2017 13th International Conference on Natural Computation, Fuzzy Systems and Knowledge Discovery (ICNC-FSKD 2017)

Binary Variational Genetic Programming for the Problem of Synthesis of Control System

Diveev, A.I.
Federal Research Centre "Computer Science and Control" of Russian Academy of Sciences
Academy of Engineering, RUDN University
Moscow, Russia
aidiveev@mail.ru

Balandina, G.I.
Academy of Engineering, RUDN University
Moscow, Russia
balandina_gi@pfur.ru

Konstantinov, S.V.
Academy of Engineering, RUDN University
Moscow, Russia
svkonstantinov@mail.ru

Abstract—**The paper describes a novel numerical symbolic regression method. It's called complete binary variational genetic programming. We use it for synthesis of optimal control. This method performs better than genetic programming at crossover, reduces the search area and speeds up search algorithm by using small variations. The efficiency of the new method is proven on the given example of control system synthesis for mobile robot.**

Keywords-synthesis of control; optimal control; symbolic regression; genetic programming.

I. Introduction

The problem of synthesis of control consists of finding of control operator or function, that depends on a vector of the state space. At the deciding the problem of synthesis we have to find the mathematical expression of the control function, because the function has to be work for different positions of control object. The structure of mathematical expression of control function can be found by methods of symbolic regression. Arguments of function of control are coordinates of state space of the control object. Integral target functional is defined in the form of minimum of error of hitting terminal point and minimum of time of target hit. In case of some constraints the penalty is defined, which increases value of integral target functional when an object is in the area of a certain constraint point.

Numerical methods of symbolic regression apply to the solution of the problem of synthesis of control long ago [1-8]. Some methods as the network operator [3-6], variational genetic programming [7], and analytic programming [8] use a variational genetic algorithm to search of optimal solution. The known methods of genetic and analytical programming, as well as other well-known method of grammatical evolution [9], have an essential shortcoming for an algorithm of evolutionary search. These methods use a genetic algorithm at search of the optimal solution. The main operation of genetic algorithm is a crossover. Performance of this operation demands exchange of subsets of codes in the possible decisions selected for the crossover. As a result, we often can receive codes of different length. It is very uncomfortable for further search. The new method uses the principle of small variations of the basic solution [10] as well as variational methods of genetic and analytical programming. We can obtain an incorrect code after some variations in methods of variational genetic and analytical programming. Then we have to spend time for code correction. It is time-consuming and contradicts the principle of small variations.

The paper presents a novel symbolical regression method of binary variational genetic programming. This method is an improving of binary genetic programming [11]. When performing crossover, this method does not change the code length, and does not require correction of code at any variations. It presents mathematical expressions as complete binary computing trees and codes them in the PC memory as ordered finite sets of integers. Binary functions of mathematical expressions are placed in the nodes of the tree-graphs. Unary operations are placed on the edges of the graph. Arguments, parameters, and unit elements for binary operations are placed in the leaves of the trees or source-nodes of the graphs. The set of functions with one argument has to include the identity function. We add constant parameters as arguments of mathematical expression to increase the number of functions used in the search. We search the values of constants as well as the structures of the mathematical expressions.

Search of the optimal solution is carried out by a variational genetic algorithm. For this purpose, we define small variations of a binary computing tree and set one basic solution. We carry out all operations of the genetic algorithm on sets of small variations of the basic solution.

II. The Problem of Synthesis of Control System

Consider the formal problem statement of synthesis of control system [4].

Set a model of control object [4]

This research was supported by Russian Foundation for Basic Research according to the research project №. 17-08-01203-a

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

where $\mathbf{x}$ is a vector of state, $\mathbf{x} \in \mathbf{R}^n$, $\mathbf{x} = [x_1 \ldots x_n]^T$, $\mathbf{u}$ is a vector of control, $\mathbf{u} \in \mathbf{R}^m$, $\mathbf{u} = [u_1 \ldots u_m]^T$, $m \le n$, t is time.

Set constraints on the control [4]

$$\mathbf{u} \in \mathrm{U} \subseteq \mathbf{R}^m, \quad (2)$$

where U is a compact set.

Given a set of initial conditions [4]

$$\mathbf{x}^0 \in \mathrm{X}_0 \subseteq \mathbf{R}^n,$$

where X_0 is a domain of the state space or a set of points of initial positions at the computation synthesis, [4]

$$\mathrm{X}_0 = \{\mathbf{x}^{0,1}, \mathbf{x}^{0,2}, \ldots, \mathbf{x}^{0,M}\}. \quad (3)$$

Set a terminal point [4]

$$\mathbf{x}^f \in \mathbf{R}^n. \quad (4)$$

Given phase constraints [4]

$$g_i(\mathbf{x}) \le 0, \; i = 1, \ldots, r. \quad (5)$$

Set a criterion of control as an integral goal function [4]

$$\tilde{J} = \int_0^T f_0(\mathbf{x}(t), \mathbf{u}(x(t)), t)dt \to \min, \quad (6)$$

where

$$T = \begin{cases} t, \text{ if } t < t^+ \text{ and } \| \mathbf{x}^f - \mathbf{x}(t) \| < \varepsilon \\ t^+ \text{ - otherwise} \end{cases}, \quad (7)$$

ε is a small positive value, t^+ is a set limited value of time,

$$\| \mathbf{x}^f - \mathbf{x}(t) \| = \max_i \{ | x_i^f - x_i(t) | : i = 1, \ldots, n\}. \quad (8)$$

At the decision of the problem we have to find a control function depending on a vector of the state space coordinates [4]

$$\mathbf{u} = \mathbf{h}(\mathbf{x}), \quad (9)$$

where $\mathbf{h}(\mathbf{x})$ - is a limited multi-dimension function $\mathbf{h}(\mathbf{x}) : \mathbf{R}^n \to \mathbf{R}^m$, $\forall \mathbf{x} \in \mathbf{R}^n$, $\mathbf{h}(\mathbf{x}) \in \mathrm{U} \subseteq \mathbf{R}^m$ [4].

In order to consider phase restrictions, we will include them in the goal function

$$\tilde{\tilde{J}} = \int_0^{t_f} (f_0(\mathbf{x}(t), \mathbf{u}(x(t))) + \sum_{i=1}^{r} \chi(g_i(\mathbf{x}) g_i(\mathbf{x})) dt, \quad (10)$$

where $\chi(a)$ is Heaviside step function [4]

$$\chi(a) = \begin{cases} 1, \text{ if } a > 0 \\ 0, \text{ otherwise} \end{cases}. \quad (11)$$

The synthesizing function (9) has to supply the achievement of control target (4) from some initial positions (3) with optimum values of the quality criterion (10). We change the criterion (10) on the sum of goal functions (10) for each initial position from the set (3)

$$J(\mathbf{u}(\mathbf{x})) = \sum_{j=1}^{M} \left(\tilde{\tilde{J}} \right)_{\mathbf{x}(0) = \mathbf{x}^{0,j}} \to \min. \quad (12)$$

To decide the task, we apply the new symbolic regression method, which has to find the function by an evolutionary algorithm.

III. Binary Variational Genetic Programming

To construct a code of binary genetic programming we set the basic sets of functions and arguments:

- a set of arguments of mathematical expression [11]

$$\mathrm{F}_0 = (x_1, \ldots, x_N, q_1, \ldots, q_P); \quad (13)$$

- a set of functions with one argument [11]

$$\mathrm{F}_1 = (f_{1,1}(y) = y, f_{1,2}(y), \ldots, f_{1,R}(y)) \text{ [11]}; \quad (14)$$

- a set of functions with two arguments [11]

$$\mathrm{F}_2 = (f_{2,1}(y_1, y_2), \ldots, f_{2,S}(y_1, y_2)); \quad (15)$$

- a set of unit elements for functions with two arguments [11]

$$\mathrm{E}_2 = (e_1, \ldots, e_S). \quad (16)$$

It is necessary that the identity function have included in the set of functions with one argument [11]

$$f_{1,1}(y) = y. \quad (17)$$

Functions with two arguments from the set (15) must have a unit element from the set (16) [11], $\forall f_{2,i}(y_1, y_2) \in \mathrm{F}_2$ $\exists e_j \in \mathrm{E}_2$

$$f_{2,i}(e_j, y_2) = y_2, \; f_{2,i}(y_1, e_j) = y_1, \quad (18)$$

where $i, j \in \{1, \ldots, S\}$.

Before obtaining a code of mathematical expression we include the set of unit elements (16) into the set of arguments (13) of mathematical expression, a set of unit elements [11]

$$F = (f_1 = x_1, \ldots, f_N = x_N, f_{N+1} = q_1, \ldots, f_{N+P} = q_P,$$

$$f_{N+P+1} = e_1, \ldots f_{N+P+S} = e_S). \quad (19)$$

Firstly, we write down mathematical expression in the form of composition of elements of mathematical expression, functions, variables and parameters

$$y = f_{\alpha_1}(f_{\alpha_2}(\ldots f_{\alpha_K})\ldots) = f_{\alpha_1} \circ f_{\alpha_2} \circ \ldots \circ f_{\alpha_K}, \quad (20)$$

where $f_{\alpha_i} \in F \cup F_1 \cup F_2$, $i = 1, \ldots, K$.

Assume we have set the most quantities of nesting of components in function composition. Let the most quantities of nesting is equal L. As a result we obtain a binary computing tree with level L. The tree with L levels has 2^L leaves. We present a mathematical expression as a computing complete binary tree (see Fig. 1)

In the fig.1 a is number of an element of set F (13) of arguments of mathematical expression, b is number of function with two arguments from the set F_2 (15), u is number of function with one argument from the F_1 (14).

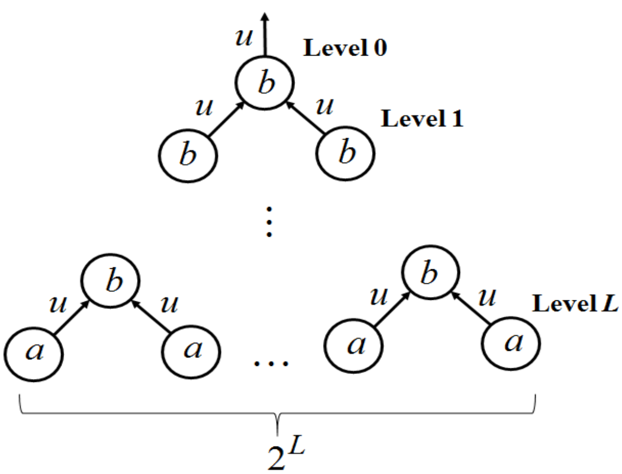

Figure 1. A comlete binary computing tree

In every level the tree has the same quantity of integers for numbers of functions with one and two arguments. The down level includes numbers of functions with one arguments and numbers of elements from the set of arguments of mathematical expression. For completing of the level we use any function with two arguments with its unit element and the identity function. As a result, we obtain code for mathematical expression in the form of ordered set of integers. Each number focuses to functions with one or two arguments, and numbers of elements from the set of arguments of mathematical expression om the last level.

$$C = (u_1, b_1, u_{2,1}, u_{2,2}, b_{2,1}, b_{2,2}, \ldots$$

$$\ldots, u_{L,1}, u_{L,2}, \ldots, u_{L,2^L}, a_{L,1}, a_{L,2}, \ldots, a_{L,2^L}). \quad (21)$$

Consider an example. Let mathematical expression be set

$$g = e^{-qx}\cos(\sin(x+q)).$$

For the mathematical expression, we have sets

$$F_1 = (f_{1,1}(y) = y, f_{1,2}(y) = e^y, f_{1,3}(y) = -y,$$

$$f_{1,4}(y) = \cos(y), f_{1,5}(y) = \sin(y)),$$

$$F_2 = (f_{2,1}(y_1, y_2) = y_1 + y_2, f_{2,2}(y_1, y_2) = y_1 y_2,),$$

$$F = (f_1 = x, f_2 = q, f_3 = 0, f_4 = 1).$$

Here 0 is a unit element for summation function and 1 is unit element for production function.

We present the mathematical expression in the form of composition of nesting functions from the sets

$$g = f_{2,2}(f_{1,2}(f_{1,3}(f_{2,2}(q,x))), f_{1,4}(f_{1,5}(f_{2,1}(x,q)))).$$

We have for our case nested depth is three. The fig. 2 shows complete binary computing tree for the mathematical expression

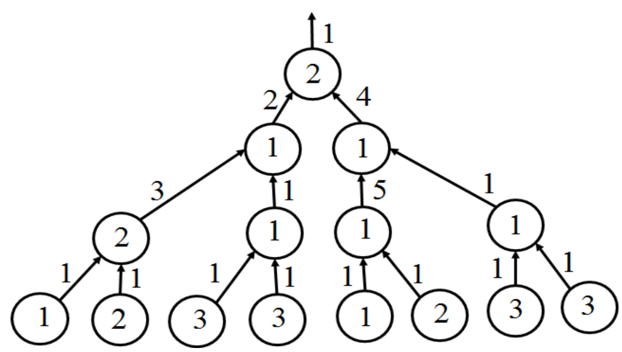

Figure 2. A comlete binary computing tree for the mathematical expression

The code of mathematical expression is

$$C = (1,2,\ 2,4,1,1,\ 3,1,5,1,2,1,1,1,\ 1,1,1,1,1,1,1,1,2,3,3,1,2,3,3).$$

For calculation of mathematical expression, we use an ordered set of real variables with the same cardinal number as the code.

$$Y = (y_1, \ldots, y_{30}).$$

Calculate mathematical expression from the end

$$Y = (y_1, \ldots, y_{14}, x, q, 0, 0, x, q, 0, 0, x, q, 0, 0, x, q, 0, 0)\ ,$$

$$Y = (y_1, \ldots, y_{10}, xq, 0, x+q, 0, \ldots)\ ,$$

$$Y = (y_1, \ldots, y_6, -xq, 0, \sin(x+q), 0, \ldots)\ ,$$

$$Y = (y_1, y_2, e^{-xq}, \cos(\sin(x+q)), -xq, \sin(x+q), \ldots)\ ,$$

$$Y = (e^{-xq}\cos(\sin(x+q)), e^{-xq}\cos(\sin(x+q)), \ldots)\ .$$

IV. A Variational Genetic Algorithm

A variational genetic algorithm uses the principle of small variations [10]. For using of this principle we define a small variation of a code in the form of the integer vector with two components [10]

$$\mathbf{w} = [w_1\ \ w_2]^T\ , \qquad (22)$$

where w_1 is a number on order of element in a code, w_2 is a new integer number for element of the code in position w_1.

Assume a code of mathematical expression

$$C = (c_1, \ldots, c_K) \qquad (23)$$

is described by a binary tree with L levels. Then

$$K = 2^{L+2} - 2\ . \qquad (24)$$

When we have (22) than we obtain a new code [10]

$$\mathbf{w} \circ C = (\overbrace{c_1, \ldots, w_2}^{w_1}, \ldots, c_K)\ . \qquad (25)$$

For small variation (22) the following rules are right

$$w_1 \in \{1, \ldots, 2^{L+2} - 2\}\ , \qquad (26)$$

$$w_1 \in \begin{cases} \{1, \ldots, |F_1|\},\ \text{if}\ \ 2^i - 1 \le w_1 \le 3 \cdot 2^{i-1} - 2 \\ \{1, \ldots, |F_2|\},\ \text{if}\ \ 3 \cdot 2^{i-1} - 1 \le w_1 \le 2^{i+1} - 2\ , \\ \{1, \ldots, |F|\},\ \text{if}\ 3 \cdot 2^{L} - 1 \le w_1 \le 2^{L+2} - 2 \end{cases} \qquad (27)$$

where $i = 1, \ldots, L$.

Genetic operations in a variational genetic algorithm are carried out on sets of variation vectors (22). A genetic algorithm constructed on the principle of small variations consists of the following steps:

1) Give a basic solution by the theoretic researches or on the expirience of the researcher

$$C_0 = (c_1^0, \ldots, c_K^0)\ ;$$

2) Generate randomly ordered sets of variation vectors

$$\Omega = \{W_1, \ldots, W_H\}\ ,$$

where W_i is an ordered set of variation vectors (22)

$$W_i = (\mathbf{w}^{i,1}, \ldots, \mathbf{w}^{i,l})\ ,$$

$\mathbf{w}^{i,j} = [w_1^{i,j}\ \ w_2^{i,j}]^T$, $j = 1, \ldots, l$, $i = 1, \ldots, H$, l is a length variation or quantity of small variations in one of the set of variations.

Operations crossover and mutation are applied to the sets of small variations [10]. We select two parents $W_i = (\mathbf{w}^{i,1}, \ldots, \mathbf{w}^{i,l})$ and $W_j = (\mathbf{w}^{j,1}, \ldots, \mathbf{w}^{j,l})$ like in any genetic algorithm, and randomly determine a point for crossover $c \in \{1, \ldots, l\}$ and then we exchange tails of the selected parents

$$\widetilde{W}_i = (\mathbf{w}^{i,1}, \ldots, \mathbf{w}^{i,c-1}, \mathbf{w}^{j,c}, \ldots, \mathbf{w}^{j,l})\ ,$$

$$\widetilde{W}_j = (\mathbf{w}^{j,1}, \ldots, \mathbf{w}^{j,c-1}, \mathbf{w}^{i,c}, \ldots, \mathbf{w}^{i,l})\ .$$

For an operation of mutation we randomly create two points of mutation $\mu, \eta \in \{1, \ldots, l\}$. Then we randomly generate two new variations vectors in the positions μ $\mathbf{w}^{i,\mu} = [w_1^{i,\mu}\ \ w_2^{i,\mu}]^T$ and η $\mathbf{w}^{i,\eta} = [w_1^{i,\eta}\ \ w_2^{i,\eta}]^T$ for both new solutions.

V. Computational Experiment

Let a mobile robot is considered as a control object [12]

$$\dot{x}_1 = \frac{u_1 + u_2}{2}\cos(x_3)\ ,$$

$$\dot{x}_2 = \frac{u_1 + u_2}{2}\sin(x_3)\ ,$$

$$\dot{x}_3 = \frac{u_1 - u_2}{2}\ ,$$

where restrictions on controls are set, $-10 \le u_{1,2} \le 10$.

We must find a control function for the object, that will move it from any initial condition

$$X_0 = \{\mathbf{x}^{0,1} = [10\ 10\ 0]^T, \mathbf{x}^{0,2} = [10\ 11\ 0]^T,$$

$$\mathbf{x}^{0,3} = [11\ 10\ 0]^T, \mathbf{x}^{0,4} = [11\ 11\ 0]^T\}$$

to the terminal condition

$$\mathbf{x}^f = [0\ 0\ 0]^T$$

for minimal time.

The goal function has the following form

$$J(\mathbf{u}(\mathbf{x})) = \sum_{j=1}^{4} (t_f(\mathbf{x}^{0,j}) + s(\mathbf{x}^{0,j})),$$

where

$$s(\mathbf{x}^{0,j}) = \int_0^{t_f} \left(\sum_{i=1}^{2} \vartheta(g_i(\mathbf{x}(t))) g_i(\mathbf{x}(t)) \right) dt,$$

t_f is a time of control process and it is defined by equation (7).

$$g_1(\mathbf{x}(t)) = r_1 - \sqrt{(x_1(t) - \tilde{x}_{1,1})^2 + (x_2(t) - \tilde{x}_{1,2})^2} \le 0,$$

$$g_2(\mathbf{x}(t)) = r_2 - \sqrt{(x_1(t) - \tilde{x}_{2,1})^2 + (x_2(t) - \tilde{x}_{2,2})^2} \le 0.$$

where $\varepsilon = 0.01$, $t^+ = 2.8$. $r_1 = 2.5$, $r_2 = 2.5$, $\tilde{x}_{1,1} = 7.5$, $\tilde{x}_{1,2} = 7.5$, $\tilde{x}_{2,1} = 2.5$, $\tilde{x}_{2,2} = 2.5$.

At search of solution we use the following functions with one argument:

$$F_1 = (f_1(y) = y,\ f_2(y) = y^2,\ f_3(y) = -y,$$

$$f_4(y) = \mathrm{sgn}(y)\sqrt{|y|},\ f_5(y) = \frac{1}{y},\ f_6(y) = e^y,$$

$$f_7(y) = \ln(|y|),\ f_8(y) = \frac{1 - e^{-y}}{1 + e^{-y}},\ f_9(y) = \begin{cases} 1, \text{ if } y \ge 0 \\ 0 \text{ - otherwise} \end{cases},$$

$$f_{10}(y) = \mathrm{sgn}(y),\quad f_{14}(y) = y^3,\ f_{15}(y) = \sqrt[3]{y},$$

$$f_{16}(y) = \begin{cases} y, \text{ if } |y| \le 1 \\ \mathrm{sgn}(y) \text{ - otherwise} \end{cases},\ f_{17}(y) = \mathrm{sgn}(y)\ln(|y| + 1),$$

$$f_{18}(y) = \mathrm{sgn}(y)\exp(|y| - 1),\ f_{23}(z) = y - y^3).$$

The set of functions with two arguments were

$$F_2 = (f_1(y_1, y_2) = y_1 + y_2,\ ,\ f_2(y_1, y_2) = y_1 y_2).$$

A basic solution was the following

$$u_{1,2} = q_1(x_1^f - x_1) + q_2(x_2^f - x_2) + q_3(x_3^f - x_3).$$

where $q_i = 1$, $i = 1, \ldots, 3$.

We used the following parameters of a genetic algorithm: cardinal number of elements in initial set is 512, number of generations in one epoch before exchange of a basic solution is 65, number of possible crossovers and mutations in one generation is 128, general number of generations is 128, length of variation or number of small variation in one possible solution is 8, probability of mutation is 0.7.

We have obtained the following code of solution: 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 3, 1, 1, 1, 1, 1, 6, 1, 1, 1, 1, 1, 2, 1, 1, 1, 10, 1, 1, 1, 3, 1, 1, 4, 1, 4, 1, 1, 1, 5, 1, 1, 2, 2, 2, 2, 2, 2, 1, 2, 2, 2, 1, 2, 2, 2, 2, 2, 1, 1, 1, 1, 1, 1, 1, 1, 1, 9, 1, 1, 1, 4, 1, 1, 1, 1, 1, 1, 1, 3, 1, 1, 1, 1, 1, 6, 1, 1, 1, 1, 0, 6, 1, 4, 2, 5, 0, 3, 4, 4, 2, 5, 0, 6, 2, 4, 2, 5, 0, 3, 1, 4, 1, 5, 0, 3, 1, 4, 2, 5, 0, 3.

The values of components of controls were in positions 3 and 4 of the obtained code. The code corresponds the following mathematical expressions for control

$$u_i = \begin{cases} \mathrm{sgn}(u_i)10, \text{if } |u_i| > 10 \\ \tilde{u}_i \text{ - otherwise} \end{cases},\ i = 1,2,$$

where

$$\tilde{u}_1 = q_3 x_3 + q_1 x_1 - \mathrm{sgn}(x_1) - q_2 x_2 +$$

$$q_3 x_3 - q_2 + x_1 + \mathrm{sgn}(x_3)\sqrt{|q_2 x_3|},$$

$$\tilde{u}_2 = q_3 x_3\ \mathrm{sgn}(x_1)\sqrt{|q_1 x_1|} + x_2 - q_2 + q_3 x_2 +$$

$$\exp\left(q_1 x_1 + \frac{1}{x_2 \exp(q_2)} \right) + q_3 x_3 + q_1 x_1,$$

$q_1 = 3.40918$, $q_2 = 0.68164$, $q_3 = 3.63184$.

Fig. 3 shows results of simulation of robot movement from four initial conditions. Red areas are dynamic phase constraints which were successfully avoided. Found control gave the following results: time of control process – $t = 2.59$, accuracy of implementation of the terminal conditions is 0.0084. Fig. 4 shows the angle of the speed direction of the mobile robot, fig. 5 and fig. 6 show values of control u_1 and u_2 respectively for one of the initial conditions.

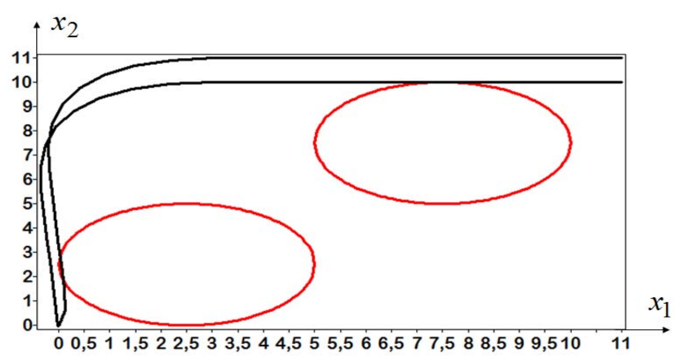

Figure 3. Trajectories of movement of robot from four intial conditions

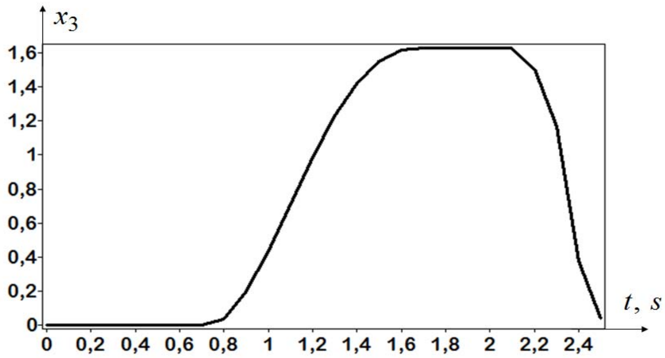

Figure 4. The angle of the direction of the mobile robot

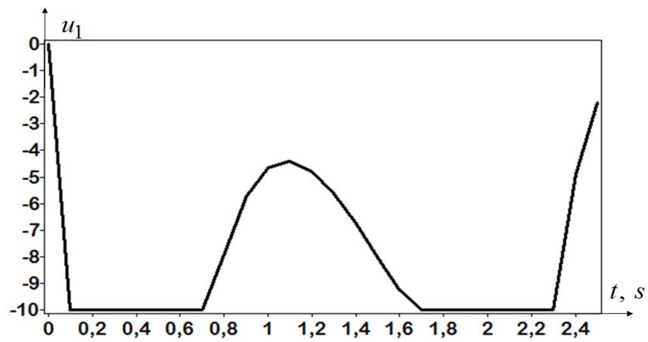

Figure 5. Control u_1

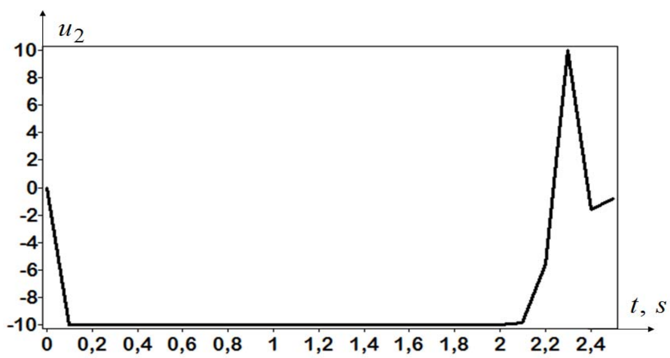

Figure 6. Control u_2

VI. CONCLUSIONS

The paper has presented a new symbolic regression method of complete binary variational genetic programming. The new method codes a mathematical expression in the form of a set of integer numbers. Every element of the code is a number of element from the sets of functions or number of variables or parameters. The order of numbers in the code connects with levels of complete binary computing tree. A computational experiment showed as the new method have solved the problem of synthesis of optimal control for mobile robot. We obtained a mathematical expression for control function depending on the state of the object The experiment showed the efficiency of the new method of symbolic regression.

REFERENCES

[1] K. Cpałka, K. Łapa, and A. Przybył G. Eason "A New Approach to Design of Control Systems Using Genetic Programming," Information Technology And Control, 2015, no. 4, vol. 44, pp. 433-442

[2] J. Imae, Y. Kikuchi, N. Ohtsuki, T. Kobayashi, and G. Zhai "Design of nonlinear control systems by means of differential genetic programming," Proceedings of the 43rd IEEE Conference on Decision and Control, December 14-17, 2004, Atlantis, Paradise Island, Bahamas, pp. 2734-2739.

[3] R. A. Maher, M. J. Mohamed. "An Enhanced Genetic Programming Algorithm for Optimal Controller Design," Intelligent Control and Automation, 2013, vol. 4, pp. 94-101.

[4] A. Diveev. And E. Sofronova "Application of Network Operator Method for Synthesis of Optimal Structure and Parameters of Automatic Control System," Proc. of 17-th IFAC World Congress, Seoul, 05.07.2008 – 12.07.2008, pp. 6106 – 6113.

[5] A. Diveev and E.Sofronova, E., "Numerical Method of Network Operator for Multi-objective Synthesis of Optimal Control System," Proc. of Seventh International Conference on Control and Automation (ICCA'09) Christchurch, New Zealand, December 9-11, pp. 701-708.

[6] 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, no 2, vol. 51, pp. 228–243.

[7] A. Diveev, S. Ibadulla, N. Konyrbaev, and E. Shmalko, "Variational Genetic Programming for Optimal Control System Synthesis of Mobile Robots," Preprints of the 11th IFAC Symposium on Robot Control, Salvador, BA, Brazil, August 26-28, pp. 113-118.

[8] A. Diveev, S. Ibadulla, N. Konyrbaev, and E. Shmalko, "Variational Analytic Programming for Synthesis of Optimal Control for Flying Robot," Preprints of the 11th IFAC Symposium on Robot Control, Salvador, BA, Brazil, August 26-28,pp. 82-87.

[9] M. O'Neill and C. Ryan, "Grammatical Evolution," IEEE Trans. Evol. Comput. 2001, 5, pp. 349 – 358.

[10] A.I. Diveev "Small Variations of Basic Solution Method for Non-numerical Optimization" IFAC-PapersOnLine, 2015, vol. 48 (25), pp. 028-033.

[11] A.I. Diveev, N.B. Konyrbaev, and E.A. Sofronova "Method of Binary Analytic Programming to Look for Optimal Mathematical Expression," XIIth International Symposium «Intelligent Systems», INTELS'16, 5-7 October 2016, Moscow, Russia, Procedia Computer Science 103, 2017, pp. 597 – 604.

[12] J. Chen, D. San, J. Yang, H. Chen. "Leader–Follower Formation Control of Multiple Non-holonomic Mobile Robots Incorporating a Receding-horizon Scheme," The International Journal of Robotics Research. 2010, no 6, vol. 29, pp. 727-747.