

Bellman Function Search by Symbolic Regression

Askhat Diveev¹ and Elena Sofronova²

Abstract—The solution of the optimal control synthesis problem based on the Bellman equation is considered. The Bellman optimal control problem allows us to obtain the control function as a function of the state space coordinates. The essential difficulty of this problem is that the function is unknown. To solve the problem, one of the symbolic regression methods, the network operator method, is used to search for the Bellman function. This approach enables an automatic search for the structure and parameters of the mathematical expression using a special genetic algorithm. The result of the solution is a mathematical expression for the Bellman function. The optimal control found includes the gradient of the Bellman function. This function is described by code in the form of a network operator matrix. Therefore, the gradient of Bellman function is calculated numerically. An example of solving the control synthesis problem via Bellman equation for a mobile robot subjected to disturbance of initial conditions is presented.

I. INTRODUCTION

The control synthesis problem is one of the important and complex problems of control theory. The control synthesis problem consists in finding control as a function of the state space coordinates. This control ensures effective control of the object under object disturbances. In most cases, the control synthesis problem is solved manually by the control system developer.

Since the early 2000s, approximate numerical methods for solving the synthesis problem have appeared based on the use of symbolic regression methods [1]. Symbolic regression methods allow for the finding of mathematical expressions of control functions based on the optimization criterion. Special evolutionary genetic algorithms are used to search for mathematical expressions [2]. One of the disadvantages of this approach is that it is impossible to determine how far the solutions obtained are from the optimal one. This article examines solutions to the control synthesis problem using the Bellman equation [3]. The control function is obtained from the Bellman equation. The optimality of the solution obtained is determined by the Bellman function. The authors suggest that, for some problems, finding the Bellman function is simpler than directly solving the control synthesis problem.

In the paper [4], eight mathematical problems are formulated in which it is necessary to find a mathematical expression for a function. One of these problems is the problem of finding the Bellman function. Nowadays, the Bellman

function is often used as a reward function for various reinforcement learning tasks, including control problems [5]. In classical control theory, the Bellman function was given in advance in the form of a mathematical expression that was defined exactly for the values of the parameters [6]. In general, the Bellman function is guessed. There are no mathematical approaches to its analytical derivation. If the Bellman function is correctly defined, it allows to obtain an adaptive control system with low sensitivity to uncertainties [7], [8].

In other works, the optimal control has been determined numerically by dynamic programming and sometimes in the control process [9]. Sometimes the Bellman function is used to theoretically justify special types of control, for example impulse control for systems described by differential inclusions [10].

Machine Learning Control (MLC) is a promising paradigm in the field of control. This term has recently appeared [11], [12], although the methods associated with MLC have been used for about twenty years. An important difference between MLC and simple Machine Learning (ML) [13] is that artificial neural networks do not play a dominant role in MLC. Throughout the book [11], artificial neural networks are not mentioned as a means of machine learning control. The main role in machine learning control is assigned to symbolic regression methods, in particular the method of genetic programming [2]. In fact, there are nearly twenty symbolic regression methods that for some tasks perform better than GP.

Symbolic regression methods are universal function approximators like artificial neural networks. The main difference is that they can find mathematical expressions for different functions. The obtained mathematical expressions can be further used for deeper study of solved problems, in particular for finding regularities. Thus, the methods of symbolic regression open possibilities to solve a large number of known problems where it is necessary to find mathematical expressions of various functions. In this paper we consider the solution of the problem of finding the Bellman function using the method of symbolic regression.

Modern control systems are programs implemented on the on-board computer of the control object. The description of transformations of input signals performed by the control system in universal form is a mathematical expression. Unlike control systems created forty or more years ago, the implementation of linear or non-linear transformations is not significantly different today. We can write x or $\sin(x)$ for the on-board processor – this is not significantly different, although previously the implementation of the sine function

¹Askhat Diveev is Chief researcher of Federal Research Center "Computer Science and Control" of the Russian Academy of Sciences, Moscow, 119333, Vavilov Str., 44/2 FRC CS RAS, Russia aidiveev@mail.ru

²Elena Sofronova is senior research of Federal Research Center "Computer Science and Control" of the Russian Academy of Sciences, Moscow, 119333, Vavilov Str., 44/2 FRC CS RAS, Russia sofronova_ea@mail.ru

was somewhat more complex. This means that today it is necessary to use methods that allow you to find mathematical expressions to create a control system. These are symbolic regression methods that find not only the parameters but also the structure of a mathematical expression. Previously, to find a mathematical expression, for example a description of the functional dependence of experimental data, a researcher would write a mathematical expression with accuracy to the values of the parameters. The parameters were found by a computational method, often the method of least squares. Note that an artificial neural network is a multidimensional function with a given structure and a large number of unknown parameters. Finding their values is called machine learning.

The ability of symbolic regression methods to search for the structure of a mathematical expression comes from the characteristics of a genetic algorithm. This optimisation algorithm does not use arithmetic operations, summation and multiplication on numbers when searching for an optimal solution. It is designed to search in the space of codes. Crossover and mutation operations don't use arithmetic operations. All symbolic regression methods use special genetic algorithms to search for the optimal solution. Each method differs from others by the form of coding and the crossover operation of the genetic algorithm for these codes. In recent years we have successfully applied symbolic regression methods for control synthesis of various control objects [14].

II. PROBLEM STATEMENT

Consider the optimal control synthesis problem according to R. Bellman statement [3].

The mathematical model of the control object is

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{u}), \quad (1)$$

where \mathbf{x} is a state space vector, $\mathbf{x} \in \mathbb{R}^n$, \mathbf{u} is a control vector, $\mathbf{u} \in U \subseteq \mathbb{R}^m$, U is a compact set, $m \leq n$.

Initial conditions are

$$X_0 = \{\mathbf{x}^{0,1}, \dots, \mathbf{x}^{0,K}\} \subseteq \mathbb{R}^n. \quad (2)$$

The terminal condition is

$$\mathbf{x}(t_f) = \mathbf{x}^f \in \mathbb{R}^n, \quad (3)$$

where t_f is a time of reaching the terminal condition, t is not given, but is limited, $t \leq t^+$.

The quality criterion is

$$J = \int_0^{t_f} f_0(\mathbf{x}, \mathbf{u}) dt \rightarrow \min_{\mathbf{u} \in U}. \quad (4)$$

The control is searched in the form

$$\mathbf{u} = \mathbf{h}(\mathbf{x}). \quad (5)$$

Bellman equation for the problem (1)–(4) is

$$-\frac{d\mu(\mathbf{x})}{dt} = \min_{\mathbf{u} \in U} \left\{ \left(\frac{\partial \mu(\mathbf{x})}{\partial \mathbf{x}} \right)^T \mathbf{f}(\mathbf{x}, \mathbf{u}) + f_0(\mathbf{x}, \mathbf{u}) \right\}. \quad (6)$$

If the Bellman function $\mu(\mathbf{x})$ is known, then the control function is found from Bellman equation (5)

$$\mathbf{u} = \operatorname{argmin} \left\{ \left(\frac{\partial \mu(\mathbf{x})}{\partial \mathbf{x}} \right)^T \mathbf{f}(\mathbf{x}, \mathbf{u}) + f_0(\mathbf{x}, \mathbf{u}) \right\}. \quad (7)$$

Let us include initial (2) and terminal (3) conditions in the quality criterion (4)

$$J = \sum_{j=1}^K \left(\int_0^{t_{f,j}} f_0(\mathbf{x}(t), \mathbf{x}^{0,j}, \mathbf{u}) dt + p_1 \sqrt{\sum_{i=1}^n (x_i^f - x_i(t_{f,j}, \mathbf{x}^{0,j}))^2} \right) \rightarrow \min_{\mu(\mathbf{x})}. \quad (8)$$

where p_1 is a weight coefficient, $\mathbf{x}(t, \mathbf{x}^{0,K})$ is a partial solution of ODE (1) with control (7) from initial condition $\mathbf{x}^{0,j}$, $t_{f,j}$ is a time of reaching the terminal condition (3) from initial condition j , $j = 1, \dots, K$.

To approximate Bellman equation and solve (1)–(7) a network operator method, as one of symbolic regression methods, is applied.

III. THE NETWORK OPERATOR METHOD

The network operator method was developed for the control synthesis problem [1]. The method encodes a mathematical expression in a form of a directed graph, which is represented in computer memory as an integer matrix. The network operator method encodes a mathematical expression as a composition of elementary functions with one or two arguments. Functions with two arguments should be commutative, associative and have a unit element. Parameters and variables are included as arguments in a mathematical expression. A set of elementary functions with one argument includes the identity function, which does not change the value of the argument. The computational graph of the network operator includes source nodes associated with arguments of the mathematical expression, edges associated with functions with one argument, and other nodes associated with functions with two arguments.

To encode a mathematical expression in the form of a network operator graph, it is necessary to represent a mathematical expression as a composition of alternating functions with one and two arguments. This is done using the identity function and functions with two arguments with the unit element as the second argument.

Let us consider an example of encoding a mathematical expression using the network operator method. Suppose a mathematical expression has the following form

$$y = -x_2 \sin(\cos(q_1 x_1 + q_2)). \quad (9)$$

where x_1, x_2 are variables, q_1, q_2 are parameters.

For coding the mathematical expression (9) it is enough to use the following sets.

The set of arguments

$$F_0 = \{f_{0,1} = x_1, f_{0,2} = x_2, f_{0,3} = q_1, f_{0,4} = q_2, \} \quad (10)$$

The set of functions of one argument

$$F_1 = \{f_{1,1}(z) = z, f_{1,2}(z) = -z, f_{1,3}(z) = \sin(z), f_{1,4}(z) = \cos(z)\}. \quad (11)$$

The set of functions of two arguments

$$F_2 = \{f_{2,1}(z_1, z_2) = z_1 + z_2, f_{2,2}(z_1, z_2) = z_1 z_2\}. \quad (12)$$

The first index of elements in the sets (10)–(12) points to a number of arguments, the second index is the element number.

Let us write the mathematical expression (9) using elements of the sets (10)–(12).

$$y = f_{2,2}(f_{1,2}(f_{0,2}), f_{1,3}(f_{1,4}(f_{2,1}(f_{2,2}(f_{0,3}, f_{0,1}), f_{0,4}))))). \quad (13)$$

The expression (13) includes coding rule violations. For example $\sin(\cos(\dots))$, $f_{2,2}(f_{0,1}, f_{0,2})$ and others. Correct the code and receive the following mathematical expression

$$y = f_{2,2}(f_{1,2}(f_{0,2}), f_{1,3}(f_{2,1}(e_1, \cos(f_{2,1}(f_{1,1}(f_{2,2}(f_{1,1}(f_{0,3}), f_{1,1}(f_{0,1}))), f_{1,1}(f_{0,4})))))); \quad (14)$$

where e_1 is a unit element for function $f_{2,1}(z_1, z_2)$. It is zero, $e_1 = 0$.

The expression (14) has four elements from the set of arguments (10) and four functions with two arguments. Therefore the network operator graph has eight nodes. Graph of the network operator for mathematical expression (9) is presented in the Fig. 1.

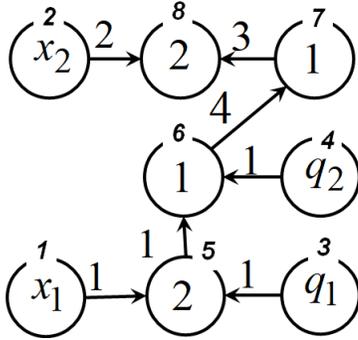


Fig. 1. Graph of the network operator for the mathematical expression

In the Fig. 1 the function numbers with two arguments are located in nodes, in source-nodes arguments of mathematical expression are located. The function numbers with one argument are located near arcs. In upper part of nodes the node numbers are pointed.

The adjacency matrix for the graph has the following form

$$\mathbf{A} = \begin{bmatrix} 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad (15)$$

To build the network operator matrix it is needed to change ones in adjacency matrix on the function numbers with one argument and to put the function numbers with two arguments on the main diagonal in the lines, that associated with corresponding nodes.

$$\Psi = \begin{bmatrix} 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 2 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 4 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 3 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 \end{bmatrix} \quad (16)$$

The network operator matrix for the mathematical expression (9) has the following form

The network operator matrix has upper triangle form when all nodes are enumerated such, that for any pair of connected by one arc nodes, the node number with exiting arc is smaller than the node number with entering arc.

To decode of network operator matrix it is needed to set a vector of node.

$$\mathbf{z} = [z_1 \dots z_L]^T, \quad (17)$$

where L is a number of lines in the network operator matrix. Each element of node vector is initialized. If the component number corresponds source-node, then a value of argument associated with this node is put in this component of node vector. If the component number points to node associated with function with two arguments, then the unit element for this function with two arguments is put in this component of node vector. Next, the algorithm passes each row of the matrix from top to bottom. If nonzero non-diagonal element is found, $\psi_{i,j} \neq 0$, $i < j$, then the calculation is performed by the formula

$$z_j \leftarrow f_{2,\psi_{j,j}}(z_j, f_{1,\psi_{i,j}}(z_i)). \quad (18)$$

The result of the mathematical expression calculation is obtained after passing through all the rows of the network operator matrix. Consider decoding of the network operator matrix (16).

Firstly, the vector of nodes is initialized

$$\mathbf{z} = [x_1 \ x_2 \ q_1 \ q_2 \ 1 \ 0 \ 0 \ 1]^T.$$

In the first row we find $\psi_{1,5} = 1$. Then $z_5 = 1 \cdot f_{1,1}(z_1) = x_1$,

$$\mathbf{z} = [x_1 \ x_2 \ q_1 \ q_2 \ x_1 \ 0 \ 0 \ 1]^T.$$

In the second row $\psi_{2,8} = 2$, then $z_8 = 1 \cdot f_{1,2}(z_2) = -x_2$,

$$\mathbf{z} = [x_1 \ x_2 \ q_1 \ q_2 \ x_1 \ 0 \ 0 \ -x_2]^T.$$

Next $\psi_{3,5} = 1$, then $z_5 = x_1 \cdot z_3 = x_1 q_1$,

$$\mathbf{z} = [x_1 \ x_2 \ q_1 \ q_2 \ x_1 q_1 \ 0 \ 0 \ -x_2]^T.$$

In the fourth row $\psi_{4,6} = 1$, then $z_6 = 0 + f_{1,1}(z_4) = q_2$,

$$\mathbf{z} = [x_1 \ x_2 \ q_1 \ q_2 \ x_1 q_1 \ q_2 \ 0 \ -x_2]^T.$$

In the fifth row $\psi_{5,6} = 1$, then $z_6 = q_2 + f_{1,1}(z_5) = x_1 q_1 + q_2$,

$$\mathbf{z} = [x_1 \ x_2 \ q_1 \ q_2 \ x_1 q_1 \ x_1 q_1 + q_2 \ 0 \ -x_2]^T.$$

In the sixth row $\psi_{6,7} = 4$, therefore $z_7 = 0 + f_{1,4}(z_6) = \cos(q - 1x_1 + q - 2)$,

$$\mathbf{z} = [x_1 \ x_2 \ q_1 \ q_2 \ x_1 q_1 \ x_1 q_1 + q_2 \ \cos = (q - 1x_1 + q_2) \ -x_2]^T.$$

And finally in the seventh row $\psi_{7,8} = 3$. We obtain the result of calculation of all mathematical expression

$$z_8 = -x_2 \cdot f_{1,3}(z_7) = -x_2 \sin(\cos(q_1 x_1 + q_2)).$$

The network operator method uses the principle of small variation of basic solution [15]. According to this principle only one possible solution is coded in the form of the network operator matrix. Other possible solution are coded as ordered multi-sets of vectors of small variations. Each small variation can change only one element in the network operator matrix. For record of small variation an integer vector with four component is used. The application of this principle is aimed at finding such small variations of the symbolic regression code that slightly change the mathematical expression according to the quality criterion used in the problem. This principle allows you to preserve the inheritance property when performing a crossover operation in symbolic regression. This provides an efficient search for an optimal solution by an evolutionary algorithm [16].

$$\mathbf{w} = [w_1 \ w_2 \ w_3 \ w_4]^T, \quad (19)$$

where w_1 is a type of small variation, $w_1 = 0$ is a replacing a non-diagonal element, $w_1 = 1$ is a replacing a diagonal element, $w_1 = 2$ is a changing the zero value of the non-diagonal element to non-zero one or adding an arc, $w_1 = 3$ is a changing non-diagonal element non-zero value to zero one or removing an arc, this variation is made only if non-zero non-diagonal elements remain in the row and in the column where changing element stood, w_2 is a number of line, w_3 is a number of column, w_4 is a new value of an element.

Consider an example of a small variation. Let variation vector has the following value

$$\mathbf{w} = [2 \ 2 \ 6 \ 4]^T.$$

After small variation, the following network operator matrix is obtained

$$\mathbf{w} \circ \Psi = \begin{bmatrix} 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 4 & 0 & 2 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 2 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 4 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 3 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 \end{bmatrix}$$

The new network operator corresponds to the following mathematical expression

$$y = -x_2 \sin(\cos(q_1 x_1 + x_2 + q_2)).$$

Any non-basic possible solution in the network operator method is coded as ordered multi-set of small variation vectors

$$W_\alpha = (\mathbf{w}^{\alpha,1}, \dots, \mathbf{w}^{\alpha,d}), \quad (20)$$

where d is a depth of the set of variations.

For crossover operations two possible solutions are selected

$$\begin{aligned} W_\alpha &= (\mathbf{w}^{\alpha,1}, \dots, \mathbf{w}^{\alpha,d}), \\ W_\beta &= (\mathbf{w}^{\beta,1}, \dots, \mathbf{w}^{\beta,d}), \end{aligned} \quad (21)$$

The crossover point is determined randomly $c \in \{1, \dots, d\}$ and two new possible solutions are obtained after exchanging small variation vectors after the crossover point of selected possible solutions

$$\begin{aligned} W_\gamma &= (\mathbf{w}^{\alpha,1}, \dots, \mathbf{w}^{\alpha,c-1}, \mathbf{w}^{\beta,c}, \dots, \mathbf{w}^{\beta,d}) \\ W_\sigma &= (\mathbf{w}^{\beta,1}, \dots, \mathbf{w}^{\beta,c-1}, \mathbf{w}^{\alpha,c}, \dots, \mathbf{w}^{\alpha,d}), \end{aligned} \quad (22)$$

IV. A COMPUTATIONAL EXPERIMENT

Consider an optimal control synthesis problem of mobile robot.

A mathematical model of control object is given

$$\begin{aligned} \dot{x}_1 &= 0.5(u_1 + u_2) \cos(x_3), \\ \dot{x}_2 &= 0.5(u_1 + u_2) \sin(x_3), \\ \dot{x}_3 &= 0.5(u_1 - u_2), \end{aligned} \quad (23)$$

where $\mathbf{x} = [x_1 \ x_2 \ x_3]^T$ is a state space vector, $\mathbf{u} = [u_1 \ u_2]^T$ is a control vector.

Constraints on control are

$$-10 = u^- \leq u_i \leq u^+ = 10, \quad i = 1, 2. \quad (24)$$

Eight initial conditions are

$$\begin{aligned} X_0 &= \{ \mathbf{x}^{0,1} = [-2 \ -2.5 \ -5\pi/12]^T, \\ &\mathbf{x}^{0,2} = [-2 \ -2.5 \ 5\pi/12]^T, \\ &\mathbf{x}^{0,3} = [-2 \ 2.5 \ -5\pi/12]^T, \\ &\mathbf{x}^{0,4} = [-2 \ 2.5 \ 5\pi/12]^T, \\ &\mathbf{x}^{0,5} = [2 \ -2.5 \ -5\pi/12]^T, \\ &\mathbf{x}^{0,6} = [2 \ -2.5 \ 5\pi/12]^T, \\ &\mathbf{x}^{0,7} = [2 \ 2.5 \ -5\pi/12]^T, \\ &\mathbf{x}^{0,8} = [2 \ 2.5 \ 5\pi/12]^T. \end{aligned} \quad (25)$$

The terminal condition is

$$\mathbf{x}^f = [0 \ 0 \ 0]^T. \quad (26)$$

The quality criterion is

$$J = \int_0^{t_f} (1 + p_2(u_1^2 + u_2^2)) dt \rightarrow \min_{\mathbf{u}}, \quad (27)$$

where $p_2 = 0.1$.

The Bellman equation is

$$\begin{aligned} -\frac{d\mu(\mathbf{x})}{dt} &= \min_{\mathbf{u} \in U} \left\{ \frac{\partial \mu(\mathbf{x})}{\partial x_1} 0.5(u_1 + u_2) \cos(x_3) + \right. \\ &\frac{\partial \mu(\mathbf{x})}{\partial x_2} 0.5(u_1 + u_2) \sin(x_3) + \\ &\left. \frac{\partial \mu(\mathbf{x})}{\partial x_3} 0.5(u_1 - u_2) + 1 + p_2(u_1^2 + u_2^2) \right\}. \end{aligned} \quad (28)$$

From Bellman equation the optimal control is found

$$u_i = \begin{cases} u^-, & \text{if } \tilde{u}_i < u^- \\ u^+, & \text{if } \tilde{u}_i > u^+ \\ \tilde{u}_i, & \text{otherwise} \end{cases}, i = 1, 2, \quad (29)$$

where

$$\tilde{u}_1 = -\cos(x_3) \frac{\partial \mu(\mathbf{x})}{4p_2 \partial x_1} - \sin(x_3) \frac{\partial \mu(\mathbf{x})}{4p_2 \partial x_2} - \frac{\partial \mu(\mathbf{x})}{4p_2 \partial x_3}, \quad (30)$$

$$\tilde{u}_2 = -\cos(x_3) \frac{\partial \mu(\mathbf{x})}{4p_2 \partial x_1} - \sin(x_3) \frac{\partial \mu(\mathbf{x})}{4p_2 \partial x_2} + \frac{\partial \mu(\mathbf{x})}{4p_2 \partial x_3}. \quad (31)$$

To find Bellman function $\mu(\mathbf{x})$ the network operator method was applied.

During the search the network operator method used the following quality criterion

$$J = \sum_{j=1}^8 \left(\int_0^{t_{f,j}} (1 + p_2(u_1^2 + u_2^2)) dt + p_1 \sqrt{\sum_{i=1}^3 (x_i^f - x_i(t_{f,j}, \mathbf{x}^{0,j}))^2} \right) \rightarrow \min_{\mu(\mathbf{x})}, \quad (32)$$

where $p_1 = 1, p_2 = 0.1$.

The basic solution was chosen as

$$\mu^0(\mathbf{x}) = q_1(x_1^f - x_1) + q_2(x_2^f - x_2) + q_3(x_3^f - x_3), \quad (33)$$

where $q_i = 1, i = 1, 2, 3$.

Finally, after the search process the following Bellman function was obtained

$$\begin{aligned} \mu(\mathbf{x}) = & 2q_3(x_3^f - x_3) + q_2(x_2^f - x_2) + \\ & q_1(x_1^f - x_1) + \text{sgn}(q_2(x_2^f - x_2)) \times \\ & (\exp(|q_2(x_2^f - x_2)|) - 1) + \sin(q_1) + \\ & \text{sgn}(2q_3(x_3^f - x_3) + q_2(x_2^f - x_2) + \\ & q_1(x_1^f - x_1) + \text{sgn}(q_2(x_2^f - x_2)) \times \\ & (\exp(|q_2(x_2^f - x_2)|) - 1) + \\ & \sin(q_1)) + \cos(q_3(x_3^f - x_3) + \\ & q_2(x_2^f - x_2)), \end{aligned} \quad (34)$$

where $q_1 = 6.569, q_2 = 7.251, q_3 = 8.314$.

Results of simulation the system (23) with control (30), (31) and Bellman function (34) are presented in the Fig. 2–5.

In the Fig. 2 the optimal trajectories of mobile robot movement on the plane $\{x_1, x_2\}$ from eight initial conditions are presented (25). As we can see the mobile robot achieves the terminal condition with sufficient accuracy from all eight initial states. The optimization criterion did not include minimizing the path length, so the paths of movement of the control object from different initial states to the terminal state is not the shortest.

In the Figures 3–5 the plots of variables of time for initial state $\mathbf{x}^{0,8} = [2 \ 2.5 \ 5\pi/12]^T$ are presented. In the Figures 6 and 7 plots of control for the initial state $\mathbf{x}^{0,8} = [2 \ 2.5 \ 5\pi/12]^T$ are presented.

Parameters of computational experiment: size of network operator – 24×24 , number of functions with one argument

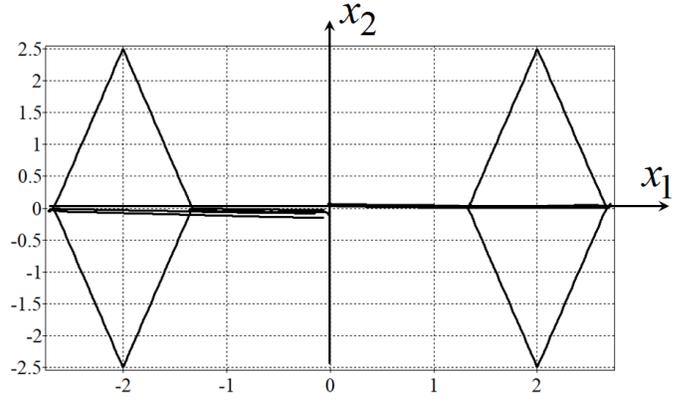


Fig. 2. Optimal trajectories on the plane $\{x_1, x_2\}$ from eight initial states

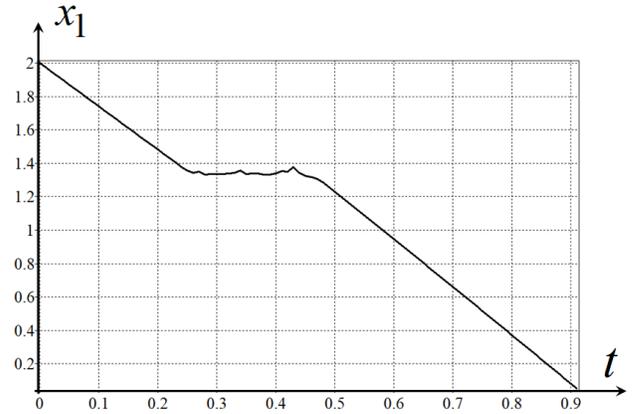


Fig. 3. Plot of the variable $x_1(t)$ for initial state $\mathbf{x}^{0,8}$

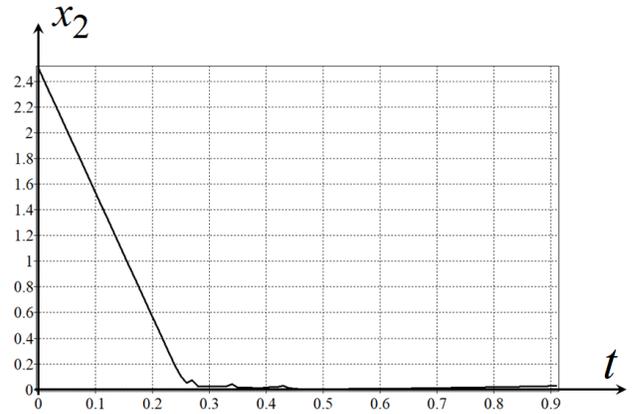


Fig. 4. Plot of the variable $x_2(t)$ for initial state $\mathbf{x}^{0,8}$

– 20, number of functions with two arguments – 2, number of parameters – 3, size of initial population – 512, number of generations – 128, number of possible crossovers at each generation – 128, depth of variation – 8, number of generations between change of basic solution – 20.

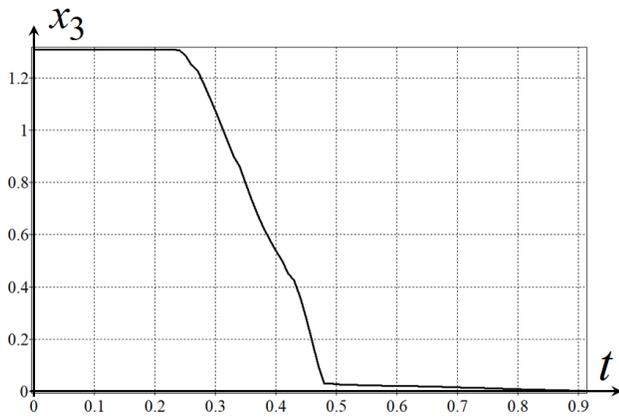


Fig. 5. Plot of the variable $x_3(t)$ for initial state $\mathbf{x}^{0,8}$

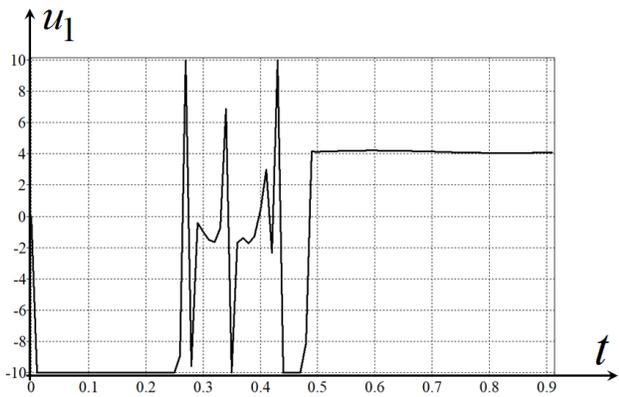


Fig. 6. Control u_1

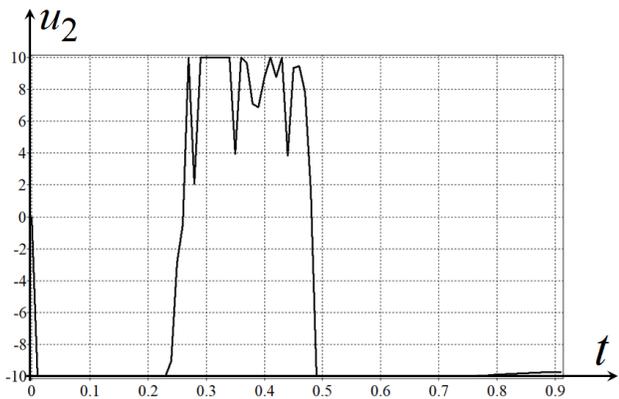


Fig. 7. Control u_2

V. CONCLUSIONS

The paper considers the problem of synthesizing optimal control in the formulation of R. Bellman. The solution to this problem is the control function, which is obtained from the Bellman equation. To obtain the Bellman equation, it is necessary to define a scalar Bellman function that must satisfy given boundary conditions. The work uses symbolic

regression to determine the Bellman function and construct the Bellman equation. The network operator method is used as the symbolic regression method. The paper provides an example of a solution to the problem of synthesizing an optimal stabilization system based on the construction of the Bellman equation by symbolic regression for a wheeled robot with a differential drive. The Bellman function found by symbolic regression made it possible to solve the problem of synthesizing the optimal stabilization system for the object under consideration, which was confirmed by the results of a computational experiment.

Further research is related to determining the dependence of the control function on the type of Bellman function, obtained for various quality criterion.

REFERENCES

- [1] A. Diveev and E. Sofronova, "Application of Network Operator Method for Synthesis Optimal Structure and Parameters of Automatic Control System," IFAC Proceedings Volumes, 2008, vol. 41, No 2, pp. 6106 — 6113.
- [2] J.R. Koza, *Genetic Programming: On the Programming of Computers by Means of Natural Selection*; MIT Press: Cambridge, MA, USA, 1992; 819p.
- [3] R.E. Bellman, I. Glicksberg, and O.A. Gross, Some Aspects of the Mathematical Theory of Control Process. Rand Corporation in Santa Monica, Calif. 1958, 263 p.
- [4] A. Diveev and E. Sofronova, "Machine search problem for mathematical expression," Doklady Mathematics, 2024, vol. 520, no. 2, pp. 22—32. (in npress)
- [5] J. Kubalik, J. Zegklitz, E. Derner, and R.Babuska, Symbolic Regression Methods for Reinforcement Learning. arXiv:1903.09688v1, March, 2019, pp. 1–12.
- [6] M.I. Zelikin and N.B. Melnikov "The Bellman function and optimal synthesis in control problems with non-smooth constraints," Journal of Mathematical Sciences, Vol. 121, No. 2, 2004, pp.2281–2294.
- [7] K. Okano and K. Hagino, "Adaptive control design approximating solution of Hamilton-Jacobi-Bellman equation for nonlinear strict-feedback system with uncertainties," 2008 SICE Annual Conference, Chofu, Japan, 2008, pp. 204–208.
- [8] J. Su, H. Cheng, H. Guo, R. Huang, and Z. Peng, "An Approximate Quadratic Programming for Efficient Bellman Equation Solution," IEEE Access, 7, 2019, p. 126077 – 126087.
- [9] S. Mehraeen and S. Jagannathan, "Decentralized nearly optimal control of a class of interconnected nonlinear discrete-time systems by using online Hamilton-Bellman-Jacobi formulation," The 2010 International Joint Conference on Neural Networks (IJCNN), 18–23 July 2010, Barcelona, Spain, pp. 1–8.
- [10] S.L. Fraga, and L.P. Fernando, "Hamilton-Jacobi-Bellman Equation and Feedback Synthesis for Impulsive Control," IEEE Transactions on automatic control, vol. 57, no. 1, January 2012, pp. 244–249.
- [11] T. Duriez, S.L. Brunton, and B.R. Noack, Machine Learning Control – Taming Nonlinear Dynamics and Turbulence. Springer International Publishing Switzerland, 2017, 211 p.
- [12] A. Bensoussan, Y. Li, D.P.C. Nguyen, M.-B. Tran, S.C.P. Yam, and X. Zhou, Machine Learning and Control Theory. arXiv:2006.05604 [cs.LG].
- [13] M.I. Jordan and T.M. Mitchell, Machine learning: Trends, perspectives, and prospects. Science, 349(6245):255 LP – 260, jul 2015.
- [14] S.M. Richards, J.-J. Slotine, N. Azizan, M. Pavone, "Learning Control-Oriented Dynamical Structure from Data," Proceedings of the 40th International Conference on Machine Learning, Honolulu, Hawaii, USA. PMLR 202, 2023.
- [15] A.I. Diveev, and E.A. Sofronova, "Universal Approach to Solution of Optimization Problems by Symbolic Regression," Applied Sciences, 2021, 11, 5081, pp. 1-19.
- [16] A. Diveev, E. Sofronova, N. Konyrbaev, O. Abdullayev, "Advanced Model with a Trajectory Tracking Stabilisation System and Feasible Solution of the Optimal Control Problem," Mathematics 2024, 12, 3193.