NEUROCONTROLLED OBJECT PARAMETERS ADJUSTMENT BY ACKERMANN'S FORMULA USAGE

Markiyan Nakonechnyi, Dr. Sc., Professor; Orest Ivakhiv, Dr. Sc., Professor; Oleksandr Viter, Ph. D., Ass.-Prof.; Yuriy Nakonechnyi, Ph. D., Ass.-Prof.
Lviv Polytechnic National University, Ukraine, e-mail: viteros45@gmail.com

https://doi.org/10.23939/istcmtm2020.01.022

Abstract. Synthesis methods of controllers based on the use of frequency characteristics or root hodographs are considered classic or traditional. Frequency methods are available in practical applications, and most control systems are designed based on various modifications to these methods. A distinctive feature of these methods is the so-called robustness, which means that the characteristics of a closed system are insensitive to the minor errors of the model of the real system. This feature is significant because of the complexity of constructing an accurate model of the real system, as well as the fact that many systems are inherent in all kinds of nonlinearities, which complicate their analysis and synthesis. In recent years, many attempts have been made to develop new methods of synthesis, commonly referred to as modern control theory. One synthesis method is like the root hodograph method, which allows positioning the poles of the closed-loop transfer function at predetermined points.

In the article on the basis of information about the desired transient characteristic of the reference, which is obtained on the basis of a dynamic neural network, using the Ackerman formula, a procedure for calculating the coefficient matrix, whose introduction in the structure of the object model provides the specified dynamics of the process. On the base of the reference mathematical model is created the architecture of the corresponding dynamic neural network. During training, there is the target function as a numerical sequence that corresponds to the desired transient characteristic of the system, and the input signal is given in the form of a numerical sequence that reproduces jump function. Using the values of the weight coefficients obtained in the course of learning the neural network, the coefficients of the mathematical model of the reference and the roots of its characteristic equation are calculated, with the following calculation using the Ackerman formula of the coefficients of the matrix, whose values are entered into the structure of the model ensuring the specified dynamics of the process in it.

Key words: Mechatronics, Robotics, System, Synthesis, Dynamics, Ackermann's formula, Neural network.

1. Introduction

The robots become more sophisticated and reliable from year to year. They often are used in the different manufacturing operations such as at the assembly line and perform a variety of tasks usually done by skilled operators. The tasks range from pick and place, welding and pointing, to the placement of engine blocks into cars. Today's manufacturing cells are designed around the capabilities of robot speed and reach, and they are becoming an essential element of manufacturing processes.

Mostly mechatronic as well as robotic systems consist of nonlinear elements that are covered by complex feedbacks, and the operation of such systems in the real world is affected by a variety of noises, interferences and other disturbing factors [1]–[3]. It significantly limits the use of modern and classical control theory when creating controls [4]–[6]. In recent decades, management strategies have used theories based on the idea of system linearization, which does not fully reflect its physical properties, and in some cases, even when the dependencies between the inputs and outputs of the system are accurately reproduced, their use cannot provide adequate control of the system. Therefore, artificial neural networks are increasingly used in synthesizing control algorithms, which take into account the features of the object that the network must reproduce, and its training is conducted on the basis of the input and output data that characterize the processes that take place in the object [7]–[15]. Because neural networks are inherently nonlinear, they can be used to identify both linear and nonlinear objects, as well as to implement control algorithms in such objects.

In recent years, the synthesis of neural controllers to control processes in dynamic objects has been performed based on methods based on the use of a root hodograph or frequency response [2], [4], [5]. It is known, when using a root hodograph synthesis of the controller can be done in two ways. In the first case, the synthesis is based on the requirements imposed on the system operation in transient mode, and the second involves the use of system characteristics in the steady-state.

One of the peculiarities of the root hodograph method is that when used, the controller synthesis is performed in the time domain, which requires the use of a mathematical model of an object with characteristics that are close to the characteristics of a real object, which in some cases is difficult to implement. In addition, the scope of the root hodograph method is limited by the fact that it provides the possibility of location in a given area of the complex plane only one pair of poles and in cases where the order of the system is higher than the second, it is almost impossible to influence the location of the other poles. Thus, if these poles in the complex plane occupy positions that do not allow providing the specified dynamics of the system, then the synthesis of the controller based on the root hodograph method is reduced to using the trial and error method. The controller synthesis using frequency characteristics is based on the Nyquist stability criterion, which allows evaluating the stability of a closed system by its frequency characteristics when open. For the synthesis of the controller using frequency characteristics, the basic dynamic parameters of the system must be predefined the open system transmission coefficient, phase stability or bandwidth, installation time, perturbation compensation efficiency, etc.

Frequency methods are widely used to synthesize controllers and most control systems are designed based on various modifications to these methods. When using frequency methods, it is possible to provide to a certain extent the insensitivity of the characteristics of a closed system to changing its parameters, which is essential for control systems with pronounced nonlinearities, the presence of which significantly complicates the process of controller synthesis.

It should be noted that frequency methods involve the transition from the time domain of signal representation to the frequency with the subsequent procedure of controller synthesis, which significantly complicates the possibility of direct estimation of temporal characteristics of the system and, as a consequence, the choice of effective ways of their improvement.

In recent years, certain methods of synthesis of controllers, the implementation of which involves the possibility of placing all the poles of the transfer function of a closed system at specified points of a complex plane, have acquired certain development [6]. The use of these methods requires the availability of information about object state variables, the measurement of which is not always possible due to the complexity of the access to them or the absence of appropriate measurement converters. In practice, the estimation of state variables, whose measurement causes significant difficulties, is performed based on data on variables that can be directly measured.

2. Task of the Research

The goal of the current paper consists of the specified dynamics of the process getting based on the desired reference transient characteristic which was taken from a dynamic neural network. The last may be obtained thanks to the Ackermann's formula using for calculating the coefficient matrix, whose introduction in the structure of the object model provides the specified dynamics of the process.

3. Formulation of the Problem

Synthesis of controllers by positioning the poles of the transfer function of a system at given points of a complex plane implies the presence of its mathematical model in state variables, which for a linearized object is as follows:

$$\begin{aligned} \dot{x}(t) &= Ax(t) + Bu(t), \\ y(t) &= Cx(t) + Du(t), \end{aligned} \tag{1}$$

where x(t) is a state vector of dimension (n×1) whose components are the *n*-order object state variables; $\dot{x}(t)$ is a time derivative of the vector x(t); A is a matrix of coefficients with (n×n) dimension; B is the input matrix with (n×r) dimension; u(t) is the input vector of dimension (r×1) with components as input object variables; y(t) is the output vector of dimension (p×1), the components of which are the output object variables; C is the output matrix with (p×n) dimension; D is the bypass matrix that determines the direct link between object input and output. In most cases, the matrix D elements are equal to zero because the connection between inputs and outputs in real physical objects is dynamic. Let us consider a control object that is described by the following transmitting function:

$$W(s) = \frac{b_{n-1} \cdot S^{n-1} + b_{n-2} \cdot S^{n-2} + \ldots + b_1 \cdot S + b_0}{S^n + a_{n-1} \cdot S^{n-1} + a_{n-2} \cdot S^{n-2} + \ldots + a_1 \cdot S + a_0}. \tag{2}$$

For such an object, the equations of state can be written as follows

$$\dot{x}(t) = \begin{bmatrix} 0 & 1 & 0 & \ldots & 0 & 0 \\ 0 & 0 & 1 & \ldots & 0 & 0 \\ \ldots & \ldots & \ldots & \ldots & \ldots & \ldots \\ 0 & 0 & 0 & \ldots & 0 & 1 \\ -a_0 & -a_1 & -a_2 & \ldots & -a_{n-1} & -a_n \end{bmatrix} x(t) + \begin{bmatrix} 0 \\ 0 \\ \ldots \\ 0 \\ 1 \end{bmatrix} u(t), \tag{3}$$

$$y(t) = [b_0\ b_1\ b_2 \ldots b_{n-2}\ \ b_{n-1}]\, x(t).$$

The peculiarity of the coefficient matrix *A* is that all its coefficients, except for coefficients above the main diagonal, are equal to zero, and the last raw is filled with the coefficients of the characteristic equation with the opposite sign. Thus, knowing the matrix of coefficients of an object, one can write its characteristic equation. Consider a synthesis procedure for a controller based on the location of the poles of the closed-loop transfer function at given points in the complex plane. It is assumed that the system has one input and one output, and the defining signal supplied to its input is equal to zero. The control law can then be represented as a functional relationship between the state vectors and the object input as follows:

$$U(t)=f\left[x(t)\right]. \tag{4}$$

Let's form a control law for our case in the following form:

$$u(t)=\left[-K_1,-K_2,...-K_n\right]\times\begin{bmatrix}X_1(t)\\X_2(t)\\......\\X_n(t)\end{bmatrix}= \tag{5}$$

$$=-K_1X_1(t)-K_2X_2(t)-...-K_nX_n(t)=K\left[X(t)\right],$$

where K is the coefficients vector of dimension [1 × n]. Substituting (5) into the first equation of system (1), one obtains:

$$\dot{x}(t)=Ax(t)-BKx(t)=(A-BK)x(t)=A_f x(t), \tag{6}$$

where A_f is the following matrix of closed-loop coefficients:

$$A_f=\begin{bmatrix}0 & 1 & 0 & ... & 0\\0 & 0 & 1 & ... & 0\\ \multicolumn{5}{c}{......................................}\\0 & 0 & 0 & & 1\\-a_0-K_1 & -a_1-K_2 & -a_2-K_3 & ... & -a_{n-1}-K_n\end{bmatrix}. \tag{7}$$

The characteristic equation of a closed system can be written in the following form:

$$\left[SI-A+BK\right]=$$
$$=\begin{bmatrix}S & 0 & 0 & ... & 0\\0 & S & 0 & ... & 0\\ \multicolumn{5}{c}{..............................}\\0 & 0 & S & ... & 0\\S & 0 & 0 & ... & S\end{bmatrix}-$$
$$-\begin{bmatrix}0 & 1 & ... & 0\\0 & 0 & ... & 0\\ \multicolumn{4}{c}{..............................}\\0 & 0 & ... & 1\\-a_0-K_1 & -a_1-K_2 & ... & -a_{n-1}-K_n\end{bmatrix}= \tag{8}$$
$$=\begin{bmatrix}0 & 1 & ... & 0\\0 & 0 & ... & 0\\ \multicolumn{4}{c}{..............................}\\0 & 0 & ... & 1\\a_0+K_1 & a_1+K_2 & ... & a_{n-1}+K_n\end{bmatrix}=$$
$$=S^n+(a_{n-1}+K_n)S^{n-1}+...+(a_1+K_2)S+(a_0+K_1)=0.$$

where I is a unitary matrix of dimension [n × n].

If considering the conditions of the given process dynamics, the roots of the characteristic equation of a closed system should acquire values $-\lambda_1$, $-\lambda_2$, ..., $-\lambda_{n-1}$, then the desired characteristic equation for such a system will be as follows:

$$\begin{aligned}F(s)&=S^n+a_{n-1}S^{n-1}+...+a_1S+a_0=\\&=(S+\lambda_1)(S+\lambda_2)...(S+\lambda_n)=0.\end{aligned} \tag{9}$$

In accordance with the procedure for the synthesis of the controller on the basis of the poles positioning method at given points of the complex plane, it is necessary to calculate a matrix *K*, which would ensure the equality of the left parts in expressions (8) and (9), i.e.:

$$\begin{aligned}&S^n+(\alpha_{n-1}+K_n)S^{n-1}+...+(\alpha_1+K_2)S+\\&+(\alpha_0+K_1)=S^n+\alpha_{n-1}S^{n-1}+...+\alpha_1S\ +\alpha_0.\end{aligned} \tag{10}$$

Equating in (10) the coefficients for equal degrees *S*, one obtains *n* linear equations with respect to *n* unknowns as follows:

$$a_{i-1}+K_i=\alpha_{i-1} \tag{11}$$

the solutions of which are the coefficients of the matrix K_i

$$K_i=\alpha_{i-1}-a_{i-1},\ \text{where}\quad i=1,2,...,n. \tag{12}$$

Relations (12) determine the overall solution of the problem of synthesis of a system with one input and one output on the basis of the poles positioning method at the setpoints of the complex plane, but it is necessary to ensure the correspondence of the model of the system in the canonical form of controllability. Remark, that it cannot always be achieved because the state variables of the model in the canonical form of control in most cases do not correspond to the state variables of the real system. Therefore, consequently, they are not those variables that determine the physical content of the processes that take place in the real system. In addition, state variables in the canonical form of control may in some cases be unavailable for measurement. Thus, in the general case, the calculation of the coefficients of the matrix K is based on the transformation of similarity, which provides the possibility of transition from a given model of any structure to a canonical form of controllability, with subsequent carrying out of the corresponding mathematical transformations. To implement this procedure can be used Ackerman's formula [16] in the following form:

$$\begin{aligned}K&=\left[0\ 0\ ...\ 0\ 1\right]\times\\&\times\left[B\ \ AB\ ...\ A^{n-2}B\ \ A^{n-1}B\right]^{-1}\times\alpha_c(A),\end{aligned} \tag{13}$$

where $\alpha_c(A)$ is a characteristic polynomial calculated on the basis of the coefficients of the desired characteristic equation of the system $\alpha_c(S)$, i.e.

$$\alpha_c(A)=A^n+\alpha_{n-1}A^{n-1}+...+\alpha_1A+\alpha_0I, \tag{14}$$

where I is a unit matrix. Consider the problem of a dynamic neural network designing to reproduce the transition characteristic, which is given in the form of a numerical sequence. Consider the problem of a dynamic neural network creation to reproduce the transient characteristic, which is given in the form of a numerical sequence. Let us suppose that a physical object or its mathematical model in the form of a differential equation, which includes input and output quantities and their derivatives in time up to the *n*-th order, is used to obtain the desired transient characteristic. Remark that the dependence of the output quantities on the input and derivatives from the input and output is unambiguous. Let us assume that the analytical representation of this equation is unknown in advance or known only in general form.

The above assumptions about the differential equation make it possible to present it in the general form:

$$F(x,x',x'',\dots,x^{(n_x)},y,y',y'',\dots,y^{(n_y)})=0, \qquad (15)$$

where n_x and n_y is the maximum order of the derivatives from the inputs and the outputs, respectively; x, x', x'', $\dots, x^{(n_x)}$ and $y, y', y'', \dots, y^{(n_y)}$ is the input and output vector and its derivatives, accordingly, i.e.

$$x=\begin{bmatrix} x_1 \\ x_2 \\ \dots \\ x_{S^m} \end{bmatrix},\ x'=\begin{bmatrix} \frac{dx_1}{dt} \\ \frac{dx_2}{dt} \\ \dots \\ \frac{dx_{S^m}}{dt} \end{bmatrix},\ \dots,\ x^{(n_x)}=\begin{bmatrix} \frac{d^{n_x}x_1}{dt^{n_x}} \\ \frac{d^{n_x}x_2}{dt^{n_x}} \\ \dots \\ \frac{d^{n_x}x_{S^m}}{dt^{n_x}} \end{bmatrix};$$

$$y=\begin{bmatrix} y_1 \\ y_2 \\ \dots \\ y_{S^n} \end{bmatrix},\ y'=\begin{bmatrix} \frac{dy_1}{dt} \\ \frac{dy_2}{dt} \\ \dots \\ \frac{dy_{S^n}}{dt} \end{bmatrix},\ \dots,\ y^{(n_y)}=\begin{bmatrix} \frac{d^{n_y}y_1}{dt^{n_y}} \\ \frac{d^{n_y}y_2}{dt^{n_y}} \\ \dots \\ \frac{d^{n_y}y_{S^n}}{dt^{n_y}} \end{bmatrix}.$$

The task is to choose a dynamic neural network architecture that, after completing the training procedure, with a given degree of accuracy, can reproduce the dynamics of the process that meets the desired transient response. As a basis, it can be taken a two-layer neural network in which the layers are arranged one after the other. Obviously, the created neural network should provide the ability to solve equation (15), and therefore its architecture is determined by the way of solving this equation, considering its kind, or other assumptions about it.

Equation (15) includes time derivatives (up to the *n*-th order inclusive), so there must be a mechanism for calculating the values of these derivatives based on the presented sets of input and output signals. For practical implementation of such a mechanism in the created neural network it is necessary to provide for the presence of several functional units, namely:

– the unit of derivatives reproduction (dynamic part), in which using the delay lines at the inputs of neurons of the first layer reproduces the structure of a given equation;

– the unit of implementation of the functional equations given by the equation, by which the network weights are calculated.

Let us consider the implementation of neural networks for partial cases of equation (15), and where it is possible, formulate not only the requirements for the architecture of the network but also ways of selecting the initial values of its coefficients, if the coefficients of the equation are known in advance.

4. Problem Solving

For the case where the input and output quantities are one-dimensional, the differential equation (15) can be represented as:

$$\begin{aligned} &a_{n_y}\frac{d^{n_y}y}{dt^{n_y}}+a_{n_y-1}\frac{d^{n_y-1}y}{dt^{n_y-1}}+\dots+a_1\frac{dy}{dt}+a_0y= \\ &=b_{n_x}\frac{d^{n_x}x}{dt^{n_x}}+b_{n_x-1}\frac{d^{n_x-1}x}{dt^{n_x-1}}+\dots+b_1\frac{dx}{dt}+b_0x. \end{aligned} \qquad (16)$$

Since the values of input and output variables are known only at certain intervals Δt, derivatives can be calculated only by approximate formulas. For example, for some *q*-th sample of variables *x* and *y* there are expressions as following:

$$\frac{dx_q}{dt}=\frac{x_q-x_{q-1}}{\Delta t}, \qquad (17)$$

. . . .

$$\begin{aligned} \frac{d^n x_q}{dt^n}&=\frac{x_q-C_n^1x_{q-1}+C_n^2x_{q-2}+\dots+C_n^m(-1)^m x_{q-m}+}{(\Delta t)^n} \\ &+\frac{\dots+C_n^{n-1}(-1)^{n-1}x_{q-n+1}+(-1)^n x_{q-n}}{(\Delta t)^n}, \end{aligned} \qquad (18)$$

$$\frac{dy_q}{dt}=\frac{y_q-y_{q-1}}{\Delta t}, \qquad (19)$$

. . . .

$$\begin{aligned} \frac{d^n y_q}{dt^n}&=\frac{y_q-C_n^1y_{q-1}+C_n^2y_{q-2}+\dots+C_n^m(-1)^m y_{q-m}+\dots}{(\Delta t)^n} \\ &+\frac{\dots+C_n^{n-1}(-1)^{n-1}y_{q-n+1}+(-1)^n y_{q-n}}{(\Delta t)^n}, \end{aligned} \qquad (20)$$

$$C_n^m = \frac{n!}{m!(n-m)!}; \tag{21}$$

Substituting expressions (17)–(21) into equation (16), reducing such terms and dividing by a factor of a variable y_q, the equation is obtained:

$$\begin{aligned} & y_q + \alpha_1 y_{q-1} + \alpha_2 y_{q-2} + \ldots + +\alpha_{n_y} y_{q-n_y} = \\ & = \beta_0 x_q + \beta_1 x_{q-1} + \beta_2 x_{q-2} + \ldots + \beta_{n_x} x_{q-n_x}, \end{aligned} \tag{22}$$

By solving relatively to value y_q, it can be obtained:

$$\begin{aligned} & y_q = \beta_0 x_q + \beta_1 x_{q-1} + \beta_2 x_{q-2} + \ldots + \beta_{n_x} x_{q-n_x} - \\ & -(\alpha_1 y_{q-1} + \alpha_2 y_{q-2} + \ldots + +\alpha_{n_y} y_{q-n_y}). \end{aligned} \tag{23}$$

Equation (23) uniquely defines the architecture of a neural network on a single neuron with a linear activation function, which can be represented [17]–[19] as a recurrent digital filter (Fig. 1). From the foregoing considerations, it follows that in the simplest case, to reproduce a linear differential equation with constant coefficients, the neural network can be implemented as a single neuron, the input of which is the current value of the input value and at least, than n_x of its values at previous samples, as well as not less than n_y of its previous values of the output value. Here there is the order of the input variable equal to n_x and the order of the output variable equal to n_y. Consider the procedure for constructing a neural network for the case where the differential equation looks like:

$$\frac{d^2y}{dt^2} + a_{E_1}\frac{dy}{dt} + a_{E_0} y = b_E x. \tag{24}$$

There is created the model for equation (24) solving in the Simulink environment [20] (Fig. 2).
Let us reduce the equation (24) to a discrete form, using expressions (17)–(21) to approximate. To do this, by presenting the first and second derivatives of the original variable in the form:

$$\frac{dy_q}{dt} = \frac{y_q - y_{q-1}}{\Delta t}, \tag{25}$$

$$\frac{d^2y}{dt^2} = \frac{y_q - 2y_{q-1} + y_{q-2}}{\Delta t^2}, \tag{26}$$

and by substituting expressions (25), (26) in the equation (24), there is obtained the following expression:

$$\begin{aligned} & y_q - 2y_{q-1} + y_{q-2} + a_{E_1}\Delta t\, y_q - \\ & - a_{E_1}\Delta t\, y_{q-1} + a_{E_0}\Delta t^2 y_q = b_E \Delta t^2 x_q. \end{aligned} \tag{27}$$

Grouping in the expression (27) the additives at the arguments y_q, y_{q-1}, y_{q-2} and multiplying both parts of the obtained equation by $1\big/\left(a_{E_0}\Delta t^2 + a_{E_1}\Delta t + 1\right)$, there is obtained the following:

$$\begin{aligned} y_q = {} & \frac{b_E \Delta t^2}{a_{E_0}\Delta t^2 + a_{E_1}\Delta t + 1} x_q + \\ & + \frac{a_{E_1}\Delta t + 2}{a_{E_0}\Delta t^2 + a_{E_1}\Delta t + 1} y_{q-1} - \\ & - \frac{1}{a_{E_0}\Delta t^2 + a_{E_1}\Delta t + 1} y_{q-2}, \end{aligned} \tag{28}$$

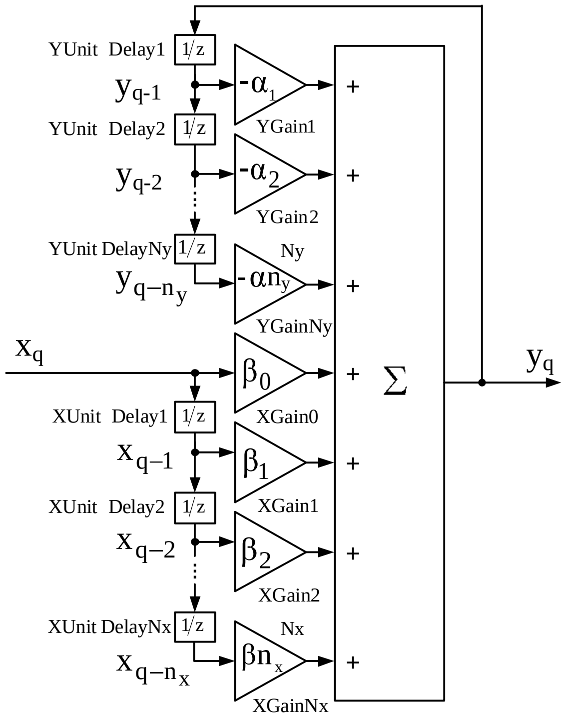


Fig. 1. Scheme for solving of the difference equation (9)

and using the following notation:

$$W_{11} = \frac{b_E \Delta t^2}{a_{E_0}\Delta t^2 + a_{E_1}\Delta t + 1}, \tag{29}$$

$$W_{12} = \frac{a_{E_1}\Delta t + 2}{a_{E_0}\Delta t^2 + a_{E_1}\Delta t + 1}, \tag{30}$$

$$W_{13} = -\frac{1}{a_{E_0}\Delta t^2 + a_{E_1}\Delta t + 1}, \tag{31}$$

It was obtained as follows:

$$y_q = W_{11} x_q + W_{12} y_{q-1} + W_{13} y_{q-2}, \tag{32}$$

which is the equation of a neural network with a linear activation function.

The equation (32) corresponds with a certain scheme of the neural network (Fig. 3). Thus, if the values of the coefficients of the differential equation are known, then the transition to the corresponding neural network can be carried out using relations (17)–(21). After carrying out the corresponding mathematical transformations, it will give approximate values of the coefficients that can be used in the construction of the digital filter. These values will be the initial coefficients for the network being created. The ability of the neural network to learn allows, based on the desired transient characteristic of the system to refine the values of these coefficients or to set the values of unknown coefficients after arbitrary selection of their initial values. This procedure is based on a training numerical sequence which is used as an expected signal.

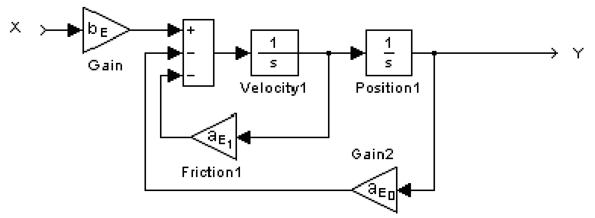


Fig. 2. Model for solving equation (10)

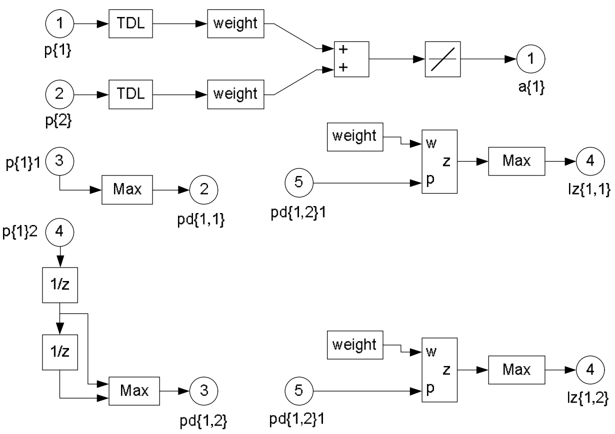


Fig. 3. A neural network scheme for solving the second-order equations

Solving the system of equations (29)–(31) with respect to unknown variables b_E, a_{E_0} and a_{E_1}, it is possible to obtain the relation as follow:

$$b_E = -\frac{W_{11}}{W_{13}T^2}, \tag{33}$$

$$a_{E_0} = \frac{W_{12} + W_{13} - 1}{W_{13}T^2}, \tag{34}$$

$$a_{E_1} = -\frac{W_{12} + 2W_{13}}{W_{13}T}. \tag{35}$$

Above mentioned variables provide the ability to calculate linear differential equation coefficients whose solution at the jumping action of the input variable corresponds to the desired transient response that was used as the expected signal when learning a dynamic neural network.

Thus, by defining the expected signal for training the neural network in the form of the desired transient characteristic of the system and selecting on the basis of information about the object the architecture of the corresponding linear dynamic neural network, it is possible to determine in the course of its learning the weighting factors. Using transformations (33)–(35) the coefficients of the linear differential equation are calculated, the solution of which reproduces the desired process dynamics in the system. The presence of information about the coefficients of the linear differential equation, which in our case can be used as a model of the reference, provides for the possibility of calculating the roots of the corresponding characteristic equation and thereby determining the position of these roots in the complex plane.

Let us consider the problem of synthesizing a control system using the method of positioning poles at given points of a complex plane for a second-order object. If taking into account the conditions of the given dynamics of the process in the system, the roots of its characteristic equation must take the value $-\lambda_1, -\lambda_2$, then the characteristic equation for such a system has the form:

$$F(S) = (S+\lambda_1)(S+\lambda_2) = S^2 + (\lambda_1+\lambda_2)S + \lambda_1\lambda_2 = 0; \tag{36}$$

By setting an object model with a transfer function in the form as follow:

$$W(S) = \frac{X(S)}{U(S)} = \frac{1}{S^2 + a_1 S + a_0}, \tag{37}$$

bringing it to the canonical form of control:

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -a_0 & -a_1 \end{bmatrix} \times \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} \times u, \tag{38}$$

and using the following notation:

$$\dot{x} = \begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix}; \ A = \begin{bmatrix} 0 & 1 \\ -a_0 & -a_1 \end{bmatrix}; \ x = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}; \ B = \begin{bmatrix} 0 \\ 1 \end{bmatrix}; \tag{39}$$

let us present the equation (37) as follows

$$\dot{x} = Ax + Bu; \tag{40}$$

4.1. Ackermann's Approach Application

Then, in our case, Ackermann's formula [16], [21] looks like:

$$K = [0\ 1] \times [B\ \ AB]^{-1} \times [A^2 + (\lambda_1+\lambda_2)A + \lambda_1\lambda_2 I]; \tag{41}$$

To calculate the coefficients K, it is necessary to make a series of transformations. Let us write the characteristic polynomial of the system in matrix form using the coefficients of the desired characteristic equation (36) as follows

$$\alpha_c(A) = A^2 + (\lambda_1+\lambda_2)A + \lambda_1\lambda_2 I = \\ = \begin{bmatrix} \lambda_1\lambda_2 - a_0 & \lambda_1+\lambda_2 - a_1 \\ [a_1(\lambda_1+\lambda_2)]a_0 & [a_1(\lambda_1+\lambda_2)]a_1 - a_0\lambda_1\lambda_2 \end{bmatrix}, \tag{42}$$

where I is a unitary matrix. Having defined the following matrix:

$$[B\ \ AB]^{-1} = \begin{bmatrix} a_1 & 1 \\ 1 & 0 \end{bmatrix}, \tag{43}$$

and substituting equations (42), (43) into the expression (41), there is obtained an expression for calculating the matrix of coefficients K:

$$K = [0 \;\; 1]\begin{bmatrix} a_1 & 1 \\ 1 & 0 \end{bmatrix}\times$$

$$\times\begin{bmatrix} \lambda_1\lambda_2 - a_0 & \lambda_1 + \lambda_2 - a_1 \\ [a_1(\lambda_1+\lambda_2)]a_0 & [a_1(\lambda_1+\lambda_2)]a_1 - a_0\lambda_1\lambda_2 \end{bmatrix} = \qquad (44)$$

$$= [\lambda_1\lambda_2 - a_0 \quad \lambda_1 + \lambda_2 - a_1].$$

Thus, the final expressions for calculating the coefficients K_1 and K_2 are as follows:

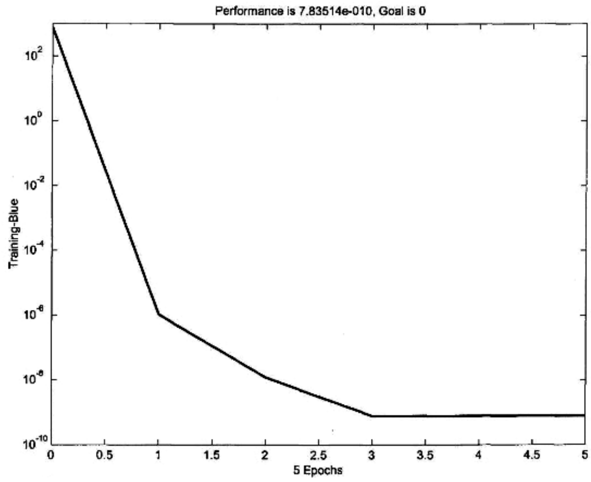
$$K_1 = \lambda_1\lambda_2 - a_0; \qquad (45)$$

$$K_2 = \lambda_1 + \lambda_2 - a_1. \qquad (46)$$

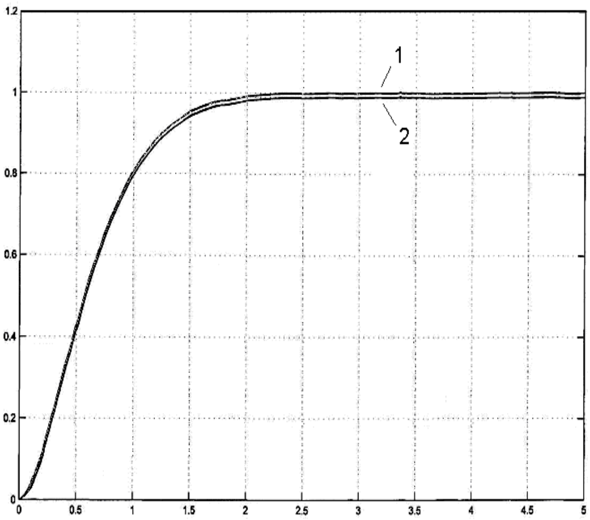
Summing the coefficients K_1 and K_2 with the corresponding values of the coefficients a_1 and a_0 of the characteristic equation of the object allows correcting the system in the direction of approximation of its transient characteristic to the transient characteristic of the reference. In practice, the procedure for determining the coefficient matrix K on the basis of the desired transition characteristic using a dynamic neural network is carried out in two stages. In the first stage, based on the features of the construction (structure) of the mathematical model of the object, the architecture of the corresponding dynamic neural network is created and trained. Here the objective target is a numerical sequence that corresponds to the desired transient characteristic of the system, and the input signal is given in the form of a numerical sequence, which reproduces the step function. In the second stage, using the values of the weighting coefficients obtained in the learning process of the neural network, the coefficients of the mathematical model of the reference and the roots of its characteristic equation are calculated, followed by the calculation of the coefficients of the matrix K.

4.2. Dynamic System Simulation

In such a way was created (Fig. 3) a block scheme of a dynamic neural network, whose weight coefficients were used to calculate the coefficients of a mathematical model of the reference with the desired transient characteristic at the training process completed. Network training was based on the identification of a dynamic object using the Levenberg-Marquardt algorithm [19]. The network was trained using the Neural Network packet offline. The input for training the network was given in the form of a numerical sequence that corresponded to the step function, and the expected output signal was specified by a sequence of numbers that reproduced the desired transient characteristic of the system. It was observed the dynamics of the learning process of the neural network in the form of dependence of the target function, given in the quadratic representation $E = 0.5\sum_{k=1}^{N}(y_k - t_k)^2$, versus the number of training cycles (Fig. 4, *a*).



a



b

Fig. 4. Dynamics of the neural network learning – (a), the output signals of the reference (1) and the adjusted object (2) – comparison – (b)

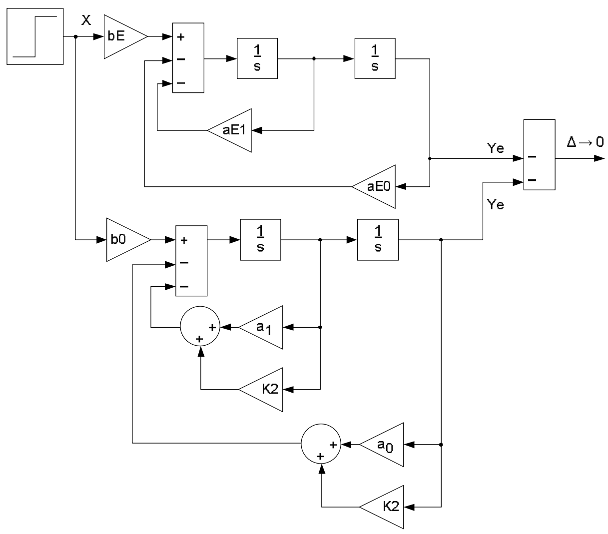


Fig. 5. Reference and adjustment models of the controlled object

On the basis of the values of the weight coefficients W_{11}, W_{12}, W_{13} obtained in the neural network training process, the coefficients b_E, a_{E_0}, b_{E_0} of the mathematical model of the reference were calculated using the equations (33)–(35). Using the mathematical model of the object with the given values of its coefficients, as well as the coefficients K_1 and K_2 were calculated based on the results of the neural network learning using Ackermann's formula. Its implementation into the structure of the object provided the identity of its transition characteristic with ones corresponded to the reference. It allows obtaining the mathematical models of the reference and the adjusted control object (Fig. 4, *b*) by the implementation of corrected coefficients K_1 and K_2 (Fig. 5).

The output signals of the reference and the adjusted object refer while acting on the inputs of the step signal.

5. Conclusions

On contrary to the known synthesis methods based on the use of frequency characteristics or root hodograph the proposed Ackermann's formula application guarantees the desired transient function obtain. The considered approach was realized according to the results of the neural network learning by using Ackermann's formula. The latter is based on the transformation of similarity, which translates a given model of an arbitrary structure into a canonical form of control. Such a procedure simplifies the determination of the desired elements of the matrix K. Comparison of the output signals of the reference and the adjusted objects simultaneously activated by step signal confirms our assumption about the expediency of the object model correction by the coefficients K_1 and K_2.

The implementation of the corrective coefficients which were obtained into the object dynamic model ensures that the processes in the models of the reference and the adjusted objects are almost completely identical. The value of the RMS error in the modeling of the system was equal to $1.87 \cdot 10^{-3}$. It indicates the high efficiency of the proposed algorithm of the object model improvement.

6. Acknowledgment

The authors are grateful to the staff and head of the Information and Measurement Technologies Department of Lviv Polytechnic National University for their support.

7. Conflict of interests

There is no conflict of interest while writing, preparing and publishing the article, as well as mutual claims by the co-authors.

References

[1] M. Shahinpoor, *A Robot Engineering Textbook*. New Mexico: University of New Mexico Press, Albuquerque, 1984.

[2] K. S. Fu, R. C. Gonzalez, C. S. G. Lee, *Robotics: Control, Sensing, Vision and Intelligence*. McGraw-Hill Education (India) Pvt Limited, 1988.

[3] G. Galkin, *Matlab & Simulink. Designing of mechatronic systems on a PC*. St. Petersburg: Korona-Vek, 2008 (in Russian).

[4] G. C. Goodwin, S. F. Graebe, Mario E. Salgado, *Control System Design*. Upper Saddle River, New Jersey: Prentice Hall, 2001.

[5] M. H. Popovich, O. V. Kovalchuk, *The theory of automatic control*. *Textbook*. Kyiv: Lybid, 2007 (in Ukrainian).

[6] R. C. Dorf, R. H. Bishop, *Modern Control Systems*, 13th Edition, Pearson Education, 2017.

[7] S. Haykin, *Neural Network. A Comprehensive Foundation*. New York: Macmillan College Publishing Company, 1994.

[8] C. L. Phillips, R. D. Harbor. *Feedback Control Systems*. Upper Saddle River, New Jersey: Prentice Hall, 2000.

[9] S. Osowski, *Sieci neuronowe do przetwarzania informacji*. Warszawa: Oficyna Wydawnicza Politechniki Warszawskiej, 2000.

[10] M. A. Novotarskiy, B. B. Nesterenko, "Artificial neural networks: calculations". *Proceedings of the Institute of Mathematics of NAS of Ukraine*, vol, 50, Kyiv: Naukova Dumka, 2004 (in Ukrainian).

[11] E. V. Bodianskyi, *Artificial neural networks: architectures, training, applications*. E. V. Bodianskyi and O. G. Rudenko, Eds. Kharkiv: Telemekh, 2004 (in Ukrainian).

[12] D. Rutkowska, M. Pilinski, L. Rutkowski, *Sieci neuronowe, algorytmy genetyczne i systemy rozmyte*. Warszawa - Lodz: Wydawnictwo Naukowe PWN, 2004.

[13] O. G. Rudenko, *Artificial neural networks: Textbook*. O. G. Rudenko and E. V. Bodianskyi, Eds. Kharkiv: SMIT Company LLC, 2006 (in Ukrainian).

[14] H. Al-Duwaish, S. Z. Rizvi, "*Design of a Neuro-Controller for Multivariable Nonlinear Time-Varying Systems*" in *WSEAS Transactions on Systems and Control*, vol. 5, Issue 9, pp. 711–720, 2010.

[15] Y. M. Rashkevich, R. O. Tkachenko, I. H. Tsmots, D. D. Peleshko, *Neural-like methods, algorithms, and structures for real-time signal and image processing*. Lviv: Lviv Polytechnic Publishing House, 2014 (in Ukrainian).

[16] J. E. Ackermann, Der Entwurf Linear Regelungs Systems in Zustandstraum, Regelungstech Process, *Datenverarb*, 7, 1972.

[17] M. Dorozhovets, O. Ivakhiv, V. Mokritskyi, *Uniform Converters of Information Support for Mechatronic Systems*. Lviv: Publishing House of Lviv Polytechnic National University, 2009 (in Ukrainian).

[18] J. Su, M. Nakonechnyi, O. Ivakhiv, A. Sachenko, "Developing the Automatic Control System Based on Neural Controller", *Information Technology and Control*, vol. 44, no. 3, pp. 262–270, 2015.

[19] M. Nakonechnyi, O. Ivakhiv, Y. Nakonechnyi, *Neural Network Control Systems for Nonlinear Objects: Monograph*. Lviv: Raster-7, Publishing House, 2017 (in Ukrainian).

[20] J. B. Dabney, T. L. Harmen, *Mastering Simulink 4*. *Upper Saddle River*. New Jersey: Prentice Hall, 2001.

[21] O. Katsuhiko, *Modern Control Engineering, 5th edition, Published by Pearson, 2010.*