Received: 17 February 2025 | Revised: 26 May 2025 | Accepted: 26 June 2025 | Published online: 15 July 2025

RESEARCH ARTICLE

Artificial Intelligence and Applications
2025, Vol. 00(00) 1–10
DOI: 10.47852/bonviewAIA52025435
BON VIEW PUBLISHING

Training of the Dynamic Systems Control: A Neural Network or a Learning Algorithm

Dmytro Kucherov[1,*], Ihnat Myroshnychenko[1], Natalia Khalimon[1], and Valerii Tkachenko[1]

[1]*Department of Intelligent Cybernetic Systems, State University "Kyiv Aviation Institute", Ukraine*

Abstract: The dynamic system control problem under conditions of a priori uncertainty regarding the parameters of the controlled object is considered. The properties of controllers typically used in control systems are studied. Among them are a neural network, a proportional–integral–derivative (PID) controller, and a learning algorithm. The sign-changing input signal is considered in dynamic systems using the minimum time criterion. A dynamic system is represented by the first-order differential equations system, which allows using the state space method in the analysis. A feature of the research is the study of the quality of the system tuning under conditions of parametric uncertainty and the presence of homogeneous non-Gaussian noise in the phase coordinate measurement channels. The system's reaction results for the studied approaches for the proposed mathematical model are compared. The learning algorithm showed an improvement over conventional methods by at least 40% in the evaluated indicators, in which the influence of interference is leveled by introducing a unique function of the "hysteresis" type. The modeling results are given in support of the conclusions made.

Keywords: neural network, PID controller, learning algorithm

1. Introduction

The task of management is widely used in all spheres of human activity, and the level of effectiveness of its solution is the most important indicator of any system, as well as an indicator of the development of the scientific, technical, industrial, and defense potential of each state. At the same time, special attention is paid to industrial models in which the management of moving or dynamic objects with variable parameters is implemented.

Unfortunately, it should be noted today that in the classic formulation of the problem, when external influences are not considered, the known mathematical model of the dynamic object with the stationary of its parameters is actual in simple cases. It does not always meet the operating conditions. When designing a control system, it should be considered that external influence is not Gauss, acts within a limited time interval, and the researcher does not have time to study his statistical characteristics. At the functioning stage, these problems have engaged no one as well. The mathematical model is not always linear. The specified condition may be fair at the narrow interval of the initial characteristics of the dynamic object. The order of differential equations describing a dynamic object is also not known. However, we can use digital computing systems for both the modeling and design of dynamic systems.

In the conditions considered for robotics and dynamic systems control tasks, it is appropriate to use methods of teaching the functioning control system to achieve the kinematic (calculated) parameters of the required movement. In this case, the control system, in addition to the usual representation as a system of "controller-dynamic object," covered by feedback, acquires the properties of intellectual due to the use of an additional system that studies the nature of input change and output parameters of motion and does not act on a dynamic plant, but to the controller, changing its parameters so that the purpose of control is achieved in the broad range conditions of the environment.

Today, we know most of the learning techniques used in control systems have been developed, and their comprehensive use can also be effective. The most used are iterative teaching methods, including gradient learning methods and neural networks. The latest techniques are implemented on programmable logic integrated schemes and with the help of special neurochips and neurocomputers. The most famous examples of neurocomputers are Synapse neurocomputer (Siemens, Germany) and Neuro-matrix processor. They are based on a software system that controls computing devices with parallel flows of identical commands and multiple data flow, the so-called Multiple Single Instruction Multiple Data (MSIMD) architecture. The main paper's goal is to consider the algorithms that can be at the heart of software systems that control specified computer devices.

2. Literature Review

An analysis of the available literature shows that neural networks are one of the most frequently used machine-learning methods for solving various problems in recent years. It is emphasized in the reviews by Mahesh [1], Ray [2], Stanley et al. [3], and Zhao et al. [4]. Thus, Mahesh [1] proposed a brief overview and some prospects for applying machine-learning algorithms to automatic data processing. Ray [2] proposed a review of the most frequently used machine-learning algorithms. In this review, the author selected an appropriate learning algorithm that meets specific task requirements. Key aspects of modern neuroevolution, including large-scale computations, novelty, diversity, and indirect coding capabilities, as well as the contribution of this field to meta-learning and architecture search, are provided in the review by Stanley et al. [3]. Zhao et al. [4] presented an overview of neural network applications using software-defined network approaches with machine learning. Detailed information about neural networks and

***Corresponding author:** Dmytro Kucherov, Department of Intelligent Cybernetic Systems, State University "Kyiv Aviation Institute", Ukraine. Email: dmytro.kucherov@npp.kai.edu.ua

© The Author(s) 2025. Published by BON VIEW PUBLISHING PTE. LTD. This is an open access article under the CC BY License (https://creativecommons.org/licenses/by/4.0/).

their applications can be found in the training course developed by Gurney [5].

A traditional solution to the construction of effective dynamic systems is the use of PID regulators, which are widely used in driving complex vehicles, such as unmanned aerial vehicles. However, setting up a proportional–integral–derivative (PID) regulator requires certain skills. Therefore, to improve the quality of the quadrotor control, Bouzid et al. [6] offered the option of using a PID controller with variable coefficients, which certainly complicates the structure of the control system and requires an increase in the accuracy of measuring its current coordinates. Hadid et al. [7] investigated various approaches to constructing a control system for a nonlinear dynamic object, such as a quadcopter, in which they noted that the PID controller, which is traditionally used in flight controllers, is inferior to classical controller implementation options in terms of dynamic performance. Guettal et al. [8] developed the control of the unmanned aerial vehicle by a backstepping neural network controller, which is currently considered an alternative to the PID controller. Cengiz et al. [9] proposed to set the weight coefficients of the neural network using a genetic algorithm. The proposed approach has an obvious drawback associated with the need for a problem solution of local minimums for the optimal configuration of the neural network since the real mechanisms for solving this problem for genetic algorithms are not yet available [10]. Murugesan and Ramasubbu [11] presented a method for evaluating the excitation current of the synchronous motor based on the learning algorithm to obtain an adequate multiple linear regression model.

Gabella [12] studied the dynamic structure of weights in training a feed-forward neural network. Shrestha and Mahmood [13] presented an overview of different types of networks, as well as several optimization methods that include deep learning for improving accuracy and reducing training time. Li et al. [14] offered deep training with the reinforcement for navigation of vehicles in free space based on training in virtual space and it is transmitted to a vehicle that moves in the real world.

Since a neural network is a rather complex and cumbersome structure, one of the available options for tuning the network parameters is genetic algorithms. Thus, a methodology for the automatic design of a neural network using a particle swarm optimization algorithm, which uses various fitness functions to avoid over-fitting and reduce the number of connections in the network, was presented by Garro and Vazquez [15]. Such et al. [16] trained a deep neural network with several million adjustable parameters by a genetic algorithm. Tsmots et al. [17] proposed a method for parallel vertical group data processing for neural algorithms and neural network structures. Chen and Liu [18] proposed an alternative option for deep learning, which involves extensive training in a flat network. Ou et al. [19] proposed improving the quality of a deep autoencoder-type network using the regularization method.

Alternatives to convolution neural networks are provided in refs. [20–23]. Leo and Kalita [20] introduced the concept of incremental learning, in which the neural network identifies unknown classes at the testing stage and updates itself autonomously if new features are detected. To ensure recognition accuracy, the authors introduce a threshold for reliable classification. Li and Zhang [21] created a self-organizing learning model for a two-layer feed-forward neural network. In this network, the network weights optimization is performed with the training errors on the training data set using a swarm algorithm. Dai et al. [22] studied distributed learning algorithms to bring the neural network closer to multi-agent reinforcement learning. Huoh et al. [23] proposed a graph neural network model that has superior sensitivity and accuracy compared to convolution neural networks (CNN) and recurrent neural networks (RNN).

The authors [24–27] propose using adaptation methods in neural networks. Sayed [24] reviewed modern achievements in adaptation, learning, and optimization in neural networks. Lin [25] proposed a learning algorithm similar to the back-propagation algorithm, which does not require feedback. In this case, neurons can adapt asynchronously and simultaneously, similar to biological neurons. Zhou et al. [26] propose an adaptive learning network based on a deep deterministic gradient policy with an adaptive neural network of fuzzy inference systems.

Liu et al. [27] discussed the method of adaptive dynamic programming in control problems. Sun et al. [28] proposed a method of adaptive dynamic programming for optimal fuel consumption in a non-linear system with unknown dynamics in continuous time.

The idea of clustering in neural networks is supported by the authors [29, 30]. For example, Heer et al. [29] focused on increasing network reliability, for which they chose the greedy and lazy greedy heuristics to maximize the clustering coefficient. Yin et al. [30] proposed local closure coefficients as a metric for clustering edges based on the general node of the neural network.

Neural networks in problems of control of the dynamics of moving systems were considered in the works [31–33]. Trischler and D'Eleuterio [31] showed the possibility of training a feed-forward neural network to reproduce the dynamics of the original system. Li et al. [33] studied the problem of rocket control with time control. They proposed a control method with training, which provides the desired control time with a restriction in the field of view. The training is based on a gradient of the error that adjusts the parameter of proportional control. Rosmann et al. [34] presented an approach to the synthesis of predictive control of a time-optimal model. Laschov and Margaliot [35] considered the problem of transferring a Boolean network from a given state to a desired one in a minimal time based on the maximum principle. di Bernardo et al. [36] proposed a discrete-time minimum control synthesis algorithm for discrete-time systems, which is advisable in controlling continuous-time discrete plants.

Wilt and Sands [37], Huang and Sands [38], and Pittella and Sands [39] propose the use of feedforward control in combination with deterministic artificial intelligence (referred to as DAI in the papers) for the control of complex systems of various types. The mathematical model of the system is assumed to be no higher than a second-order differential equation. Deterministic artificial intelligence is defined as the computation of the current parameter matrix via pseudo-inversion of a linear equation, where both the current control signal and the parameter vector of the system are treated as unknowns. However, the presence of errors in the system is noted by Huang and Sands [38]. These errors are most likely caused by the direct nature of control, which is susceptible to interfering signals in the coordinate measurement channels, as well as by the use of the singular value decomposition (SVD) algorithm for control signal computation.

The analysis of the reviewed literature indicates the relevance of investigating both existing and novel artificial intelligence algorithms for enhancing the control performance of dynamic systems under interference conditions and for mitigating the impact of such disturbances within feedback control systems.

3. Problem Statement

A dynamic system is described by a system of first-order differential equations in matrix form as follows:

$$\begin{aligned} \dot{x} &= Ax + Bu, \\ y &= Cx. \end{aligned} \tag{1}$$

In Equation (1), the following notations are introduced: x is the state vector, $x \in R^n$, u is the control vector, $u \in R^m$, y is the output vector, $y \in R^p$, and A, B, and C are matrices, $A \in R^{n \times n}$, $B \in R^{n \times m}$, $C \in R^{p \times n}$. The listed vectors change in time, the time variable t in Equation (1) is omitted to simplify the notation, the components of matrices A and C are considered to be given, do not depend on time t, which corresponds to a stationary system, and matrix B contains unknown parameters associated with the gain coefficients in the system. Thus, the system of equations (Equation (1)) can include a dynamic object and a controller that provides the required shape of the control signal.

Also known are the initial state of the system at the moment $t = 0$, i.e. $x(0) \in R^n \backslash 0$, which assumes that the system initial state is not at the origin of the coordinate grid, and the final state $x(t_f)$, which is in the ε-neighborhood of the origin, i.e. $\{x(t_f) : d(x(t_f),\ 0) < \varepsilon\}$, which corresponds to the consideration of the system dynamics concerning the error.

It is further assumed that the coordinates of the state vector x are measured by a limited set of noisy sensors, i.e. there is noise ξ of unknown nature in the measurement channels, i.e.

$$\widehat{x} = x + \xi, \tag{2}$$

In Equation (2) $\widehat{x}$ is the measured state vector. This assumption is because the researcher does not have enough data to determine the static properties of the noise or does not have enough time to study its statistical properties. All that is known about the noise signal ξ is that it is limited in amplitude, i.e. the inequality is satisfied:

$$|\xi| \leq \Xi. \tag{3}$$

In Equation (3) Ξ is the maximum noise level in the sensors used in the measurement system, which is true in most practical cases, except for those where the statistical properties of the noise signal are specified in advance.

Then, Equation (2) transform the initial equations system (Equation (1)) into the following:

$$\begin{aligned} \widehat{x} &= A\widehat{x} + Bu, \\ \hat{y} &= C\widehat{x}. \end{aligned} \tag{4}$$

It is assumed that the control system (Equation (4)) ensures the movement of the state vector x from the start $x(0)$ to the final state $x(t_k)$ in a minimal, limited time, i.e.

$$T\left(\widehat{x}\right) \underset{u \in U}{\longrightarrow} t_{min}, \tag{5}$$

In Equation (5) $T\left(\widehat{x}\right)$ is the function of displacement of the coordinates of the dynamic system (Equation (4)) under the influence of the control signal u such that the minimum control time t_{min} is ensured under the conditions (Equation (3)) of the problem statement, and U is the admissible control signals in the system.

4. Problem Solution

Traditionally, this type of problem is assumed to start for the case of known parameters, and then the initial formulation of the problem with unknown parameters is considered.

4.1. Problem with known parameters

The classical formulation of the problem for the case with known parameters assumes the search for a solution in the class of control actions of maximum amplitude of the opposite sign, which corresponds to the problem of maximum speed, i.e. $u \in \{+U,\ -U\}$ [39]. In this case, two types of solution are possible, one of them is in the time plane, when it is necessary to determine the moments of switching the control signal, and the second is sought in the phase plane, when the change in the sign of the control action is determined by the so-called switching function:

$$F(x,\ c) = 0, \tag{6}$$

separating the region of control signals of opposite signs. In Equation (6), x is the vector of phase coordinates of the dynamic system, and c is the vector of parameters. The structure of the control system is in Figure 1 and its output signal without noise signal is in Figure 2. The decision in analytical form can be found for systems of the Equations (1) and (4), the order of which is not higher than the third.

It is assumed that the solution in the phase plane looks simpler and is more accurate since the control signal is switched based on measurements of the system's phase coordinates, which are represented by the position of the control object and its derivatives.

The search for a solution is somewhat complicated by the interference of Equation (3) in the phase coordinate measurement channels since interference leads to an erroneous switching of the sign of the control signal. The consequence of an error in switching the sign of the control is a delay in the control process due to the sliding mode during early switching or cyclic switching due to a later switching of

Figure 1
Control system structure

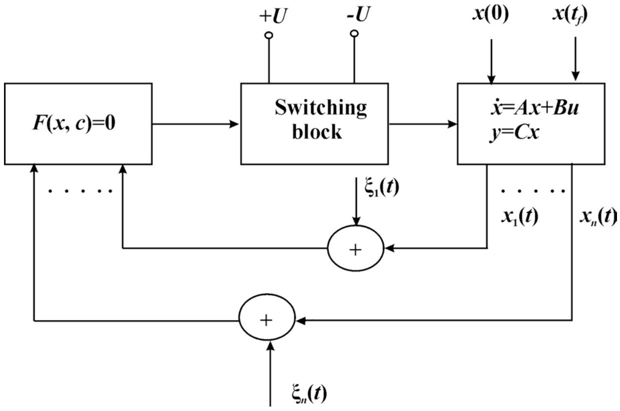
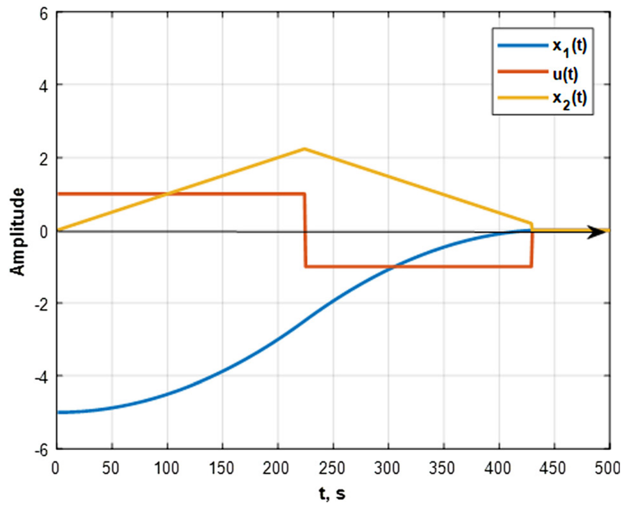


Figure 2
Control without interference ($\xi_1(t) = 0$, $\xi_2(t) = 0$) in the phase coordinate measurement channels, $x_1(t)$ is the output value, $x_2(t)$ is its derivative



the sign of the control signal to the opposite one. Nevertheless, this problem can be solved by introducing hysteresis in Equation (6) based on the phase coordinates in the form:

$$\widehat{x}' = \widehat{x} \mp \Xi,\ if\, F(x,c) \gtrless 0,\ u = \pm U \tag{7}$$

which ensures that the dynamic system reaches the target region, defined by the required sign, in the shortest possible time, as specified in Equation (5). The consequence phase point falls into the ε-neighborhood of the origin, the dimensions of which must be consistent with the initial conditions $x(0)$ and the noise level Ξ.

The described cases for a control system with a regulator as a separating function are shown in Figures 3 and 4.

Introducing coordinate-based hysteresis, as expressed in Equation (7), enables control of a dynamic system with known parameters in the minimum time specified by Equation (5).

Figure 3
Control in the presence of interference ($\xi_1(t) \neq 0,\ \xi_2(t) \neq 0$) in the channels for measuring phase coordinates without hysteresis

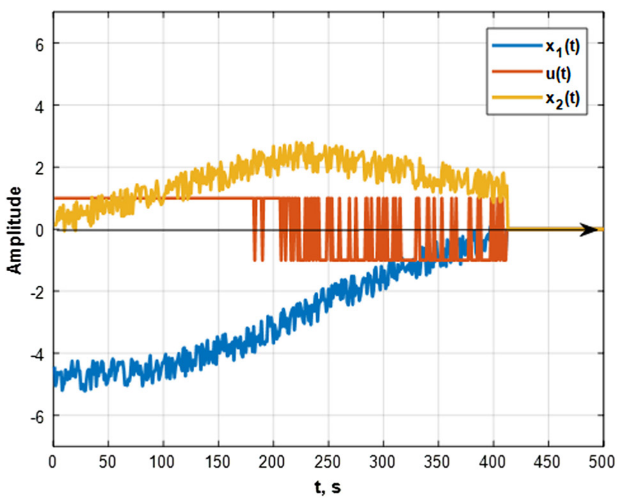
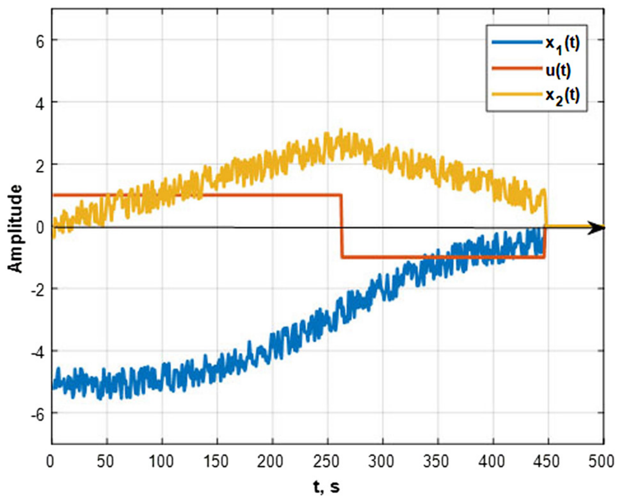


Figure 4
Control with interference ($\xi_1(t) \neq 0,\ \xi_2(t) \neq 0$) in the phase coordinate measurement channels with hysteresis, $x_1(t)$ is the output value, $x_2(t)$ is its derivative



4.2. Control of a dynamic object using a neural network

One of the powerful tools in conditions of parametric uncertainty is a neural network. A neural network is a distributed and parallel system capable of adaptive learning by changing its parameters, which are adjusted by analyzing positive and negative influences. The general element of a neural network is an artificial neuron, the algorithm of which is described by the formulas:

$$s = \sum_{i=1}^{n} w_i x_i + b, \qquad y = f(s), \tag{8}$$

where w_i is the weighting factor, $i = 1 \ldots n$; n is the number of neuron inputs; b is the displacement value; s is the summation result; x_i is the component of the input vector (input signal), $I = 1 \ldots n$; y is the output signal of the neuron; f is a nonlinear transformation function (activation function).

An artificial neuron in neural networks is an elementary data converter. According to Equation (8), it consists of multipliers, an adder, and a nonlinear converter. Numbers, so-called weighting factors, which change during setting up the neural network, multiply the components of the input signal. The adder adds the signals coming from the multipliers. The nonlinear converter transforms the received signal from the adder into a decision regarding the situation at the network's input.

In the general case, the input signal, weighting coefficients, and shift can take actual values and in many practical problems — only some fixed values. The output can be both real and integer and is determined by the type of the activation function.

4.3. Control via the PID controller

Control systems with PID controllers in the control loop are of considerable user and research interests [40]. It is determined primarily by the simplicity of the regulator design, the possibility of industrial implementation, transparent functionality, and even the possibility of implementation with microprocessors. In addition, these devices have good technical potential, such as control power and mechanical stability, and satisfy the cost-effectiveness criterion.

Along with the listed advantages, PID regulators have a serious drawback, primarily related to the complexity of the setting [41]. Since its device has parallel-connected components that have their regulation parameters, their change has a contradictory effect on the regulated process. Thus, an increase in the gain of the proportional link from one side leads to a decrease in the system error, and from the other side, it reduces the system stability. Increasing the time constant of the integrating link, on the one hand, reduces overshooting, increases the stabilizing properties of the regulator, and reduces the influence of interference but delays the control process. The differentiating block minimizes the duration of the transient process but increases the risk of high-frequency interference affecting the system's performance.

A PID controller is a device, in which the variable u forms according to the following expression:

$$u(t) = Ke(t) + \frac{1}{T_i}\int_0^t e(\tau)d\tau + T_d\frac{de(t)}{dt} \tag{9}$$

In Equation (9) t is the adjustment time, K, T_i, and T_d are the proportional coefficient, integration constant, and differentiation constant, respectively.

4.4. Control via learning

The algorithmic approach to learning assumes the presence of a changing vector of system parameters. Then, the goal of teaching is to achieve such a state of the system that the system should reach a final state. This state is considered optimal among all available alternatives. Isolation of the predominant state essentially boils down to the selection of some functionals, the extremum of which would correspond to this state. If the vector c is a vector of variable parameters, and the functional $J(c)$, then the condition of the extremum of the functional, which for certainty we will consider the minimum given by the equation:

$$\Delta J(c) = 0 \tag{10}$$

The physical meaning of the algorithmic approach consists of finding the optimal vector $c = c^*$ that satisfies Equation (10). A search algorithm can be represented in the discrete form as follows:

$$c[n] = c[n-1] - \Gamma(n)\,\nabla J(c[n-1]) \tag{11}$$

In Equation (11) Γ is a square N-dimensional matrix, the elements of which are constant or, generally speaking, depend on the current value of the vector $c\,[n-1]$. Proper selection of the matrix Γ should ensure the convergence of $c[n]$ to the optimal value of c^*.

5. Simulation

To study the considered algorithms, the object was presented in the form of a system of second-order differential equations of the Equation (1), in which the matrix A and vector B have the form [42]:

$$A = \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix},\ B = \begin{pmatrix} 0 \\ k \end{pmatrix} \tag{12}$$

In Equation (12), k is the conversion coefficient, which is considered unknown. It is assumed that the control system is equipped with a position and speed sensor, which allows measuring the output signal $x_1(t)$ and its derivative $x_2(t)$. The sensors are noisy, i.e., the measurement occurs under conditions of interfering noise. The input signal $u(t)$ is also available for measurement. According to the comments made, the matrix C has the form:

$$C = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \tag{13}$$

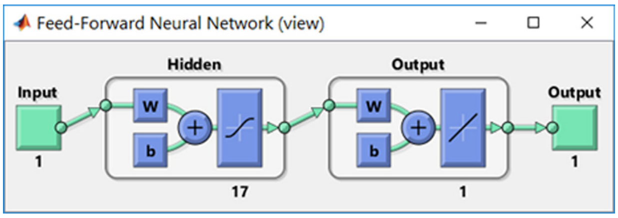
It is also assumed that the initial position of the object that should be moved to the final state corresponding to some neighborhood of the origin is known. The start state of the control object is $x_1(0) = -5$, $x_2(t) = 0$. When the matrix C is written in the Equation (13), the output vector y whole coincides with the state vector x.

5.1. Neural network

The training of the neural network (see Figure 4) using signals distorted by noise, which in the interests of the task was described by a uniform distribution, took place in the Matlab environment using the feedforwardnet function.

In the modeling, two layers represent the neural network, in which the first hidden layer has 17 neurons with a nonlinear activation function, and the output layer has one neuron with a linear activation function is sufficient. The neural network diagram is shown in Figure 5. The

Figure 5
The neural network structure in the Matlab environment



Levenberg-Marquardt algorithm was used as the training algorithm, and the criterion for the quality of neural network training was the mean square error.

The output signals of the system when controlling the neural network, assuming that the measurement of the output signals occurred without interference, are shown in Figure 6.

Analysis of Figure 6 shows that training the neural network using signals distorted by interference has significant errors, although the control signal is recognized satisfactorily.

5.2. PID-controller

The simulation of the PID controller, represented by Equation (9), with the same system in Simulink is shown in Figures 7 and 8. Figure 7 shows the measured signals and the control signal with the PID controller without interference.

From Figure 7, it is evident that the tuned PID controller tightens the processes in the system with an alternating input signal. Similar signals in the control system with a PID controller and measurement of the output value by a noisy sensor are shown in Figure 8. The noise is modeled by a separate generator with a uniform distribution in the interval $(-0.5 \ldots 0.5)U$.

From Figure 8, it is clear that the PID controller copes quite well with the problem of control in noisy conditions.

5.3. Learning algorithm

According to the problem conditions, the position of the switching function (Equation (6)) in the phase coordinate space is unknown. Its

Figure 6
Controlling a system configured by a neural network

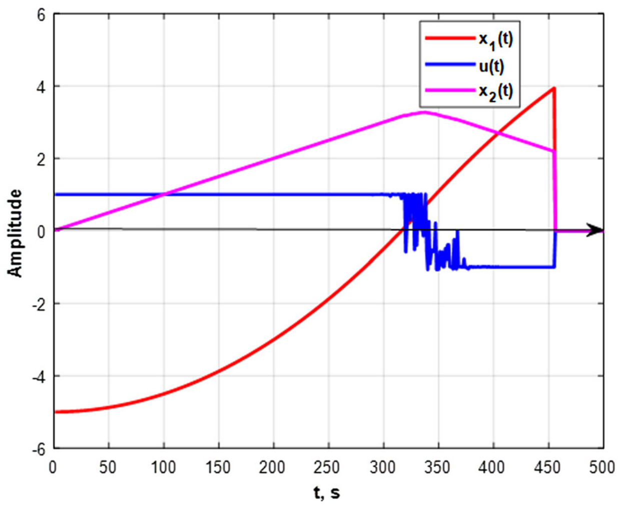


Figure 7
Controlling a system configured by a PID controller

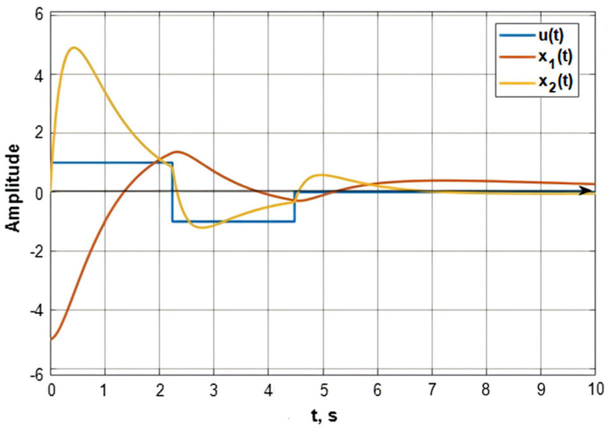
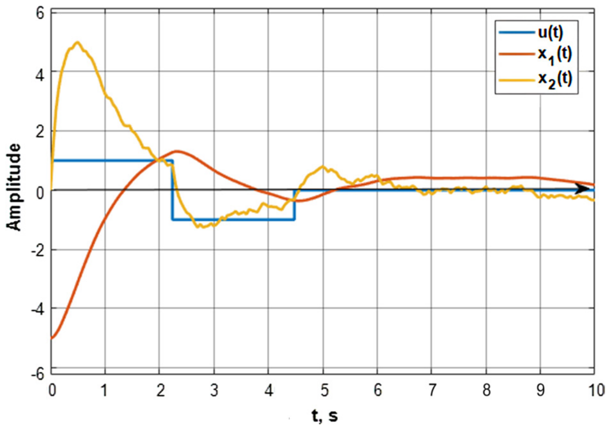


Figure 8
Signals $x_1(t)$, $x_2(t)$ and $u(t)$ in a control system with a PID controller with the noisy sensor in the output signal measurement channel



position is determined by the parameter vector $c = (c_1, c_2)^T$, and the unknown components of which are subject to determination. In the problem under consideration, the unknown parameters are corrected following an iterative algorithm that takes into account the position of the first switching point relative to the optimal position of the switching line, which is taken to be the position at which the phase point falls into the ε-neighborhood of the origin. The complete learning algorithm is a modification of the general Equation (11), in which the matrix Γ is replaced by a unit coefficient. The Equation (11), which includes the coordinates of the desired switching point and accounts for the hysteresis defined by Equation (7), is expressed by the following equations:

$$c[n] = c[n-1] + w(t[n]) \tag{14}$$

if $F(c[n-1], w(t[n]-0)) > 0$ and $|x(t[n])| > \varepsilon_1$, $|z(t[n])| \le \varepsilon_2$ with $l(T_n) = 1$ for some $T[n] > t[n]$ or if $F(c[n-1], w(t[n]+0)) > 0$ and for $l(t) > 1$ for $t > t[n]$;

$$c[n] = c[n-1] - w(t[n]) \tag{15}$$

if $F(c[n-1], w(t[n]-0)) < 0$ and $|x(t[n])| > \varepsilon_1$, $|z(t[n])| \le \varepsilon_2$ with $l(T_n) = 1$ for some $T[n] > t[n]$ or if $F(c[n-1], w(t[n]+0)) < 0$ and for $l(t) > 1$ for $t > t[n]$;

$$c_1[n] \begin{cases} c_1[n] & if\, c_1[n] > 0, \\ 0.1, & otherwise, \end{cases} \tag{16}$$

$$c_2[n] \begin{cases} c_2[n] & if\, c_2[n] > 0, \\ 0, & otherwise, \end{cases} \tag{17}$$

$$c[n] = c[n-1] \tag{18}$$

if $|x(t[n])| < \varepsilon_1$, $|z(t[n])| \le \varepsilon_2$ with $l(T[n]) = 1$ for some $T[n] > t[n]$.

In this algorithm, ε_1 and ε_2 are given values that define a given ε-neighborhood of the origin in space $\{w\} \subset R^2$, $z(t) = x_1|x_2|$, and $T[n]$ acts as the duration of the movement from the state $x(0)$ to $x(t_f)$ in this ε-neighborhood at each n-th learning step. In Equations (14)–(18), the authors propose a modification of the one in Kucherov et al. [43].

The results of the algorithm are presented in Figures 9 and 10.

Figure 9
Signals $x_1(t)$, $x_2(t)$, and $u(t)$ in a control system with a learning controller with noise in the output signal measurement channel and incorrect vector c on the first learning step

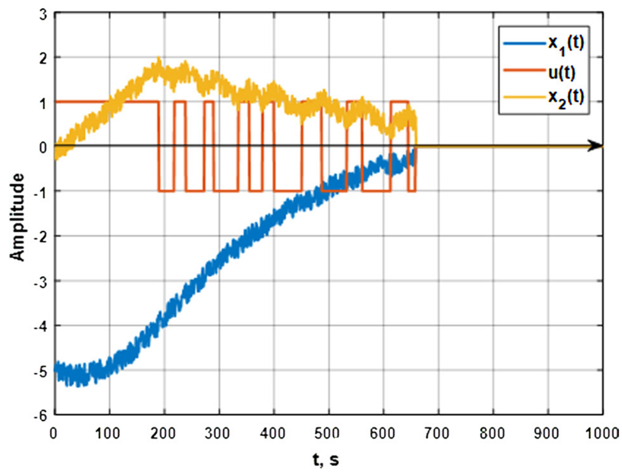
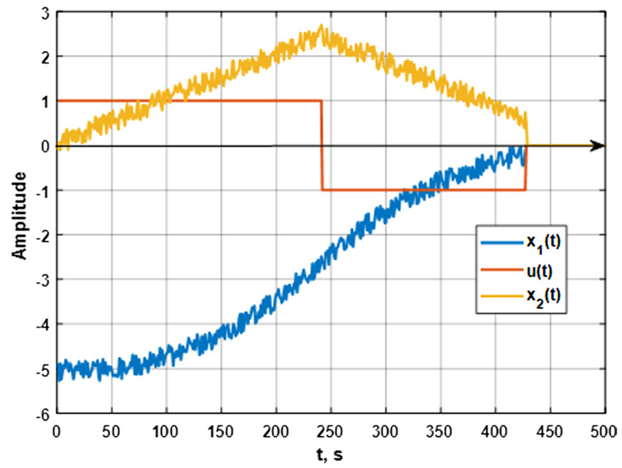


Figure 10
Signals $x_1(t)$, $x_2(t)$, and $u(t)$ in a control system with a learning controller with noise in the output signal measurement channel on the last learning step



The evaluation of the correction results was carried out using the Lyapunov-type convergence criterion in the form:

$$J(i) = (100 - c_1)^2 + (50 - c_2)^2 \tag{19}$$

The numbers in Equation (19) correspond to the actual values of the uncertainty parameters, increased by 100 times. The function J(i) is shown in Figure 11.

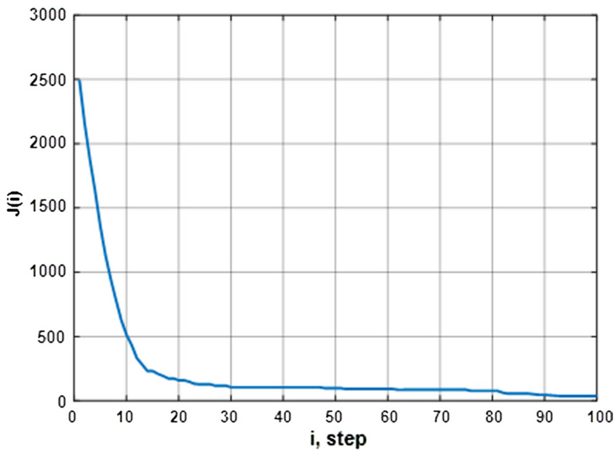
The monotonically decreasing nature of the curve constructed based on the results of calculating Equation (19) indicates satisfactory learning results.

6. Discussion

The methods considered allow controlling a dynamic system; however, several comments should be made. It would seem that fine-tuning a control system with unknown parameters against the interfering noise can be ensured by means such as, for example, a PID controller. It has become a standard in the field of control systems. However, these devices require precise tuning by algorithms, as presented in Kucherov et al. [41]. In addition, the presence of an integrator introduces additional inertia into the system, which leads to a delay in the control process and overshooting, and the effect of noise increases dynamic and static errors in control.

The nature of the noise is usually unknown and difficult to predict, and for this reason, there is no universal method of countermeasure it.

Figure 11
Function *J*(*i*)



Therefore, the paper presents one of the methods of counteraction as presented in Kucherov et al. [43], based on hysteresis, for a low signal-to-noise ratio (SNR).

Similar disadvantages are also present in the neural controller based on trained neurons. Due to the simple mathematical model of the control object (2–3 orders of differential equation), there is no point in complicating the structure of the neural network by increasing the layers or the number of neurons due to the increase in the "cost" of the controller. The gain in control accuracy and time does not compensate for the rise in control value since noise is still present. Nevertheless, the learning algorithm synthesized on simple mathematical rules allows us to reduce the impact of a priori uncertainty caused by the lack of information about the parameters of the object and the nature of noise.

Providing sufficient control accuracy in the time domain by calculating switching moments is also an erroneous control strategy due to their dependence on the object parameters, which can change. A possible option for constructing an effective control system can be implemented in the phase plane by adjusting unknown coefficients of the switching line according to the proposed learning algorithm. The presented statements confirm the quantitative indicators of the regulator adjustment based on independent measurements, which are presented in the Table 1.

The estimates are given for the same dynamic object. However, the PID controller was tuned in the absence of interference, and then the interference level changed stepwise, so the number of tuning iterations is constant and is determined by the performance of the controller, the role of which in this case was performed by a computer with installed software and a controller-tuning program. The PID controller showed stable indicators under interference conditions, but the duration of the transient process exceeds adaptive controllers by approximately 3–4 times.

Although the neural network obtained better results in the index of setup time compared to the learning algorithm, the accuracy is approximately 2 times less. In addition, the simplicity of the technical implementation of the learning algorithm allows it to be recommended for use in a promising adaptive regulator to obtain an effective control system for dynamic processes.

The main characteristics of the control processes implemented by different regulators applied to the same object, under identical initial conditions and noise levels, are presented in Table 2.

The error values presented in Table 2 are calculated solely for the output coordinate, defined as the ratio of the last value to the initial state. The time metric refers to the interval within which the output coordinate reaches the required value. The results demonstrate the superiority of the proposed controller based on the learning algorithm.

Table 1
Parameters of the dynamic system for the studied types of regulators

	Error, percentage		Time, s		Number of iterations	
Regulator	**SNR=−6 dB**	**SNR=−20 dB**	**SNR=−6 dB**	**SNR=−20 dB**	**SNR=−6 dB**	**SNR=−20 dB**
PID	8	15	15	18	15	15
Neural net	25	49	4.08	3.72	19	8
Learning algorithm	10	30	4.32	4.18	10	40

Table 2
Main output parameters of the dynamic system for the studied types of regulators

Regulator	Error, %	Time, s	Number of switching intervals
Optimal control	0.01	4.42	2
PID	3.8	5.2	2
Neural net	78	4.52	2
Learning algorithm	0.7	4.5	2

Table 3
Table of abbreviations and variables

Abbreviations	Meaning	Variable	Meaning
CNN	Convolution neural networks	A, B, C	Matrices
DAI	Deterministic Artificial Intelligence	b	Displacement value
MSIMD	Multiple Single Instruction Multiple Data	c	Vector parameters
PID	Proportional-integral-derivative	$e(t)$	Error in time plane
RNN	Recurrent neural networks	$F(\cdot)$	Switching function
SNR	Signal to noise ratio	i	Number of iteration
		$J(\cdot)$	Quality functional
		K, T_i, T_d	PID controller parameters
		l	number of switching
		m, n, p	Integer value, size of matrix
		s	Neuron solving function
		t	Time
		u	Control vector
		U	Set of control signal
		w	Weighting vector
		x, $\hat{x}$	State vector and its evaluation
		y, $\hat{y}$	Output vector and its evaluation
		z	Additional vector
		ε	Error in phase plane
		ξ	Vector noise
		Ξ	Maximum noise level

7. Conclusion

This paper examines modern approaches to the design of control systems operating under interference signals acting in coordinate measurement channels. These approaches are based on the use of artificial intelligence elements, including a PID controller with intelligent tuning, a neural network used as a feedforward controller, and a controller employing a learning algorithm. Their performance is compared to that of an optimal controller.

Controllers based on classical and PID algorithms need to be adjusted to obtain the given parameters of the transient process, but interference worsens the dynamics of the control process.

A controller based on a neural network needs a reference signal, but interference significantly degrades the output signal and does not provide acceptable indicators of the transient process. At the same time, there is difficulty in modifying the neural network to obtain better indicators.

Learning algorithms can be used independently in the controller, the regulator structure is greatly simplified, and it is possible to obtain acceptable transition process results. To get better results, the learning algorithm needs to be complicated due to the refinement of previous results and the accumulation of knowledge about the operation of the dynamic system.

For the reader's convenience, all abbreviations and variables used in the paper are summarized in Table 3.

Acknowledgement

The study was carried out on an initiative basis in the interest of obtaining objective evidence of the advantages of learning algorithms in solving specific problems.

Ethical Statement

This study does not contain any studies with human or animal subjects performed by any of the authors.

Conflicts of Interest

The authors declare that they have no conflicts of interest to this work.

Data Availability Statement

The data that support the findings of this study are openly available at https://drive.google.com/drive/u/0/folders/1gqzZ6Tla1RDaJAl86rp7yHHbs6Ggj_wp.

Author Contribution Statement

Dmytro Kucherov: Conceptualization, Methodology, Formal analysis, Investigation, Supervision, Project administration, Funding acquisition. **Natalia Khalimon:** Resources, Data curation, Writing - original draft, Writing - review & editing, Visualization. **Ihnat Myroshnychenko:** Software, Validation. **Valerii Tkachenko:** Validation.

References

[1] Mahesh, B. (2020). Machine learning algorithms—A review. *International Journal of Science and Research*, *9*(1), 381–386. https://doi.org/10.21275/ART20203995

[2] Ray, S. (2019). A quick review of machine learning algorithms. In *2019 International Conference on Machine Learning, Big Data,*

Cloud and Parallel Computing, 35–39. https://doi.org/10.1109/COMITCon.2019.8862451

[3] Stanley, K. O., Clune, J., Lehman, J., & Miikkulainen, R. (2019). Designing neural networks through neuroevolution. *Nature Machine Intelligence*, *1*(1), 24–35. https://doi.org/10.1038/s42256-018-0006-z

[4] Zhao, Y., Li, Y., Zhang, X., Geng, G., Zhang, W., & Sun, Y. (2019). A survey of networking applications applying the software defined networking concept based on machine learning. *IEEE Access*, *7*, 95397–95417. https://doi.org/10.1109/ACCESS.2019.2928564

[5] Gurney, K. (2018). An introduction to neural networks. *UK: CRC Press*. https://doi.org/10.1201/9781315273570

[6] Bouzid, Y., Derrouaoui, S. H., & Guiatni, M. (2021). PID gain scheduling for 3D trajectory tracking of a quadrotor with rotating and extendable arms. In *2021 International Conference on Recent Advances in Mathematics and Informatics*, 1–4. https://doi.org/10.1109/ICRAMI52622.2021.9585973

[7] Hadid, S., Boushaki Zamoum, R., & Refis, Y. (2025). Linear and nonlinear control design for a quadrotor. *Bulletin of Electrical Engineering and Informatics*, *14*(2), 940–955. https://doi.org/10.11591/eei.v14i2.8234

[8] Guettal, L., Chelihi, A., Ajgou, R., & Touba, M. M. (2022). Robust tracking control for quadrotor with unknown nonlinear dynamics using adaptive neural network based fractional-order backstepping control. *Journal of the Franklin Institute*, *359*(14), 7337–7364. https://doi.org/10.1016/j.jfranklin.2022.07.043

[9] Cengiz, K., Lipsa, S., Dash, R. K., Ivković, N., & Konecki, M. (2024). A novel intrusion detection system based on artificial neural network and genetic algorithm with a new dimensionality reduction technique for UAV communication. *IEEE Access*, *12*, 4925–4937. https://doi.org/10.1109/ACCESS.2024.3349469

[10] Ma, Z., Wu, G., Suganthan, P. N., Song, A., & Luo, Q. (2023). Performance assessment and exhaustive listing of 500+ nature-inspired metaheuristic algorithms. *Swarm and Evolutionary Computation*, *77*, 101248. https://doi.org/10.1016/j.swevo.2023.101248

[11] Murugesan, K., & Ramasubbu, R. (2025). Driving training-based optimization technique for estimating synchronous motor excitation current. *Bulletin of Electrical Engineering and Informatics*, *14*(2), 813–822. https://doi.org/10.11591/eei.v14i2.8579

[12] Gabella, M. (2021). Topology of learning in feedforward neural networks. *IEEE Transactions on Neural Networks and Learning Systems*, *32*(8), 3588–3592. https://doi.org/10.1109/TNNLS.2020.3015790

[13] Shrestha, A., & Mahmood, A. (2019). Review of deep learning algorithms and architectures. *IEEE Access*, *7*, 53040–53065. https://doi.org/10.1109/ACCESS.2019.2912200

[14] Li, Z., Shang, T., & Xu, P. (2025). Multi-modal attention perception for intelligent vehicle navigation using deep reinforcement learning. *IEEE Transactions on Intelligent Transportation Systems*, *26*(6), 8657–8669. https://doi.org/10.1109/TITS.2025.3535885

[15] Garro, B. A., Vazquez, R. A. (2015). Designing artificial neural networks using particle swarm optimization algorithms. *Computational Intelligence and Neuroscience*, *2015*(1), 369298. https://doi.org/10.1155/2015/369298

[16] Such, F. P., Madhavan, V., Conti, E., Lehman, J., Stanley, K. O., Clune, J. (2019). Deep neuroevolution: Genetic algorithms are a competitive alternative for training deep neural networks for reinforcement learning. In *International Conference on Learning Representations*, 1-23.

[17] Tsmots, I., Teslyuk, V., Teslyuk, T., Ihnatyev, I. (2018). Basic components of neuronetworks with parallel vertical group data real-time processing. In *Advances in Intelligent Systems and Computing II: Selected Papers from the International Conference on Computer Science and Information Technologies*, *689*, 558–576. https://doi.org/10.1007/978-3-319-70581-1_39

[18] Chen, C. L. P., & Liu, Z. (2018). Broad learning system: An effective and efficient incremental learning system without the need for deep architecture. *IEEE Transactions on Neural Networks and Learning Systems*, *29*(1), 10–24. https://doi.org/10.1109/TNNLS.2017.2716952

[19] Ou, C., Zhu, H., Shardt, Y. A. W., Ye, L., Yuan, X., Wang, Y., & Yang, C. (2025). Quality-driven regularization for deep learning networks and its application to industrial soft sensors. *IEEE Transactions on Neural Networks and Learning Systems*, *36*(3), 3943–3953. https://doi.org/10.1109/TNNLS.2022.3144162

[20] Leo, J., & Kalita, J. (2022). Incremental deep neural network learning using classification confidence thresholding. *IEEE Transactions on Neural Networks and Learning Systems*, *33*(12), 7706–7716. https://doi.org/10.1109/TNNLS.2021.3087104

[21] Li, H., & Zhang, L. (2021). A bilevel learning model and algorithm for self-organizing feed-forward neural networks for pattern classification. *IEEE Transactions on Neural Networks and Learning Systems*, *32*(11), 4901–4915. https://doi.org/10.1109/TNNLS.2020.3026114

[22] Dai, P., Liu, H., Yu, W., & Wang, H. (2023). Distributed neural learning algorithms for multiagent reinforcement learning. *IEEE Internet of Things Journal*, *10*(23), 21039–21060. https://doi.org/10.1109/JIOT.2023.3284510

[23] Huoh, T.-L., Luo, Y., Li, P., & Zhang, T. (2023). Flow-based encrypted network traffic classification with graph neural networks. *IEEE Transactions on Network and Service Management*, *20*(2), 1224–1237. https://doi.org/10.1109/TNSM.2022.3227500

[24] Sayed, A. H. (2014). Adaptive networks. *Proceedings of the IEEE*, *102*(4), 460–497. https://doi.org/10.1109/JPROC.2014.2306253

[25] Lin, F. (2022). Supervised learning in neural networks: Feedback-network-free implementation and biological plausibility. *IEEE Transactions on Neural Networks and Learning Systems*, *33*(12), 7888–7898. https://doi.org/10.1109/TNNLS.2021.3089134

[26] Zhou, Q., Zhao, D., Shuai, B., Li, Y., Williams, H., & Xu, H. (2021). Knowledge implementation and transfer with an adaptive learning network for real-time power management of the plug-in hybrid vehicle. *IEEE Transactions on Neural Networks and Learning Systems*, *32*(12), 5298–5308. https://doi.org/10.1109/TNNLS.2021.3093429

[27] Liu, D., Xue, S., Zhao, B., Luo, B., & Wei, Q. (2021). Adaptive dynamic programming for control: A survey and recent advances. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, *51*(1), 142–160. https://doi.org/10.1109/TSMC.2020.3042876

[28] Sun, T., & Sun, X.-M. (2021). An adaptive dynamic programming scheme for nonlinear optimal control with unknown dynamics and its application to turbofan engines. *IEEE Transactions on Industrial Informatics*, *17*(1), 367–376. https://doi.org/10.1109/TII.2020.2979779

[29] Heer, H., Streib, L., Schäfer, R. B., & Ruzika, S. (2020). Maximising the clustering coefficient of networks and the effects on habitat network robustness. *Plos One*, *15*(10), e0240940. https://doi.org/10.1371/journal.pone.0240940

[30] Yin, H., Benson, A. R., & Leskovec, J. (2019). The local closure coefficient: A new perspective on network clustering. In *Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining*, 303–311. https://doi.org/10.1145/3289600.3290991

[31] Trischler, A. P., D'Eleuterio, G. M. T. (2016). Synthesis of recurrent neural networks for dynamical system simulation.

Neural Networks, *80*, 67–78. https://doi.org/10.1016/j.neunet.2016.04.001
[32] Wang, Y. (2017). A new concept using LSTM Neural Networks for dynamic system identification. In *2017 American Control Conference*, 5324–5329. https://doi.org/10.23919/ACC.2017.7963782
[33] Li, G., Li, S., Li, B., & Wu, Y. (2024). Deep reinforcement learning guidance with impact time control. *Journal of Systems Engineering and Electronics*, *35*(6), 1594–1603. https://doi.org/10.23919/JSEE.2024.000111
[34] Rosmann, C., Makarow, A., Hoffmann, F., & Bertram, T. (2017). Time-optimal nonlinear model predictive control with minimal control interventions. In *2017 IEEE Conference on Control Technology and Applications*, 19–24. https://doi.org/10.1109/CCTA.2017.8062434
[35] Laschov, D., & Margaliot, M. (2013). Minimum-time control of Boolean networks. *SIAM Journal on Control and Optimization*, *51*(4), 2869–2892. https://doi.org/10.1137/110844660
[36] di Bernardo, M., di Gennaro, J. O. F., Olm, J. M., & Santini, S. (2010). Discrete-time minimal control synthesis adaptive algorithm. *International Journal of Control*, *83*(12), 2641–2657. https://doi.org/10.1080/00207179.2010.536916
[37] Wilt, E., & Sands, T. (2022). Microsatellite uncertainty control using deterministic artificial intelligence. *Sensors*, *22*(22), 8723. https://doi.org/10.3390/s22228723
[38] Huang, B.-R., & Sands, T. (2023). Novel learning for control of nonlinear spacecraft dynamics. *Journal of Applied Math*, *1*(1), 42. https://doi.org/10.59400/jam.v1i1.42
[39] Pittella, A., Sands, T. (2023). Proposals for surmounting sensor noises. *Sensors*, *23*(6), 3169. https://doi.org/10.3390/s23063169
[40] Oustry, A., Tacchi, M. (2023). Minimal time nonlinear control via semi-infinite programming. *arXiv Preprint*: 2307.00857. https://doi.org/10.48550/arXiv.2307.00857
[41] Kucherov, D., Kozub, A., Tkachenko, V., Rosinska, G., & Poshyvailo, O. (2021). PID controller machine learning algorithm applied to the mathematical model of quadrotor lateral motion. In *2021 IEEE 6th International Conference on Actual Problems of Unmanned Aerial Vehicles Development*, 86–89. https://doi.org/10.1109/APUAVD53804.2021.9615438
[42] Kucherov, D., Shmelova, T., Poshyvailo, O., Tkachenko, V., Miroshnichenko, I., & Ogirko, I. (2023). Mathematical model of damping of UAV oscillations in the cargo delivery problem. In *2023 IEEE 4th KhPI Week on Advanced Technology*, 1–6. https://doi.org/10.1109/KhPIWeek61412.2023.10312855
[43] Kucherov, D. P., Jiang, G., Liu, H., & Fu, M. (2024). UAV group control protocol with adaptive consensus. *International Journal of Adaptive Control and Signal Processing*, *38*(9), 3177–3194. https://doi.org/10.1002/acs.3868

How to Cite: Kucherov, D., Myroshnychenko, I., Khalimon, N., & Tkachenko, V. (2025). Training of the Dynamic Systems Control: A Neural Network or a Learning Algorithm. *Artificial Intelligence and Applications*. https://doi.org/10.47852/bon-viewAIA52025435