mathematics

Article

Universal Stabilisation System for Control Object Motion along the Optimal Trajectory

Askhat Diveev † and Elena Sofronova *,†

Federal Research Center "Computer Science and Control" of the Russian Academy of Sciences, 44, build. 2, Vavilova Str., Moscow 119333, Russia; aidiveev@mail.ru
* Correspondence: sofronova_ea@mail.ru
† These authors contributed equally to this work.

Abstract: An attempt to construct a universal stabilisation system that ensures the object motion along specified trajectory from certain class is presented. If such a stabilisation system is constructed, then only the problem of optimal control is solved, but for a model of the object, which includes a stabilisation system and a subsystem with a reference model for generating a specified trajectory. In this case, the desired control is the control in the reference model. Statement of complete optimal control problem includes two problems, optimal control problem and stabilisation system synthesis problem for motion along given trajectory in the state space. Numerical methods for solving these problems based on evolutionary computation and symbolic regression are described. It is shown that when solving the stabilisation system synthesis problem, it is possible to obtain a universal system that provides stabilisation of the object motion relative to any trajectory from a certain class. Therefore, it is advisable to formulate an optimal control problem for an object with a motion stabilisation system. A computational example of solving the problem for the spatial motion of a quadrocopter is given.

Keywords: optimal control; control synthesis; stabilisation system; evolutionary algorithm; symbolic regression

MSC: 49M25; 68W50

check for updates

Citation: Diveev, A.; Sofronova, E. Universal Stabilisation System for Control Object Motion along the Optimal Trajectory. *Mathematics* **2023**, *11*, 3556. https://doi.org/10.3390/math11163556

Academic Editor: Liliya Demidova

Received: 15 July 2023
Revised: 1 August 2023
Accepted: 14 August 2023
Published: 17 August 2023

Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

When solving the optimal control problem in the classical Pontryagin statement [1], a solution is obtained in the form of a time-dependent control function. Such a solution cannot be directly implemented in the real control object, since it leads to the construction of an open-loop control system that is not sensitive to the real current position of the control object. The control system without feedback is sensitive to small perturbations. The control object with an open-loop control system cannot reach the terminal state with a given accuracy and provide the optimum value of the given quality criterion. Therefore, in most cases, the optimal control problem is considered as an initial problem for obtaining an optimal program control and an optimal program trajectory. In order to implement the obtained solution in a real control object, it is necessary to further solve the problem of stabilisation system synthesis for the control object motion along the obtained optimal program trajectory.

Two things should be noted here. First, solving the control synthesis problem is often no less complicated than solving the original optimal control problem, because in the synthesis problem, the control function is sought as a function of the state vector. Second, solving the control synthesis problem changes the dynamics of the control object. The mathematical model of the control object in the form of differential equations contains the control function in the right side, so the optimal control found for the original mathematical model may no longer be optimal for the mathematical model of the object with a motion stabilisation system. In any case, a system that provides feedback control is necessary to implement the solution of the optimal control problem, although this does not follow from the most classical statement of the optimal control problem.

The statement of the control problem, where it is necessary to find the control function of the state vector, was also proposed by R. Bellman a long time ago [2]. To solve this problem, a dynamic programming method was developed that allows numerically find the value of the control vector for each value of the state vector. If we solve the problem in the Bellman statement using the dynamic programming method for an initial state, as in the Pontryagin statement, the resulting solution will also be sensitive to changes in the initial conditions, and the dependence of the control function on the state vector will not provide an adequate response to disturbances. Solving the Bellman optimal control problem by dynamic programming for a set of initial states faces the problem of the curse of dimensionality.

In [3], a refined statement of the optimal control problem is formulated. The classical statement is complemented by the requirement that the resulting optimal trajectory has a non-empty neighbourhood with attractor properties. It means that the resulting optimal control should ideally be a special solution of the differential equation, an attractor. To achieve this, the control is first sought as a function of time and state, and to implement the stabilisation system, the initial state is replaced by the initial state domain.

Ensuring additional requirements can be obtained in various ways, for example, by reformulating the optimal control problem into a problem of general control synthesis for a given area of initial states [3]; then, each particular solution from a given area provides the optimal value of a given quality criterion. Such a task is computationally difficult. Another approach is to solve the control synthesis problem in order to ensure stability with respect to the terminal state, but in this case, we do not guarantee that the optimal value of the given quality criterion can be obtained.

To stabilise the movement of the control object along the optimal trajectory, theoretical works [4] propose linearising the model relative to the trajectory and obtaining a linear non-stationary model of the object. For the stability of such an object, linear feedback is proposed. In general, stability for a non-stationary object is not an unsolved problem. In most practical works, points are set on the tracked trajectory and the control object is made stable relative to these points. PI and PID controllers are used for this purpose [5–8]. Movement on stable trajectory points slows down the movement of the object near the stability point, so the optimum value of the criterion is not maintained. In the work [9], a system for tracking the trajectory of a quadcopter is built based on stabilization of the quadcopter speed in the horizontal plane along a straight line. For this purpose, a proportional regulator is built based on the Lyapunov function. Movement along the trajectory of straight segments with a constant speed is not optimal. It is necessary to track the trajectory not only in space but also in time.

In [10], an approach to solving the optimal control problem by the synthesized control method is considered. According to this approach, first the synthesis problem to ensure the stability of the object relative to the equilibrium point in the state space is solved, and the optimal control problem in the original statement is solved at the second stage, where the optimal positions of the stable equilibrium points are found. Optimal control is achieved by changing the positions of stable equilibrium points after a given time interval. Synthesised control is a universal approach to solving the optimal control problem in the class of feasible systems, but in any particular case it may have several solutions. Each of the solutions may be differently sensitive to disturbances of the initial conditions. The practical advantage of the synthesized control is that the synthesis problem of ensuring the stability of the equilibrium point is solved at a preliminary stage, i.e., at the stage of creating a control system, and the solution of the optimal control problem by choosing the position of the equilibrium points can be solved on the on-board computer for a specific current situation. Solving the problem of synthesising a control system on an on-board computer is usually complicated due to the high computational cost.

The problem of synthesis of motion stabilisation along the optimal trajectory was first considered in [11], where the method of symbolic regression was applied to solve the synthesis problem. Before that, the symbolic regression method was used to solve the problem of general control synthesis, without solving the optimal control problem. As a result, we obtain a control system that includes a reference model for generating the optimal trajectory in time. Studies have shown that the stabilisation system depends on the type of optimal trajectory. In practice, it means that the use of such a control system is difficult, that is, when the situation changes and a new optimal control problem arises, it is necessary not only to solve the optimal control problem, but to re-solve the stabilisation system synthesis problem, which is unacceptable for the on-board computer.

In contrast to the previous works, in this paper, it is proposed to use a universal stabilisation system. The universality of the stabilisation system is that one system of motion stabilisation for a particular object is obtained for different types of trajectories. The obtained stabilisation system can be used for other trajectories as well. There may be trajectories for which this stabilisation system is not suitable, but this requires additional research, which is currently underway. The stabilisation system should ensure the stabilisation of control object motion along the given optimal trajectories from a certain class. For this purpose, first the synthesis problem is solved for one stabilisation system for several given trajectories. The class of trajectories is considered as a training set, and the stabilisation system synthesis is the learning of control system for a given training set. Next, this stabilisation system is applied to the motion along the trajectory that was not included in the training set. To solve the stabilisation system synthesis problem, methods of symbolic regression are applied [12,13].

The rest of the paper is organized as follows. The statements of optimal control problems and the synthesis of a motion stabilisation system along a given trajectory, as well as a new optimal control problem for an object that includes a reference model and a motion stabilisation system relative to the trajectory obtained using the reference model in the control system, are presented in Section 2. Such stabilisation system is called universal. Next, the network operator method, one of methods of symbolic regression, is described in Section 3. An example of the universal stabilisation system synthesis for the spatial movement of a quadcopter and the use of this system when the quadcopter is moving along a complex trajectory is given in Section 4. Computational experiments are followed by a conclusion in Section 5 and a discussion in Section 6, respectively.

2. Statement of Complete Optimal Control Problem

We consider some statements of the optimal control problem that make it complete in terms of implementing of the control problem solution in a real object. To implement a solution, it is necessary to obtain a closed-loop control system, so that a found control function depends on the state space vector.

The classical statement of the optimal control problem does not allow obtaining a closed-loop control system, so a found control function depends only on time. In the following, a classical statement of the optimal control problem is considered.

2.1. Optimal Control Problem

The mathematical model of a control object is given in the form of an ordinary differential equation system

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

where $\mathbf{x}$ is a state vector of control object, $\mathbf{x} \in \mathbb{R}^n$, $\mathbf{x} = [x_1 \dots x_n]^T$, $\mathbf{u}$ is a control vector, $\mathbf{u} \in \mathrm{U} \subseteq \mathbb{R}^m$, U is a compact set that defines control constraints. For example, values of control vector components can be bounded from above and below

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

$\mathbf{u}^- = [u_1^- \dots u_m^-]^T$, $\mathbf{u} = [u_1 \dots u_m]^T$, $\mathbf{u}^+ = [u_1^+ \dots u_m^+]^T$.

For System (1), the initial state is given:

$$\mathbf{x}(0) = \mathbf{x}^0. \tag{3}$$

The terminal state is given as

$$\mathbf{x}(t_f) = \mathbf{x}^f, \tag{4}$$

where t_f is a terminal time of achievement of the terminal state. The terminal time is not specified, but it is limited; $t_f \leqslant t^+$, t^+ is a given time limit.

The quality criterion is given in the common integral form

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

When solving the problem by direct numerical approach, the terminal state (4) is reached with a certain accuracy, which is included in the quality criterion (5). Therefore, the quality criterion for the numerical solution of the problem has the following form:

$$J_1 = \int_0^{t_f} f_0(\mathbf{x}, \mathbf{u})dt + p_1\|\mathbf{x}^f - \mathbf{x}(t_f)\| \to \min_{\mathbf{u}\in \mathrm{U}}, \tag{6}$$

where p_1 is a weight coefficient;

$$t_f = \begin{cases} t, \text{ if } t < t^+ \text{and } \|\mathbf{x}^f - \mathbf{x}(t)\| \leqslant \varepsilon_1 \\ t^+, \text{ otherwise} \end{cases}, \tag{7}$$

where ε_1 is an accuracy of achievement of the terminal state (4).

In the classical optimal control problem, a control function is sought as a function of time:

$$\mathbf{u} = \mathbf{v}(t) \in \mathrm{U}. \tag{8}$$

To implement the solution of the optimal control problem, it is necessary to synthesise the stabilisation system of a motion along the obtained optimal trajectory. This stabilisation system should change the mathematical model of the control object in such a way that the optimal trajectory in the state space acquires a non-zero neighbourhood with attractor properties. In the statement of the optimal control problem, either these requirements should be included and then the implementation of this property is conducted at the discretion of the control system designer, or it is necessary to add a statement of the stabilisation system synthesis problem of the motion along the optimal trajectory, and then solve these two problems together sequentially.

Thus, the control synthesis problem for stabilising the motion along the optimal trajectory is described as follows.

2.2. Stabilisation System Synthesis

The same mathematical model of control object as in the optimal control problem (1) is used.

Instead of the one initial state (3), a domain of initial states is specified. For a numerical solution, the initial state domain is given in the form of a finite set of points

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

The terminal state is given (4).

The optimal trajectory is a time function

$$\mathbf{x}^*(t),\ t \in (0; t_f). \tag{10}$$

It is necessary to find the optimal control as a function of the deviation of the state space vector from the optimal trajectory

$$\mathbf{u} = \mathbf{h}(\mathbf{x}^* - \mathbf{x}). \tag{11}$$

If the control function (11) is placed in the right part of ODE system (1), then the following system is obtained:

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{h}(\mathbf{x}^* - \mathbf{x})). \tag{12}$$

The control function (11) should minimize the sum of maximum deviations of all particular solutions of the system (12) from initial states of the given domain (9)

$$J_2 = \sum_{i=1}^{K} \left(\max_{t \in (0;t_{f,i})} \|\mathbf{x}^* - \mathbf{x}(t, \mathbf{x}^{0,i})\| + p_1 \|\mathbf{x}^f - \mathbf{x}(t_{f,i})\| \right) \to \min_{\mathbf{u} \in \mathrm{U}}, \tag{13}$$

where $t_{f,i}$ is a time of achievement the terminal state of the particular solution from the initial state $\mathbf{x}^{0,i}$ defined by Equation (7), $\mathbf{x}(t, \mathbf{x}^{0,i})$ is a particular solution of the system (12) from the initial state $\mathbf{x}^{0,i}$.

There is an ambiguity in this problem statement: How to obtain the value of the optimal trajectory (10) at a given time? When solving the optimal control problem in the first stage, the optimal control function is sought as a function of time, but not an optimal trajectory. When searching by a direct approach, the optimal control function is usually approximated by a piece-wise continuous function. The result is an analytical mathematical expression for the control function. The optimal trajectory in the general case has no mathematical expression and it is obtained numerically after simulation of the control object model (1) with the optimal control function. To obtain the optimal trajectory and use it to solve the stabilisation system synthesis problem in the second stage, the results of the simulation of the control object model (1) with the optimal control function can be kept as an array of time points and values of the state space vector at this moment. Another way is to simulate the control object model (1) with the optimal control function together with the control object model used for the synthesis of the stabilisation system. Then, instead of the control object model (1) and the optimal trajectory (10) in the statement of the stabilisation system synthesis problem, a model with two subsystems is used:

$$\begin{aligned} \dot{\mathbf{x}} &= \mathbf{f}(\mathbf{x}, \mathbf{u}), \\ \dot{\mathbf{x}}^* &= \mathbf{f}(\mathbf{x}^*, \mathbf{u}^*). \end{aligned} \tag{14}$$

The first subsystem is the mathematical model of the control object with a sought-after control function for stabilising the movement relative to the optimal trajectory. The second subsystem is the reference model that generates the optimal trajectory.

It should be noted that the stabilisation system changes the dynamics of the control object and the optimal control for the control object without a stabilisation system can be non-optimal for the control object with a stabilisation system. In order to solve the synthesis problem of the stabilisation system, a machine learning control by symbolic regression is used. The stabilisation system in a common case cannot provide stabilisation for any trajectory, and therefore it should be synthesised for each new optimal trajectory.

2.3. Optimal Control Problem for Object with Motion Stabilisation System

In this work, a universal stabilisation system synthesis problem is considered. This universal stabilisation system provides motion of a control object along any trajectory from some class. To solve the universal stabilisation system synthesis problem, machine learning control by symbolic regression is performed for the same model of the control object and for some given trajectories at the same time. Suppose such a universal stabilisation system is obtained. If it is known that an optimal trajectory belongs to the class of trajectories stabilised by the universal stabilisation system, then the optimal control problem can be solved for the control object with a stabilisation system.

The problem statement of the optimal control problem for a control object with a universal stabilisation system of movement along the trajectory has the following description.

The mathematical model of the control object with a universal stabilisation system and with a reference model for the generation of the program trajectory is given as

$$\begin{aligned} \dot{\mathbf{x}} &= \mathbf{f}(\mathbf{x}, \mathbf{h}(\mathbf{x}^* - \mathbf{x})), \\ \dot{\mathbf{x}}^* &= \mathbf{f}(\mathbf{x}^*, \mathbf{u}), \end{aligned} \tag{15}$$

where the function of stabilisation system $\mathbf{h}(\mathbf{x}^* - \mathbf{x})$ satisfies the constraints on control for any values of its arguments

$$\mathbf{h}(\mathbf{x}^* - \mathbf{x}) \in \mathrm{U}. \tag{16}$$

The initial state is given in (3). The terminal state is given in (4). The quality criterion is as follows:

$$J_3 = \int_0^{t_f} f_0(\mathbf{x}, \mathbf{h}(\mathbf{x}^* - \mathbf{x}))dt + p_1\|\mathbf{x}^f - \mathbf{x}(t_f)\| \to \min_{\mathbf{u}\in\mathrm{U}}. \tag{17}$$

It is necessary to find a control function as a time function

$$\mathbf{u} = \mathbf{v}^*(t) \in \mathrm{U}, \tag{18}$$

with the constraint (16) that the particular solution of the system (15) from the given initial state (3) reaches the given terminal state (4) with the optimal value of the given criterion (17).

In this problem, a control function is sought as a function of time and a closed-loop system with feedback control is obtained.

3. Symbolic Regression for Solving the Control Synthesis Problem

Symbolic regression is a unique computational technique that allows finding the mathematical expressions of the desired functions. Note that artificial neural networks can also approximate any function, but they do not find the structure of the function. The structure of an artificial neural network is determined by the type of network. It can vary regularly over a certain range by changing the number of layers and the number of neurons in each layer.

The universal approximation of functions by an artificial neural network is provided by a large number of parameters. The determination of parameter values as a result of network training allows obtaining the required values of the desired function for the entire set of specified values of its arguments. The technique of using an artificial neural network is similar to the technique of digging a hole using only a shovel. Obviously, any hole can be dug with a shovel if a large number of workers is used; in the neural network, these are parameters. But why is it impossible to use more advanced mechanisms, such as excavators, in approximation problems? These are nonlinear transformations. They should not be used just because they are difficult to use and require qualification.

Note that nonlinear effects are characteristic of physical and natural phenomena. Many models of physical processes are nonlinear. Only for nonlinear differential equations it is possible to obtain a stable limit cycle or an attractor property for a manifold of non-zero dimension.

There are problems that are difficult to solve with an artificial neural network. One example is the control synthesis problem. In this problem, it is necessary to find a function that is part of the mathematical model of the control object. Each new function changes the dynamic properties of the object, so we cannot define the behaviour of the object with the optimal control function in advance since the form of the desired function is unknown at the time of the search. The lack of a training example complicates the application of an artificial neural network. The function should be searched only by the value of the quality criterion. Paper [14] explicitly states that the future of artificial intelligence is not connected with incomprehensible neural networks, but with quite clear symbolic regression.

Here, the symbolic regression method is used to solve the control synthesis problem. All symbolic regression methods encode mathematical expressions in the form of special codes. To encode a mathematical expression, first, the alphabet of elementary functions is defined. The search for a mathematical expression is performed by a special genetic algorithm on the space of codes of mathematical expressions. In a special genetic algorithm, the crossover operation is performed taking into account the code of the mathematical expression so that after the crossover operation, two correct codes of new mathematical expressions are obtained.

The best known symbolic regression method is genetic programming (GP) [15]. GP encodes a mathematical expression in the form of a computational tree. On the leaves of the tree are the arguments of the mathematical expression. Each node of the tree represents an elementary function. The number of branches leaving the node is equal to the number of arguments of the elementary function. The crossover operation in GP involves randomly selecting nodes in the parent trees and swapping the subtrees originating from those nodes. GP is not the most convenient method of symbolic regression because after the crossover operation the codes of mathematical expressions change length and the number of identical arguments of a mathematical expression should be equal to the number of occurrences of that argument in the desired mathematical expression. GP has also been applied to solve control problems [16,17].

There are now about twenty symbolic regression methods. In this paper, one of them, the network operator method (NOP) [12], is used to solve the control synthesis problem. NOP uses only functions with one or two arguments in the alphabet of elementary functions. It encodes a mathematical expression in the form of a directed graph. Source nodes of the graph are associated with the arguments of the mathematical expression. Other nodes of the graph are associated with the functions of two arguments. The edges of the graph are associated with functions of one argument.

We consider an example of encoding a mathematical expression by the network operator method. A mathematical expression is given:

$$y = a\exp(-bx_1)(\sin(cx_2) + \cos(dx_2)), \tag{19}$$

where a, b, c, d are constant parameters, x_1, x_2 are variables. Parameters and variables are arguments of mathematical expression (19).

To encode the mathematical expression, we use the following functions:

(1) functions with one argument:

$$\begin{aligned} \mathrm{F}_1 \quad = \quad & \{f_{1,1}(z) = z, f_{1,2}(z) = -z, f_{1,3}(z) = \exp(z), \\ & f_{1,4}(z) = \sin(z), f_{1,5}(z) = \cos(z)\}; \end{aligned} \tag{20}$$

(2) functions with two arguments:

$$\mathrm{F}_2 = \{f_{2,1}(z_1, z_2) = z_1 + z_2, f_{2,2}(z_1, z_2) = z_1 \cdot z_2\}. \tag{21}$$

Functions with two arguments should be commutative, associative and have a unit element,

$$f_{2,i}(e_i, z) = f_{2,i}(z, e_i) = z,$$

where e_i is a unit element of function $f_{2,i}(z_1, z_2)$.

Figure 1 shows the directed graph of NOP for mathematical expression (19).

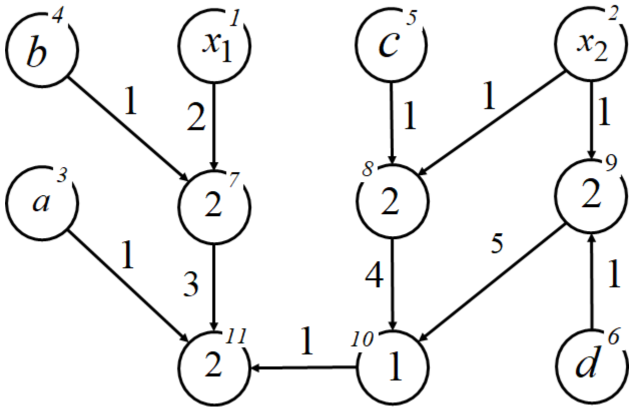

Figure 1. The network operator graph for the mathematical expression.

In the network operator graph, the arguments of the mathematical expression are shown in the source node. The numbers of functions with two arguments are shown in other nodes. Numbers of functions with one argument are displayed next to the edges. The nodes are indexed in their upper parts. If the node indices are sorted such that the index of node where the edge comes out is of a lesser value than that of the index of nodes where the edge comes in, then the network operator matrix is upper triangular.

In PC memory, the network operator is presented as an integer matrix, that has a structure like that of the network operator graph adjacency matrix. The network operator matrix for the graph in Figure 1 has the following form:

$$\mathbf{\Psi} = [\psi_{i,j}] = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 2 & 0 & 0 & 0 & 3 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 & 0 & 4 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 & 5 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 \end{bmatrix}, \; i,j = 1,\ldots,L = 11, \quad (22)$$

where L is a number of nodes in the network operator graph, $\dim(\mathbf{\Psi}) = L \times L$.

In the network operator matrix, lines with zeros on the main diagonal are linked with source-nodes. Other non-zero elements on the main diagonal are the function numbers with two arguments. Non-zero elements above the main diagonal $\psi_{i,j} \neq 0$ are function numbers with one argument.

For calculation values of the mathematical expression by network operator, a vector of nodes is defined. Initially, the vector of nodes consists of mathematical expression arguments and unit elements of the corresponding functions with two arguments.

For considered mathematical expression, the initial vector of nodes has the following form:

$$\mathbf{z}^{(0)} = [x_1\ x_2\ a\ b\ c\ d\ 1\ 1\ 1\ 0\ 1]^T, \quad (23)$$

where 1 is a unit element for function of multiplication $f_{2,2}(z_1, z_2) = z_1 \cdot z_2$, 0 is a unit element for function of summary $f_{2,1}(z_1, z_2) = z_1 + z_2$.

The calculation of mathematical expression is performed by the following equation:

$$z_j^{(i)} \leftarrow \begin{cases} f_{2,\psi_{j,j}}(z_j^{(i-1)}, f_{1,\psi_{i,j}}(z_i^{i-1})), \text{ if } \psi_{i,j} \neq 0 \\ z_j^{(i-1)}, \text{ otherwise} \end{cases}, \; i = 1,\ldots,L-1,\ j = i+1,\ldots,L. \quad (24)$$

To find an optimal mathematical expression by the network operator method, the variation genetic algorithm is used. This algorithm implements the principle small variations of basic solution [18]. According to this principle, small variations of code are defined and only one basic solution is encoded. Other possible solutions are coded as sets of small variations. A code of small variation is an integer vector of four components,

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

where w_1 is a type of small variation, w_2 is the line number, w_3 is the column number, $w_3 \geq w_2$, w_4 is a new value of a network operator matrix element.

In the network operator, four types of small variations are used: $w_1 = 0$ is an exchange of the function with one argument: if $\psi_{w_2,w_3} \neq 0$, then $\psi_{w_2,w_3} \leftarrow w_4$; $w_1 = 1$ is an exchange of the function with two arguments: if $\psi_{w_2,w_2} \neq 0$, then $\psi_{w_2,w_2} \leftarrow w_4$; $w_1 = 2$ is an insertion of the additional function with one argument: if $\psi_{w_2,w_3} = 0$, then $\psi_{w_2,w_3} \leftarrow w_4$; $w_1 = 3$ is an elimination of the function with one argument: if $\psi_{w_2,w_3} \neq 0$ and $\exists \psi_{w_2,j} \neq 0$, $j > w_2$, $j \neq w_3$ and $\exists \psi_{i,w_3} \neq 0$, $i \neq w_2$, then $\psi_{w_2,w_3} \leftarrow 0$.

In the search algorithm, a population of possible solutions is used. Any possible solution other than the basic solution is encoded as a set of small variation vectors

$$\mathrm{W}_i = \{\mathbf{w}^{1,1}, \ldots, \mathbf{w}^{i,d}\}, \ i = 1, \ldots, H, \tag{26}$$

where d is a depth of variation, a parameter of the search algorithm, H is a number of possible solutions in the population.

Any possible solution $\mathbf{\Psi}_i$ is obtained by small variations of the basic solution $\mathbf{\Psi}_0$. Variation vector is an operator changing of the network operator matrix. Therefore, for any possible solution, one can write the following equation:

$$\mathbf{\Psi}_i = \mathrm{W}_i \circ \mathbf{\Psi}_0 = \mathbf{w}^{i,d} \circ \mathbf{w}^{i,d-1} \circ \cdots \circ \mathbf{w}^{i,1} \circ \mathbf{\Psi}_0. \tag{27}$$

To perform a crossover operation, two possible solutions are selected randomly:

$$\begin{aligned} \mathrm{W}_\alpha &= \{\mathbf{w}^{\alpha,1}, \ldots, \mathbf{w}^{\alpha,d}\}, \\ \mathrm{W}_\beta &= \{\mathbf{w}^{\beta,1}, \ldots, \mathbf{w}^{\beta,d}\}. \end{aligned} \tag{28}$$

The crossover point is selected randomly, $c \in \{1, \ldots, d\}$. Two new possible solutions are obtained by the exchange of tails after the crossover point of selected possible solutions

$$\begin{aligned} \mathrm{W}_{H+1} &= \{\mathbf{w}^{\alpha,1}, \ldots, \mathbf{w}^{\alpha,c-1}, \mathbf{w}^{\beta,c}, \ldots, \mathbf{w}^{\beta,d}\}, \\ \mathrm{W}_{H+2} &= \{\mathbf{w}^{\beta,1}, \ldots, \mathbf{w}^{\beta,c-1}, \mathbf{w}^{\alpha,c}, \ldots, \mathbf{w}^{\alpha,d}\}. \end{aligned} \tag{29}$$

4. Computation Experiment

Consider the optimal control problem of the spatial motion of a quadcopter. The mathematical model of the control object is

$$\begin{aligned} \dot{x}_1 &= x_4, \\ \dot{x}_2 &= x_5, \\ \dot{x}_3 &= x_6, \\ \dot{x}_4 &= u_4(\sin(u_3)\cos(u_2)\cos(u_1) + \sin(u_1)\sin(u_2)), \\ \dot{x}_5 &= u_4\cos(u_3)\cos(u_1) - g, \\ \dot{x}_6 &= u_4(\cos(u_2)\sin(u_1) - \cos(u_1)\sin(u_2)\sin(u_3)), \end{aligned} \tag{30}$$

where $g = 9.80665$.

For a given model of the control object, it is necessary to build a system for stabilising the motion along a given spatial trajectory, where the shape of the trajectory is not known in advance. For this purpose, we first create a universal stabilisation system. Solving the optimal control problem, we obtain several different trajectories, and then solve the stabilisation system synthesis problem for all the trajectories.

4.1. Synthesis of Universal Stabilisation System

Usually, the development of a system for stabilising the movement of an object along a given trajectory [19] is to study the mathematical model of the control object, determine the control channels, determine the deviation of the object from the trajectory or from the nearest point [7] located on the trajectory and inserting the regulator into the control channel for qualitative compensation of the deviation. Sometimes model predictive controls with a simplified model of the control object are used to quickly determine the deviation [20]. In this work, an attempt is made to develop a universal system for stabilising the movement along a given trajectory based on machine learning control by symbol regression. In this approach, the analytical study of the mathematical model of the control object is entrusted to the computer, which itself finds the necessary control channels and inserts the necessary regulators there.

To generate program trajectories, we formulate an optimal control problem for a given object (30).

Initial and terminal states are given as

$$\mathbf{x}(0) = \mathbf{x}^0 = [0\ 5\ 0\ 0\ 0\ 0]^T, \tag{31}$$

$$\mathbf{x}(t_f) = \mathbf{x}^f = [10\ 5\ 10\ 0\ 0\ 0]^T, \tag{32}$$

where t_f is a time of achievement of the terminal state, t_f is not given, but limited, $t_f \leqslant t^+ = 5.6$.

Phase constraints are included in the quality criterion,

$$\varphi_i(\mathbf{x}) = r_i - \sqrt{(x_{1,i} - x_1)^2 + (x_{3,i} - x_3)^2} \leqslant 0,\ i = 1, \ldots, M, \tag{33}$$

where M is a number of obstacles, $M = 2$, r_i is a radius of obstacle i, $r_1 = 2$, $r_2 = 2$, $(x_{1,i}, x_{3,i})$ are coordinates of their centers, $x_{1,1} = 2.5$, $x_{3,1} = 2.5$, $x_{1,2} = 7.5$, $x_{3,2} = 7.5$.

The control object should move through some specified areas. Changing the position of the these areas affects the optimal trajectory shape. This condition is also included in the quality criterion,

$$\delta_i^{(k)}(\mathbf{x}(t)) = \min_{0 \leqslant t \leqslant t_f} \left\{ \sqrt{(z_{1,i}^{(k)} - x_1(t))^2 + (z_{3,i}^{(k)} - x_3(t))^2} - d_i^{(k)} \right\} \leqslant 0, \tag{34}$$

where $(z_{1,i}^{(k)}, z_{3.i}^{(k)})$ are coordinates of the area centers on the horizontal plane, $i = 1, \ldots, S$, $k = 1, \ldots, P$, $d_i^{(k)}$ is a size of area, S is a number of areas, $S = 4$, $d_i^{(k)} = 0.6$, P is a number of optimal control problems, $P = 4$, k is a current optimal control problem.

The quality criterion is

$$J_4^{(k)} = t_f + p_1\|\mathbf{x}^f - \mathbf{x}(t_f)\| + p_2 \sum_{i=1}^{M} \int_0^{t_f} \vartheta(\varphi_i(\mathbf{x}))dt + p_3 \sum_{j=1}^{S} \vartheta(\delta^{(k)}(\mathbf{x}(t))) \to \min_{\mathbf{u} \in \mathrm{U}}, \tag{35}$$

where $p_1 = 2$, $p_2 = 3$, $p_3 = 3$, $\vartheta(\alpha)$ is the Heaviside step function

$$\vartheta(\alpha) = \begin{cases} 1, \text{ if } \alpha > 0 \\ 0, \text{ otherwise} \end{cases}. \tag{36}$$

To obtain a variety of trajectories, we consider the conditions for the object to pass through specified areas when solving the optimal control problem. Various optimal trajectories were obtained as a result of different locations of the specified areas.

The training set contained four trajectories in defined locations in the required areas.

The coordinates of required area centers are

$$z_{1,1}^{(1)} = 2.5,\ z_{3,1}^{(1)} = 0.4,\ z_{1,2}^{(1)} = 5.0,\ z_{3,2}^{(1)} = 2.0,\ z_{1,3}^{(1)} = 7.5,\ z_{3,3}^{(1)} = 4.5,\ z_{1,4}^{(1)} = 9.6,\ z_{3,4}^{(1)} = 7.5,$$
$$z_{1,1}^{(2)} = 2.5,\ z_{3,1}^{(2)} = 0.4,\ z_{1,2}^{(2)} = 4.5,\ z_{3,2}^{(2)} = 2.5,\ z_{1,3}^{(2)} = 5.5,\ z_{3,3}^{(2)} = 7.5,\ z_{1,4}^{(2)} = 7.5,\ z_{3,4}^{(2)} = 9.6,$$
$$z_{1,1}^{(3)} = 0.0,\ z_{3,1}^{(3)} = 2.0,\ z_{1,2}^{(3)} = 2.0,\ z_{3,2}^{(3)} = 5.0,\ z_{1,3}^{(3)} = 5.0,\ z_{3,3}^{(3)} = 8.0,\ z_{1,4}^{(3)} = 8.0,\ z_{3,4}^{(3)} = 10.0,$$
$$z_{1,1}^{(4)} = 0.4,\ z_{3,1}^{(4)} = 2.5,\ z_{1,2}^{(4)} = 2.5,\ z_{3,2}^{(4)} = 4.5,\ z_{1,3}^{(4)} = 7.5,\ z_{3,3}^{(4)} = 5.5,\ z_{1,4}^{(4)} = 9.6,\ z_{3,4}^{(4)} = 7.5.$$

When solving optimal control problems, we used a direct approach. For this purpose, we divided the time axis into equal intervals and looked for the values of constant parameters at the boundaries of the intervals. Taking into account the control constraints, the piecewise linear approximation of the control function has the following form:

$$u_i^{(k)} = \begin{cases} u_j^+, \text{ if } \hat{u}_j^k > u_j^+ \\ u_j^-, \text{ if } \hat{u}_j^k < u_j^- \\ \hat{u}_j^k, \text{ otherwise} \end{cases} \tag{37}$$

where

$$\hat{u}_j^{(k)} = q_{j+mi}^{(k)} + (q_{j+m(i+1)}^{(k)} - q_{j+mi}^{(k)})\frac{t - i\Delta t}{\Delta t}, \tag{38}$$

$i = 1, \ldots, N$, $j = 1, \ldots, m$, N is a number of time intervals, Δt is a time interval,

$$N = \left\lfloor \frac{t^+}{\Delta t} \right\rfloor = \left\lfloor \frac{5.6}{0.4} \right\rfloor = 14. \tag{39}$$

For numerical solution a hybrid algorithm [21] was applied. A hybrid algorithm is based on three well-known algorithms: the genetic algorithm [22], particle swarm optimization [23], and the grey wolf optimizer [24].

The total $(N+1)m = 15 \times 4 = 60$, $\mathbf{q}^{(k)} = [q_1 \ldots q_{60}]^T$ parameters were found.

The solutions of the optimal control problem obtained by the hybrid evolutionary algorithm are given in the Data Availability section.

The parameters of hybrid evolutionary algorithm are the number of possible solutions in population—1024, the number of generations—512, the number of evolutionary transformations in each population—512. For GA, the number of bits for integer part—4, the number of bits for fructional part—12, the probability of mutation—0.75. For GWO, the number of leaders—4. For PSO, the parameters are $k_\alpha = 0.729$, $k_\beta = 0.85$, $k_\gamma = 0.15$, $k_\sigma = 1$, and the number of randomly selected solutions to choose the informant—4.

The projections of optimal trajectories on the horizontal plane of the four optimal control problems are shown in Figures 2–5. In the figures, red circles indicate obstacles that define phase constraints of the problem. Small dotted circles indicate the specified areas mandatory for trajectories to follow.

After obtaining the optimal program trajectories, the problem of synthesis of the object motion stabilisation system along these trajectories was solved (1)–(2), (9)–(13). It was necessary to find one stabilisation system for all four program trajectories from the training set according to criterion (13). The domain of initial conditions (9) contained $K = 26$ vectors of initial states.

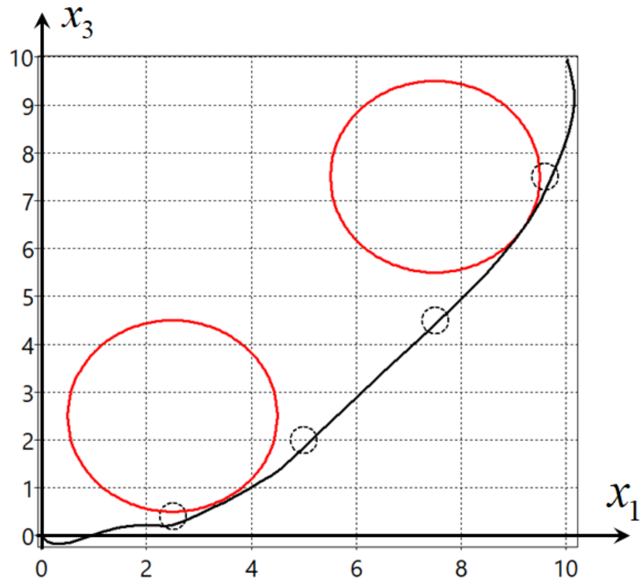

Figure 2. Projection of optimal trajectory 1 on the horizontal plane.

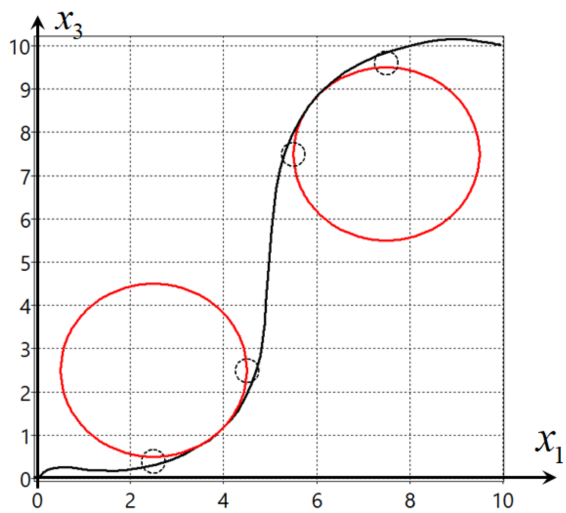

Figure 3. Projection of optimal trajectory 2 on the horizontal plane.

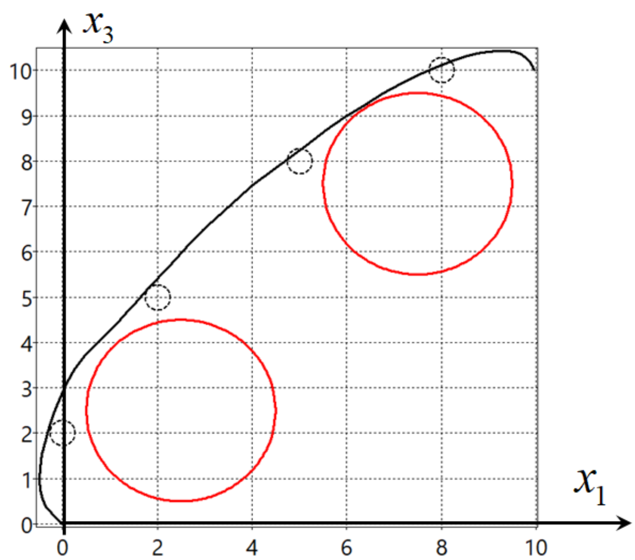

Figure 4. Projection of optimal trajectory 3 on the horizontal plane.

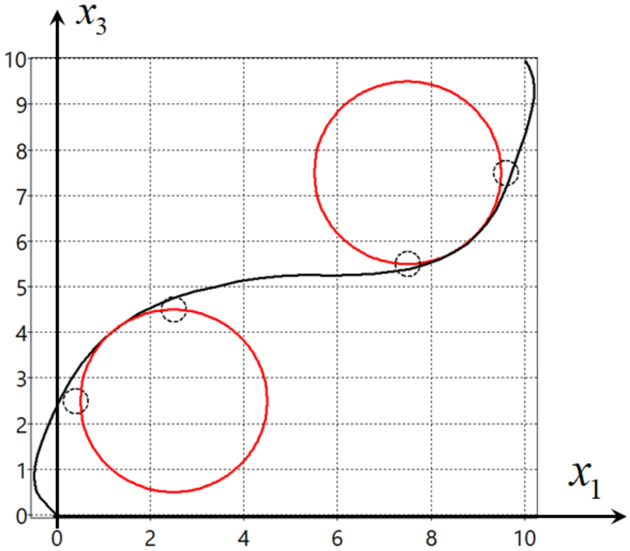

Figure 5. Projection of optimal trajectory 4 on the horizontal plane.

To solve the synthesis problem, the network operator method [12] implemented in the software package developed by the authors was used. The parameters of the algorithm were the number of chromosomes in the initial population—512; the number of generations—128; the number of couples in one generation—128; the dimension of network operator 36×36; the number of variations of one possible solution—5. The computational time for the universal stabilisation system synthesis on PC with CPU Intel Corei7 2.8 GHz for four trajectories was approximately 10 min.

The following solution was obtained:

$$u_i = \begin{cases} u_i^+, \text{ if } \tilde{u}_i > u_i^+ \\ u_i^-, \text{ if } \tilde{u}_i < u_i^- \\ \tilde{u}_i, \text{otherwise} \end{cases}, \quad (40)$$

where

$$\tilde{u}_1 = \mu(A) + \rho_{19}(q_2), \quad (41)$$

$$\tilde{u}_2 = (\tilde{u}_1 - \tilde{u}_1^3)\vartheta(q_6(x_6^* - x_6) + q_3(x_3^* - x_3)), \quad (42)$$

$$\tilde{u}_3 = \tilde{u}_2 + \rho_{17}(q_1 \arctan(x_1^* - x_1) + q_4(x_4^* - x_4)), \quad (43)$$

$$\begin{aligned} \tilde{u}_4 \quad = \quad & \tilde{u}_3^2 + \operatorname{sgn}(B + \mu(A) + \rho_{19}(q_2))\sqrt{|B + \mu(A) + \rho_{19}(q_2)|} + \\ & \vartheta(C) + \sin(D) + \operatorname{sgn}(-A \arctan(E)F) + \arctan(G) + \sqrt[3]{F} + \\ & \sin(q_4(x_4^* - x_4)) + \exp(q_2(x_2^* - x_2)) + \sqrt{q_1}, \end{aligned} \quad (44)$$

$$A = q_6(x_6^* - x_6) + q_3(x_3^* - x_3) + \vartheta(q_6(x_6^* - x_6)),$$

$$B = H + \tanh(D) + \exp(E) + \sqrt[3]{F} + \exp(q_5(x_5^* - x_5)),$$

$$C = D + \tanh(-A \arctan(E)F) + \rho_{18}(F),$$

$$D = -A \arctan(E)F + \sqrt[3]{G} + \operatorname{sgn}(A) +$$

$$\sin(q_1 \arctan(x_1^* - x_1) + q_4(x_4^* - x_4)) + \cos(q_3(x_3^* - x_3)),$$

$$E = F^3 + q_1 \arctan(x_1^* - x_1) + q_4(x_4^* - x_4) + \vartheta(q_5(x_5^* - x_5)) +$$

$$\arctan(q_4) - (x_6^* - x_6) + (x_5^* - x_5)^2,$$

$$F = \sin(q_6(x_6^* - x_6)) + q_5(x_5^* - x_5) + (q_2 + 1)(x_2^* - x_2) + \cos(q_1) - (x_2^* - x_2)^3,$$

$$G = \rho_{19}(A) + E + \ln(|q_6(x_6^* - x_6) + q_3(x_3^* - x_3)|) + \exp(F) +$$
$$\vartheta(q_5(x_5^* - x_5)) + \operatorname{sgn}(x_5^* - x_5) + (x_2^* - x_2)^3,$$

$$H = \exp(C) + \cos(q_6(x_6^* - x_6)) + \operatorname{sgn}(C)\sqrt{|C|} + \operatorname{sgn}(D)\sqrt{|D|} + \sin(q_5(x_5^* - x_5)),$$

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

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

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

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

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

$q_1 = 12.10181$, $q_2 = 4.23291$, $q_3 = 15.55688$, $q_4 = 14.70337$, $q_5 = 7.75635$, $q_6 = 10.45923$.

Figures 6–9 show projections of one optimal trajectory (in blue) and eight perturbed trajectories (in black) from the domain of initial conditions (9). The figures show that the same stabilisation system (40)–(44) provides the object motion in the neighbourhood of all four optimal trajectories. It should be considered universal.

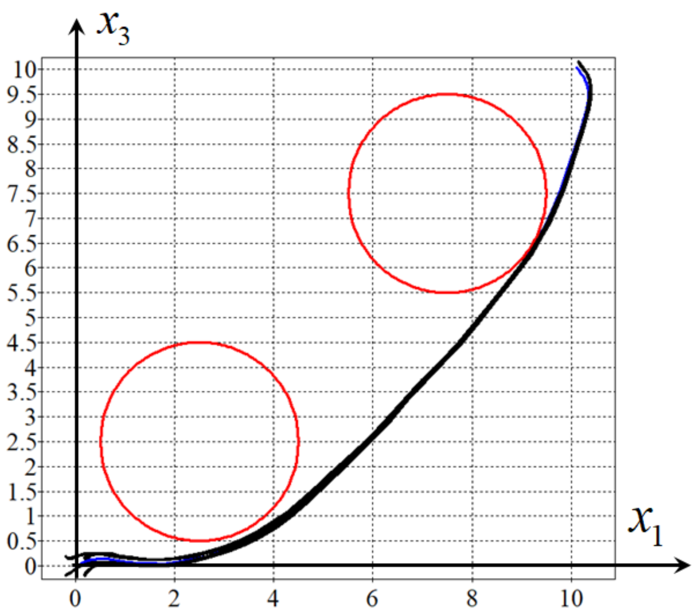

Figure 6. Projections of optimal trajectory 1 and perturbed trajectories from eight initial states on the horizontal plane.

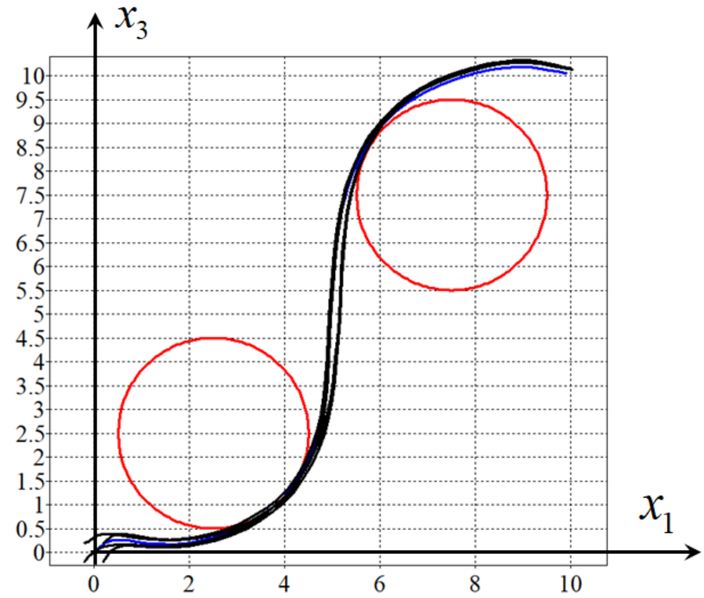

Figure 7. Projections of optimal trajectory 2 and perturbed trajectories from eight initial states on the horizontal plane.

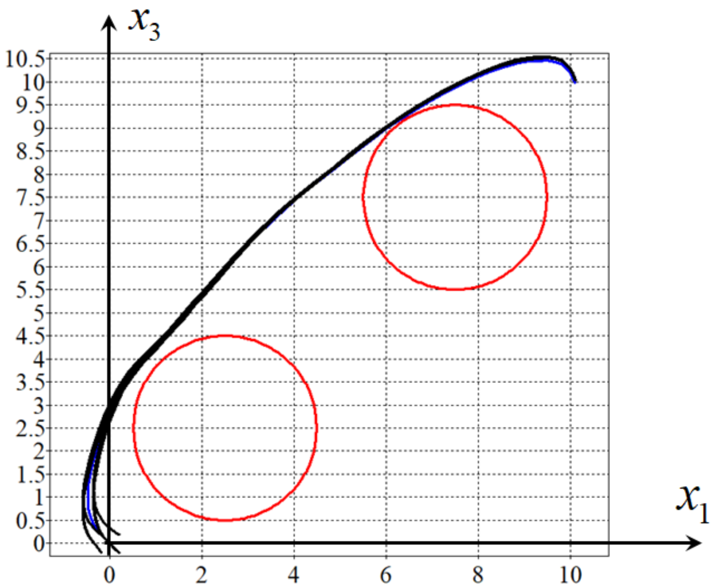

Figure 8. Projections of optimal trajectory 3 and perturbed trajectories from eight initial states on the horizontal plane.

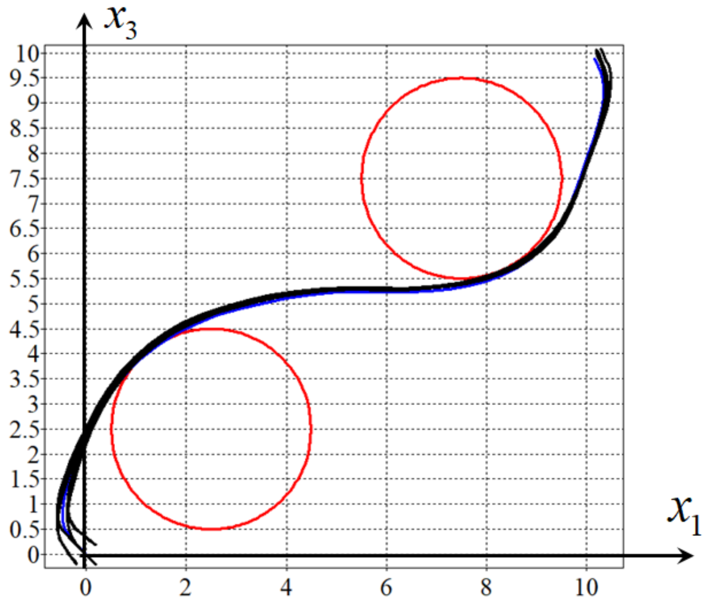

Figure 9. Projections of optimal trajectory 4 and perturbed trajectories from eight initial states on the horizontal plane.

4.2. Solution of Complete Optimal Control Problem for Object with Motion Stabilisation System

In the second computational experiment, we solved the complete optimal control problem for an object with a universal stabilisation system (40)–(44). The spatial motion of the quadrotor along a closed trajectory was considered. The mathematical model of the control object with a universal stabilisation system has the following form:

$$\begin{aligned}
\dot{x}_1 &= x_4, \\
\dot{x}_2 &= x_5, \\
\dot{x}_3 &= x_6, \\
\dot{x}_4 &= h_4(\mathbf{x}^* - \mathbf{x})(\sin(h_3(\mathbf{x}^* - \mathbf{x}))\cos(h_2(\mathbf{x}^* - \mathbf{x}))\cos(h_1(\mathbf{x}^* - \mathbf{x})) \\
&\quad + \sin(h_1(\mathbf{x}^* - \mathbf{x}))\sin(h_2(\mathbf{x}^* - \mathbf{x}))), \\
\dot{x}_5 &= h_4(\mathbf{x}^* - \mathbf{x})\cos(h_3(\mathbf{x}^* - \mathbf{x}))\cos(h_1(\mathbf{x}^* - \mathbf{x})) - g, \\
\dot{x}_6 &= h_4(\mathbf{x}^* - \mathbf{x})(\cos(h_2(\mathbf{x}^* - \mathbf{x}))\sin(h_1(\mathbf{x}^* - \mathbf{x})) \\
&\quad - \cos(h_1(\mathbf{x}^* - \mathbf{x}))\sin(h_2(\mathbf{x}^* - \mathbf{x}))\sin(h_3(\mathbf{x}^* - \mathbf{x}))), \\
\dot{x}_1^* &= x_4^*, \\
\dot{x}_2^* &= x_5^*, \\
\dot{x}_3^* &= x_6^*, \\
\dot{x}_4^* &= u_4(\sin(u_3)\cos(u_2)\cos(u_1) + \sin(u_1)\sin(u_2)), \\
\dot{x}_5^* &= u_4\cos(u_3)\cos(u_1) - g, \\
\dot{x}_6^* &= u_4(\cos(u_2)\sin(u_1) - \cos(u_1)\sin(u_2)\sin(u_3)),
\end{aligned} \tag{45}$$

where $\mathbf{x} = [x_1 \dots x_6]^T$, $\mathbf{x}^* = [x_1^* \dots x_6^*]^T$.

For the control object, the initial conditions coincide with the terminal conditions

$$\mathbf{x}^0 = \mathbf{x}^f = [0\ 5\ 0\ 0\ 0\ 0]^T. \tag{46}$$

Four phase constraints are

$$\varphi_i(\mathbf{x}) = r_i - \sqrt{(x_{1,i} - x_1)^2 + (x_{3,i} - x_3)^2} \leqslant 0,\ i = 1, \dots, M, \tag{47}$$

where M is the number of obstacles, $M = 4$, r_i is the radius of obstacle i, $r_i = 2$, $i = 1, \dots, 4$, $(x_{1,i}, x_{3,i})$ are coordinates of their centers, $x_{1,1} = 5$, $x_{3,1} = 0$, $x_{1,2} = 10$, $x_{3,2} = 5$, $x_{1,3} = 5$, $x_{3,3} = 10$, $x_{1,4} = 0$, $x_{3,4} = 5$.

The three specified areas are as follows:

$$\delta_i(\mathbf{x}(t)) = \min_{0 \leqslant t \leqslant t_f} \left\{ \sqrt{(z_{1,i} - x_1(t))^2 + (z_{3,i} - x_3(t))^2} - d_i \right\} \leqslant 0, \tag{48}$$

where $(z_{1,i}, z_{3.i})$ are the coordinates of the area centers on the horizontal plane, $i = 1, \ldots, S$, S is the number of areas, $S = 3$, d_i is the size of the area, $d_i = 0.6$.

The coordinates of the required area centers are $z_{1,1} = 10$, $z_{3,1} = 0$, $z_{1,2} = 10$, $z_{3,2} = 10$, $z_{1,3} = 0$, $z_{3,3} = 10$.

The optimal control problem was solved on the basis of the direct approach by a hybrid evolutionary algorithm [21]. To solve the problem, we set $t^+ = 15.2$ and time interval $\Delta t = 0.4$. There were 38 intervals in total. For each interval boundary, $M = 4$ controls had to be found. Thus, the total $(N+1)m = 39 \times 4 = 156$, $\mathbf{q} = [q_1 \ldots q_{156}]^T$ parameters were found.

The values of the found parameters for optimal control are given in the Data Availability section. The value of the quality criterion (35) is $J_4 = 15.1221$. Figure 10 shows the projection of the new found optimal trajectory on the horizontal plane.

Figure 11 shows a new optimal trajectory (in blue) and eight perturbed trajectories (in black) for an object with a universal stabilisation system. As it can be seen from the figure, all perturbed trajectories are in the neighbourhood of the optimal trajectory and satisfy the phase constraints. A value of the quality criterion (35) for the object with a universal stabilisation system without perturbations (blue) is $J_4 = 16.341$.

To estimate the results, a comparable experiment was performed. For models with and without a stabilisation system, the initial states were subjected to random perturbations,

$$x_i(0) = x_i^0 + 2\beta_0(\xi(t) - 1), \tag{49}$$

where $\xi(t)$ is a random numbers generator. Function $\xi(t)$ returns a random number from 0 to 1 after each call, β_0 is a level of perturbations.

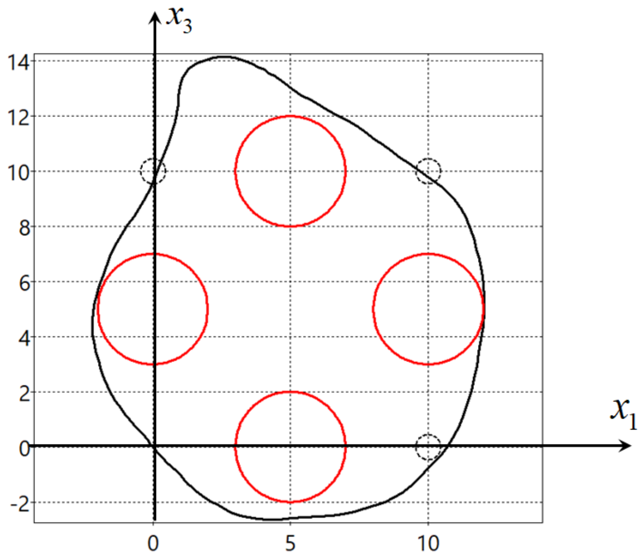

Figure 10. Projection of a new optimal trajectory on the horizontal plane.

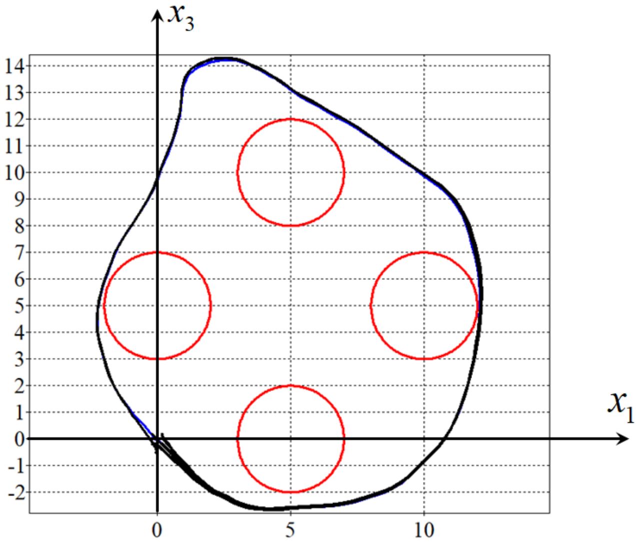

Figure 11. Projections of a new optimal trajectory (blue) and perturbed trajectories (black) on the horizontal plane.

The values of quality criterion (35) for perturbed initial states at $\beta_0 = 0.1$ for object with and without stabilisation system are given in Table 1.

The last two lines show average value and standard deviation (SD) on experiments. As it can be seen from Table 1, an object model without a stabilisation system is essentially more sensitive to the perturbation of initial conditions.

Table 1. Sensitivity of solutions to perturbations of initial states.

No	Direct	Stabilisation
1	20.5112	16.4147
2	18.5402	16.4715
3	19.3551	16.4456
4	18.3839	16.3356
5	20.0805	16.4408
6	18.7307	16.4417
7	19.0710	16.4456
8	21.3337	16.4120
9	20.3913	16.4495
10	18.5464	16.4710
Avg.	19.4940	16.4323
SD	1.02170	0.04012

5. Conclusions

The paper presents a statement of the complete optimal control problem in which, according to the classical statement, it is necessary to find the control function and the optimal program trajectory, and to implement the solution, it is necessary to solve the synthesis problem of motion stabilisation along the program trajectory. To solve the stabilisation system synthesis problem, machine learning of control by the symbolic regression method is used.

For the first time, it is proposed to synthesise a universal stabilisation system that ensures the motion of an object along different trajectories from some class. To synthesise a universal stabilisation system, a training set of different trajectories is generated. As a result of solving the problem of stabilisation system synthesis, we obtain one universal stabilisation system which provides object motion along all trajectories from the training set.

The obtained solution was tested on stabilisation of the object motion along the trajectory, which was not included in the training set. An example of solving the complete optimal control problem for quadcopter motion in space with four obstacles was given. The optimal trajectory was a closed curve, which passed through the specified areas and avoided the obstacles.

Training of the stabilisation system was performed on trajectories that differed significantly from the example. The training set included four trajectories that were obtained as a result of solving the optimal control problem of quadcopter spatial motion from a given initial point to a given terminal point in space with two phase constraints. The trajectories differed in that they avoided obstacles from different sides. The computational experiment showed that the universal stabilisation system provided qualitative motion of the object along the closed optimal trajectory, which was not included in the training set.

6. Discussion

Future research is aimed at expanding of the class of trajectories that are included in the training set. It is important to identify and investigate the properties of trajectories that cannot be stabilised by the obtained stabilisation system, i.e., to determine the limits of applicability of the proposed universal stabilisation system.

Furthermore, the construction of universal stabilisation systems for different control objects will exclude the most time-consuming stage of synthesis of the trajectory motion stabilisation system from the solution of the optimal control problem. If for some object it is necessary to construct several stabilisation systems for different classes of trajectories, it is necessary to synthesise such stabilisation systems and further use them to solve different optimal control problems.

Author Contributions: Conceptualization, A.D. and E.S.; methodology, A.D.; software, A.D. and E.S.; validation, E.S.; formal analysis, A.D.; investigation, E.S.; resources, A.D.; writing—original draft preparation, A.D. and E.S.; writing—review and editing, E.S.; visualization, A.D.; supervision, A.D. All authors have read and agreed to the published version of the manuscript.

Funding: This research was supported by the Ministry of Science and Higher Education of the Russian Federation, project No. 075-15-2020-799.

Data Availability Statement: The data that support the findings of this study are openly available in https://cloud.mail.ru/public/4CBU/sBEA3B2wE, access date: 1 July 2023.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Pontryagin, L.S.; Boltyanskii, V.G.; Gamkrelidze, R.V.; Mishchenko, E.F. *The Mathematical Theory of Optimal Process*; Gordon and Breach Science Publishers: New York, NY, USA; London, UK; Paris, France; Montreux, Switzerland; Tokyo, Japan, 1985; 360p.
2. Bellman, R.E.; Dreyfus, S.E. *Applied Dynamic Programming*; Princeton Legacy Library: Princeton, NJ, USA, 1962.
3. Diveev, A.I. Refinement of Optimal Control Problem for Practical Implementation of Its Solution. *Dokl. Math.* **2023**, *107*, 28–36. [CrossRef]
4. Brockett, R.W. Asymptotic Stability and Feedback Stabilization. In *Differential Geometric Control Theory*; Brockett, R.W., Millman, R.S., Sussmann, H.J., Eds.; Birkhauser: Boston, MA, USA, 1983; pp. 181–191.
5. Walsh, G.; Tilbury, D.; Sastry, S.; Murray, R.; Laumond, J.P. Stabilization of Trajectories for Systems with Nonholonomic Constraints. *IEEE Trans. Autom. Control* **1994**, *39*, 216–222. [CrossRef]
6. Samir, A.; Hammad, A.; Hafez, A.; Mansour, H. Quadcopter Trajectory Tracking Control using State-Feedback Control with Integral Action. *Int. J. Comput. Appl.* **2017**, *168*, 1–7. [CrossRef]
7. Nguyen, A.T.; Xuan-Mung, N.; Hong, S.-K. Quadcopter Adaptive Trajectory Tracking Control: A New Approach via Backstepping Technique. *Appl. Sci.* **2019**, *9*, 3873. [CrossRef]
8. Mohamed, M.J.; Abbas, M.Y. Design a Fuzzy PID Controller for Trajectory Tracking of Mobile Robot.*Eng. Technol. J.* **2018**, *36A*, 100–110. [CrossRef]
9. Ma, T.; Wong, S. Trajectory Tracking Control for Quadrotor UAV. In Proceedings of the 2017 IEEE International Conference on Robotics and Biomimetics (ROBIO), Macau, Macao, 5–8 December 2017.

10. Diveev, A.; Sofronova, E. Synthesized Control for Optimal Control Problem of Motion Along the Program Trajectory. In Proceedings of the 2022 8th International Conference on Control, Decision and Information Technologies (CoDIT), Istanbul, Turkey, 17–20 May 2022; pp. 475–480.
11. Diveev, A.; Sofronova, E.; Konyrbaev, N.; Bexeitova, A. Stabilization of Movement Along an Optimal Trajectory and Its Solution. *Eng. Proc.* **2023**, *33*, 12.
12. Diveev, A.I.; Sofronova, E.A. The network operator method for search of the most suitable mathematical equation. In *Bio-Inspired Computational Algorithms and Their Applications*; InTech: Rijeka, Croatia, 2012; pp. 19–42.
13. Diveev, A.; Kazaryan, D.; Sofronova, E. Symbolic regression methods for control system synthesis. In Proceedings of the 2014 22nd Mediterranean Conference on Control and Automation, MED 2014, Palermo, Italy, 16–19 June 2014; pp. 587–592.
14. Quade, M.; Abel, M.; Isele, T. Machine learning control—Explainable and analyzable methods. *Phys. D Nonlinear Phenom.* **2020**, *412*, 132582. [CrossRef]
15. Koza, J.R. *Genetic Programming: On the Programming of Computers by Means of Natural Selection*; MIT Press: Cambridge, MA, USA, 1992; 819p.
16. Koza, J.; Keane, M.; Yu, J.; Bennett, F., III; Mydlowec, W. Automatic Creation of Human-Competitive Programs and Controllers by Means of Genetic Programming. *Genet. Program. Evolvable Mach.* **2000**, *1*, 121–164. [CrossRef]
17. Marchetti, F.; Minisci, E. Genetic Programming Guidance Control System for a Reentry Vehicle under Uncertainties. *Mathematics* **2021**, *9*, 1868. [CrossRef]
18. Sofronova, E.A.; Diveev, A.I. Universal Approach to Solution of Optimization Problems by Symbolic Regression. *Appl. Sci.* **2021**, *11*, 5081. [CrossRef]
19. Karnani, C.; Raza, S.; Asif, A.; Ilyas, M. Adaptive Control Algorithm for Trajectory Tracking of Underactuated Unmanned Surface Vehicle (UUSV). *J. Robot.* **2023**, *2023*, 4820479. [CrossRef]
20. Chipofya, M.; Lee, D.J.; Chong, K.T. Trajectory Tracking and Stabilization of a Quadrotor Using Model Predictive Control of Laguerre Functions. *Abstr. Appl. Anal.* **2015**, *2015*, 916864. [CrossRef]
21. Diveev, A. Hybrid evolutionary algorithm for optimal control problem. In *Intelligent Systems and Applications. IntelliSys 2022*; Arai K., Ed.; Lecture Notes in Networks and Systems; Springer: Cham, Switzerland, 2023; Volume 543, pp. 726–738.
22. Davis, L. *Handbook of Genetic Algorithms*; Van Nostrand Reinhold: New York, NY, USA, 1991.
23. Kennedy, J.; Eberhart, R. Particle swarm optimization. In Proceedings of the IEEE International Conference on Neural Networks, Perth, Australia, 27 November–1 December 1995; Volume IV, pp. 1942–1948.
24. Mirjalili, S.; Mirjalili, S.M.; Lewis, A. Grey Wolf Optimizer. *Adv. Eng. Softw.* **2014**, *69*, 46–61. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.