mathematics

Article

# A Stabilisation System Synthesis for Motion along a Preset Trajectory and Its Solution by Symbolic Regression

**Askhat Diveev [1], Elena Sofronova [1] and Nurbek Konyrbaev [2,*]**

1 Federal Research Center "Computer Science and Control" of the Russian Academy of Sciences, Vavilova Str., 44, Build. 2, 119333 Moscow, Russia; aidiveev@mail.ru (A.D.); sofronova_ea@mail.ru (E.S.)
2 Institute of Engineering and Technology, Korkyt Ata Kyzylorda University, Aiteke bi Str. 29A, Kyzylorda 120014, Kazakhstan
* Correspondence: n.konyrbaev@mail.ru

**Abstract:** The problem of a stabilisation system synthesis for the motion of a control object along a given spatial trajectory is considered. The complexity of the problem is that the preset trajectory is defined in the state subspace and not in time. This paper describes a stabilisation system synthesis for motion along a trajectory specified in time and along a trajectory specified in the form of a manifold in a state space. In order to construct a stabilisation system, it is necessary to determine a distance between an object and the given trajectory at each moment in time. For trajectories that are not given in time, the determination of this distance can be ambiguous. An object may be exactly on a trajectory but at a different time. This paper proposes some approaches to solve the problem. One of the approaches is to transform a given trajectory in a state subspace into a trajectory given in time. A description of a universal method to perform this transformation is presented. In order to solve the synthesis problem automatically, without having to analyse the mathematical model of the control object, it is suggested that machine learning control by symbolic regression is used. In computational experiments, examples of stabilisation system syntheses for quadcopter motion along a given spatial trajectory are presented.

**Keywords:** machine learning control; optimal control; control synthesis; stabilisation system; symbolic regression; quadcopter

**MSC:** 49M25; 68W50

check for updates

**Citation:** Diveev, A.; Sofronova, E.; Konyrbaev, N. A Stabilisation System Synthesis for Motion along a Preset Trajectory and Its Solution by Symbolic Regression. *Mathematics* **2024**, *12*, 706. https://doi.org/10.3390/math12050706

Academic Editors: Liliya Demidova and Ioannis G. Tsoulos

Received: 3 December 2023
Revised: 7 February 2024
Accepted: 21 February 2024
Published: 28 February 2024

## 1. Introduction

The problem of stabilising the motion of an object along a given trajectory is very common for almost all autonomous robots moving in geometric space. If a mathematical model of a control object is an ODE system with a free control vector in the right part, then the stabilisation system is a control function that changes the right part so that the ODE system, as a parametric mapping of the state space into itself, has an attractor property in the neighbourhood of the given trajectory. An argument of the control function of a stabilisation system is a deviation of the control object from the given trajectory. An ideal stabilisation system has such a control function that the ODE system of the mathematical model of the control object has a stable particular solution or a singular particular solution in the form of an attractor in the neighbourhood of the given trajectory.

Basically, two approaches are used to solve the tracking problem. The first is analytical, when the inverse problem is solved (a trajectory is specified in time and a control is sought that ensures movement along the trajectory). This approach is widely used for simple low-dimensional models and, as a rule, control is included linearly in the object model. In many papers related to trajectory tracking [1–5], the control system is built based on the analysis of the control object model. The developer of the control system studies the mathematical model of the control object, defines control channels that affect the movement of the object and determines the components of the control vector that affect a particular direction of movement of the object. Further, regulators are inserted into the control channels, which qualitatively work out movement errors along the trajectory.

The second approach is applied when the trajectory is given in space. A stabilisation system is built relative to a point in state space and then these points are placed on the trajectory [6–8]. Switching points occur sequentially and the object moves from one point to another along a trajectory. This process is known as point tracking. This approach ensures accurate movement along the trajectory providing the intervals for switching points and their locations are optimally selected. Note that the movement of a control object from one stable point to another is not uniform. The speed of the object slows down as it approaches a stable point of equilibrium. At the equilibrium point, the object stops. Therefore, point tracking is not optimal if the quality criterion depends on the time of the control process. Each of these approaches, when executed carefully enough, can produce a satisfactorily accurate trajectory.

To ensure the stability of the object with respect to the equilibrium point in point tracking, it is necessary to solve the control synthesis problem. It is necessary to search for a control function as a function of a state space vector. At present this problem does not have an exact universal numerical solution. The most general approach is to linearise the model with respect to the equilibrium point and to find the control as a linear feedback function, so that the matrix of the control system has eigenvalues in the left half of the complex plane. Such an approach does not provide any quality in the control process. Solutions of differential equations in the neighbourhood of a stable equilibrium point can be asymptotic or periodic.

The control synthesis problem is so complex that it can only be solved when the control system is created. Trajectories can be given in the process of robotic operation. The motion stabilisation system should be universal, so that this system can be used to stabilise a wide class of trajectories, i.e., one stabilisation system should be applied to movement along any trajectory from a set. In [9], a universal stabilisation system for motion along an optimal program trajectory is obtained as a result of solving the optimal control problem. The stabilisation system is constructed on the basis of machine learning control by symbolic regression. The optimal trajectory is a function of time, so the deviation of the control object from the given trajectory is simply calculated as a difference between the state vector of the control object and the current coordinate of the trajectory point in the geometric subspace at any time. This paper continues the study of the construction of a universal stabilisation system for the motion of a control object along a given trajectory. In contrast to [9], here we consider the trajectories that are not functions of time but are given in the state space in the form of a one-dimensional manifold.

Another aspect of the study deals with the automation of control system development. For a real control system, the problem of control synthesis is solved manually by studying the mathematical model of the control object, determining the control channels and inserting the regulators there. This paper presents the approach of constructing an object motion stabilisation system along the trajectory based on machine learning control by symbolic regression. Machine learning (ML) is a branch of artificial intelligence that focuses on the learning through experience and decision making of computer programs similar to humans. ML has already been effectively applied to various fields [10], for example, to develop accurate, interpretable and generalisable nonlinear models for complex natural and engineered systems [11]. A survey on the application of ML to solve large control problems is presented in [12]. In [13], the term machine learning control (MLC) was introduced as a framework to discover effective control laws for complex, nonlinear dynamics, mainly by genetic programming [14].

The machine, or more precisely the program running on the machine itself, builds a system for stabilising the movement of the object along the trajectory. Developers do not need to study the mathematical model of the control object and define control channels. A computer does it all for them. The developer does not even need to know the trajectory itself, so the stabilisation system to be created is first trained to follow any trajectory consisting of straight segments. The developer then splits the trajectory into segments and passes them to the stabilisation system in this form. The object under the control of the stabilisation system moves along a given trajectory. Thus, the novelty of the research is to propose a universal approach to stabilisation system synthesis for motion along a trajectory not given in time.

In this paper MLC is used to solve the stabilisation system synthesis problem for quadcopter motion along a trajectory given in the state subspace by a symbolic regression method, a network operator method. To estimate the performance of the proposed approach, a linear trajectory containing sharp corners, which pose difficulties for the movement of the quadcopter, was chosen.

## 2. Statement of the Stabilisation System Synthesis of an Optimal Motion

### *2.1. Stabilisation System along Trajectory Given in Time*

The mathematical model of a control object in the form of an ODE system is

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

where $\mathbf{x}$ is a state vector of the control object, $\mathbf{x} \in \mathbb{R}^n$, $\mathbf{x} = [x_1 \dots x_n]^T$; the state space vector consists of two subvectors

$$\mathbf{x}^T = [\mathbf{y}^T \vdots \mathbf{z}^T],\ \mathbf{y} = [x_1 \dots x_k]^T,\ \mathbf{z} = [x_{k+1} \dots x_n]^T, \tag{2}$$

$\mathbf{y} \in \mathbb{R}^k$, $\mathbb{R}^k$ is a geometrical subspace, $k \in \{2, 3\}$, $\mathbf{z}$ is a subvector of state vector that contains components that are not included into geometrical subspace, $\mathbf{u}$ is a control vector, $\mathbf{u} \in \mathrm{U} \subseteq \mathbb{R}^m$ and U is a compact set that defines control constraints. For example, control vector component values may be constrained

$$\mathbf{u}^- \leqslant \mathbf{u} \leqslant \mathbf{u}^+, \tag{3}$$

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

For system (1), an initial state domain is

$$\mathrm{X}_0 \subseteq \mathbb{R}^n. \tag{4}$$

The terminal state is given in geometrical subspace

$$\mathbf{y}(t_f) = \mathbf{y}^f = [x_1^f \dots x_k^f]^T, \tag{5}$$

where $t_f$ is a limited terminal time to reach the terminal state and $t_f \leqslant t^+$, $t^+$ is a given value.

The quality criterion is given in the following common integral form

$$J_0 = \int_0^{t_f} f_0(\mathbf{x}, \mathbf{u}) dt \to \min_{\mathbf{u} \in \mathrm{U}}. \tag{6}$$

The spatial trajectory is presented as a one-dimensional manifold

$$\theta_i(\mathbf{y}) = 0,\ i = 1, \dots, k-1. \tag{7}$$

The terminal state (5) is located on the given trajectory,

$$\theta_i(\mathbf{y}^f) = 0,\ i = 1, \dots, k-1. \tag{8}$$

The control function is searched as

$$\mathbf{u} = \mathbf{h}(\mathbf{y}^* - \mathbf{y}(t)) \in \mathrm{U}, \tag{9}$$

where $\mathbf{y}^*$ is the point on the manifold (7) closest to the current value of the state vector

$$\min \|\mathbf{y}^* - \mathbf{y}(t)\|, \ \theta_i(\mathbf{y}^*) = 0, \ i = 1, \ldots, k-1. \tag{10}$$

For any $P$ initial states from the set (4), the following quality criterion reaches the minimum value

$$J_1 = \sum_{i=1}^{P} \left( \int_0^{t_{f,i}} (f_0(\mathbf{x}(t, \mathbf{x}^{0,i}), \mathbf{h}(\mathbf{y}^* - \mathbf{y}(t, \mathbf{x}^{0,i}))) + p_1 \|\mathbf{y}^* - \mathbf{y}(t, \mathbf{x}^{0,i})\|)dt + p_2 \|\mathbf{y}^f - \mathbf{y}(t_{f,i}, \mathbf{x}^{0,i})\| \right) \to \min, \tag{11}$$

where $p_1$, $p_2$ are weight coefficients, $\mathbf{y}(t, \mathbf{x}^{0,i})$ is a particular solution of ODE system

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

from initial state $\mathbf{x}^{0,i} \in \mathrm{X}_0$, $i = 1, \ldots, P$, and $t_{f,i}$ is a time to reach the terminal state (5), which is calculated as

$$t_{f,i} = \begin{cases} t, \text{ if } t < t^+ \text{ and } \|\mathbf{y}^f - \mathbf{y}(t, \mathbf{x}^{0,i})\| \leqslant \varepsilon_1 \\ t^+, \text{otherwise} \end{cases}, \ i = 1, \ldots, P, \tag{13}$$

$\varepsilon_1$ is the given small positive value.

The solution of the problem is a control function (9). To solve the problem, it is necessary to find the point $\mathbf{y}^*$ on the manifold (7) closest to the current state of the control object at each moment of time. It is a rather time-consuming computational process because it requires solving an optimisation problem.

### 2.2. *Stabilisation System along Trajectory Given in Space*

Let us consider another approach to solve this problem. Suppose that any one-dimensional manifold can be divided into straight segments. Let $\mathbf{y}^{*,j}$, $j = 1, \ldots, N$ be the join points of the straight segments in the geometric space, $\mathbb{R}^k$. Then, the length of straight segment between points $\mathbf{y}^i$ and $\mathbf{y}^{i+1}$ is

$$L_j = \|\mathbf{y}^{*,j+1} - \mathbf{y}^{*,j}\|. \tag{14}$$

The length of the manifold is

$$L = \sum_{j=1}^{N-1} L_i = \sum_{j=1}^{N-1} \|\mathbf{y}^{*,j+1} - \mathbf{y}^{*,j}\|. \tag{15}$$

If $t^+$ is the maximum time for motion on the manifold, then a module of motion speed on the manifold is

$$v = \frac{L}{t^+}, \tag{16}$$

and a time of movement on a straight segment $j$, $j = 1, \ldots, N-1$, is

$$t_j = \frac{L_j}{L} t^+. \tag{17}$$

Now, let us build a reference model

$$\dot{\mathbf{y}}^* = \mathbf{v}, \tag{18}$$

where $\mathbf{v} = [v_1 \dots v_k]^T$,

$$v_i = \frac{y_i^{*,j+1} - y_i^{*,j}}{t_j}, \tag{19}$$

where $y_i^{*,j}$ is a coordinate $i$ of the point $j$, $i = 1, \dots, k$, $j = 1, \dots, N-1$.

The mathematical model of the control object includes the reference model for trajectory generation

$$\begin{aligned} \dot{\mathbf{x}} &= \mathbf{f}(\mathbf{x}, \mathbf{u}), \\ \dot{\mathbf{y}}^* &= \mathbf{v}. \end{aligned} \tag{20}$$

The task requires the search for a control function, presented as (9), to minimise the value of the quality criterion (11). To address this control synthesis problem, the approach of machine learning control through symbolic regression is employed.

## 3. Network Operator Method

Symbolic regression allows finding a mathematical expression in the form of special code. Coding depends on the method chosen. To code a mathematical expression, symbolic regression uses an alphabet of elementary functions. Genetic programming (GP) [14] is the most popular symbolic regression method. GP codes mathematical expressions in the form of computation trees. Arguments of mathematical expressions are the leaves of the trees; functions are located in the nodes. The number of branches leaving the node is equal to the number of arguments of the function associated with this node. When performing the crossover operation, two codes exchange the branches exiting from the nodes selected as crossover points. After crossover, the length of the GP code may change which requires additional computational effort for analysis. Now, there are about twenty symbolic regression methods.

In this study the network operator (NOP) method developed by the authors is used [15]. The network operator method uses codes of mathematical expressions presented as directed graphs. Functions with one or two arguments form the alphabet of elementary functions. Functions with two arguments are commutative and associative, and have unit elements, so these functions with two arguments can be potentially used as functions with any number of arguments. If a function with two arguments has one argument then the second argument is a unit element of this function. In the network operator method, the source nodes of the directed graph contain arguments of the mathematical expression. The remaining nodes in the graph contain functions that require two arguments. The edges between the nodes in the graph are associated with functions that only require one argument.

### *3.1. Coding*

To illustrate this concept, let us explore an example of encoding a mathematical expression using the network operator method. Consider mathematical expressions

$$\begin{aligned} y_1 &= x_1 \exp(-q_1 x_2 + q_2) \sin(q_2 x_1 + q_1), \\ y_2 &= x_2 \exp(-q_2 x_1 + q_1) \cos(q_1 x_2 + q_2), \end{aligned} \tag{21}$$

where $x_1$ and $x_2$ are variables, and $q_1$ and $q_2$ are constant parameters.

In order to transform the mathematical expression into a code, let us employ the alphabet of functions:

(1) Functions with one argument

$$\begin{aligned} \mathrm{F}_1 = \{ & f_{1,1}(z) = z, f_{1,2}(z) = -z, f_{1,3}(z) = \exp(z), \\ & f_{1,4}(z) = \sin(z), f_{1,5}(z) = \cos(z)\}; \end{aligned} \tag{22}$$

(2) Functions with two arguments

$$\mathrm{F}_2 = \{f_{2,1}(z_1, z_2) = z_1 + z_2, f_{2,2}(z_1, z_2) = z_1 \cdot z_2\}. \tag{23}$$

The identity function $f_{1,1}(z)$, which outputs the same value as its input argument, should be among the functions with one argument.

Functions with two arguments should be commutative and 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 the function $f_{2,i}(z_1, z_2)$. In this case, 0 is a unit element for the summation function $f_{2,1}(z_1, z_2)$ and 1 is a unit element for the multiplication function $f_{2,2}(z_1, z_2)$.

In Figure 1, the directed graph of the NOP for a given mathematical Expression (21) is depicted.

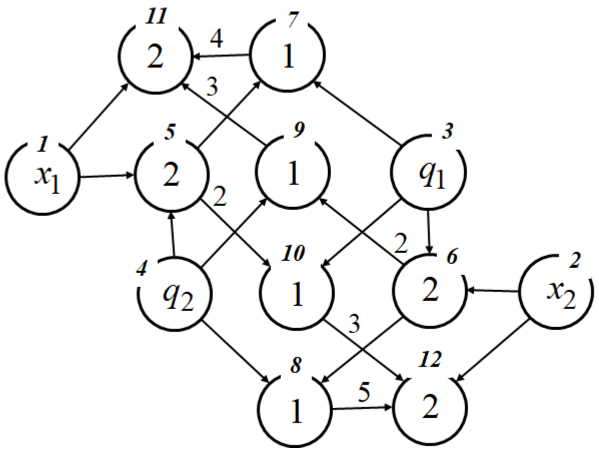

**Figure 1.** The graph of the NOP for (21).

The nodes are enumerated in their upper parts. The arguments are in the source nodes №1–№4. Nodes other than the source node contain numbers of functions with two arguments. Numbers of functions with a single argument are displayed over the edges. Nodes №11 and №12 are the sink nodes. When the node indices are arranged such that the index of the node where the edge originates is lower than the indices of the nodes where the edge terminates, the network operator matrix becomes upper triangular.

In the memory of a personal computer, the network operator is represented as an integer matrix that follows a structure similar to the adjacency matrix of the network operator graph. The network operator matrix corresponding to the graph shown in Figure 1 takes the following form

$$\mathbf{\Psi} = [\psi_{i,j}] = \begin{bmatrix} 0&0&0&0&1&0&0&0&0&0&1&0\\ 0&0&0&0&0&1&0&0&0&0&0&1\\ 0&0&0&0&0&1&1&0&0&1&0&0\\ 0&0&0&0&1&0&0&1&1&0&0&0\\ 0&0&0&0&2&0&1&0&0&2&0&0\\ 0&0&0&0&0&2&0&1&2&0&0&0\\ 0&0&0&0&0&0&1&0&0&0&4&0\\ 0&0&0&0&0&0&0&1&0&0&0&5\\ 0&0&0&0&0&0&0&0&1&0&3&0\\ 0&0&0&0&0&0&0&0&0&1&0&3\\ 0&0&0&0&0&0&0&0&0&0&2&0\\ 0&0&0&0&0&0&0&0&0&0&0&2 \end{bmatrix}, \; i,j = 1,\ldots,D = 12, \qquad (24)$$

where $D$ is the number of nodes in the graph and $\dim(\mathbf{\Psi}) = D \times D$.

In the matrix, rows that have zeros on the main diagonal are connected to source nodes. The other elements on the main diagonal that are not zero represent the numbers of functions with two arguments. The elements above the main diagonal, denoted as $\psi_{i,j} \neq 0$, are numbers of functions with one argument.

## 3.2. Decoding

To calculate the mathematical expression using the network operator, a vector of nodes is defined. Initially, this vector consists of the arguments of the mathematical expression and the unit elements of the corresponding functions with two arguments.

The initial vector of nodes for (24) is

$$\mathbf{z}^{(0)} = [x_1\ x_2\ q_1\ q_2\ 1\ 1\ 0\ 0\ 0\ 0\ 1\ 1]^T. \tag{25}$$

Further, all rows in the network operator matrix are followed sequentially and when a non-zero element occurs components of the vector of nodes are changed

$$z_j^{(i)} \leftarrow \begin{cases} f_{2,\psi_{j,j}}(z_j^{(i-1)}, f_{1,\psi_{i,j}}(z_i^{(i-1)})), \text{ if } \psi_{i,j} \neq 0 \\ z_j^{(i-1)}, \text{ otherwise} \end{cases}, \ i = 1,\ldots,D-1,\ j = i+1,\ldots,D, \tag{26}$$

where $\psi_{i,j}$ is an element of the network operator matrix in row $i$ and column $j$.

Each component of the vector of nodes contains the results of intermediate calculations. After viewing the $(i-1)$-th row of the component, the $z_i$ does not change.

For example, in row 1, the first non-zero element is $\psi_{1,5}$ =1. This means that the component $z_5$ of the vector of nodes changes. We define a function with two arguments by the matrix of the network operator, $\psi_{5,5} = 2$. Therefore, it is a function of $f_{2,2}(z_1, z_2)$ multiplication. The first argument of this function is the value of the current $z_5$ component of the node vector. The second component is determined by the value of a non-zero element, $\psi_{1,5} = 1$. This is a single argument function with the number 1, $f_{1,1}(z)$. The argument to this function is the value of the $z_1$ component of the node vector. As a result, we obtain

$$z_5^{(1)} \leftarrow f_{2,2}(z_5^{(0)}, f_{1,1}(z_1^{(0)})) = f_{2,2}(1, f_{1,1}(x_1)) = 1 \cdot x_1 = x_1.$$

The vector of nodes after viewing all rows of the network operator matrix has the following form

$$\mathbf{z}^{(11)} = \begin{bmatrix} x_1 \\ x_2 \\ q_1 \\ q_2 \\ q_2 x_1 \\ q_1 x_2 \\ q_2 x_1 + q_1 \\ q_1 x_2 + q_2 \\ -q_1 x_2 + q_2 \\ -q_2 x_1 + q_1 \\ x_1 \exp(-q_1 x_2 + q_2) \sin(q_2 x_1 + q_1) \\ x_2 \exp(-q_2 x_1 + q_1) \cos(q_1 x_2 + q_2) \end{bmatrix}. \tag{27}$$

The last two components are equal to the mathematical expressions (21).

## 3.3. Variational Genetic Algorithm

In order to find the mathematical expression that is optimal in some criterion using the network operator method, the variational genetic algorithm (VarGA) is employed. VarGA follows the principle of making slight changes, so-called small variations, to the initial solution [16]. The symbolic regression method is utilised to code one potential solution, known as the basic solution. Other solutions are presented as small variations of the basic solution. These variations are represented by an integer vector of four components

$$\mathbf{w} = [w_1\ w_2\ w_3\ w_4]^T, \tag{28}$$

where $w_1$ is a type of variation, $w_2$ is the row number, $w_3$ is the column number, $w_2 \leqslant w_3$ and $w_4$ is the new value of an element in the matrix.

Four types of small variations are defined for the network operator. The first type, denoted by $w_1 = 0$, involves substituting the function with a single argument: if $\psi_{w_2,w_3} \neq 0$, then $\psi_{w_2,w_3} \leftarrow w_4$.

The second type, $w_1 = 1$, is an exchange of the function with two arguments: if $\psi_{w_2,w_2} \neq 0$, then $\psi_{w_2,w_2} \leftarrow w_4$.

The third type, $w_1 = 2$, is an insertion of the additional function with one argument: if $\psi_{w_2,w_3} = 0$, then $\psi_{w_2,w_3} \leftarrow w_4$.

The fourth type, $w_1 = 3$, 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 $\psi_{w_2,w_3} \leftarrow 0$.

Consider an example of the vector of small variations for (24)

$$\mathbf{w} = [2\ 5\ 8\ 4]^T. \tag{29}$$

The first component shows that it is a small variation of type 3 changing a zero non-diagonal element. In the network operator matrix (24) element in the fifth row $w_2 = 5$, in the eighth column, $w_3 = 8$ is equal to zero, $\psi_{5,8} = 0$. After the small variation (29), this element is changed to $\psi_{5,8} = w_4 = 4$. The edge from node 5 to node 8 appears in the graph (see Figure 2). The new edge is dash lined.

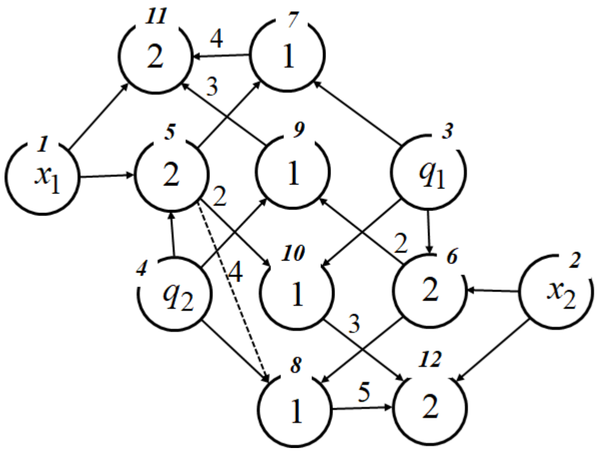

**Figure 2.** The graph of the NOP after the small variation with a new edge (dash line).

As a result a new mathematical expression is obtained

$$y_2 = x_2 \exp(-q_2 x_1 + q_1) \cos(q_1 x_2 + q_2 + \sin(x_1 q_2)). \tag{30}$$

All other possible solutions that originated from the basic solution are coded by ordered sets of vectors of small variations

$$\mathrm{W}_i = (\mathbf{w}^{i,1}, \ldots, \mathbf{w}^{i,d}), \tag{31}$$

where $i = 1, \ldots, H$, $H$ is a number of possible solutions in the population and $d$ is a depth of variation.

The crossover in VarGA is performed over ordered sets of small variations. Two possible solutions are selected randomly or as a result of tournament

$$\mathrm{W}_i = (\mathbf{w}^{i,1}, \ldots, \mathbf{w}^{i,d}),\ \mathrm{W}_j = (\mathbf{w}^{j,1}, \ldots, \mathbf{w}^{j,d}),\ i, j \in \{1, \ldots, H\}. \tag{32}$$

The crossover point is selected randomly, $c \in \{1, \ldots, d\}$. New possible solutions are obtained after the exchange of small variation vectors after the crossover point in the selected possible solutions

$$\begin{array}{rcl} W_{H+1} & = & (\mathbf{w}^{i,1}, \ldots, \mathbf{w}^{i,c}, \mathbf{w}^{j,c+1}, \ldots, \mathbf{w}^{j,d}), \\ W_{H+2} & = & (\mathbf{w}^{j,1}, \ldots, \mathbf{w}^{j,c}, \mathbf{w}^{i,c+1}, \ldots, \mathbf{w}^{i,d}). \end{array} \tag{33}$$

## 4. Computation Experiment

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

$$\begin{array}{rcl} \dot{x}_1 & = & x_4, \\ \dot{x}_2 & = & x_5, \\ \dot{x}_3 & = & x_6, \\ \dot{x}_4 & = & u_4(\sin(u_3)\cos(u_2)\cos(u_1) + \sin(u_1)\sin(u_2)), \\ \dot{x}_5 & = & u_4\cos(u_3)\cos(u_1) - g, \\ \dot{x}_6 & = & u_4(\cos(u_2)\sin(u_1) - \cos(u_1)\sin(u_2)\sin(u_3)), \end{array} \tag{34}$$

where $g = 9.80665$.

The initial state is

$$\mathbf{x}(0) = \mathbf{x}^0 = [0\ 5\ 0\ 0\ 0\ 0]^T. \tag{35}$$

The terminal state is

$$\mathbf{x}(t_f) = \mathbf{x}^f = [10\ 5\ 10\ 0\ 0\ 0]^T, \tag{36}$$

where

$$t_f = \begin{cases} t, \text{ if } t < t^+ \text{and } \|\mathbf{x}^f - \mathbf{x}\| \le \varepsilon_1 = 0.01 \\ t^+ = 14, \text{ otherwise} \end{cases}. \tag{37}$$

The quality criterion is

$$J_2 = \int_0^{t_f} 1dt = t_f \to \min. \tag{38}$$

For a given model of the control object, it is necessary to develop a stabilisation system for movement along a given spatial trajectory. The given trajectory consists of straight segments in 3D space. To define a spatial trajectory of straight connected segments, it is enough to define the locations of connection points. In the considered example, the coordinates of the connection points are as follows

$$\begin{array}{rcl} Y^* & = & \{\mathbf{y}^{*,1} = [0\ 5\ 0]^T,\ \mathbf{y}^{*,2} = [2\ 5\ 2]^T,\ \mathbf{y}^{*,3} = [8\ 5\ 2]^T, \\ & & \mathbf{y}^{*,4} = [2\ 5\ 8]^T,\ \mathbf{y}^{*,5} = [8\ 5\ 8]^T,\ y^{*,6} = [10\ 5\ 10]^T\} \end{array} \tag{39}$$

According to the proposed approach, initially, a reference model (18) is created.
The length of the trajectory is

$$L = \sum_{i=1}^{5} L_i = 2.82843 + 6 + 8.48528 + 6 + 2.82843 = 26.14213. \tag{40}$$

Equation (17) is used to calculate the values of the reference time intervals. The following time intervals were obtained: $t_1 = 1.515$, $t_2 = 3.213$, $t_3 = 4.544$, $t_4 = 3.213$ and $t_5 = 1.515$.

The control object with the reference model is

$$\begin{array}{rcl}
\dot{y}_1^* & = & v_1, \\
\dot{y}_2^* & = & v_2, \\
\dot{y}_3^* & = & v_3, \\
\dot{y}_4^* & = & 0, \\
\dot{y}_5^* & = & 0, \\
\dot{y}_6^* & = & 0, \\
\dot{x}_1 & = & x_4, \\
\dot{x}_2 & = & x_5, \\
\dot{x}_3 & = & x_6, \\
\dot{x}_4 & = & h_4(\mathbf{y}^* - \mathbf{x})(\sin(h_3(\mathbf{y}^* - \mathbf{x}))\cos(h_2(\mathbf{y}^* - \mathbf{x}))\times \\
 & & \cos(h_1(\mathbf{y}^* - \mathbf{x})) + \sin(h_1(\mathbf{y}^* - \mathbf{x}))\sin(h_2(\mathbf{y}^* - \mathbf{x}))), \\
\dot{x}_5 & = & h_4(\mathbf{y}^* - \mathbf{x})\cos(h_3(\mathbf{y}^* - \mathbf{x}))\cos(h_1(\mathbf{y}^* - \mathbf{x})) - g, \\
\dot{x}_6 & = & h_4(\mathbf{y}^* - \mathbf{x})(\cos(h_2(\mathbf{y}^* - \mathbf{x}))\sin(h_1(\mathbf{y}^* - \mathbf{x})) - \\
 & & \cos(h_1(\mathbf{y}^* - \mathbf{x}))\sin(h_2(\mathbf{y}^* - \mathbf{x}))\sin(h_3(\mathbf{y}^* - \mathbf{x}))),
\end{array} \tag{41}$$

where $\mathbf{h}(\mathbf{y}^* - \mathbf{x})$ is a required control function,

$$v_i = \frac{y_i^{*,j+1} - y_i^{*,j}}{t_j}, \text{ if } t_{j-1} \leqslant t < t_j, j = 1, \ldots, 5,\ t_0 = 0. \tag{42}$$

The current objective is to address the control synthesis problem and determine a control function based on the deviation between the state vector of the control object and that of the reference model

$$\mathbf{h}(\mathbf{y}^* - \mathbf{x}) = [h_1(\mathbf{y}^* - \mathbf{x}) \ldots h_4(\mathbf{y}^* - \mathbf{x})]^T. \tag{43}$$

Machine learning control by the network operator method is used. When solving the synthesis problem, the initial state (35) was replaced by a set of initial states

$$\mathrm{X}_0 = \{\mathbf{x}^{0,1}, \ldots, \mathbf{x}^{0,P}\}, \tag{44}$$

where

$$\mathbf{x}^{0,i+3j+9k+1} = [x_1^0 + (i-1)\Delta\ x_2^0 + (j-1)\Delta\ x_3^0 + (k-1)\Delta\ x_4^0\ x_5^0\ x_6^0]^T, \tag{45}$$

$i \in \{0,1,2\}, j \in \{0,1,2\}, k \in \{0,1,2\}, \Delta = 0.5$, for $i = j = k = 2$, $P = 27$.

When solving the synthesis problem, the error of movement along the given trajectory and the accuracy of reaching the terminal state for all initial states are additionally included in the criterion

$$J_3 = \sum_{i=1}^{P} \left( t_{f,i} + p_1 \int_0^{t_{f,i}} \|\mathbf{y}^* - \mathbf{y}(t, \mathbf{x}^{0,i})\| dt + p_2 \|\mathbf{x}^f - \mathbf{x}(t, \mathbf{x}^{0,i})\| \right) \to \min, \tag{46}$$

where $p_1 = 1$, $p_2 = 1$ and $t_{f,i}$ is a time to reach the terminal state from the initial state $\mathbf{x}^{0,i}$ according to (13).

Machine learning control by network operator method was performed with the following parameters: size of NOP matrix $36 \times 36$, graph of NOP had 12 source nodes, including 6 nodes for variables and 6 nodes for searched parameters, and 4 sink nodes for output components of control vector. Parameters of variational genetic algorithm were: number of possible solutions in initial population—512, number of crossover operations in one generation—64, number of generations—64, depth of variation—5, number of generations between change of basic solution—20, number of bits for coding a parameter—16.

Computations were performed on CPU Intel core i7, 2.8 GHz. Computational time was approx. 30 min.

The following solution was found by the network operator method

$$u_i = \begin{cases} u_i^+, \text{if } \hat{u}_i > u_i^+ \\ u_i^-, \text{if } \hat{u}_i < u_i^- \\ \hat{u}_i, \text{otherwise} \end{cases}, i = 1, \ldots, m = 4, \tag{47}$$

where

$$\hat{u}_1 = \mu(C), \tag{48}$$

$$\hat{u}_2 = \hat{u}_1 - \hat{u}_1^3, \tag{49}$$

$$\hat{u}_3 = \hat{u}_2 + \rho_{19}(W + \mu(C)) + \rho_{17}(A), \tag{50}$$

$$\hat{u}_4 = \hat{u}_3 + \ln(|\hat{u}_2|) + \text{sgn}(W + \mu(C))\sqrt{|W + \mu(C)|} + \rho_{19}(W) +$$

$$\arctan(H) + \text{sgn}(F) + \arctan(E) + \exp(q_2(y_2^* - x_2)) + \sqrt{q_1}, \tag{51}$$

$$C = q_6(y_6^* - x_6) + q_3(y_3^* - x_3),\ W = V + \tanh(G) + \exp(D),$$

$$A = q_1(y_1^* - x_1) + q_4(y_4^* - x_4),\ H = G + \tanh(F) + \rho_{18}(B),$$

$$F = E + C + \arctan(D) - B,\ E = D + \text{sgn}(y_5^* - x_5) + (y_2^* - x_2)^3,$$

$$V = \exp(H) + \cos(q_6(y_6^* - x_6)) + \text{sgn}(D)\sqrt{|D|},\ G = F + \sqrt[3]{E} + \sin(A),$$

$$B = \sin(q_6(y_6^* - x_6)) + q_5(y_5^* - x_5) + q_2(y_2^* - x_2) + \cos(q_1) + \vartheta(y_2^* - x_2),$$

$$D = \rho_{17}(C) + B^3 + A + \vartheta(q_5(y_5^* - x_5)) + (y_5^* - x_5)^2,$$

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

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

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

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

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

$q_1 = 7.26709$, $q_2 = 11.46143$, $q_3 = 12.77026$, $q_4 = 3.20630$, $q_5 = 8.38501$ and $q_6 = 5.56250$.

The projections of optimal trajectory (black line) and the given trajectory (blue line) on the horizontal and vertical planes are shown in Figures 3 and 4.

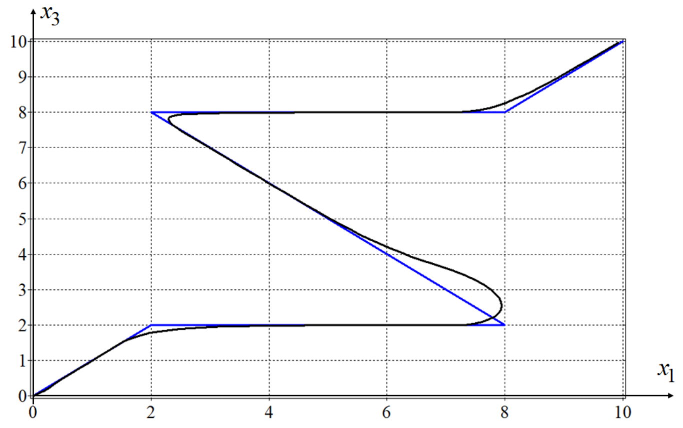

**Figure 3.** Projection of optimal trajectory (black line) and the given trajectory (blue line) on the horizontal plane $\{x_1; x_3\}$.

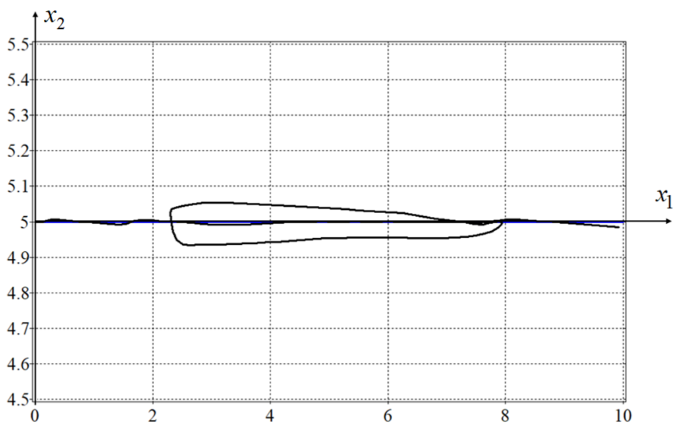

**Figure 4.** Projection of optimal trajectory (black line) and the given trajectory (blue line) on the vertical plane $\{x_1; x_2\}$.

To check the feasibility property of the obtained solution of the optimal control problem, a simulation of the control system from eight different initial states was performed: $\mathbf{x}^{0,1} = [-0.5\ 4.5\ \ -0.5\ 0\ 0\ 0]^T$, $\mathbf{x}^{0,2} = [-0.5\ 4.5\ 0.5\ 0\ 0\ 0]^T$, $\mathbf{x}^{0,3} = [-0.5\ 5.5\ \ -0.5\ 0\ 0\ 0]^T$, $\mathbf{x}^{0,4} = [-0.5\ 5.5\ 0.5\ 0\ 0\ 0]^T$, $\mathbf{x}^{0,5} = [0.5\ 4.5\ \ -0.5\ 0\ 0\ 0]^T$, $\mathbf{x}^{0,6} = [0.5\ 4.5\ 0.5\ 0\ 0\ 0]^T$, $\mathbf{x}^{0,7} = [0.5\ 5.5\ \ -0.5\ 0\ 0\ 0]^T$, $\mathbf{x}^{0,8} = [0.5\ 5.5\ 0.5\ 0\ 0\ 0]^T$.

The results of the simulation are shown in Figures 5 and 6.

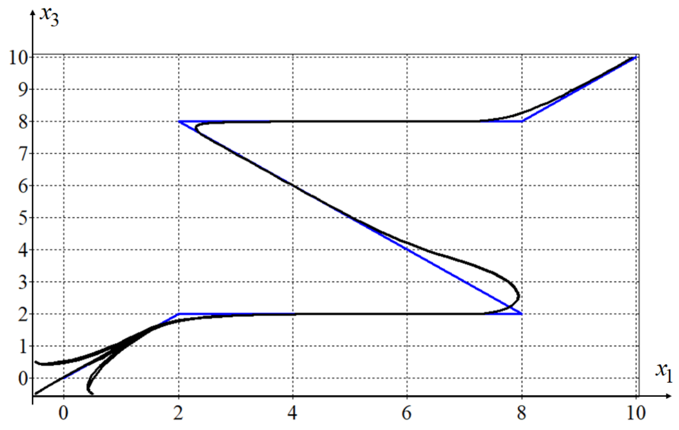

**Figure 5.** Projections of trajectories of control object from eight initial states (black lines) and the given trajectory (blue line) on the horizontal plane $\{x_1; x_3\}$.

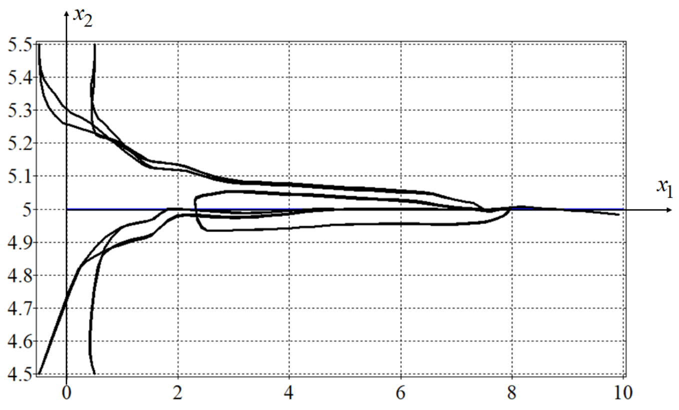

**Figure 6.** Projections of trajectories of control object from eight initial states (black lines) and the given trajectory (blue line) on the vertical plane $\{x_1; x_2\}$.

The results of the simulation show that the synthesised stabilisation system of the control object motion along the given spatial trajectory is not sensitive to disturbances; therefore it is realisable in a real object.

## 5. Discussion

The main difference of the method considered in this paper is that the trajectory stabilisation system is built entirely on the basis of machine learning by symbolic regression. Typically, a person analyses the mathematical model of the control object, identifies the control channels and determines which channels influence a particular movement of the object. Controllers are then installed in these channels and only then the parameters of the controllers are adjusted. This process is manual. A feature of machine learning control by symbolic regression is that the program is only provided with a model of the control object with a control function in the right part. The machine performs all other stages independently, without human intervention. It finds the control function in a coded form. The authors could not find a similar machine approach for the problem in other publications. Comparing the machine-learning-based approach discussed in the paper with control systems manually developed by specialists is not entirely correct, although it can be noted that the machine built a control system of equal quality. The disadvantage of the resulting control system is the complexity of the resulting mathematical expression, because the machine does not sense the complexity of calculations. We still insist that a machine search for solutions is more promising for building complex control systems than the manual approach.

In comparison to previous papers by the authors [9], in the present work, the complexity of the optimal control problem is that the given trajectory is defined in space and not in time. Instead of determining points on a given trajectory to calculate the deviation of an object from that trajectory as in prior approaches, a reference model is constructed that determines the reference motion along the trajectory. To solve the control synthesis problem, a machine learning control by symbolic regression is applied. The main goal of machine learning is to create a program to automatically write a control function. In our opinion, this approach will allow artificial intelligence to be created. Computational experiments have shown that the resulting control system has a feasibility property.

However, the proposed numerical method as all numerical methods has certain drawbacks. To obtain a solution it uses numerical techniques that include discretisation of the problem domain and approximation of the solution through an iterative computational process. Errors are accumulated and affect the final solution which is not exact, but it is still the only way to solve complex problems, such as the one presented in this paper.

Furthermore, the limitation of the proposed approach, namely, the usage of a reference model for trajectory generation, is that the velocity can be easily obtained for straight segments but might become a tricky task for differentiable trajectories.

## 6. Future Work

Future research is aimed at applying the presented approach to trajectories of different types. The applicability of this approach to differentiable trajectories, where the trajectory does not consist of straight segments, should be investigated. It is also necessary to investigate how the accuracy of the approximation of a smooth trajectory by straight segments affects the accuracy of the movement of the object along the trajectory, when the control synthesis problem has been solved for a reference model of movement on straight segments.

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

**Funding:** This research was funded by the Science Committee of the Ministry of Science and Higher Education of the Republic of Kazakhstan (Grant No. AP14869851).

**Data Availability Statement:** The data that support the findings of this study are openly available in https://cloud.mail.ru/public/rApd/Tv43xQ2QY, accessed on 1 December 2023.

**Conflicts of Interest:** The authors declare no conflicts of interest.

## References

1. 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]
2. 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]
3. Zhang, T.; Xue, J.; Jiao, X.; Wang, Z. Adaptive Finite Time Trajectory Tracking Control of Autonomous Vehicles. In Proceedings of the 2020 4th CAA International Conference on Vehicular Control and Intelligence (CVCI), Hangzhou, China, 18–20 December 2020; pp. 684–688.
4. Cui, C.; Zhu, D.; Sun, B. Trajectory Re-planning and Tracking Control of Unmanned Underwater Vehicles on Dynamic Model. In Proceedings of the 2018 Chinese Control and Decision Conference (CCDC), Shenyang, China, 9–11 June 2018; pp. 1971–1976.
5. Wang, R.; Xue, H.; Li, Z.; Li, H.; Wang, N.; Zhao, H. Fixed-Time Trajectory Tracking Control of an Unmanned Surface Vehicle. In Proceedings of the 2020 International Conference on System Science and Engineering (ICSSE), Kagawa, Japan, 31 August–3 September 2020; pp. 1–4.
6. 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]
7. 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]
8. Ma, T.; Wong, S. Trajectory Tracking Control for Quadrotor. UAV. In Proceedings of the 2017 IEEE International Conference on Robotics and Biomimetics (ROBIO), Macao, China, 5–8 December 2017.
9. Diveev, A.; Sofronova, E. Universal Stabilisation System for Control Object Motion along the Optimal Trajectory. *Mathematics* **2023**, *11*, 3556. [CrossRef]
10. Jordan, M.I.; Mitchell, T.M. Machine learning: Trends, perspectives, and prospects. *Science* **2015**, *349*, 255–260. [CrossRef] [PubMed]
11. Brunton, S. Machine Learning for Scientific Discovery. *Bull. Am. Phys. Soc.* **2023**, *68*. Available online: https://meetings.aps.org/Meeting/MAR23/Session/S13.1 (accessed on 1 December 2023).
12. Bensoussan, A.; Li, Y.; Nguyen, D.P.C.; Tran, M.-B.; Yam, S.C.P.; Zhou, X. Machine Learning and Control Theory. *arXiv* **2020**, arXiv:2006.05604.
13. Duriez, T.; Brunton, S.L.; Noack, B.R. *Machine Learning Control—Taming Nonlinear Dynamics and Turbulence*; Springer International Publishing: Cham, Switzerland, 2017.
14. Koza, J.R. *Genetic Programming: On the Programming of Computers by Means of Natural Selection*; MIT Press: Cambridge, MA, USA, 1992; 819p.
15. 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.
16. Sofronova, E.A.; Diveev, A.I. Universal Approach to Solution of Optimization Problems by Symbolic Regression. *Appl. Sci.* **2021**, *11*, 5081. [CrossRef]