Extended Model Approach for Solving Optimal Control Problem in Class of Implemented Control Functions

Askhat Diveev[1] Elena Sofronova[2], and Artem Barabash[3]

Abstract— **It is known that solving the classical problem of optimal control leads to obtaining a control function, as a function of time, which cannot be implemented directly in the control system of a real object, since the resulting control system is open-loop one. It is proposed to use the method of the extended model of the control object. Initially, a universal system for stabilizing the movement of an object along any trajectory in the space of states from a certain class is synthesized for the object model. This stabilization system is built into the control object. The original reference model of the control object is then added to the object with a free control vector on the right side. Thus, the extended object model includes an object model with a motion stabilization system and a reference model. The optimal control problem is solved for the extended model. In the synthesis of a universal stabilization system, machine learning by symbolic regression is used. An example of solving the problem of optimal control of a wheel robot with a differential drive is given.**

I. INTRODUCTION

The implementation of the solution of the optimal control problem [1] at a real facility is one of the most important problems of control theory. It is believed that the optimal control problem is solved to obtain optimal trajectories in the state space. To realize the movement of an object along an optimal trajectory, it is necessary to solve the problem of synthesizing a system for stabilizing the movement of an object along a trajectory. It should be borne in mind that the model of the control object with the trajectory stabilization system differs from the model for which the optimal control problem was solved. This means that the conditions of the optimal control problem are violated and the optimal trajectory for a model without a stabilization system is no longer optimal for an object model with a stabilization system. The only solution here is to build a stabilization system for the control object before solving the optimal control problem. In this case, the optimal control problem is solved for an object with a stabilization system, and therefore, the solution of such a problem can be directly implemented on a real control object. Here the problem arises, how to build a stabilization system for a control object, the form of movement of which is not known in advance.

For robotic systems, very often before creating a control system, the object is made stable relative to a point in the state space [2]. Then, stability points are arranged along preset trajectory. With sequential activation of points, the control object moves from one point to the next, thereby ensuring the movement of the object along the path [3], [4]. Here it should be pointed out that the exact movement of the control object along the trajectory set in the space of states does not correspond to the exact implementation of the movement of the object along the optimal trajectory obtained as a result of solving the optimal control problem, since in order to maintain the optimality property, it is necessary to track the trajectory not only in state space, but also in time.

In synthesized optimal control, the optimal control problem [5] is solved for a control object with a stabilization system at the point of state space. In the problem, it is necessary to find point positions in the state space so, that when they are sequentially activated at regular intervals, the control object moves according to the optimal control problem from the initial specified state to the terminal state with the optimal value of the specified quality criterion. In the process of movement, the control object does not reach an active point of stable equilibrium, since at this point, according to the theory, it should stop

In synthesized optimal control, machine learning by symbolic regression is used to solve the problem of synthesizing a stabilization system at a point in state space, as a rule this is the method of a network operator [6] or any other variation method of symbolic regression [7], for example, the method of variation Cartesian programming [8]. The quality of solving the problem of optimal control by the synthesized control method depends on the nature of the stability point. Oscillatory or asymptotic stability points have different forms of optimal motion of the control object and different sensitivity to external perturbations and inaccuracies of the model. Synthesized optimal control with asymptotic points of stable equilibrium are less sensitive to external perturbations and inaccuracies.

Another method of solving the optimal control problem is the extended control object model method. Initially, for a given model of a control object, a universal system is built to stabilize the movement of the object relative to any trajectory from a certain class in an extended state space, including the time axis. Then, a reference model is included in the control object model together with the universal motion stabilization system to generate an optimal trajectory. For the extended

[1]Askhat 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

[2]Elena Sofronova is Senior 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 sofronova_ea@mail.ru

[3]Artem Barabash is post graduate student of Federal Research Center "Computer Science and Control" of the Russian Academy of Sciences, Moscow, 119333, Vavilov Str., 44/2 FRC CS RAS, Russia artew44@gmail.com

model of the control object, the problem of optimal control in the classical formulation is solved. For the reference model, a control function is found as a function of time. A reference model with control as a function of time generates optimal trajectories in an extended state space.

In this approach, the main problem is the synthesis of a universal motion stabilization system. To solve the synthesis problem, it is necessary to determine a basic solution and a set of trajectories in the space of states, relative to which the control object should be stabilized.

In the paper [9], the network operator method is used to solve the problem of synthesizing a universal stabilization system, and many trajectories were determined by solving the problem of optimal control with phase restrictions and different placement of areas of mandatory passage.

In this work, we continue to study the extended control object model method. Here we apply another new method of symbolic regression, the universal code method, and build trajectories as solutions to the optimal control problem for a group of two robots that should swap places in space with phase constraints.

The computational experiment provides examples of solving the problem of optimal control using the extended model method.

II. EXTENDED MODEL FOR SOLVING THE OPTIMAL CONTROL PROBLEM

A. The Optimal Control Problem

Consider the optimal control problem.

The mathematical of control object is given in the form of ordinary differential equations system with a free control vector in the right side.

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

where $\mathbf{x}$ is a vector of the states space, $\mathbf{x} \in \mathbb{R}^n$, $\mathbf{u}$ is a vector of control, $\mathbf{u} \in \mathrm{U} \subseteq \mathbb{R}^m$, U is a compact set, determining often restrictions on control

$$\mathbf{u}^- \leq \mathbf{u} \leq \mathbf{u}^+, \tag{2}$$

$\mathbf{u}^-$, $\mathbf{u}^+$ given vectors of lower and upper control constraints.

The initial state is given

$$\mathbf{x}^0 = [x_1^0 \dots x_n^0]^T. \tag{3}$$

The terminal state is given

$$\mathbf{x}^f = [x_1^f \dots x_n^f]^T. \tag{4}$$

The quality criterion is given

$$J_0 = \int_0^{t_f} f_0(\mathbf{x}, \mathbf{u}) dt \to \min_{\mathbf{u} \in \mathrm{U}}, \tag{5}$$

where t_f is a time of achievement of the terminal state. It isn't given, but it is limited $t_f \leq t^+$, t^+ is a given limited time of control process.

In the problem it is necessary to find a control function of time

$$\mathbf{u} = \mathbf{v}(t), \tag{6}$$

with take account of restrictions on control (2) such, that if the found control function is replaced in the right side of the differential equations system (1) instead of the free control vector, then the differential equations system

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{v}(t)), \tag{7}$$

will have the particular solution from the given initial state (3), that reaches the given terminal state (4) for time not increasing the limited time t^+, with optimal value of the quality criterion (5). To implement a solution of the optimal control problem in a real control object directly, the extended model is used.

B. The Extended Model

Firstly, the control synthesis problem is solved for the model of the control object (1) in order to provide a movement of the control object along some trajectories are generated by a reference model. At solving the control synthesis problem, the following problem statement is considered.

The mathematical model of control object is given (1).

Some programs of control functions are given

$$\mathrm{V} = \{\mathbf{v}^{*,1}(t), \dots, \mathbf{v}^{*,M}(t)\}. \tag{8}$$

The set initial points of states are given

$$\mathrm{X}_0 = \{\mathbf{x}^{0,1}, \dots, \mathbf{x}^{0,K}\}. \tag{9}$$

Any program control inserted in the right side of the model (1) instead of free control vector is generated a program trajectory as a particular solution of the differential equations system from the given initial state (3),

$$\dot{\mathbf{x}}^{*,j}(t, \mathbf{x}^0) = \mathbf{f}(\mathbf{x}(t, \mathbf{x}^0), \mathbf{v}^{*,j}(t),\ j = 1, \dots, M. \tag{10}$$

It is necessary to construct a control system of motion stabilization along all program trajectories for all initial states to minimize the following quality criterion

$$J_1 = \sum_{i=0}^{K} \sum_{j=1}^{M} \int_0^{t_i} \|\mathbf{x}^{*,j}(t) - \mathbf{x}(t, \mathbf{x}^{0,i})\| dt \to \min \tag{11}$$

where $\mathbf{x}^{*,j}(t)$ is a program trajectory (10) generated by the program control (8).

To construct the motion stabilization system a control function in the following form is used

$$\mathbf{u} = \mathbf{h}(\mathbf{x}^{*,j}(t) - \mathbf{x}(t)) \in \mathrm{U}. \tag{12}$$

To find the control function (14) symbolic regression is used.

After that the control function is found the extended model of control object is used

$$\begin{aligned} \dot{\mathbf{x}}^* &= \mathbf{f}(\mathbf{x}^*, \mathbf{u}), \\ \dot{\mathbf{x}} &= \mathbf{f}(\mathbf{x}, \mathbf{h}(\mathbf{x}^* - \mathbf{x})), \end{aligned} \tag{13}$$

where the first equation is a reference model.

Further the optimal control problem is solved for the extended model (13), (2)–(6). In that problem the control vector is found for the reference model, and the quality criterion (5) is calculated for the states space vector of the second equation from (13).

III. SYMBOLIC REGRESSION FOR SYNTHESIS OF MOTION STABILIZATION SYSTEM

To solve the control synthesis problem and to receive a control function in the form of function of the state space vector

$$\mathbf{u} = \mathbf{h}(\mathbf{x}^* - \mathbf{x}) \in \mathrm{U}, \quad (14)$$

the numerical method of symbolic regression is used.

Symbolic regression codes a mathematical expression of the desired function in the form of special code and then searches for the optimal possible solution of a mathematical expression in the form of code in the code space. Here the new symbolic regression method is presented, variation universal code.

To code a mathematical expression variation universal code uses an alphabet of elementary functions. athe alhabet includes function without arguments or arguments of desired mathematical expression

$$\mathrm{F}_0 = \{f_{0,1} = x_1, \ldots, f_{0,n} = x_n, f_{0,n+1=q_1,\ldots,f_{0,n+p}=q_p}\}. \quad (15)$$

where x_1 – x_n are variables, q_1 – q_P are parameters.

The set of function with one argument

$$\mathrm{F}_1 = \{f_{1,1}(z) = z, f_{1,2}(z), \ldots f_{1,W}(z)\}. \quad (16)$$

The set of functions with two arguments

$$\mathrm{F}_2 = \{f_{2,1}(z_1, z_2), \ldots, f_{2,V}(z_1, z_2\}. \quad (17)$$

To code a mathematical expression each elementary function is codded by integer vector of two components. the first component is a number of arguments, and the second component is the function number in the set of functions with the same of number of arguments.

All functions with one argument and arguments of a mathematical expression are coded sequence in the same order as they stay in the mathematical expression. Functions with two arguments is coded in a prefix form and them function number code sets before codes of arguments

Consider an example. Let we have a mathematical expression

$$y = \sin(q_1 x_1 + q_2) + \sqrt{x_1^2 + x_2^2}. \quad (18)$$

To code the example of mathematical expression it is enough to use the following alphabet of elementary functions

$$\mathrm{F}_0 = \{f_{0,1} = x_1, f_{0,2} = x_2, f_{0,3} = q_1, f_{0,4} = q_2\},$$

$$\mathrm{F}_1 = \{f_{1,1}(z) = z, f_{1,2}(z) = z^2, f_{1,2}(z) = \sin(z), f_{1,4}(z) = \sqrt{z}\},$$

$$\mathrm{F}_2 = \{f_{2,1}(z_1, z_2) = z_1 + z_2, f_{2,2} = z_1 \cdot z_2\}.$$

A record of the mathematical expression (18) with using elementary functions is

$$y = f_{2,1}(f_{1,3}(f_{2,1}(f_{2,2}(f_{0,3}, f_{0,1}), f_{0,4})), f_{1,4}(f_{2,1}(f_{1,2}(f_{0,1}), f_{1,2}(f_{0,2})))). \quad (19)$$

The universal code of the function(19) has the following form

$$\tilde{y} = \left(\begin{bmatrix}2\\1\end{bmatrix}, \begin{bmatrix}1\\3\end{bmatrix}, \begin{bmatrix}2\\1\end{bmatrix}, \begin{bmatrix}2\\2\end{bmatrix}, \begin{bmatrix}0\\3\end{bmatrix}, \begin{bmatrix}0\\1\end{bmatrix}, \begin{bmatrix}0\\4\end{bmatrix}, \begin{bmatrix}1\\4\end{bmatrix}, \begin{bmatrix}2\\1\end{bmatrix}, \begin{bmatrix}1\\2\end{bmatrix}, \begin{bmatrix}0\\1\end{bmatrix}, \begin{bmatrix}1\\2\end{bmatrix}, \begin{bmatrix}0\\2\end{bmatrix}\right). \quad (20)$$

Components of the integer vectors in the code (20) coincide with indexes of elementary functions in the mathematical expression (19).

To check a correct of a code the index of element is used. The index of element k is calculated by the following equation

$$T(k) = T(k-1) + a_1(k) - 1, \quad (21)$$

To check a correct of a code the index of element is used. The index of elements is calculated by the following equation where $T(k-1)$ is an index of previous element $k-1$, $a_1(k)$ is a first component of a code of elementary function, that corresponding element k.

To calculate an index of the first element we consider the index of previous or zero element is zero, $T(0) = 0$,

Any universal code is correct, if index of any its element except of the last element is bigger zero, and index of the last element is equal zero. The index of element shows, minimal number of elements must be after the current element.

The index of the element allows you to find any subexpression in the expression, if we consider any element in the expression as the first element of the subexpression, then any element after it with an index equal to zero is the end of the subexpression.

The last element of the universal code always has the first component of the element code equal to zero.

To calculate value of a mathematical expression firstly we turn on a counter of arguments $j = -1$. Then sequent go all elements from the last one to the first one, $k = N, \ldots, 1$. If the first component of an element code is zero, $a_1(k) = 0$, we increase a current value of the counter $j \leftarrow j + 1$ and consider the value of argument as a current value of the mathematical expression,

$$r_j = f_{0,a_2(k)}, \quad (22)$$

where a_2 is the argument number of the mathematical expression. If the first component of an code element is one, then we calculate value of corresponding function with one argument and this result place instead of the current value of the mathematical expression

$$r_j = f_{1,a_2(k)}(r_j). \quad (23)$$

If the first component of an code element is two, then we calculate value of corresponding function with two argument with previously current value of mathematical expression as the second argument and this result place instead of the current value of the mathematical expression, and we decrease a value of the counter

$$r_j = f_{2,a_2(k)}(r_j, r_{j-1}),\ j \leftarrow j - 1, \quad (24)$$

As a result, the arguments of a mathematical expression or function with zero quantity of arguments increase the index

j by one, and the functions with two arguments decrease the index j by one. Functions with one argument do not change the j index. In any mathematical expression encoded by the universal code method, the number of arguments of the mathematical expression must be one more than the number of functions with two arguments.

Calculations are performed correctly, if at $k = 1$, $j = 0$ then result of calculations is in r_0.

The variation universal code uses the principle of small variation of basic solution. One possible solution is coded by universal code and other possible solutions in the initial population are coded a set of small variation vectors. To increase code, we use set of codes as vector of codes. Each code has possible maximal of length L and real length $L_i \leq L$, $i = 1, \dots, M$, where M is a number of codes or a dimension of code vector. After calculation of each code its result is placed in a set of arguments and it can be used in next calculations.

A small variation vector contains five components

$$\mathbf{w} = [w_1\ w_2\ w_3\ w_4\ w_5]^T, \tag{25}$$

where w_1 is a type of variation, w_2 is the code number, w_3 is a position in the code w_2, w_4, w_5 are additional components of small variation vector.

In universal code four type variation are used. The type $w_1 = 0$ corresponds to changing of a secomd elements of the code

$$a_{w_2,2}(w_3) \leftarrow w_4. \tag{26}$$

The type $w_1 = 1$ corresponds to removing function with one or two arguments. If $a_{w_2,1}(w_3) = 1$, then the function with one argument is removed. If $a_{w_2,1}(w_3) = 2$, then the function with two arguments is removed. When deleting the function with two arguments the next subexpression is removed too, begin with element after the function element w_3.

The type $w_1 = 2$ corresponds to inserting function with one argument. If $L_{w_2} < L$, then all elements after position w_3 are moved to the right on one position, and then the code of function with one argument is inserted,

$$a_{w_2,1}(w_3) = 1,\ a_{w_2,2}(w_3) = w_4. \tag{27}$$

The type $w_1 = 3$ corresponds to inserting function with two arguments. If $L_{w_2} < L - 1$ then all elements after position w_3 are moved on two positions to the right, and then the code function with two arguments is inserted in the position w_3 and the code of argument of mathematical expression is inserted in the position $w_3 + 1$,

$$\begin{array}{rclrcl} a_{w_2,1}(w_3) & \leftarrow & 2, & a_{w_2,2}(w_3) & \leftarrow & w_4, \\ a_{w_2,1}(w_3+1) & \leftarrow & 0, & a_{w_2,2}(w_3+1) & \leftarrow & w_5. \end{array} \tag{28}$$

Every possible solution except the basic solution is coded by a set of small variation vectors.

$$\mathrm{W}_i = (\mathbf{w}^{i,1}, \dots, \mathbf{w}^{i,d}), \tag{29}$$

where d is a number of small variation vectors in one a set or the depth of variations of the basic solution.

To find the optimal solution variation genetic algorithm is used. Crossover and mutation operation are performed on sets of small variation vectors. After some generations the basic solution is changed on the best found current possible solution. This process is named change of epoch.

IV. COMPUTATIONAL EXPERIMENT

Consider the optimal control problem for group of two robots with differential drive. The mathematical model of control objects has the following form

$$\begin{array}{rcl} \dot{x}_1 & = & 0.5(u_1 + u_2)\cos(x_3), \\ \dot{x}_2 & = & 0.5(u_1 + u_2)\sin(x_3), \\ \dot{x}_3 & = & 0.5(u_1 - u_2), \\ \dot{x}_4 & = & 0.5(u_3 + u_4)\cos(x_6), \\ \dot{x}_5 & = & 0.5(u_3 + u_4)\sin(x_6), \\ \dot{x}_6 & = & 0.5(u_3 - u_4), \end{array} \tag{30}$$

where $\mathbf{x}^{(1)} = [x_1\ x_2\ x_3]^T$ is a vector of the state space of the first robot, $\mathbf{x}^{(2)} = [x_4\ x_5\ x_6]^T$ is a vector of the state space of the second robot, $\mathbf{u}^{(1)} = [u_1\ u_2]^T$ is a control vector of the first robot, $\mathbf{u}^{(2)} = [u_3\ u_4]^T$ is a control vector of the second robot.

Control of robots are restricted

$$-10 = u^- \leq u_i \leq u^+ = 10,\ i = 1, 2, 3, 4. \tag{31}$$

The initial state is given

$$\mathbf{x}^0 = [0\ 0\ 0\ 10\ 10\ \pi]^T. \tag{32}$$

The terminal state is given

$$\mathbf{x}^f = [10\ 10\ 0\ 0\ 0\ \pi]^T. \tag{33}$$

The phase constraints are given

$$\begin{array}{c} \varphi_i(\mathbf{x}^{(j)}) = R_i - \\ \sqrt{(x_{1,i} - x_{1+3(j-1)})^2 + (x_{2,i} - x_{2+3(j-1)})^2} \leq 0, \end{array} \tag{34}$$

where $i = 1, 2$, $j = 1, 2$, $R_1 = R_2 = 2$, $x_{1,1} = 5$, $x_{2,1} = 2$, $x_{1,2} = 5$, $x_{2,2} = 8$.

The condition of avoiding collision is given

$$\chi(\mathbf{x}) = R_0 - \sqrt{(x_1 - x_4)^2 + (x_2 - x_5)^2} \leq 0, \tag{35}$$

where $R_0 = 1$.

The quality criterion is given. For numerical solution this criterion can include a checking the phase constraints and achievement to the terminal state.

$$J = t_f + p_1 \sum_{i=1}^{2} \sum_{j=1}^{2} \int_0^{t_f} \Big(\vartheta(\chi(\mathbf{x})) + \vartheta(\varphi_i(\mathbf{x}^{(j)})) \Big) dt +$$

$$p_2 \|\mathbf{x}^f - \mathbf{x}(t_f)\| \to \min_{\mathbf{u}} \tag{36}$$

where $p_1 = 4$, $p_2 = 1$, $t_f = max\{t_{f,1}, t_{f,2}\}$, $t_{f,j}$ is a time of achievement of the terminal state of robot j,

$$\vartheta(\alpha) = \begin{cases} 1, \text{ if } \alpha > 0 \\ 0, \text{ otherwise} \end{cases}. \tag{37}$$

To solve the optimal control problem in the class of directly implementing control functions, the extended model is used. Initially, we synthesize a universal stabilization system. For this purpose, we use the method of symbolic regression, variational universal code. As a result, the following control function was found

$$u_{i+2(j-1)} = \begin{cases} u^+, \text{ if } u^+ \leq \tilde{u}_{i+2(j-1)} \\ u^-, \text{ if } \tilde{u}_{i+2(j-1)} \leq u^- \\ \tilde{u}_{i+2(j-1)}, \text{ otherwise} \end{cases}, \quad (38)$$

where $i = 1, 2,\ j = 1, 2,$

$$u_{1+\tilde{2(j-1)}} = \mu(r_{10}) + r_9 + \rho_{19}(r_8) + \rho_{17}(r_5) + \text{sgn}(\rho_{17}(r_4)) + r_3^2 + \exp(r_2) + \vartheta(q_2\Delta_2\text{sgn}(\Delta_1)) + r_2^2 - \Delta_2, \quad (39)$$

$$u_{2+\tilde{2(j-1)}} = \vartheta(r_{11}) + \text{sgn}(r_{10})\sqrt{|r_{10}|} + \rho_{19}(r_8) - r_7 + (q_3)^{-1} + \vartheta(\Delta_1), \quad (40)$$

$$r_{10} = r_9 - r_9^3 + r_6^2 + \text{sgn}(r_4) - r_2 + \ln(|q_3\exp(q_2)|)\Delta_3\cos(\Delta_1),$$

$$r_9 = r_7 + q_1 + \cos(r_7) + \sin(q_3\exp(q_2)) + \arctan(q_2\Delta_2\text{sgn}(\Delta_1)\Delta_1),$$

$$r_8 = \rho_{18}(r_6) + (r_5q_2r_6\Delta_2\text{sgn}(\Delta_1 + \tanh(\Delta_3)))^{-1},$$

$$r_7 = \vartheta(r_4)\exp(q_2\text{sgn}(\Delta_1)\rho_{17}(q_1)),$$

$$r_6 = \rho_{19}(r_4) + r_5 + (q_3)^{-1} + \rho_{18}(\Delta_3),$$

$$r_5 = r_4^2 - r_3\text{sgn}(r_1)\rho_{19}(q_3)\rho_{18}(\Delta_3)\text{sgn}(\Delta_2)\sqrt{|\Delta_2|},$$

$$r_4 = q_3(r_2 - q_2),$$

$$r_3 = 2\Delta_1 + \mu(\Delta_3^3) + \Delta_2^3,$$

$$r_2 = q_3\exp(\Delta_3)q_2 + \Delta_2\Delta_1,$$

$$r_1 = q_3(q_1\Delta_1 + \sin(q_1) + \Delta_2^3),$$

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

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

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

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

$$\Delta_k = x^*_{k+3(j-1)}(t) - x_{k+3(j-1)}(t).\ k = 1, 2, 3,\ j = 1, 2,$$

$\mathbf{x}^*(t)$ is an optimal trajectory, $q_1 = 13.84326$, $q_2 = 2.27808$, $q_3 = 1.24121$.

Subsequently, a universal stabilization system (38) was inserted into the extended model along with the model. (30), which for the extended model was the reference. In total, the extended model included twelve equations.

The projections of the robots paths onto a horizontal plane from eight perturbed initial states are shown in Figures 1 for the first robot and Figure 2 for the second robot. They are also given there. optimal trajectories (points).

As can be seen from the results of experiments, universal stabilization, constructed automatically by the method of symbolic regression, ensures the movement of both robots in the vicinity of given program trajectories. The trajectories of robots from perturbed initial states approach the optimal trajectory and aim to reach the final state. The deviation error from the given trajectory decreases over time. As a result of embedding the equations of the universal motion stabilization system along the trajectory from a certain class into the right side of the model, it provides such qualitative changes in the solutions of the differential equations of the system model that a special solution appears in the form of a program trajectory that has areas of attraction of other solutions.

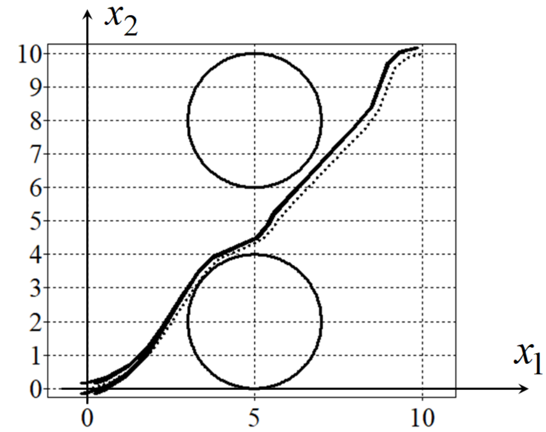

Fig. 1. Projections of the trajectories of the first robot onto a horizontal plane from eight perturbed initial states. Optimal trajectory (points)

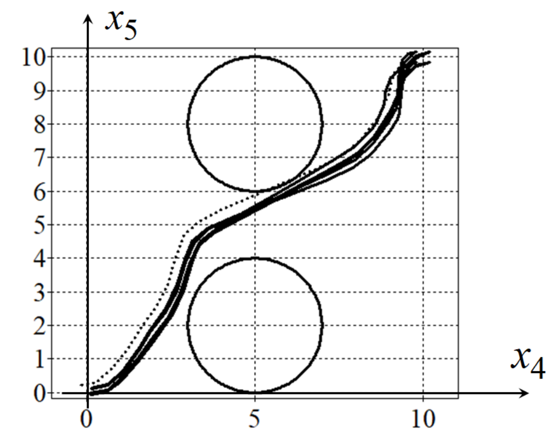

Fig. 2. Projections of the trajectories of the second robot onto a horizontal plane from eight perturbed initial states. Optimal trajectory (points)

Since small initial state perturbations practically do not change the solutions of the system model, it is obvious that these solutions can be directly implemented in a real control object.

To check the quality of the universal stabilization system (38) with the same extended model, another problem of optimal control was solved. In the new problem, phase constraints of a different size and another located were considered, $R_1 = R_2 = 2.2$, $x_{1,1} = 2$, $x_{2,1} = 6$, $x_{1,2} = 8$, $x_{2,2} = 4$. Robots started from other initial states, also moved towards each other to change places. Results of experiments are presented in the figures 3 and 4.

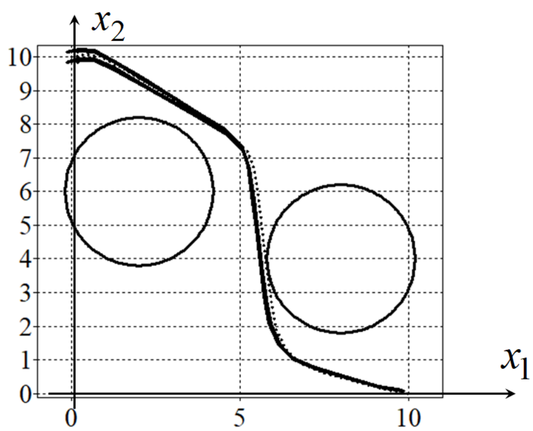

Fig. 3. Projections of the trajectories of the first robot onto a horizontal plane from eight perturbed initial states in the second task.

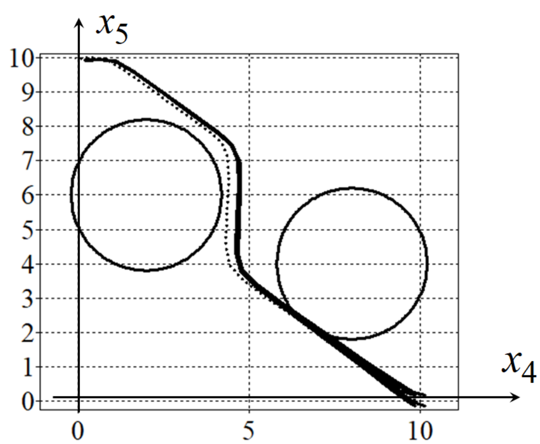

Fig. 4. Projections of the trajectories of the second robot onto a horizontal plane from eight perturbed initial states in the second task

As can be seen from the graphs of the results of experiments in the new optimal control problem, it is also possible to stabilize the movement of robots in the vicinity of other optimal trajectories. This means that embedding the mathematical equations for the control function of the universal stabilization system in the right side of the system of the control object model under consideration provides a qualitative change in the solutions of the system so that the optimal trajectory becomes a special solution and has an area of attraction in its neighborhood.

The results of experiments showed that universal stabilization ensures high-quality movement of the object near the optimal trajectory, while the same motion stabilization system can be used for various problems of optimal control of this object. The initial state perturbations in the experiments amounted to ± 0.15 of all state vector components.

The optimal control problem requires much less computational time than the control synthesis problem. As a result of using a universal motion stabilization system, the mathematical expression of which is at the design stage of the control system, it is possible to solve the problems of optimal control directly during operation of the control object with a sufficiently high-quality on-board processor.

V. CONCLUSIONS

The paper presents a method of an extended model for solving the problem of optimal control in a class of implemented functions. The method initially solves the problem of synthesizing a system for stabilizing the movement of an object in the vicinity of a given trajectory. Symbolic reshgression is used to solve the synthesis problem. This paper presents a new method of symbolic regression, the variational universal code. The method includes two well-known symbolic regression methods, Cartesian genetic programming and genetic programming. For the synthesis of a universal motion stabilization system, a multiplicity of trajectories and a set of initial states near to given trajectories are given.‘

The extended model is a control object model with a system for stabilising movement along a given trajectory from a plurality of trajectories of a certain class and a reference model for generating an optimal trajectory. As a result, the dimension of the extended model of the control object is twice the dimension of the model. The paper presents the solution of two problems of optimal control with static and dynamic phase restrictions for a group of two robots with differential drive. Disturbances in the initial states of robots with a universal stabilization system did not significantly change the trajectory of movement. Therefore, the solution of the optimal control problem for robots is feasible in real objects

REFERENCES

[1] L.S. Pontryagin, V.G. Boltyanskii, R.V. Gamkrelidze, E.F. Mishchenko, The Mathematical Theory of Optimal Process, Gordon and Breach Science Publishers, New York, London, Paris, Montreux, Tokyo: 1985. L.S. Pontryagin, Selected works, Vol. 4. 360 p.

[2] M. Egerstedt, Motion Planning and Control of Mobile Robots. Ph.D. Thesis, Royal Institute of Technology, Stockholm, Sweden, 2000.

[3] G. Walsh, D. Tilbury, S. Sastry, R. Murray, J.P. Laumond, Stabilization of trajectories for systems with nonholonomic constraints. IEEE Trans. Autom. Control 1994, 39, pp. 216–222.

[4] A. Samir, A. Hammad, A. Hafez, H. Mansour, Quadcopter Trajectory Tracking Control using State-Feedback Control with Integral Action. Int. J. Comput. Appl. 2017, 168, pp. 1–7.

[5] A.I. Diveev, E.Yu. Shmalko, V.V. Serebrenny, P. Zentay, Fundamentals of Synthesized Optimal Control. Mathematics 2021, 9, 21.

[6] A. Diveev, E. Shmalko, Stabilization System Quality of Synthesized Optimal Control. Mathematics 2024, 1,0.

[7] A. Diveev, E. Shmalko, Machine Learning Control by Symbolic Regression. Springer International Publishing, Cham, 2021.

[8] A. Diveev, E. Shmalko, Machine-Made Synthesis of Stabilization System by Modified Cartesian Genetic Programming// IEEE Transactions on Cybernetics, 2022, 52(7), pp. 6627–6637.

[9] A. Diveev, E. Sofronova, Universal Stabilisation System for Control Object Motion along the Optimal Trajectory. Mathematics 2023, 11, 3556.