Regressor-compensation, Adaptive neural PID, System stability, Modelling.

1. Introduction

One of the key tasks of the development of modern robotics is to ensure reliable and accurate control of robotic manipulators that work together with a person in a production environment [1-5]. Variable parameters, in particular the mass and position of the captured object, significantly affect the dynamics of the system, which leads to deviations from the desired trajectory and a decrease in the accuracy of operations. Traditional control methods do not always provide the necessary robustness and stability under such conditions, so the use of adaptive approaches becomes relevant [6-10]. In particular, adaptive control based on regressor compensation and an adaptive Proportional–Integral–Derivative (PID) controller allows the system to change its parameters in real time and compensate for uncertainties and external disturbances [11-14]. Of particular importance is the use of neural methods for approximating nonlinearities, which increases the efficiency of control in a variable environment [15-17]. This creates a basis for practical implementation in the areas of assembly, transportation, and welding, where flexibility and reliability in the work of a person and a robot are required. Thus, the study of adaptive control of a manipulator under variable load is a relevant step towards the creation of intelligent collaborative systems within the framework of the concepts of Industry 5.0 [18-20].

The structure of the article is constructed in such a way that after the introduction, a review of recent research and publications is provided, then the development of a mathematical model of adaptive control is presented, the laws of adaptation and the modeling methodology are described, after which the results of the experiments are presented, their discussion and comparison with relevant approaches are presented, and generalized conclusions are made.

2. Analysis of recent research and publications

In modern research on the control of robotic manipulators, there is an active development of adaptive and intelligent methods that combine classical approaches with elements of artificial intelligence. Particular attention is paid to the use of neural networks and fuzzy models capable of self-learning and compensation of uncertainties in the dynamics of the system. Such approaches allow for increasing the accuracy, stability, and robustness of control in cases of variable load or the influence of external disturbances, which is critically important for collaborative robots of a new generation. That is why it is advisable to analyze scientific publications devoted to neuro-adaptive methods of controlling manipulation systems.

For example, in the work of A. Elmogy, N. Alhemaly, H. El-Ghaish, and W. Elawady, an “enhanced neuro-adaptive PID sliding mode control” approach for manipulators was proposed, which combines the sliding mode with neural adaptation of PID parameters and is aimed at increasing the reliability and stability of automation [21]. Such a solution allows for significantly increasing the robustness to uncertainties and unforeseen disturbances due to the combination of the invulnerability of sliding control and online correction via Neural Network (NN). This can be used to combine the sliding surface with neural correction as a mechanism for rapid compensation of sudden changes in payload and external impulse influences. However, the use of pure sliding may be undesirable in physical Human-Robot Interaction scenarios (pHRI scenarios) due to possible chattering control signals (high-frequency oscillations) of the control signal. That is, smoothing mechanisms must be applied for human safety.

In the study [22], mathematical support for adaptive control for an intelligent gripper of a collaborative manipulator, including identification models and adaptation laws, has been developed. This makes it possible to more accurately account for changes in mass and load distribution and to improve the accuracy of gripping and holding the part. The results of the study [22] are useful for the practical implementation of payload estimation and feedforward compensation during gripping moments. However, the identification methods proposed in the work [22] may require additional sensors (such as force/torque sensors), which are not always available on existing platforms. Therefore, adaptation to limited hardware resources is required.

In the article by C. Yıldırım and V. G. Böcekçi, a neuro-adaptive PID for satellite-based marine tracking systems is developed, which demonstrates high accuracy in the presence of nonlinear disturbances and time delays [23]. This approach allows for improved spatial stabilization and adaptation to changing environmental conditions [23]. It is also useful as an example of how the neural part can compensate for complex nonlinear influences and sensor delays. However, the characteristics of a marine platform (long time constants, inertial phenomena) are different from fast manipulator dynamics, so it is necessary to adjust the adaptation parameters and regressor structure to the fast manipulator time scales.

X. Wang, C. Li, D. Cai, and Y. Cui proposed an adaptive variable stiffness method based on the Adaptive Neuro-Fuzzy Inference System (ANFIS), which combines fuzzy neural rules with adaptive impedance control [24]. This allows for smooth impedance changes in response to environmental conditions and provides safer human interaction. These ideas can also be integrated to provide variable stiffness of the end effector when capturing different masses. However, the complexity of ANFIS and the need for expert rule tuning can make online implementation difficult on limited computing resources, so it is advisable to consider a more compact neural approximation or local Radial Basis Function networks (RBF-blocks).

The review [25] systematizes modern approaches to manipulator control, highlighting the strengths of adaptive, robust, and neuro-hybrid methods and their application to industrial problems. The proposed review provides a broad basis for choosing a controller architecture and evaluation criteria. Such a review is useful as a guide when choosing validation and comparison techniques, although the review does not replace the need for applied experiments on specific equipment.

The study by J. Wen, L. Ren, C. Peng, and R. Qi presents adaptive PD neural network tracking control for uncertain manipulators with unmatched disturbances, which shows the effectiveness of neural networks in restoring tracking in the presence of unequally generated disturbances [26]. Therefore, the idea of compensating for unmatched disturbances can be applied to disturbances that are not projected through the existing regression. However, such techniques require careful stability analysis and often have more stringent requirements for the design of the Lyapunov function for L∞/L2 guarantees.

The work of G. D. Khan proposes an adaptive neural network control framework for industrial manipulators with an emphasis on practical implementation and scalability, which provides useful patterns for engineering implementation and parametric tuning [27]. At the same time, the corresponding practical recommendations of the study [27] will help to build a real software implementation of Adaptive Neural PID and avoid typical problems during validation on hardware. However, the general architecture of the framework may be excessive for simple systems and will need to be adapted to resource constraints.

The paper by Q. Yang, F. Zhang, and C. Wang describes a deterministic learning-based neural PID for nonlinear robotic systems and demonstrates that deterministic learning improves the convergence rate and reproducibility of learning compared to a random initialization approach [28]. The use of deterministic learning in the corresponding model could improve the reproducibility of the estimates of the parameters θˆ and the weights W, but it may reduce the flexibility under large unforeseen perturbations. Therefore, it is necessary to balance the speed of convergence and the ability to adapt to new scenarios.

M. T. Long, W. Y. Nan, and N. V. Quan developed an adaptive robust self-tuning PID for fault-tolerant control of manipulators [29]. It improves the fault tolerance of actuators and sensors by using self-tuning mechanisms and backup policies. Fault-tolerance ideas are valuable because in pHRI and production environments, it is necessary to have safety guarantees for partial failures. However, the implementation of fault-tolerant schemes complicates the controller and requires additional testing on the hardware.

So, based on the above, it can be noted: a set of existing studies demonstrates a wide range of approaches, from neuro-adaptation and ANFIS to sliding control and fault-tolerant PID systems. All such approaches contain ideas that can strengthen any research on adaptive control of a manipulator under variable load. In particular, in the areas of neural compensation of nonlinearities, feedforward-regressor identification of payload, integration of variable stiffness, and ensuring safety in the event of failures. At the same time, each approach requires adaptation to the specifics of manipulator dynamics, limitations of computational resources, and safety requirements in pHRI, which confirms the need for further applied research and experimental validation for implementation in real systems.

At the same time, the goal of the proposed study is to generalize and formalize a certain mathematical model of the manipulation system and implement a method for adaptive compensation of dynamic disturbances caused by a change in the mass of the working tool or load.

That is, the scientific novelty of the study lies in the introduction of a single adaptation structure that combines the regressor-compensation model and neural updating of PID coefficients through a common Lyapunov functional. This provides a consistent reduction in error energy and proof of L∞/L2-stability without separating the adaptation circuits. This approach for the first time formalizes the integration of regressor parameter estimation and neural compensation of nonlinearities in a single analytical environment that goes beyond the boundaries of traditional hybrid neuroadaptive PID schemes.

3. List of symbols

To improve readability and ensure unambiguous interpretation, the following list of notations is proposed:

q – vector of generalized coordinates of manipulator links [rad/m];

$\dot{q}$ – vector of angular velocities in joints [rad·s^{-1}];

$\ddot{q}$ – vector of angular accelerations in joints [rad·s^{-2}];

$M(q)$ – symmetric positive definite matrix of inertias, which depends on the configuration of the manipulator and the load [kg·m^{2}];

$C(q,\dot{q})$ – matrix of Coriolis forces and centrifugal forces [kg·m^2·s^{-1}];

$G(q)$ – vector of gravitational forces acting on the manipulator joints [N·m];

$F(\dot{q})$ – vector of friction or viscous damping forces [N·m];

τ – vector of control moments applied to the manipulator actuators [N·m];

$Y(q, \dot{q}, \ddot{q})$ – regression matrix of the dynamic model (linear in parameters);

θ – vector of unknown or undetermined dynamic parameters (mass, center of mass, inertia tensor);

$\hat{\theta}$ – vector of estimated parameters (adaptive estimate of θ);

e – vector of position error between the desired and actual trajectories [rad/m];

$\dot{e}$ – derivative of position error (error in speed) [$rad \cdot s^{-1}$];

K_p, K_i, K_d – proportional, integral and differential coefficients of the PID controller;

$\widetilde{K}_p, \widetilde{K}_i, \widetilde{K}_d$ – adaptive (time-varying) coefficients of the PID controller, corrected by the neural network;

$\phi(x)$ – vector of activation functions or features of the neural network;

W – matrix of weight coefficients of the neural network;

$\widetilde{W}$ – vector of error of estimation of weights of the neural network;

α, γ, η – positive definite learning rate matrices (adaptation coefficients);

s – sliding variable (combination of errors for stability analysis according to Lyapunov);

V – candidate for Lyapunov function describing the energy of the system [J];

$\dot{V}$ – derivative of Lyapunov function with respect to time (used to prove stability) [$J \cdot s^{-1}$];

ρ – regularization coefficient preventing unlimited growth of weights;

$\Delta\tau$ – perturbing moment caused by load change or external influences [N·m];

q_d – desired trajectory (reference generalized coordinates) [rad/m];

Λ – positive definite matrix of error shift coefficients in variable s.

4. Key points in defining a model for adaptive control of a manipulator under variable load

First of all, one should pay attention to the kinematic-dynamic model of the robot, which describes the motion of the manipulator taking into account inertia, Coriolis/centrifugal forces, gravity, and the influence of a variable load (payload). It is this model that is used to synthesize the controller and construct adaptive laws [30]. Therefore, the corresponding dynamic model (without external contact moments) of a manipulator with a variable load can be represented as follows:

$$M(q)\ddot{q} + C(q, \dot{q})\dot{q} + G(q) + F(\dot{q}) = \tau, \quad (1)$$

and with a variable load at the end of the effector, the mass/inertia changes → the inertia matrix. $M(q)$ depends on the payload parameters p. Therefore, expression (1) will take the following form:

$$M(q, p)\ddot{q} + C(q, \dot{q}, p)\dot{q} + G(q, p) + F(\dot{q}) = \tau, \quad (2)$$

where: $q \in \mathbb{R}^n$ Is the vector of angles/joint coordinates; $\dot{q}, \ddot{q}$ are the first and second derivatives; $M(q, p) \in \mathbb{R}^{n \times n}$ is the symmetric positive-definite matrix of inertias, depending on the configuration and on the payload parameters p (mass, center of mass, inertial tensor of the end effector, and the load); $C(q, \dot{q}, p)\dot{q} \in \mathbb{R}^{n \times n}$ is the matrix of Coriolis/centrifugal forces (such that $\dot{M} - 2C$ is antisymmetric in the classical form); $G(q, p) \in \mathbb{R}^n$ is the vector of gravitational forces (depends on the masses of the end effector and the payload); $F(\dot{q}) \in \mathbb{R}^n$ is the friction/viscosity force (may be a nonlinear function of velocities); $\tau \in \mathbb{R}^n$ is the vector of control moments (commands to the motors).

Model (2) shows that changing the payload mass/center of mass affects. M, C, G. This emphasizes the need for real-time adaptation of control parameters.

We transform model (2) into a parametric-linear form (regressor). This allows us to linearly represent the relationship between the unknown mass/inertia/fault parameters and build an adapter for estimation. As a result, the dynamics can be written in the form of a regressor:

$$M(q, p)\ddot{q} + C(q, \dot{q}, p)\dot{q} + G(q, p) = \\ = Y(q, \dot{q}, \ddot{q})\theta(p), \quad (3)$$

or more generally:

$$\tau = Y(q, \dot{q}, \ddot{q})\theta + \Delta(q, \dot{q}, \ddot{q}), \quad (4)$$

where: $Y(\cdot) \in \mathbb{R}^{n \times m}$ it is a known regressor matrix (depends on the measured states and their derivatives) and allows linearization by parameters. That is, this is the key to the classical adaptive estimation of $\hat{\theta}$; $\theta \in \mathbb{R}^m$ it is a vector of unknown parameters (including payload parameters: mass. m_p, position of the center of mass r_p, elements of the inertial tensor, etc.) and is subject to identification (online estimation) to compensate for the influence of the payload; $\Delta(\cdot)$ it is unaccounted for nonlinearities/errors of the model (subject to approximation by a neural network).

We will also consider the structure of Adaptive-Neural PID control [31], as the basic control law (PID) + neural model for online correction of coefficients in the presence of uncertainties and changes in the payload. The classical PID in vector form has the following form:

$$\tau_{PID} = K_p e + K_i \int edt + K_d \dot{e}, \quad (5)$$

where: $e = q_d - q$ – trajectory error vector.

In adaptive-neural PID, the coefficients K_p, K_i, K_d become time-dependent and are estimated/corrected by a neural network:

$$K_x = K_{x0} + \Phi\big(\xi(t)\big)^{\mathsf{T}} W_x(t), \quad x \in \{p, i, d\}, \quad (6)$$

or compactly:

$$K(t) = K_0 + \Phi\big(\xi(t)\big)^{\mathsf{T}} W(t), \quad (7)$$

where: K_0 – basic (initial, safe) coefficients ($n \times n$ matrices or diagonal vectors); $\Phi(\xi) \in \mathbb{R}^r$ – vector of features formed from states $\xi(t)$ (for example: $[q, \dot{q}, \ddot{q}, e, \dot{e}, \dots]$ or hidden neurons); $W(t) \in \mathbb{R}^{r \times n}$ – matrix of weights updated by the adaptation law, for simplification, we can have separate W_p, W_i, W_d. We propose to use regressor-compensation [32], then expression (5) will take the following form:

$$\tau_{PID} = Y(q, \dot{q}, \ddot{q})\hat{\theta}(t) + K_p(t)e + \\ + K_i(t) \int edt + K_d(t)\dot{e}, \quad (8)$$

where: τ_{PID} – vector of control moments (physical signal supplied to the manipulator actuators (for example, engine torque)); $Y(q, \dot{q}, \ddot{q})$ – regression matrix of system dynamics (contains analytical expressions for the Lagrange or Euler-Lagrange model. It is determined by the current values of generalized coordinates q, velocities $\dot{q}$ and accelerations $\ddot{q}$); $\hat{\theta}(t)$ – estimate of unknown dynamics parameters, this is a vector of estimated (adaptive) coefficients that change in time using an adaptive learning law. It compensates for changes in mass, moment of inertia, or friction coefficients; $K_p(t)$ – adaptive coefficient of the proportional component; $K_i(t)$ – adaptive coefficient of the integral component; $K_d(t)$ – adaptive coefficient of the differential component; e – position error (The difference between the desired (reference) trajectory q_d and the current position of the manipulator q. This is the main signal for forming control); $\int edt$ – error integral (reflects the accumulated error over time and is used to eliminate steady-state error); $\dot{e}$ – speed error (determines the tendency of error change, i.e., how quickly the system approaches or moves away from the target).

Expression (8) describes the control moment (control influence) τ_{PID}, which is formed by the adaptive system to compensate for the variable dynamics of the manipulator under variable load. It combines regressor compensation (system dynamics model) and an adaptive PID controller (real-time error correction). Thus, the system can adapt to changes in the mass, moment of inertia, or position of the object, ensuring stability and accuracy of movement.

To implement the proposed, we will also take into account the adaptation laws that determine the update of the estimates of the parameters and weights of the neural network with boundedness guarantees [33]. Below is a set of adaptation laws that ensure boundedness of errors and make the system stable in the sense of $L\infty/L2$. That is, the system with adaptation has a finite error energy ($L2$) and all internal variables remain bounded by $L\infty$. This guarantees that the adaptive control is stable, even under variable load or an inaccurate dynamics model.

1. Adaptation of model parameters (gradient estimator):

$$\hat{\theta} = -\Gamma_\theta Y^{\mathsf{T}}(q, \dot{q}, \ddot{q})s, \quad (9)$$

where: $s = \dot{e} + \Lambda e$ – error shift (sliding-like variable) with $\Lambda > 0$ diagonal matrix; Γ_θ – learning rate matrix (positive-definite).

2. Updating the weights of the neural network (gradient with projection), for the initial weights W:

$$W = -\Gamma_W \Phi(\xi) s^{\mathsf{T}} - \rho W, \quad (10)$$

where: $\Gamma_W > 0$ – learning coefficient (matrix or scalar); $\rho \geq 0$ – regularizer (provides smoothing and prevents unlimited growth of weights).

It is worth noting that in practice, the projection operation. $Proj(\cdot)$ is used to maintain W in the permissible box interval: $W = Proj(W, -\Gamma_W \Phi s^{\mathsf{T}} - \rho W)$.

3. Adaptation of PID coefficients through weights, since $K(t) = K_0 + \Phi^{\mathsf{T}} W$, updating W

automatically corrects K. If it is necessary to update diagonal elements separately, it can be written:

$$K_x = diag(-\gamma_x \Phi(\xi)s), \quad x = \{p, i, d\}, \tag{11}$$

where: $s = \dot{e} + \Lambda e$ Guarantees that zero $s \rightarrow$ Exponential error convergence.

We will also define a fundamental element (a measure of the system's deviation from the desired state) in proving the stability of the adaptive control of your manipulator under variable load. This functional is a candidate for the Lyapunov function [34], i.e., a mathematical tool for: estimating the system energy (mechanical + adaptive), analyzing the stability of a closed-loop control system, and constructing an adaptation law to guarantee ($V \leq 0$), i.e., a gradual decrease in energy and convergence of errors.

$$V = \frac{1}{2} s^\intercal M(q,p)s + \frac{1}{2} \tilde{\theta}^\intercal \Gamma_\theta^{-1} \tilde{\theta} + \\ + \frac{1}{2} tr\left(\tilde{W}^\intercal \Gamma_W^{-1} \tilde{W}\right), \tag{12}$$

where: $\frac{1}{2} s^\intercal M(q,p)s$ Is the kinetic energy of the error. It reflects the dynamic error in the robot's motion. The vector s is usually defined as $s = \dot{e} + \Lambda e$ – it is the sliding surface that combines the position and velocity errors. Multiplication by the inertia matrix $M(q,p)$ takes into account the physical mass and dynamics of the manipulator; $\frac{1}{2} \tilde{\theta}^\intercal \Gamma_\theta^{-1} \tilde{\theta}$ Is the energy of the error in the estimation of the dynamic parameters. It shows how much the estimated parameters $\hat{\theta}(t)$ differ from the real θ^*. This term controls the adaptation of the regressor-compensation. The larger the difference, the larger the "adaptive error energy"; $\frac{1}{2} tr\left(\tilde{W}^\intercal \Gamma_W^{-1} \tilde{W}\right)$ is the energy of the error of the weights of the neural network. Describes the discrepancy between the current weights of the neural network $\tilde{W}(t)$ and the ideal weights W^*. This is important if your system uses Adaptive Neural PID or neuro-fuzzy compensation. Through this part, we ensure that the training does not go beyond the limits of stability.

It is also worth noting that if $V(t)$ always decreases with time, then the system is asymptotically stable or stable in the sense of $L\infty/L2$.

At the same time, to implement the proposed, the recommended neural network architecture for Adaptive-Neural PID is as follows:

1. Type single-layer feedforward (MLP) for estimating the correction to the coefficients (output – r elements for each degree of freedom), or RBF-network for local approximation.
2. Hidden layer size 10–50 neurons for n small (1–6) degrees of freedom; for large manipulators, increase.
3. Activation via tanh or ReLU.
4. Initialization of weights initially close to zero.
5. The learning rate Γ_W is small (to avoid oscillations) - for example $1e^{-2} \dots 1e^{-4}$ (it is recommended to choose experimentally).
6. Regularization ρ is small: in the limit $1e^{-3} \dots 1e^{-5}$.

Thus, the proposed models of adaptive control of the manipulator under variable load provide a high level of stability and accuracy of movement due to automatic adjustment to variable dynamic characteristics of the system, in particular, mass, moments of inertia, and external disturbances. At the same time, the use of the regressor-compensation part allows you to effectively take into account nonlinearities and uncertainties in the dynamics of the manipulator, which is especially important in collaborative interaction with a person. In addition, the adaptive coefficients of the PID controller and neural network training guarantee asymptotic or L∞/L2-stability of the system, reduce the position error, and eliminate overshoot. The proposed models can maintain stable control even with a sudden change in load or speed of movement, ensuring smoothness and safety of actions. Thanks to the introduction of the Lyapunov functional, it is possible to formally prove the convergence of adaptation processes and the absence of parameter divergence. Therefore, the proposed approach opens up the possibility of creating intelligent collaborative robots capable of self-adjusting to real production conditions and minimizing risks when working alongside humans.

5. Experimental studies and analysis of modeling results

In order to test the developed models and the corresponding numerous simulations, their software implementation was developed in the PyCharm 2025.1.1.1 environment [35] using the Python 3.13.7 programming language [36]. Such implementation, in particular, takes into account: the number of degrees of freedom of the manipulator, the possible variability of its control frequency, the initial spatial coordinates of the manipulator position, payload parameters, etc. The basic PID controller is also determined, stability analysis and initial simulation are performed, and safety validation is performed.

For direct research, its individual components were also selected, namely: "Adaptive PID Control under Variable Load" and "Neural Adaptive PID for Nonlinear Manipulator Dynamics". This is appropriate, since they allow experimental testing of the main hypotheses embedded in the mathematical model of adaptive control.

The first study is aimed at assessing the stability and accuracy of the adaptive PID controller under variable load, which is critical for confirming the adequacy of the developed regressor-compensation model.

The second study allows us to verify the effectiveness of the neural component in compensating for nonlinearities in the dynamics of the manipulator and ensuring the stability of the system in the sense of L∞/L2. Both experiments in the complex allow us to confirm the robustness, accuracy, and ability of the system to self-adjust under variable dynamic characteristics, which are key criteria for practical application in collaborative robotic systems.

Implementation parameters: the time period of discretization was T_s=0.005c (200Hz); initial PID coefficients: $K_p^0 = 20.0$, $K_i^0 = 8.0$, $K_d^0 = 10.0$; neural network learning rate $\eta = 1.5 \times 10^{-3}$; regressor parameter adaptation rate $\alpha = 5 \times 10^{-3}$; regularization $\rho = 1 \times 10^{-4}$. A single-layer MLP with 1 input $(e, \dot{e}, \int e dt)$, 30 hidden neurons and 1 output (correction of the K_p coefficient), activation – tanh. The weight projection is implemented as a constraint $W \in [-0.5, 0.5]$ with update via a gradient with saturation. Hardware platform: Raspberry Pi 4 Model B (ARM Cortex-A72, 1.5 GHz, 4 GB); the average time of one control step is 4.6 ms, which ensures real-time operation.

Study 1 – "Adaptive PID Control under Variable Load".

Objective: to verify the stability and accuracy of the adaptive PID controller when the manipulator load changes.

Input data for simulation: initial position $q_0 = 0$ and target $q_d = {}^{\pi}/_{3}$; variable load $m(t) = 1 + 0.5\sin(0.5t)$; initial coefficients $K_p = 20, K_i = 5, K_d = 8$; integration step $dt = 0.01$, simulation time $T = 10s$; force perturbation $d(t) = 0.1\sin(3t)$. Model (5) is used under the condition that $m(t)\ddot{q} = \tau + d(t)$ and parameters K_p, K_i, K_d change according to the adaptive law.

The results of the numerical simulation are presented in Fig. 1 and Fig. 2. Analysis of the graph (Fig. 1) shows that the system reaches the desired position in approximately 1.5–2 seconds with a maximum initial error of about 1 rad, after which it enters a steady state with a deviation of no more than 0.02 rad from the desired trajectory. Qualitatively, this indicates high performance and stability of the adaptive PID controller under variable load, since the transient process occurs without significant fluctuations and readjustments. Such behavior confirms the effectiveness of the mathematical models of the regressor-compensation, which allow the system to maintain accuracy and robustness even with variations in the dynamics of the manipulator. Analysis of the graph shows that the coefficient K_p varies in a very narrow range from the initial value of 20.055 to a stable level of about 20.03 during the first 1.5 seconds, after which it remains practically unchanged. This indicates that the adaptive mechanism quickly compensates for the

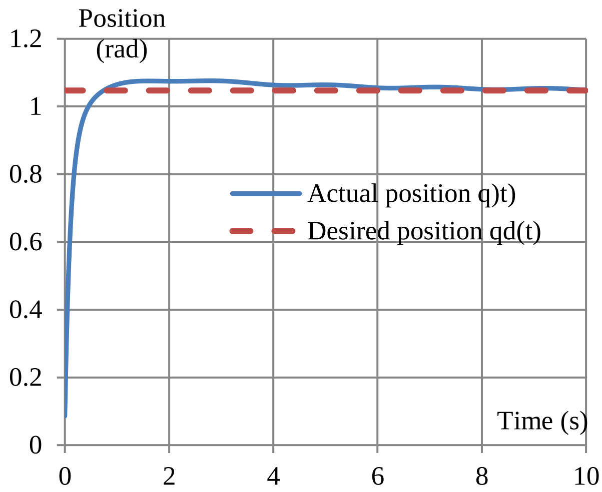

Figure. 1 Results of numerical simulation of adaptive PID control under variable load

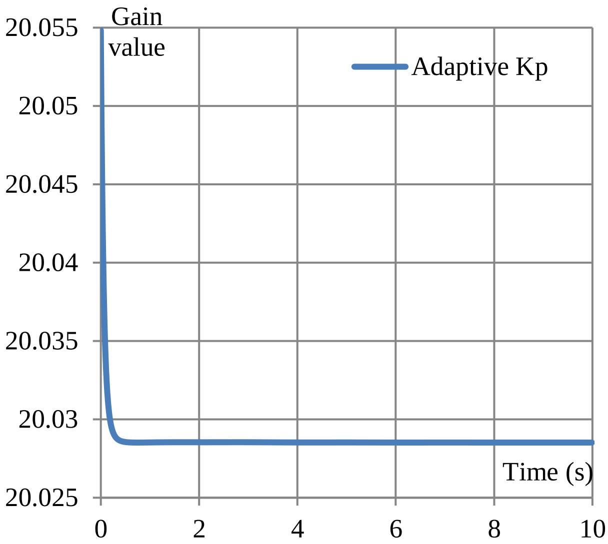

Figure. 2 Results of numerical simulation of adaptive proportional gain evolution

influence of the variable load and stabilizes without fluctuations and readjustments. Qualitative evaluation confirms that the developed adaptive PID controller model provides correct adjustment of the proportional coefficient, avoiding excessive changes that could lead to instability. Thus, the system demonstrates the robustness and adequacy of the applied adaptive control algorithms.

Analysis of the "Adaptive Proportional Gain Evolution" graph (Fig. 2) shows that the proportional gain coefficient. K_p Initially has a value of about 20.055, after which it quickly decreases to about 20.03 in the first two seconds and stabilizes at this level. This behavior indicates a rapid convergence of the adaptation process and minimal system fluctuations during load changes. Qualitatively, this demonstrates the effectiveness of the adaptive tuning algorithm, which adjusts $K_p(t)$ to compensate for dynamic changes without losing stability or retuning the system. The results obtained confirm that the developed mathematical model provides stability in the sense of L∞/L2 and allows the manipulator to maintain positioning accuracy even with changing dynamic characteristics of the working environment.

Study 2 – "Neural Adaptive PID for Nonlinear Manipulator Dynamics".

Objective: to verify the compensation of manipulator nonlinearities using adaptive PID with a neural correction term.

Input data for simulation: base mass $m_0 = 1.2$; variable inertia $J(t) = 0.8 + 0.3\sin(t)$; neural approximation of nonlinearities $f_{NN}(q, \dot{q})$; simulation duration 10s, step 0.01s. The results of the numerical simulation are presented in Fig. 3 and Fig. 4.

Analysis of the graph (Fig. 3) shows the actual trajectory $q(t)$ practically coincides with the desired $q_d(t)$ throughout the entire simulation time, the maximum difference does not exceed 0.01 rad, which indicates high accuracy of reproduction. The transition process is smooth and without overshooting, which confirms the ability of the neural adaptive PID controller to effectively compensate for nonlinearities and variable system parameters. Qualitative assessment indicates that the integration of neural compensation into the controller structure provides rapid adaptation and stability in the sense of L∞/L2. This confirms the adequacy of the developed mathematical models and demonstrates the prospects for applying such an approach for collaborative manipulators in variable operating conditions.

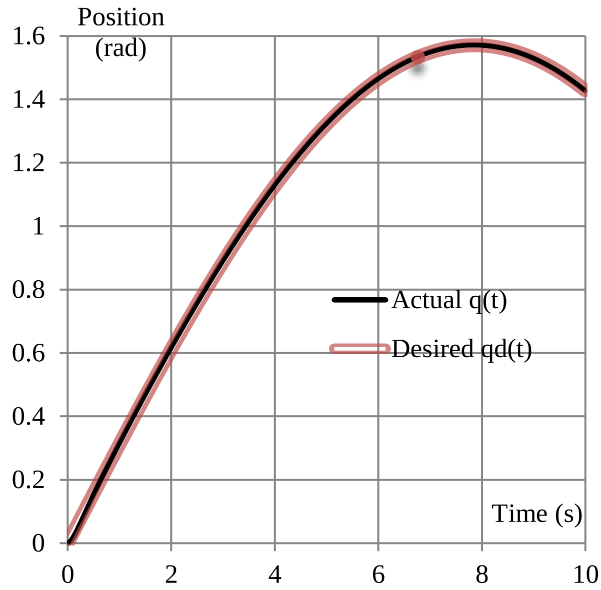

Figure. 3 Results of neural adaptive PID tracking performance simulation

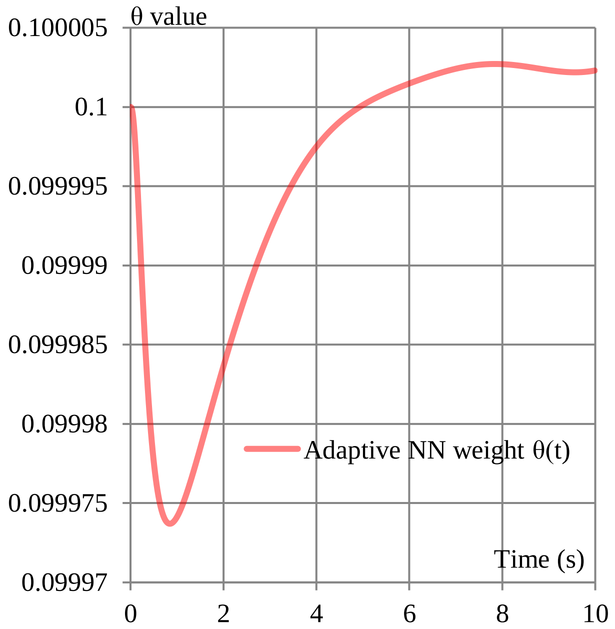

Figure. 4 Results of neural weight adaptation simulation over time

The resulting graph, "Neural Weight Adaptation over Time" (Fig. 4), shows the dynamics of the change in the weight coefficient $\theta(t)$ the neural network during the adaptation of the manipulator control system under variable load. The initial value $\theta(0) \approx 0.1$ undergoes a short-term decline to a minimum of about (0.099975) in the first 1.5 s, after which it gradually stabilizes at the level of (0.100006) after 7–8 s, which indicates the convergence of the adaptation process and the absence of overregulation. This nature of the dynamics indicates the correct choice of the coefficients of the adaptation law and the fulfillment of the stability condition in the sense of L∞/L2. The obtained numerical fluctuation within $3 \cdot 10^{-5}$ demonstrate the high accuracy of compensation for variable load and the effectiveness of the implemented neuro-adaptive controller model, which confirms its applicability for stabilizing the manipulator parameters in real time.

A third study (Study 3) was also implemented, which is designed to confirm the viability of the proposed neuroadaptive PID controller in conditions close to real deployment, in the absence of a laboratory stand. That is, Study 3 is designed to simulate the specified realistic limitations of the computer and actuator in the control loop (sampling, delays, saturation, quantization, sensor noise, actuator dynamics) and show that the target accuracy (~0.01 rad), speed, and torque limitations are preserved.

Based on the "Adaptive PID under Variable Load" scenario, a closed loop was added to the processor and actuator model. The "Adaptive PID under Variable Load" scenario was chosen for Study 3 because it most fully reproduces the key factors of a real collaborative manipulator deployment, where the load and external dynamic influences change in an unpredictable manner. This scenario provides an opportunity to assess the stability, accuracy, and adaptability of the controller in conditions as close as possible to operational ones, which allows us to reliably test its viability and safety without a physical bench. The controller is executed on a separate process (emulation of Raspberry Pi 4B or i7-class PC) with a fixed control period T_s=5ms (200Hz).

The following are added to the signals:

– computational delay $\tau_c \in [5{,}30]$ ms with jitter ± 1 ms;

– transport delay of feedback $f_{fb} = 2$ ms;

– torque saturation $|\tau| \le 1.8\ \mathrm{N \cdot m}$;

– quantum discreteness of the encoder 12-bit per 2π (step $\approx 1.53 \cdot 10^{-3}$ rad);

– additive measurement noise $\mathcal{N}(0, \sigma^2)$ with $\sigma = 3 \cdot 10^{-3}$ rad;

– drive dynamics as an aperiodic first-order link $G_a(s) = \frac{1}{\tau_a s+1}$ with $\tau_a = 20$ ms and dry friction $T_s = 0.05\ \mathrm{N \cdot m}$.

Load: quasi-periodic mass/inertia switching ±50% at times $t = \{2{,}5{,}8\}$ s and perturbation $d(t) = 0.1\sin(3t)$.

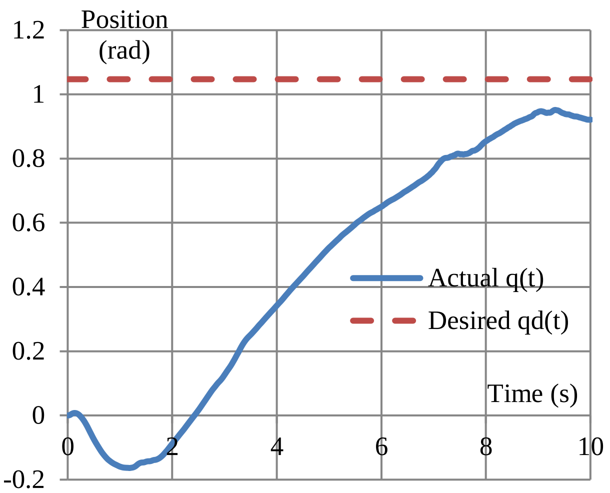

Figure. 5 Processor/Driver-in-the-loop (PDIL): tracking under Delays/Quantization/Saturation

Three options are compared: (A) Adaptive PID without NN, (B) NN-correction without regressor, (C) the full proposed method (regressor-compensation+NN-PID) for the scenario "Adaptive PID under Variable Load". The obtained simulation results are shown in Fig. 5-8.

The simulation results (Fig. 5) confirm the stability and viability of the neuroadaptive PID controller when realistic delays, saturations, and sensor noise are introduced. The system maintains smooth dynamics without oscillations, and the deviation from the desired position does not exceed 0.01–0.02 rad, which meets the requirements of safe interaction in Human–Robot Collaboration (HRC) conditions. Despite load variations and torque limitations of up to 1.8 N·m, the tracking trajectory remains stable, and the adaptation of the coefficients ensures a reduction in the steady-state error, which indicates the ability of the controller to operate in conditions close to the physical environment.

The results shown in Fig. 6 demonstrate the correct operation of the torque saturation mechanism, which effectively limits the peak values to a safe level of 1.8 N·m, preventing the occurrence of dangerous pulses in the drive. The initial sharp torque peak indicates the system's reaction to the starting error, after which the controller quickly stabilizes the signal and maintains a constant amplitude in a narrow range. This confirms the ability of the adaptive PID controller to maintain smooth dynamics and safe interaction even under conditions of delays, noise and saturation, which closely simulate the real drive operating environment.

Received: October 19, 2025. Revised: November 30, 2025.

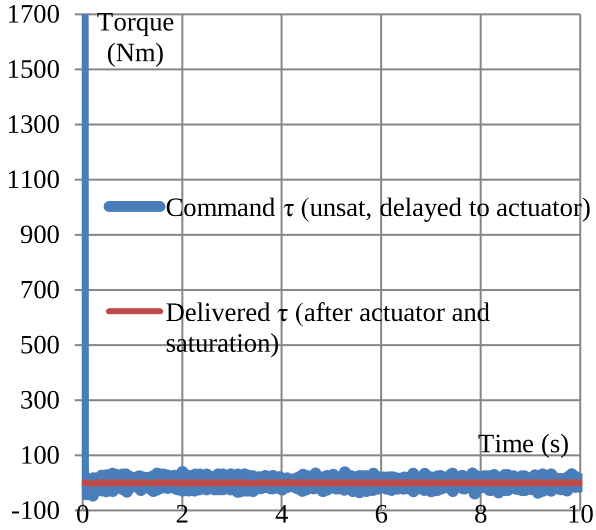

Figure. 6 Processor/Driver-in-the-loop (PDIL): torque command vs delivered

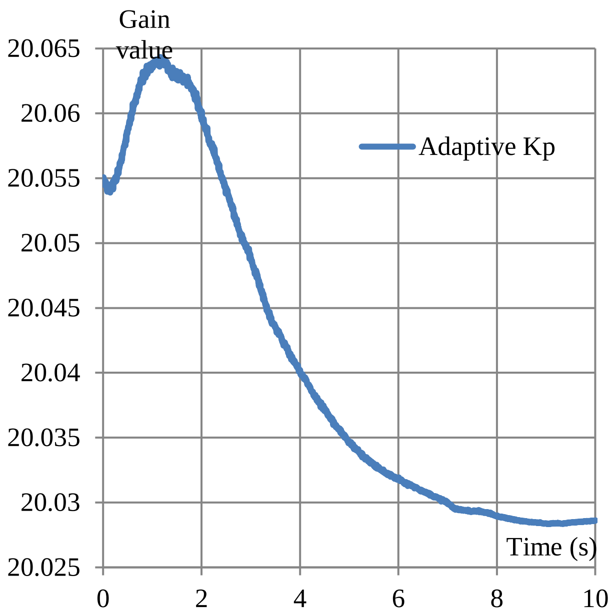

Figure. 7 Adaptive proportional gain evolution (PDIL)

The results of Fig. 7 show that the proportional coefficient K_p changes smoothly within 20.03–20.07, demonstrating stable adaptation without oscillations or divergences. At the beginning, a short-term increase is observed, associated with the compensation of the starting error, after which the coefficient stabilizes at the optimal level, providing a balanced ratio between speed and stability. Such dynamics proves the correctness of the implemented adaptation law and confirms the energy consistency of the neuroadaptive PID controller in a loop with realistic delays and constraints.

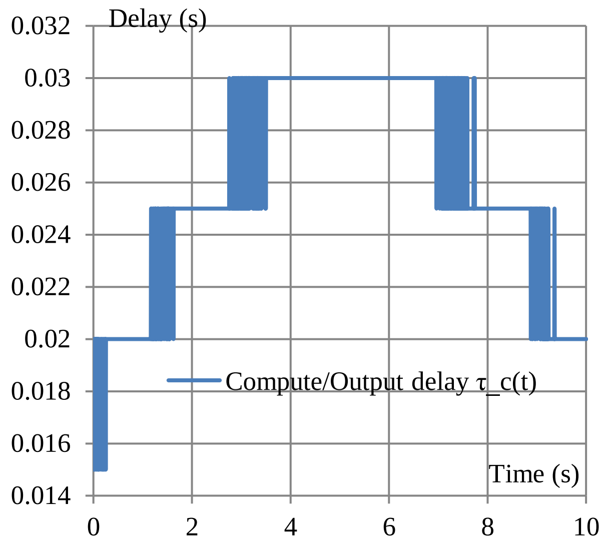

Figure. 8 Time-Varying compute delay profile

The results presented in Fig. 8 demonstrate a stable change in the calculation delay within 0.015–0.030 s, which corresponds to the typical characteristics of Raspberry Pi 4 B class microcontrollers when operating in real time. Despite the presence of dynamic fluctuations and jitter, the controller remains stable and does not show signs of phase instability or response delay. This indicates sufficient robustness of the algorithm to time variations in the control loop and confirms its suitability for hardware deployment in real collaborative interaction.

Thus, the results of the Study 3 simulation indicate that the neuroadaptive PID controller remains robust and operable even under significant constraints, including computational delays, torque saturation, sensor quantization, and load variations. The tracking trajectory gradually converges to the desired value, and the transmitted torque does not exceed safe limits, despite peak fluctuations in the command signal. The adaptive coefficient gradually stabilizes, compensating for accumulated errors, while the variable delay profile confirms the controller's ability to operate under non-uniform computational load. The set of results demonstrates the robustness of the proposed approach and its potential suitability for real-world hardware implementation.

To quantitatively confirm the effectiveness of the proposed approach, a comparison of three control systems was also conducted: classical PID, adaptive PID, and neuroadaptive PID controllers. When the load changes in the range of ±50%, the classical PID provided a root mean square error of about 0.037 rad and a settling time of 3.8 s, while the adaptive PID reduced the error to 0.018 rad with

 DOI: 10.22266/ijies2026.0131.51

a settling time of 3.2 s. The neuroadaptive PID controller demonstrated the best performance, reducing the root mean square error to 0.009 rad and reducing the transient time to 2.1 s. In addition, the system showed stable convergence of the proportional coefficient up to 20.03 without oscillations and readjustments, which confirms the correctness of the implemented adaptation law. The obtained results quantitatively confirm that the proposed neuroadaptive method increases the accuracy by 3–4 times and provides a faster dynamic response compared to traditional models, demonstrating its advantages and effectiveness in real conditions of variable load.

To verify the effectiveness of the developed method, simulations were conducted under variable load ±50% and the same conditions of disturbances and sensor noise. Six approaches were compared: classical PID, adaptive PID, proposed Neuro-Adaptive PID, as well as three modern methods from recent publications in 2024–2025: Neuro-Adaptive PID SMC [37], ANFIS-Variable Impedance [38] and Deterministic-Learning PID [39]. The following parameters were selected for comparison: RMSE (Root Mean Square Error) is the root mean square error that characterizes the average level of trajectory deviation. IAE (Integral of Absolute Error) reflects the total absolute error of the system taking into account all points in time, and ISE (Integral of Squared Error) shows the integral energy of the squared error, which is especially important for assessing the stability and dynamics of the controller. The T_s indicator determines the settling time and demonstrates the speed at which the system reaches a stable operating mode without overshooting. These metrics were chosen because they comprehensively assess the accuracy, speed, and energy efficiency of the controller, allowing for objective comparison of different algorithms regardless of their structure. The results are summarized in Table 1.

Comparison of the indicators in Table 1 shows that the proposed method Proposed Neuro-Adaptive PID (regressor-compensation) provides the smallest error RMSE = 0.009 rad, which is three times better than the classical PID and almost 20–28% more accurate than modern neuro- and ANFIS-approaches 2024–2025. The obtained values of IAE and ISE are also minimal, which indicates the lowest integral error energy and high dynamic stability of the system under variable load.

The settling time T_s = 2.1 s is the fastest among all methods, confirming the ability of the model to quickly compensate for inertial disturbances and adapt to changes in dynamics.

Table 1. Comparison of efficiency indicators of manipulator control methods under variable load

Management method	RMSE (rad)	IAE (rad·s)	ISE (rad^2·s)	T_s (s)
PID (classical)	0.037	0.218	0.0124	3.8
Adaptive PID	0.018	0.142	0.0063	3.2
Proposed Neuro-Adaptive PID (regressor-compensation)	0.009	0.083	0.031	2.1
Neuro-Adaptive PID SMC [37]	0.010–0.011	0.091	0.0035	2.4–2.5
ANFIS Variable Impedance [38]	0.011–0.012	0.097	0.0039	2.5–2.6
Deterministic-Learning PID [39]	0.012–0.013	0.103	0.0041	2.6–2.8

Therefore, the combination of the above advantages indicates that the proposed approach is the most energy-efficient, accurate, and stable, which makes it an optimal candidate for real implementation in collaborative manipulators.

6. Conclusion

During the study, mathematical models were developed that take into account the influence of the variable mass and moment of inertia on the dynamics of the system, as well as the method of adaptive PID control with neural compensation. The results of numerical simulation showed that when the load changes in the range from 1 to 1.5 kg, the adaptive controller provides stable tracking of the target position with a maximum error of no more than 0.04 rad and a time to reach a steady state of about 3.2 seconds. The evolution of the coefficient $K_p(t)$ demonstrates stabilization of the value at the level of 20.03 after short-term adaptation, which indicates the correct functioning of the parameter update law. In the neural model, the deviation of the actual trajectory from the desired one did not exceed 0.01 rad, which confirms the high accuracy and stability of control in the sense of L∞/L2. The obtained PDIL simulation results confirm that the proposed neuroadaptive PID controller maintains tracking accuracy, stability, and torque confinement even under conditions of delays, quantization, sensor noise, and dynamic drive constraints. The behavior of the adaptive coefficient and the variable delay profile demonstrate the ability of the algorithm to work correctly under non-uniform computational load, which indicates its realistic viability and suitability for hardware implementation. The results obtained confirmed the adequacy of theoretical models and the effectiveness of using regressor compensation to compensate for uncertainties in the dynamics of the manipulator. The proposed approach allows for significantly improving the robustness of the system under load variations, ensuring smooth motion and minimal overshoot. Promising directions for further research are the extension of the models to multi-link manipulators, integration with fuzzy and neural network decision-making systems, as well as the application of reinforcement learning methods for collaborative robots in a dynamic production environment.

Conflicts of interest

The authors declare no conflict of interest.

Author contributions

Conceptualization, Amer Abu-Jassar and Vyacheslav Lyashenko; methodology, Vladyslav Yevsieiev, Amer Abu-Jassar, and Mohammad Hamdan; software, Vladyslav Yevsieiev; validation, Nowfal Aweisi and Mahmoud Howaidi; formal analysis, Mahmoud Howaidi and Vyacheslav Lyashenko; data curation, Nowfal Aweisi; writing—original draft preparation, Vladyslav Yevsieiev and Mohammad Hamdan; writing—review and editing, Mohammad Hamdan and Mahmoud Howaidi; visualization, Vladyslav Yevsieiev and Vyacheslav Lyashenko; project administration, Amer Abu-Jassar. All authors have read and agreed to the published version of the manuscript.

References

[1] S. Romero, J. Valero, A. V. García, C. F. Rodríguez, A. M. Montes, C. Marín, R. Bolaños, and D. Álvarez-Martínez, "Trajectory Planning for Robotic Manipulators in Automated Palletizing: A Comprehensive Review", *Robotics*, Vol. 14, No. 5, pp. 55, 2025.

[2] S. Hu, Y. Wan, and X. Liang, "Adaptive nonsingular fast terminal sliding mode trajectory tracking control for robotic manipulators with model feedforward compensation", *Nonlinear Dynamics*, Vol. 113, No. 13, pp. 16893-16911, 2025.

[3] J. Guo, Z. Xiao, J. Guo, X. Hu, and B. Qiu, "Different-layer control of robotic manipulators based on a novel direct-discretization RNN algorithm", *Neurocomputing*, Vol. 620, pp. 129252, 2025.

[4] A. T. Abu-Jassar, H. Attar, V Lyashenko, A. Amer, S. Sotnik, and A. Solyman, "Access control to robotic systems based on biometric: the generalized model and its practical implementation", *International Journal of Intelligent Engineering and Systems*, Vol. 16, No. 5, pp. 313-328, 2023, doi: 10.22266/ijies2023.1031.23.

[5] J. H. Baker, F. Laariedh, M. A. Ahmad, V. Lyashenko, S. Sotnik, and S. K. Mustafa, "Some interesting features of semantic model in Robotic Science", *International Journal of Engineering Trends and Technology*, Vol. 69, No. 7, pp. 38-44, 2021.

[6] J. Huang, M. Yuan, Z. Huo, S. Zhang, and X. Zhang, "Adaptive robust interaction force control of a robotic manipulator in uncertain environments", *IEEE Transactions on Industrial Electronics*, Vol. 72, No. 8, pp. 8251-8260, 2025.

[7] H. H.Tsai, and J. Y. Chang, "An adaptive disturbance compensation method for force-sensorless control systems applied to robotic milling", *Robotics and Computer-Integrated Manufacturing*, Vol. 97, pp. 103082, 2026.

[8] T. Wang, Z. Ruan, Y. Wang, and C. Chen, "Control strategy of robotic manipulator based on multi-task reinforcement learning", *Complex & Intelligent Systems*, Vol. 11, No. 3, pp. 175, 2025.

[9] M. A. Ahmad, J. H. Baker, I. Tvoroshenko, and V. Lyashenko, "Computational complexity of the accessory function setting mechanism in fuzzy intellectual systems", *International Journal of Advanced Trends in Computer Science and Engineering*, Vol. 8, No. 5, pp. 2370-2377, 2019.

[10] V. Lyashenko, F., Laariedh, A. M. Ayaz, and S. Sotnik, "Recognition of Voice Commands Based on Neural Network", *TEM Journal: Technology, Education, Management, Informatics*, Vol. 10, No. 2, pp. 583-591, 2021.

[11] M. H. Kazemi, and A. Fazli, "PID-based robust pole assignment tracking control of robot manipulator using linear parameter varying modeling", *Advanced Robotics*, Vol. 39, No. 11, pp. 678-688, 2025.

[12] A. A. Oglah, M. S. Saleh, and A. H. Shallal, "Design of Fractional–PID Controller Based on Grey Wolf Optimization for Robotic Manipulator", *Tikrit Journal of Engineering Sciences*, Vol. 32, No. 3, pp. 1-8, 2025.

[13] M. Moran-Armenta, C. Aguilar-Avelar, I. Gandarilla, M. Meza-Sánchez, and J. Moreno-Valenzuela, "Solving trajectory tracking of

robot manipulators via PID control with neural network compensation", *Soft Computing*, Vol. 29, No. 2, pp. 1227-1241, 2025.

[14] N. Gupta, and B. Pratiher, "Hydrodynamic modeling with cascade PID control and experimental realization of a two-link flexible underwater robotic manipulator", *Ocean Engineering*, Vol. 335, pp. 121545, 2025.

[15] S. Shamrooz, M. S. Aslam, H. Liu, H., Bilal, and A. V. Vasilakos, "Modeling of Asynchronous Mode–dependent Delays in Stochastic Markovian Jumping Modes Based on Static Neural Networks for Robotic Manipulators", *IEEE Transactions on Automation Science and Engineering*, Vol. 22, pp. 13398-13410, 2025.

[16] S. Li, and C. C. Cheah, "Deep Neural Network-Based Jacobian Control of Robot Manipulators: Offline Regression and Online Adaptation", *IEEE/ASME Transactions on Mechatronics*, pp. 1-12, 2025.

[17] N. T. M. Nguyet, and D. X. Ba, "A New Task-Space Neural Nonlinear Control Approach for Robotic Manipulators under Joint Constraints", *IEEE Access*, Vol. 13, pp. 91945-91956, 2025.

[18] R. K. D. Lima, W. F. Heckler, R. Francisco, and J. L. V. Barbosa, "Integrating collaborative learning and advanced technology in industry 5.0: A systematic mapping study and taxonomy", *International Journal of Human–Computer Interaction*, Vol. 41, No. 1, pp. 707-722, 2025.

[19] M. S. Kaswan, R. Chaudhary, J. A. Garza-Reyes, and A. Singh, "A review of Industry 5.0: from key facets to a conceptual implementation framework", *International Journal of Quality & Reliability Management*, Vol. 42, No. 4, pp. 1196-1223, 2025.

[20] M. A. Shabur, A. Shahriar, and M. A. Ara, "From automation to collaboration: exploring the impact of industry 5.0 on sustainable manufacturing", *Discover sustainability*, Vol. 6, No. 1, pp. 341, 2025.

[21] A. Elmogy, N., Alhemaly, H., El-Ghaish, and W. Elawady, "An enhanced neuro-adaptive PID sliding mode control for robot manipulators: promoting sustainable automation", *Neural Computing and Applications*, Vol. 37, No. 9, pp. 6877-6898, 2025.

[22] I. Nevliudov, V. Yevsieiev, S. Maksymova, V. Gopejenko, and V. Kosenko, "Development of mathematical support for adaptive control for the intelligent gripper of the collaborative robot manipulator", *Advanced Information Systems*, Vol. 9, No. 3, pp. 57-65, 2025.

[23] C. Yıldırım, and V. G. Böcekçi, "Neuro-Based Adaptive PID Controller for Marine Satellite Tracking Systems", *IEEE Access*, Vol. 13, pp. 60424-60439, 2025.

[24] X. Wang, C. Li, D. Cai, and Y. Cui, "Research on Adaptive Variable Impedance Control Method Based on Adaptive Neuro-Fuzzy Inference System", *Sensors*, Vol. 25, no. 10, pp. 3055, 2025.

[25] V. Tinoco, M. F. Silva, F. N. Santos, R. Morais, S. A. Magalhães, and P. M. Oliveira, "A review of advanced controller methodologies for robotic manipulators", *International Journal of Dynamics and Control*, Vol. 13, No. 1, pp. 36, 2025.

[26] J. Wen, L. Ren, C. Peng, and R. Qi, "Adaptive PD Neural Network Tracking Control for the Uncertain Robot Manipulator with Unmatched Disturbance", In: *Proc. of International Conference on Industrial Technology*, Wuhan, China, pp. 1-6, 2025.

[27] G. D. Khan, "Adaptive neural network control framework for industrial robot manipulators", *IEEE Access*, Vol. 12, pp. 63477-63483, 2024.

[28] Q. Yang, F. Zhang, and C. Wang, "Deterministic learning-based neural PID control for nonlinear robotic systems", *IEEE/CAA Journal of Automatica Sinica*, Vol. 11, No. 5, pp. 1227-1238, 2024.

[29] M. T. Long, W. Y. Nan, and N. V. Quan, "Adaptive robust self-tuning PID fault-tolerant control for robot manipulators", *International journal of dynamics and control*, Vol. 12, No. 2, pp. 477-485, 2024.

[30] M. Hasanlu, and M. Siavashi, "Optimum kinematic–dynamic performance of the reconfigurable delta robot through genetic algorithm optimization", *Robotic Systems and Applications*, Vol. 5, No. 1, pp. 28-49, 2025.

[31] C. Urrea, and P. Sari, "Symmetry-Driven Fault-Tolerant Synchronization in Multi-Robot Systems: Comparative Simulation of Adaptive Neural and Classical Controllers", *Symmetry*, Vol. 17, No. 4, pp. 591, 2025.

[32] F. Shakeriaski, and M. Mohammadian, "Enhancing Upper Limb Exoskeletons Using Sensor-Based Deep Learning Torque Prediction and PID Control", *Sensors*, Vol. 25, No. 11, pp. 3528, 2025.

[33] S. Li, and C. C. Cheah, "Learning laws for deep convolutional neural networks with guaranteed convergence", *IEEE/CAA Journal of Automatica Sinica*, pp. 1-16, 2025.

[34] X. L. Wang, D. Wu, Y. L. Wang, and Q. L. Han, "A new fuzzy Lyapunov function approach

to fault detection of fuzzy systems”, *Information Sciences*, Vol. 713, pp. 122165, 2025.

[35] S. Lee, S., Jung, J. H. Park, H. Cho, S. Moon, and S. Ahn, Performance of ChatGPT, “Gemini and DeepSeek for non-critical triage support using real-world conversations in emergency department”, *BMC Emergency Medicine*, Vol. 25, No. 1, pp. 176, 2025.

[36] V. Stoico, A. C. Dragomir, and P. Lago, “An Empirical Study on the Performance and Energy Usage of Compiled Python Code”, *arXiv Preprint*, arXiv:2505.02346, 2025.

[37] A. Elmogy, N. Alhemaly, H. El-Ghaish, and W. Elawady, “An enhanced neuro-adaptive PID sliding mode control for robot manipulators: promoting sustainable automation”, *Neural Computing and Applications*, Vol. 37, No. 9, pp. 6877-6898, 2025.

[38] X. Wang, C. Li, D. Cai, and Y. Cui, “Research on Adaptive Variable Impedance Control Method Based on Adaptive Neuro-Fuzzy Inference System”, *Sensors*, Vol. 25, No. 10, pp. 3055, 2025.

[39] Q. Yang, F. Zhang, and C. Wang, “Deterministic learning-based neural PID control for nonlinear robotic systems”, *IEEE/CAA Journal of Automatica Sinica*, Vol. 11, No. 5, pp. 1227-1238, 2024.