Combined Symbolic Regression Approach and its Application for Synthesized Optimal Control

Askhat Diveev[1]

Abstract— **The paper considers the machine mathematical expression search problem for a control function. To solve this problem machine learning of control by symbolic regression numerical method is used. In difference from other works, where searching mathematical expression is performed by symbolic regression, here not one and some symbolic regression methods are applied sequence for search of one a mathematical expression. The paper contains a description of new symbolic regression method, that is named universal code. The method is constructed on the base the genetic programming and Cartesian genetic programming and it uses the principle small variation of basic solution. The paper presents an example of application of the universal code together with the network operator for stabilization of wheeled robot in the point of the state space with given quality of stabilization, which is needed for small sensitivity of a control function to external disturbances.**

I. INTRODUCTION

Solving the control system synthesis problem is almost always necessary to perform in order to the control system had feedback control and was implemented in real object. As a rule, this problem is solved by manually. A researcher studies a mathematical model of control object, defines control channels and inserts there controllers for compensating control errors. In seldom cases, if the problem and the model of control object allow to make that, researcher uses analytical methods, the backstepping integrator [1] or the analytical design of aggregate regulators [2], [3].

With beginning of twenty-one century the numerical method of control synthesis by symbolic regression became to use [4]. This approach allowed to automate of control synthesis problems solving. Symbolic regression includes evolutionary algorithm of search and it can find a solution of control synthesis problem in the form of mathematical expression special code for desired control function. Studies of symbolic regression showed, that the genetic programming [5], used as a search tool of the optimal solution in symbolic regression, not always provides saving property of inheritance needed for successful work of evolutionary algorithm.

At the crossover operation in genetic programming very often two new possible solutions are received not like parents even if both parents are the same. This means, that crossover operation not saving required inheritance property and generate two new possible solutions randomly. To fix this situation the principle small variation of basic solution was created [6], that firstly was applied in new symbolic regression method the network operator. According this principle only one possible solution, which on opinion of researcher is nearest to the optimal solution is codded by symbolic regression. Other possible solutions are codded as sets of small variation vectors of the basic solution. All genetic operations, crossover and mutation are performed on sets of small variation vectors, at this crossover operation is performed as in classical genetic algorithm.

The basis solution undergoes changes, that depend on the number of small variations in the set of each possible solution and the small variations themselves. A number of small variation vectors in set of each of possible solution is named the depth of variation. After given number of cycles generations the basis solution is exchanged onto the best current found possible solution. This solution is the new basis solution and this process is named epoch change. The depth of variation and a number of generations between of the epoch change are parameters of search of the optimal solution. The principle of small variation of basic solution can be used in any symbolic regression. Then this symbolic regression receives in its named word "variation" as the first word. Application of the principle of small variation in control allowed to solve many control synthesis problems and automate process of construction control system practically, that is to exclude a human from that. This means, that a computer program of control system is written itself or is learned. Here human takes part in creating only basic solution, as a parent of his child. Further the child or a program is changed or is grown itself according to given quality criterion. Obviously, that such approach is base for creating of artificial intelligence.

In this work continue to study application of symbolic regression for solving the control synthesis problem. As any evolutionary algorithm symbolic regression finds well solution on quality criterion value, but it can't define as far this solution from the optimal one. Other feature of evolutionary computations consists of that if algorithm found some good solution, corresponding a local minimum, then other solutions of the population are concentrated in near of this local minimum. It is very hard to force the algorithm to look for optimal solution in new place far from the local optimization. The paper presents a new combine approach to solving the control synthesis problem. To solve the problem two symbolic regression method are used. Firstly, one symbolic regression method is used, and then after that it found some not bad solution, other method is used.

[1] 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

II. VARIATION UNIVERSAL CODE

The variation universal code is constructed on the base of genetic programming and Cartesian genetic programming [8]. It also uses the principle of small variation of basic solution. The code consists of codes of elementary functions. Each code of elementary function is an integer vector of two components. The first component is a number of function arguments. The second component is the function number. The code is a sequence of elementary functions. Parameters and arguments of a mathematical expression are a functions without arguments and is codded as an integer vector with the first component is equaled zero.

Every code of elementary function is an argument of function locating left. Calculating of the function code is performed with last code to first code from right to left. If the function is calculated, then its result is an argument next function to left, until there is a code to left. If there is not a code to left, then this is a value of codded mathematical expression.

Consider an example. Assume, the following alphabet of elementary functions is.

Functions without arguments or variables and parameters of mathematical expression:

$$\begin{aligned} \mathrm{F}_0 &= \{f_{0,1} = x_1 = [0\ 1]^T,\ f_{0,2} = x_2 = [0\ 2]^T, \\ &\quad f_{0,3} = q_1 = [0\ 1]^T,\ f_{0,4} = q_2 = [0\ 4]^T\}. \end{aligned} \tag{1}$$

Functions with one argument:

$$\begin{aligned} \mathrm{F}_1 = \{&f_{1,1}(z) = z \rightarrow [1\ 1]^T, \\ &f_{1,2}(z) = -z \rightarrow [1\ 2]^T, \\ &f_{1,3}(z) = \exp(z) \rightarrow [1\ 3]^T, \\ &f_{1,4}(z) = \sin(z) \rightarrow [1\ 4]^T\}. \end{aligned} \tag{2}$$

Functions with two arguments:

$$\begin{aligned} \mathrm{F}_2 = \{&f_{2,1}(z_1, z_2) = z_1 + z_2 \rightarrow [2\ 1]^T, \\ &f_{2,2}(z_1, z_2) = z_1 \cdot z_2 \rightarrow [2\ 2]^T. \end{aligned} \tag{3}$$

If it is necessary to code the following mathematical expression

$$y = \exp(-x_2)\sin(q_1x_1 + q_2), \tag{4}$$

then firstly we overwrite the mathematical expression (4) in the prefix form with account of the alphabet of elementary functions

$$y = f_{2,2}(f_{1,3}(f_{1,2}(x_2)), f_{1,4}(f_{2,1}(f_{2,2}(q_1, x_1), q_2))). \tag{5}$$

Then we exchange elementary functions of them codes.

$$\tilde{y} = \left(\begin{bmatrix}2\\2\end{bmatrix} \begin{bmatrix}1\\3\end{bmatrix} \begin{bmatrix}1\\2\end{bmatrix} \begin{bmatrix}0\\2\end{bmatrix} \begin{bmatrix}1\\4\end{bmatrix} \begin{bmatrix}2\\1\end{bmatrix} \begin{bmatrix}2\\2\end{bmatrix} \begin{bmatrix}0\\3\end{bmatrix} \begin{bmatrix}0\\1\end{bmatrix} \begin{bmatrix}0\\4\end{bmatrix}\right). \tag{6}$$

In common case a code of mathematical expression has N two components integer vectors

$$\tilde{y} = \left(\begin{bmatrix}a_1(1)\\a_2(1)\end{bmatrix} \begin{bmatrix}a_1(2)\\a_2(2)\end{bmatrix} \cdots \begin{bmatrix}a_1(k)\\a_2(k)\end{bmatrix} \cdots \begin{bmatrix}a_1(N)\\a_2(N)\end{bmatrix}.\right). \tag{7}$$

To define correct of code the index of code symbol is used. The index of code symbol shows, how many codes must stay to right of the current code. The index of code symbol is calculated by equation

$$T(k) = T(k-1) + a_1(k) - 1. \tag{8}$$

Correct code has the following values of the index code symbol

$$\begin{aligned} T(0) &= 1, \\ T(k) &\geq 0, k = 1, \ldots, N-1, \\ T(N) &= 0. \end{aligned} \tag{9}$$

If the conditions (9) is fulfill, then the code is correct.

Let's check the code (6).

$T(1) = 1 + 2 - 1 = 2$, $T(2) = 2 + 1 - 1 = 2$, $T(3) = 2 + 1 - 1 = 2$, $T(4) = 2 + 0 - 1 = 1$, $T(5) = 1 + 1 - 1 = 1$, $T(6) = 1 + 2 - 1 = 2$, $T(7) = 2 + 2 - 1 = 3$, $T(8) = 3 + 0 - 1 = 2$, $T(9) = 2 + 0 - 1 = 1$, $T(10) = 1 + 0 - 1 = 0$.

To calculate a mathematical expression by its code the following equation is used

$$\begin{gathered} j = 0,\ k = N, \ldots, 1 \\ \text{if } a_1(k) = 0, \text{ then} \\ j \leftarrow j + 1,\ r_j \leftarrow s_{0,a_2} \\ \text{else, if } a_1(k) = 1, \text{ then} \\ r_j \leftarrow f_{1,a_2(k)}(r_j), \\ \text{else, if } a_1(k) = 2, \text{ then } j \leftarrow j - 1, \\ r_j \leftarrow f_{2,a_2(k)}(r_j, r_{j+1}). \end{gathered} \tag{10}$$

Result of calculations is stored in the variable r_1 at $k = 1$.

Consider an example of the code (6)

$$\begin{gathered} j \leftarrow 0,\ k = 10, \ldots, 1, \\ a_1(10) = 0,\ j \leftarrow j + 1 = 0 + 1 = 1, \\ r_1 \leftarrow q_2, \\ a_1(9) = 0,\ j \leftarrow j + 1 = 1 + 1 = 2, \\ r_2 \leftarrow x_1, \\ a_1(8) = 0,\ j \leftarrow j + 1 = 2 + 1 = 3, \\ r_3 \leftarrow q_1, \\ a_1(7) = 2,\ j \leftarrow j - 1 = 3 - 1 = 2, \\ r_2 \leftarrow r_2 \cdot r_3 = x_1q_1 \\ a_1(6) = 2,\ j \leftarrow j - 1 = 2 - 1 = 1, \\ r_1 \leftarrow r_2 + r_1 = q_1x_1 + q_2 \\ a_1(5) = 1, r_1 \leftarrow \sin(q_1x_1 + q_2) \\ a_1(4) = 0,\ j \leftarrow j + 1 = 1 + 1 = 2, \\ r_2 \leftarrow x_2, \\ a_1(3) = 1, r_2 \leftarrow -x_2 \\ a_1(2) = 1, r_2 \leftarrow \exp(-x_2) \\ a_1(1) = 2,\ j \leftarrow j - 1 = 2 - 1 = 1, \\ r_1 \leftarrow r_2 \cdot r_1 = \exp(-x_2)\sin(q_1x_1 + q_2). \end{gathered} \tag{11}$$

To increase the complexity of the desired mathematical expression, the universal code uses vector notation. Each component of the vector of the universal code is the code of a separate mathematical expression. All component code lengths are limited and can contain no more than N symbols.

$$\mathrm{Y} = (\mathbf{y}^1, \ldots, \mathbf{y}^L), \tag{12}$$

where

$$\mathbf{y}^i = \left(\begin{bmatrix}a_{i,1}(1)\\a_{i,2}(1)\end{bmatrix} \cdots \begin{bmatrix}a_{i,1}(N)\\a_{i,2}(N)\end{bmatrix}\right), \tag{13}$$

$i = 1, \ldots, L$.

Real length of each component i is $N_i \leq N$. All codes of components are calculated sequence. After calculation of the component, its value is adding to the set argument F_0. Values of codes of components are calculated sequence. After calculation of the code component, its value is adding to the set of arguments F_0 (1) of the alphabet of elementary function as in the Cartesian genetic programming, This value can be used in next code components.

A vector of small variation for universal code has dimension 5

$$\mathbf{w} = [w_1 \; w_2 \; w_3 \; w_4 \; w_5]^T, \tag{14}$$

where w_1 is a type of variation, w_2 is the component number, w_3 is the symbol number, w_4, w_5 are addition parameters for small variation.

A vector of small variation has the following types.

$w_1 = 0$ is a change of the second component of symbol code

$$a_{w_2,2}(w_3) = w_4 \tag{15}$$

$w_1 = 1$ is an elimination of a code element.

$$\begin{gathered} \text{If } a_{w_2,1}(w_3) = w_4 = 1 \text{ then} \\ a_{w_2,1}(w_3 + i - 1) \leftarrow a_{w_2,1}(w_3 + i), \\ a_{w_2,2}(w_3 + i - 1) \leftarrow a_{w_2,2}(w_3 + i), \\ i = 1, \ldots, N_{w_2} - w_3, \\ N_{w_2} \leftarrow N_{w_2} - 1. \end{gathered} \tag{16}$$

$$\begin{gathered} \text{If } a_{w_2,1}(w_3) = w_4 = 2 \text{ then} \\ j = 1, \; T = 1, \\ \text{while } T > 0 \text{ do} \\ T \leftarrow T + a_{w_2,1}(w_3 + j) - 1, \; j \leftarrow j + 1 \\ a_{w_2,1}(w_3 + i - 1) \leftarrow a_{w_2,1}(w_3 + i + j), \\ a_{w_2,2}(w_3 + i - 1) \leftarrow a_{w_2,2}(w_3 + i + j), \\ i = 1, \ldots, N_{w_2} - j, \\ N_{w_2} \leftarrow N_{w_2} - j. \end{gathered} \tag{17}$$

$w_1 = 2$ is an insert of function with one argument.

$$\begin{gathered} \text{If } N_{w_2} < N, \; \texttt{then} \\ a_{w_2,1}(i) \leftarrow a_{w_2,1}(i - 1), \\ a_{w_2,2}(i) \leftarrow a_{w_2,2}(i - 1), \\ i = N_{w_2} + 1, \ldots, w_3 + 1, \\ a_{w_2,1}(w_3) = 1, \\ a_{w_2,1}(w_3) = w_4 . N_{w_2} \leftarrow N_{w_2} + 1. \end{gathered} \tag{18}$$

$w_1 = 3$ is an insert of function with two arguments.

$$\begin{gathered} \text{If } N_{w_2} < N - 1, \; \texttt{then} \\ a_{w_2,1}(i) \leftarrow a_{w_2,1}(i - 2), \\ a_{w_2,2}(i) \leftarrow a_{w_2,2}(i - 2), \\ i = N_{w_2} + 2, \ldots, w_3 + 2, \\ a_{w_2,1}(w_3) = 2, \\ a_{w_2,2}(w_3) = w_4, \\ a_{w_2,1}(w_3 + 1) = 0, \\ a_{w_2,2}(w_3 + 1) = w_5. \end{gathered} \tag{19}$$

According to the rules of correct code in a universal code, you can always remove or insert function with one argument. Deleting a function with two arguments removes the function itself and the subexpression that defines the first argument of that function. Thus, this variation removes at least one argument of the entire argument of the mathematical expression. When inserting a function with two arguments, if the length of the code record allows, then the new function itself with two arguments and the argument of the mathematical expression are inserted. Thus, when this small variation is performed, the argument of the entire mathematical expression is inserted.

Consider examples. Let a variation vector is

$$\mathbf{w} = [0 \; 1 \; 3 \; 4 \; 1]^T.$$

Accurding to a small variation, the second component in third element ia replaced onto 3,

$$[0 \; 1 \; 3 \; 4 \; 1]^T \circ \tilde{y} = \\ \left(\begin{bmatrix} 2 \\ 2 \end{bmatrix} \begin{bmatrix} 1 \\ 3 \end{bmatrix} \begin{bmatrix} 1 \\ 4 \end{bmatrix} \begin{bmatrix} 0 \\ 2 \end{bmatrix} \begin{bmatrix} 1 \\ 4 \end{bmatrix} \begin{bmatrix} 2 \\ 1 \end{bmatrix} \begin{bmatrix} 2 \\ 2 \end{bmatrix} \begin{bmatrix} 0 \\ 3 \end{bmatrix} \begin{bmatrix} 0 \\ 1 \end{bmatrix} \begin{bmatrix} 0 \\ 4 \end{bmatrix} \right).$$

The received code corresponds the following mathematical expression

$$\tilde{y} = \exp(\sin(x_2)) \sin(q_1 x_1 + q_2).$$

Let a small variation is

$$\mathbf{w} = [1 \; 1 \; 5 \; 1 \; 1]^T.$$

According to this variation function with one argument is deleting in position 5. In result we receive the following code

$$[1 \; 1 \; 5 \; 1 \; 1]^T \circ \tilde{y} = \\ \left(\begin{bmatrix} 2 \\ 2 \end{bmatrix} \begin{bmatrix} 1 \\ 3 \end{bmatrix} \begin{bmatrix} 1 \\ 2 \end{bmatrix} \begin{bmatrix} 0 \\ 2 \end{bmatrix} \begin{bmatrix} 2 \\ 1 \end{bmatrix} \begin{bmatrix} 2 \\ 2 \end{bmatrix} \begin{bmatrix} 0 \\ 3 \end{bmatrix} \begin{bmatrix} 0 \\ 1 \end{bmatrix} \begin{bmatrix} 0 \\ 4 \end{bmatrix} \right).$$

The code corresponds the following mathematical expression

$$\tilde{y} = \exp(-x_2)(q_1 x_1 + q_2).$$

Let small variation is

$$\mathbf{w} = [1 \; 1 \; 7 \; 2 \; 1]^T.$$

This is deleting function with two arguments from position 7

$$[1 \; 1 \; 7 \; 2 \; 1]^T \circ \tilde{y} = \\ \left(\begin{bmatrix} 2 \\ 2 \end{bmatrix} \begin{bmatrix} 1 \\ 3 \end{bmatrix} \begin{bmatrix} 1 \\ 2 \end{bmatrix} \begin{bmatrix} 0 \\ 2 \end{bmatrix} \begin{bmatrix} 1 \\ 4 \end{bmatrix} \begin{bmatrix} 2 \\ 1 \end{bmatrix} \begin{bmatrix} 0 \\ 1 \end{bmatrix} \begin{bmatrix} 0 \\ 4 \end{bmatrix} \right).$$

The code corresponds the following mathematical expression

$$\tilde{y} = \exp(-x_2) \sin(x_1 + q_2).$$

Let small variation is

$$\mathbf{w} = [2 \; 1 \; 3 \; 4 \; 1]^T.$$

This is inserting function with one argument in position 3 with a code $[1 \; 4]^T$.

$$[2 \; 1 \; 3 \; 4 \; 1]^T \circ \tilde{y} = \\ \left(\begin{bmatrix} 2 \\ 2 \end{bmatrix} \begin{bmatrix} 1 \\ 3 \end{bmatrix} \begin{bmatrix} 1 \\ 4 \end{bmatrix} \begin{bmatrix} 1 \\ 2 \end{bmatrix} \begin{bmatrix} 0 \\ 2 \end{bmatrix} \begin{bmatrix} 1 \\ 4 \end{bmatrix} \begin{bmatrix} 2 \\ 1 \end{bmatrix} \begin{bmatrix} 2 \\ 2 \end{bmatrix} \begin{bmatrix} 0 \\ 3 \end{bmatrix} \begin{bmatrix} 0 \\ 1 \end{bmatrix} \begin{bmatrix} 0 \\ 4 \end{bmatrix} \right).$$

The code corresponds the following mathematical expression

$$\tilde{y} = \exp(\sin(-x_2)) \sin(q_1 x_1 + q_2).$$

Let small variation is

$$\mathbf{w} = [3\ 1\ 3\ 2\ 1]^T.$$

This is inserting function with two arguments $[2\ \ 1]^T$ in position 3 with additional argument $[0\ 1]^T$.

$$[2\ 1\ 3\ 4\ 1]^T \circ \tilde{y} =$$
$$\left(\begin{bmatrix}2\\2\end{bmatrix} \begin{bmatrix}1\\3\end{bmatrix} \begin{bmatrix}2\\1\end{bmatrix} \begin{bmatrix}0\\1\end{bmatrix} \begin{bmatrix}1\\2\end{bmatrix} \begin{bmatrix}0\\2\end{bmatrix} \begin{bmatrix}1\\4\end{bmatrix} \begin{bmatrix}2\\1\end{bmatrix} \begin{bmatrix}2\\2\end{bmatrix} \begin{bmatrix}0\\3\end{bmatrix} \begin{bmatrix}0\\1\end{bmatrix} \begin{bmatrix}0\\4\end{bmatrix} \right).$$

The code corresponds the following mathematical expression

$$\tilde{y} = \exp(x_1 - x_2)\sin(q_1 x_1 + q_2).$$

According to principle of small variation all possible solutions except a basic solution are set in the form as an order set of small variation vectors. All genetic operations are performed over these sets. For crossover operation two possible solutions or parents are selected

$$\begin{array}{l} \mathrm{W}_\alpha = (\mathbf{w}^{\alpha,1}, \dots, \mathbf{w}^{\alpha,d}), \\ \mathrm{W}_\beta = (\mathbf{w}^{\beta,1}, \dots, \mathbf{w}^{\beta,d}). \end{array} \qquad (20)$$

A crossover point is determined randomly, $c \in \{1, \dots, d\}$/ After that two new possible solutions are received by replacement components after the crossover point

$$\begin{array}{l} \mathrm{W}_{H+1} = (\mathbf{w}^{\alpha,1}, \dots, \mathbf{w}^{\alpha,c}, \mathbf{w}^{\beta,c+1}, \dots, \mathbf{w}^{\beta,d}), \\ \mathrm{W}_{H+2} = (\mathbf{w}^{\beta,1}, \dots, \mathbf{w}^{\beta,c}, \mathbf{w}^{\alpha,c+1}, \dots, \mathbf{w}^{\alpha,d}), \end{array} \qquad (21)$$

where H is a number possible solutions in an initial population, d - is a depth of variation.

III. SYNTHESIZED OPTIMAL CONTROL

The optimal control problem, formulated in the middle of the twentieth century, is classical and is often considered when creating complex control systems that must optimally move in the space of states. The main disadvantage of this problem is the complexity not so much of the problem itself as of implementing its solution on board a real object. This is because the solution to a problem in its classical formulation is a control function as a function of time. This means that with the direct implementation of the control function, as a function of time, we get an open loop control system that is very sensitive to disturbances in the initial state and inaccuracy of the mathematical model.

The problem specifies a mathematical model of the control object in the form of a system of ordinary differential equations with a free control vector on the right side.

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{u}), \qquad (22)$$

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

Control restrictions have been set.

$$\mathbf{u}^- \leq \mathbf{u} \leq \mathbf{u}^+, \qquad (23)$$

where $\mathbf{u}^-$, $\mathbf{u}^+$ are given low and upper vectors of restriction.

The initial and terminal states are given

$$\mathbf{x}^0 = [x_1^0 \dots x_n^0]^T, \qquad (24)$$

$$\mathbf{x}^f = [x_1^f \dots x_n^f]^T, \qquad (25)$$

The quality criterion is given

$$J = \int\limits_0^{t_f} f_0(\mathbf{x}, \mathbf{u})dt \rightarrow \min_{\mathbf{u}}, \qquad (26)$$

where t_f, as a rule not given, but it is limited and it is defined by time of achievement of terminal state

$$t_f = \begin{cases} t, \text{ if } t < t^+, \text{ and } \|\mathbf{x}^f - \mathbf{x}(t)\| \leq \varepsilon_1, \\ t^+, \text{ otherwise} \end{cases} . \qquad (27)$$

where ε_1 and t^+ are given positive values.

To solve the problem in the classical formulation, it is necessary to find a control function as a function of time that satisfies the given restrictions (23)

$$\mathbf{u} = \mathbf{v}(t). \qquad (28)$$

The system jf differential equations of mathematical model (22) with a control function (28) in the right side should have a particular solution from the given initial state (24), that hits to the terminal state (25) with optimal value of the quality criterion (26).

To solve the problem in the classical formulation, it is necessary to find a control function as a function of time that satisfies the given restrictions (**??**). The solution of this problem in the form of function of time (28) cannot be directly realized on the control object. A control system with function of time is open loop and it is sensitivity to small external disturbances and uncertainties of the model.

One of approaches to solving the optimal control problem in class of realizable control function is a synthesized optimal control. According to this approach firstly the control object is made stable relatively a point in the state space. After that the optimal control problem is solving for the stable control object. In this problem it is necessary to find positions of stable equilibrium points such, that after sequence activation of these point over equal time interval the control object from initial state (24) reached the terminal state (25) with optimal value of the given quality criterion (26).

IV. COMPUTATIONAL EXPERIMENT

Consider the stabilization system synthesis problem relatively a point in the state space for wheeled robot with differential drive.

The mathematical model of control object has the following form

$$\begin{array}{rcl} \dot{x}_1 & = & 0.5(u_1 + u_2)\cos(x_3), \\ \dot{x}_2 & = & 0.5(u_1 + u_2)\sin(x_3), \\ \dot{x}_3 & = & 0.5(u_1 - u_2), \end{array} \qquad (29)$$

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

The control vector has restrictions

$$-10 = u^- \leq u_i \leq u_i^+ = 10,\ i = 1, 2. \qquad (30)$$

The set of initial states is given and has twenty for points

$$\mathrm{X}_0 = \{\mathbf{x}^{0,1}, \dots, \mathbf{x}^{0,27}\}, \qquad (31)$$

where

$$\mathbf{x}^{0,i} = \begin{bmatrix} -4 + (i-1 \mod 3)\Delta_1 \\ -4 + (\lfloor (i-1)/6 \rfloor \mod 3)\Delta_2 \\ -5\pi/12 + (\lfloor (i-1)/9 \rfloor \mod 3)\Delta_3 \end{bmatrix}, \qquad (32)$$

$i = 1, \ldots, 27$, $\Delta_1 = 4$, $\Delta_2 = 4$, $\Delta_3 = 5\pi/12$.

The terminal state is

$$\mathbf{x}^f = [0\ 0\ 0]^T. \qquad (33)$$

The quality criterion requires to minimize length of summarize projection of trajectories of movement on the horizontal plane $x_1; x_2$ and accuracy achievement of the terminal state.

$$J_1 = \sum_{i=1}^{27} \left(\int_0^{t_{f,i}} \frac{|u_1 + u_2|}{2} dt + p_1 \Delta_f(t_{f,i}) \right) \to \min_{\mathbf{u}}, \qquad (34)$$

where p_1 is a penalty coefficient, $p_1 = 3$,

$$t_{f_i} = \begin{cases} t, \text{ if } t < t^+, \text{ and } \Delta_f(t) \le \varepsilon_1 \\ t^+, \text{ otherwise} \end{cases}, \qquad (35)$$

$$\Delta_f(t) = \sqrt{(x_1^f - x_1(t, \mathbf{x}^0))^2 + (x_2^f - x_2(t, \mathbf{x}^0))^2}, \qquad (36)$$

$\mathbf{x}(t, \mathbf{x}^0)$ is a particle solution of the differential equation (29) from the initial state $\mathbf{x}^0$. Firstly the network operator method[7] was used for solving this problem, The network operator was found the following solution

$$u_i = \begin{cases} u_i^+, \text{ if } u_i^+ \le \tilde{u}_i \\ u_i^-, \text{ if } \tilde{u}_i \le u_i^- \\ \tilde{u}_i, \text{ otherwise} \end{cases}, i = 1, 2, \qquad (37)$$

where

$$\tilde{u}_1 = \rho_{19}(A) + B + \rho_{17}(C) + \rho_{17}(D + E) +$$

$$\operatorname{sgn}(F) + \arctan(D), \qquad (38)$$

$$\tilde{u}_2 = \tilde{u}_1 - \tilde{u}_1^3 + A + G + \vartheta(C) + H + \mu(F) + q_1(x_1^f - x_1), \qquad (39)$$

$$A = G + H + \operatorname{sgn}(W)\sqrt{|W|} + \rho_{17}(D + E) + \mu(E) + D +$$

$$\rho_{19}(x_3^f - x_3) + x_2^f - x_2 - (x_2^f - x_2)^3 + (x_1^f - x_1)^{-1},$$

$$B = \operatorname{sgn}(G) + \mu(H) + \operatorname{sgn}(D + E)\sqrt{|D + E|} +$$

$$C + V^{-1} + \arctan(F) + \operatorname{sgn}(D),$$

$$C = W + \rho_{17}(D + E) + \rho_{18}(x_3^f - x_3),$$

$$D = q_2 \sin(x_3^f - x_3)(x_2^f - x_2)\mu(x_1^f - x_1),$$

$$E = \ln(|q_1(x_1^f - x_1)|)q_3(x_3^f - x_3),$$

$$F = E + q_1(x_1^f - x_1) + \cos(x_3^f - x_3) + (x_1^f - x_1)^{-1},$$

$$G = \mu(C) + H + \exp(W) + V + \mu(D + E) + F + \exp(x_3^f - x_3),$$

$$H = W - W^3 + V + \arctan(D) + \rho_{19}(x_3^f - x_3),$$

$$W = \arctan(D + E) + \sin(x_1^f - x_1),$$

$$V = F + q_1(x_1^f - x_1) - q_1^3(x_1^f - x_1)^3,$$

$$\mu(\alpha) = \begin{cases} \alpha, \text{ if } |\alpha| \le 1 \\ \operatorname{sgn}(\alpha), \text{ otherwise} \end{cases},$$

$$\vartheta(\alpha) = \begin{cases} 1, \text{ if } \alpha > 0 \\ 0, \text{ otherwise} \end{cases},$$

$$\rho_{17}(\alpha) = \operatorname{sgn}(\alpha)\ln(|\alpha| + 1),$$

$$\rho_{18}(\alpha) = \operatorname{sgn}(\alpha)(\exp(|\alpha|) - 1),$$

$$\rho_{19}(\alpha) = \operatorname{sgn}(\alpha)\exp(-|\alpha|),$$

$q_1 = 15.93579$, $q_2 = 14.04517$, $q_3 = 2.88135$.

Projections of trajectories from eight initial states on the horizontal plane for control object with the stabilization system are presented in the Figures 1 and 2. As you can see from the figures of the trajectory of movement from corner points, the ones shown in Fig. 1 are not straight. Further attempts to improve stabilization trajectories by the network operator method did not lead to a noticeable improvement.

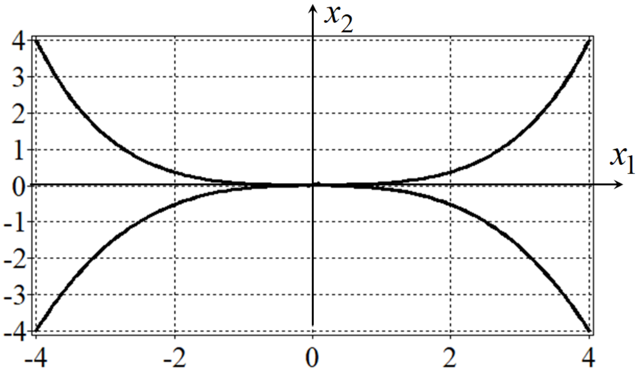

Fig. 1. Projections of trajectories from eight angular initial states for control system received by the network operator method

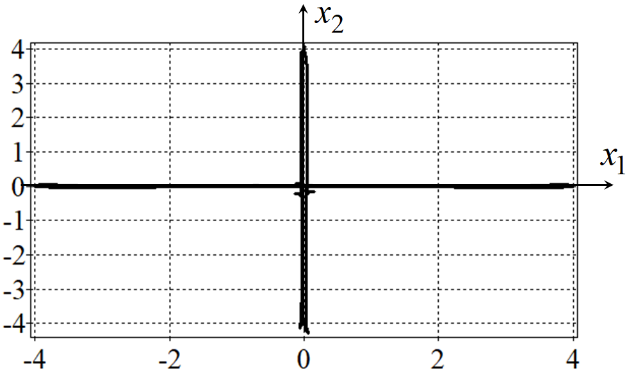

Fig. 2. Projections of trajectories from eight direct initial states for control system received by the network operator method

Further search for a better solution by the network operator method did not lead to more improve control system. To improve the solution found, the combined method of symbolic regression was applied. The solution received by the network operator was encoded with a universal code and was used as a basic solution. Variation universal code found the following solution after that.

$$\tilde{u}_1 = \rho_{19}(z_9) + z_8 + \rho_{17}(z_6) +$$

$$\rho_{17}(\mu(z_6) + z_6 + z_1 z_0 + \mathrm{sgn}(z_2) + \arctan(z_0), \qquad (40)$$

$$\tilde{u}_2 = \tilde{u}_1 - \tilde{u}_1^3 + z_9 + z_7 + \vartheta(z_6)\sqrt{|z_6|} + z_5 +$$

$$\mathrm{sgn}(z_2) + q_1(x_1^f - x_1), \qquad (41)$$

where

$$z_0 = q_2 \cos(x_3^f - x_3)\mu(\mu(x_2^f - x_2))\mathrm{sgn}(\mu(x_1^f - x_1)),$$

$$z_1 = \ln(|q_1(x_1^f - x_1)|)q_3(x_3^f - x_3),$$

$$z_2 = z_1 + q_1(x_1^f - x_1) + \cos(x_3^f - x_3) + (x_1^f - x_1)^{-1},$$

$$z_3 = z_2 + \mu(q_1(x_1^f - x_1) + (x_1^f - x_1)^3),$$

$$z_4 = \arctan(z_1 + z_0) + \sin(x_1^f - x_1),$$

$$z_5 = \mu(z_4 - z_4^3) + (x_1^f - x_1) + z_3 + \arctan(\rho_{19}(x_3^f - x_3)),$$

$$z_6 = z_4 + \rho_{17}(z_1 + \arctan(z_0)) + \rho_{18}(x_3^f - x_3),$$

$$z_7 = \mu(z_6) + z_5 + \exp(z_4) + z_3 + \mu(z - 1 + z_0) +$$

$$z_2 + \exp(x_3^f - x_3),$$

$$z_8 = \mathrm{sgn}(z_7) + z_6 + \mu(z_5) + (z_3)^{-1} + \mathrm{sgn}(z_1 + z_0)\sqrt{|z_1 + z_0|} +$$

$$\arctan((x_2^f - x_2)z_2) + \mathrm{sgn}(x_1^f - x_1),$$

$$z_9 = z_7 + z_5 + \mathrm{sgn}(z_4)\sqrt{|z_4|} + \rho_{17}(z_1 + z_0) +$$

$$\mu(z_1) + z_0 + \rho_{19}(x_3^f - x_3) + x_2^f - x_2 - (x_2^f - x_2)^3 + (x_1^f - x_1)^{-1},$$

$q_1 = 15.93481$, $q_2 = 15.96313$, $q_3 = 2.88135$.

Results of experiments are presented in the Figures 3 and 4.

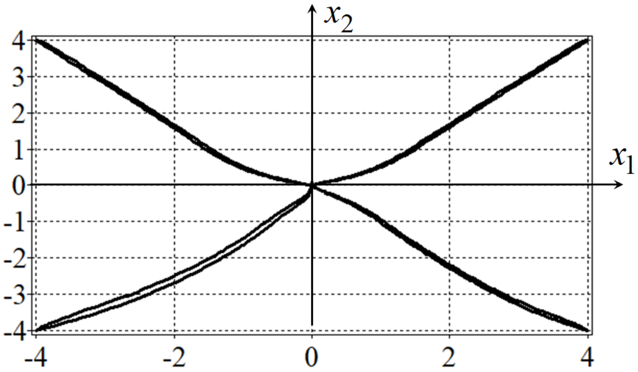

Fig. 3. Projections of trajectories from eight angular initial states for control system received by combined approach

How you can see in figures the combine approach has improved the stabilization system. As you can see from Figure 3, the use of another symbolic regression method after the network operator improved the trajectories from the corner points. They have become more direct.

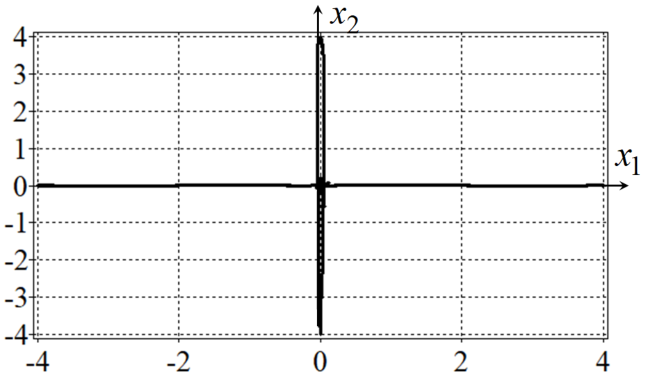

Fig. 4. Projections of trajectories from eight angular initial states for control system received by combined approach

V. Conclusions

The paper describes in detail the new method of symbolic regression. The work also proposes to improve the solutions found by the method of symbolic regression, to use a combined approach in which the problem is initially solved by one method of symbolic regression, and then solved by another method that uses the previously found solution. In a computational experiment, a combined approach is used to search for a wheel robot stabilization system with a differential drive. Experiments have shown the effectiveness of the combined approach.

References

[1] M. Krstic, I. Kanellakopoulos, and P.R. Kokotovic. Nonlinear and adaptive control design. John Wiley and Sons, 1995, 592 p.

[2] A.A. Kolesnikov, Al.A. Kolesnikov, A.A. Kuz'menko, Backstepping and ADAR Method in the Problems of Synthesis of the Nonlinear Control Systems. Mekhatronika, Avtomatizatsiya, Upravlenie. 2016;17(7), pp. 435-445. (In Russ.) https://doi.org/10.17587/mau.17.435-445

[3] A.A. Kolesnikov, Al.A. Kolesnikov, A.A. Kuz'menko, ADAR Method and Theory of Adaptive Control in the Tasks of Synthesis of the Nonlinear Control Systems. Mekhatronika, Avtomatizatsiya, Upravlenie. 2017;18(9), pp. 579-589. (In Russ.) https://doi.org/10.17587/mau.18.579-589.

[4] A. Diveev. A Numerical Method for Network Operator for Synthesis of a Control System with Uncertain Initial Values. Journal of Computer and Systems Sciences International, 2012, 51(2), pp. 228-243. https://doi.org/10.1134/s1064230712010066
Diveev, A. I. (2012). A numerical method for network operator for synthesis of a control system with uncertain initial values. Journal of Computer and Systems Sciences International, 51(2), 228–243. doi:10.1134/s1064230712010066.

[5] J.R. Koza, M.A. Keane, M.J. Streeter, W. Mydlowec, J. Yu, G. Lanza. Genetic Programming IV Routine Human-Competitive Machine Intelligence, 2005, Springer, 590 p. doi: 10.1007/b137549.

[6] . A. Diveev. Small Variations of Basic Solution Method for Non-numerical Optimization. IFAC-PapersOnLine, 2015, 48(25), pp. 28–33. doi:10.1016/j.ifacol.2015.11

[7] A. Diveev, E. Shmalko. Machine Learning Control by Symbolic Regression, 2021, Springer, 158 p.

[8] J.F. Miller. Cartesian Genetic Programming. Chapter 2 in Book in Natural Computing Series, 2003 Springer, pp. 12-34.