




Stabilization of Movement along an Optimal Trajectory and Its Solution [†]

Askhat Diveev ^{1,*} , Elena Sofronova ¹ , Nurbek Konyrbaev ²  and Ainur Bexeitova ²

¹ Federal Research Center “Computer Science and Control” of the Russian Academy of Sciences, Vavilova Str., 44/2, 119333 Moscow, Russia

² Institute of Engineering and Technology, Korkyt Ata Kyzylorda University, Aiteke bi Str. 29A, Kyzylorda 120014, Kazakhstan

* Correspondence: aidiveev@mail.ru; Tel.: +7-905-711-4427

[†] Presented at the 15th International Conference “Intelligent Systems” (INTELS’22), Moscow, Russia, 14–16 December 2022.

Abstract: The extended optimal control problem is considered. It is necessary to find an optimal control function, that not only provides the achievement of terminal state with optimal value of the given quality criterion, but also is implemented in the control system of a real object. It means, that the control function should depend on the state space vector, and the optimal solution should keep optimality property at small perturbations of the found solution. To solve this problem machine learning control by symbolic regression is used. In the extended optimal control problem, the problem statement of stabilization system synthesis for movement along the optimal trajectory is included. Synthesis problem is solved by the network operator method. In the synthesis problem a domain of initial conditions is considered instead of one point of initial state. It provides less sensitivity of found solution to perturbations of initial states. An example of solving the extended optimal control problem with complex phase constraints in the form of bottleneck for four quadcopters is presented.

Keywords: optimal control; stabilization system; control synthesis; symbolic regression; evolutionary computations



Citation: Diveev, A.; Sofronova, E.; Konyrbaev, N.; Bexeitova, A. Stabilization of Movement along an Optimal Trajectory and Its Solution. *Eng. Proc.* **2023**, *33*, 12. <https://doi.org/10.3390/engproc2023033012>

Academic Editors: Ivan Zelinka, Arutun Avetisyan and Alexander Ilin

Published: 8 June 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

Experts in the field of control are aware that the solution of the optimal control problem in the classical statement [1] cannot be implemented directly in a real control object, even in the presence of a sufficiently accurate mathematical model of the control object. The main reason for it is that the solution to the classical optimal control problem is a time function independent from coordinates of the state space vector. Therefore, the use of the obtained solution in the control object results in an open-loop control system that is sensitive even to small disturbances. The solution of the optimal control problem is only needed to obtain the optimal program trajectory in the state space. To provide a movement of a control object along this optimal trajectory, it is necessary to develop an additional control system with feedback control. It should be noted here that the development of an additional control system changes the mathematical model of the object that was used in the optimal control problem. Thus, the optimal trajectory, obtained for a control object, when solving an optimal control problem, is no longer optimal for an object with a control system that provides movement along an optimal trajectory.

There are several ways to solve the optimal control problem and provide the movement of an object along the optimal trajectory in the state space. In practice, engineers first make these robots stable relative to some point in space, and then control the movement of the robots by repositioning these stable points [2,3]. Usually, they set these points along a given trajectory to allow robots to move along that trajectory. They do not solve the optimal problem, because the control object stops near a stable point, and the task is to follow these

points subsequently. To ensure stability in the state space, engineers usually apply linear P-, PI- or PID-regulators in a feedback loop.

To provide practical feasibility of the solution, it is possible to solve the general control synthesis problem instead of the optimal control problem. For this purpose, it is necessary to change one initial state into a domain of initial states and to look for a control function of the state space vector. The search for a structure of function of many arguments is a difficult task. Here, it is possible to use methods of symbolic regression [4,5]. This perspective direction allows the search of mathematical expressions in coded form by means of a special evolutionary genetic algorithm. As a result, we obtain rather complex mathematical expressions, but they can be directly applied in a real control object.

Another way to solve the optimal control problem that can be implemented in a real object is to use a synthesized control specially developed for this purpose [6]. According to this approach, firstly, the control synthesis problem is solved, and a control object becomes stable in the state space relative to some equilibrium point. After that, the positions of this stable equilibrium point are defined by the quality criterion of the optimal control problem so that, if these positions are switched over equal time intervals, a control object every time trends to the stable equilibrium point and moves along the optimal trajectory. As a result, it achieves the given terminal state with the optimal value of the given quality criterion.

In this work, the third way of solution is considered, when after solving the optimal control problem a stabilization system for motion along the optimal trajectory is obtained. This problem is a control system synthesis for tracking the given trajectory. According to this approach, a control object is included in the control system as a reference model for the trajectory generation needed for tracking. Unlike some known works on tracking [7–10], here the optimal trajectories obtained by solving the optimal control problem are considered and a tracking system synthesis problem is solved by means of symbolic regression. In this work, mathematical statements of the extended optimal control problem [11] and the tracking system synthesis problem are presented. In the experimental part, the optimal control problem for a group of four quadcopters is presented.

2. The Extended Optimal Control Problem

Consider the optimal control problem with additional requirements to the optimal trajectory.

The mathematical model of control object is given

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

where \mathbf{x} is a state-space vector of control object, $\mathbf{x} \in \mathbb{R}^n$, \mathbf{u} is a control vector, $\mathbf{u} \in U \subseteq \mathbb{R}^m$, U is a compact set.

A compact set U determines restrictions on the control and is often presented in the form of algebraic inequalities

$$u_i^- \leq u_i \leq u_i^+, \quad i = 1, \dots, m, \quad (2)$$

where u_i is a component of the control vector, $\mathbf{u} = [u_1 \dots u_m]^T$, u_i^-, u_i^+ are the given values, $i = 1, \dots, m$.

The initial state is given:

$$\mathbf{x}(0) = \mathbf{x}^0 \in \mathbb{R}^n. \quad (3)$$

The terminal state is given:

$$\mathbf{x}(t_f) = \mathbf{x}^f \in \mathbb{R}^n, \quad (4)$$

where t_f is a terminal time not given but limited, $t_f \leq t^+$, t_f is defined on achievement of the terminal state, t^+ is the given limit time of control process. If the control object does not reach the terminal state in t^+ , then it is considered that the control object will never reach the terminal state (4).

The quality criterion is given

$$J = \int_0^{t_f} f_0(\mathbf{x}, \mathbf{u}) dt \rightarrow \min_{\mathbf{u} \in U}. \quad (5)$$

It is necessary to find a control function in the following form

$$\mathbf{u} = \mathbf{v}(t), \quad t \in (0; t_f). \quad (6)$$

If we replace a control vector \mathbf{u} by the found control function $\mathbf{v}(t)$ in the right part of ODE system (1), then the obtained ODE system

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

will have a particular solution that reaches the given terminal state (4) from the given initial state (3) with the optimal value of the quality criterion (5).

Let $\mathbf{v}^*(t)$ be the optimal control function. In the second stage, consider the following ODE system

$$\begin{aligned} \dot{\mathbf{x}}^* &= \mathbf{f}(\mathbf{x}^*, \mathbf{v}^*(t)), \\ \dot{\mathbf{x}} &= \mathbf{f}(\mathbf{x}, \mathbf{u}). \end{aligned} \quad (8)$$

In (8), the first subsystem is the reference model that generates optimal trajectory. For the reference model, the initial state (3) is given.

For the second subsystem, a domain of initial states is given

$$X_0 \subseteq \mathbb{R}^n, \quad (9)$$

where X_0 is a compact set.

It is necessary to find a control function in the following form

$$\mathbf{u} = \mathbf{h}(\mathbf{x}^* - \mathbf{x}) \in U, \quad (10)$$

where \mathbf{x}^* is the given optimal trajectory or the particular solution of the reference system in (8) from the given initial state (3).

The control function should minimize the following quality criterion

$$J_1 = \int_{X_0} \cdots \int \max_t \|\mathbf{x}^*(t) - \mathbf{x}(t, \mathbf{y})\| dt \rightarrow \min_{\mathbf{h}(\mathbf{x}^* - \mathbf{x}) \in U}, \quad (11)$$

where $\mathbf{y} \in X_0$.

3. Symbolic Regression

To solve the synthesis problem and find the control function (10), symbolic regression is used.

Symbolic regression is a perspective computational method that allows to find mathematical expressions fully automatically. If someday humanity manages to create artificial intelligence, that is, write a program that will reflect and form the goals of the calculation, then this will be performed not manually but automatically. Symbolic regression is such an approach. It allows finding the structure and parameters of mathematical expressions automatically in a coded form. Previously, when the problem of finding a mathematical expression arose, for example, for deriving the physical laws, humans wrote mathematical expressions with accuracy to values of some constant parameters. Only for finding these parameters on experimental data, numerical methods were used, for example, the least-squares method. Note, that any artificial neural network is a function with a given structure and a big number of unknown parameters. The search of these parameters by optimization algorithms is called machine learning.

In this work, a network operator method is used to solve the tracking system synthesis problem and to find the control function as a function of a state space vector.

To code mathematical expressions, symbolic regression uses an alphabet of elementary functions. The network operator uses an alphabet of unary and binary functions and codes a mathematical expression in the form of an oriented computational graph. Consider an example of coding a mathematical expression by the network operator.

Let us be given a mathematical expression

$$y = q_1 \sin(x_1 + q_2 \cos(x_1 - x_2)). \quad (12)$$

The expression has two parameters q_1, q_2 and two variables x_1, x_2 . The alphabet of the following elementary functions is enough to present this mathematical expression.

A set of unary functions

$$F_1 = \{f_{1,1}(z) = z, f_{1,2}(z) = -z, f_{1,3}(z) = \sin(z), f_{1,4}(z) = \cos(z)\}, \quad (13)$$

A set of binary functions

$$F_2 = \{f_{2,1}(z_1, z_2) = z_1 + z_2, f_{2,2}(z_1, z_2) = z_1 z_2\}. \quad (14)$$

Elementary function designations contain two indexes. The first index is the number of arguments of the function, and the second index is its number in the set. Using elementary functions, expression (12) can be written as

$$y = f_{2,2}(q_1, f_{1,3}(f_{2,1}(x_1, f_{2,2}(q_2, f_{1,4}(f_{2,1}(x_1, f_{1,2}(x_2))))))). \quad (15)$$

In the network operator method, functions with two arguments $f_{2,i}(z_1, z_2)$ must be commutative, associative and have a unit element, e_i . If one of the arguments is equal to a unit element, then the result of the calculations of the function is equal to the second argument, $f_{2,i}(e_i, z_2) = z_2$. For the addition function, the unit element is zero, $e_1 = 0$, and for multiplication, the unit element is one, $e_2 = 1$.

Figure 1 presents the network operator for the mathematical expression (15). Arguments of the mathematical expression are placed in the source nodes of the graph, the numbers of the binary functions are placed in other nodes of the graph and the numbers of the unary functions are located over arcs. To present the network operator in the PC memory, all nodes must be numerated so that the node number from which the arc outputs is less than the node number where the arc inputs. It can be done if the graph has no loops. Node numbers are pointed in the upper parts of the nodes.

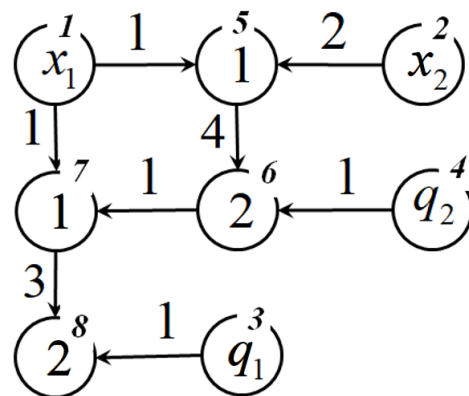


Figure 1. The network operator of the mathematical expression.

The network operator is presented in the PC memory in the form of an upper triangular integer network operator matrix. The network operator matrix for the considered mathematical expression is

$$\mathbf{\Psi} = \begin{bmatrix} 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 4 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 2 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 3 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 \end{bmatrix} \quad (16)$$

In the network operator matrix, the zero elements on the main diagonal point to the lines corresponding to the source nodes. The non-zero elements of the main diagonal are the binary function numbers. Non-zero elements above the main diagonal are the unary function numbers.

In order to calculate the mathematical expression by the network operator matrix, it is necessary to set a nodes vector of a dimension equal to the number of nodes of the network operator graph

$$\mathbf{z} = [z_1 \dots z_L]^T, \quad (17)$$

where $L \times L$ is a dimension of the network operator matrix.

Firstly, it is necessary to initialize the nodes vector. The components of the nodes vector corresponding to the source nodes are equal to the arguments of the mathematical expression, and the remaining components are equal to the unit elements of the corresponding binary functions.

Further, the values of the components of the nodes vector change after passing each line of the matrix and finding a non-zero non-diagonal element in it, as

$$z_j^{(i)} \leftarrow \begin{cases} f_{2,\psi_{i,j}}(z_j^{(i-1)}, f_{1,\psi_{i,j}})(z_i^{(i-1)}), & \text{if } \psi_{i,j} \neq 0 \\ z_j^{(i-1)} \end{cases}, \quad i = 1, \dots, L-1, \quad j = i+1, \dots, L, \quad (18)$$

where $\psi_{i,j}$ is an element of the network operator matrix $\mathbf{\Psi} = [\psi_{i,j}], i, j = 1, \dots, L$.

For the network operator matrix (16), a nodes vector has the following changes

$$\begin{aligned} \mathbf{z}^{(0)} &= [x_1 \ x_2 \ q_1 \ q_2 \ 0 \ 1 \ 0 \ 1]^T, \\ \mathbf{z}^{(1)} &= [x_1 \ x_2 \ q_1 \ q_2 \ x_1 \ 1 \ x_1 \ 1]^T, \\ \mathbf{z}^{(2)} &= [x_1 \ x_2 \ q_1 \ q_2 \ x_1 - x_2 \ 1 \ x_1 \ 1]^T, \\ \mathbf{z}^{(3)} &= [x_1 \ x_2 \ q_1 \ q_2 \ x_1 - x_2 \ 1 \ x_1 \ q_1]^T, \\ \mathbf{z}^{(4)} &= [x_1 \ x_2 \ q_1 \ q_2 \ x_1 - x_2 \ q_2 \ x_1 \ q_1]^T, \\ \mathbf{z}^{(5)} &= [z_1^{(4)} \dots z_5^{(4)} \ q_2 \cos(x_1 - x_2) \ x_1 \ q_1]^T, \\ \mathbf{z}^{(6)} &= [z_1^{(5)} \dots z_6^{(5)} \ \sin(x_1 + q_2 \cos(x_1 - x_2)) \ q_1]^T, \\ \mathbf{z}^{(7)} &= [z_1^{(6)} \dots z_7^{(6)} \ q_1 \sin(x_1 + q_2 \cos(x_1 - x_2))]^T. \end{aligned} \quad (19)$$

For more details on the network operator and the genetic algorithm that searches for the mathematical expression using the network operator, see the monographs [4,5].

4. Computational Experiment

Consider the optimal control problem for spatial movement of four similar quadcopters.

The mathematical model of control object is

$$\begin{aligned}
 \dot{x}_1^j &= x_4^j, \\
 \dot{x}_2^j &= x_5^j, \\
 \dot{x}_3^j &= x_6^j, \\
 \dot{x}_4^j &= u_4^j (\sin(u_3^j) \cos(u_2^j) \cos(u_1^j) + \sin(u_1^j) \sin(u_2^j)), \\
 \dot{x}_5^j &= u_4^j \cos(u_3^j) \cos(u_1^j) - g_c, \\
 \dot{x}_6^j &= u_4^j (\cos(u_2^j) \sin(u_1^j) - \cos(u_1^j) \sin(u_2^j) \sin(u_3^j)),
 \end{aligned} \tag{20}$$

where $j = 1, \dots, 4$, $g_c = 9.80665$.

The control is constrained

$$\begin{aligned}
 -\pi/12 &\leq u_1^j \leq \pi/12, \\
 -\pi &\leq u_2^j \leq \pi, \\
 -\pi/12 &\leq u_3^j \leq \pi/12, \\
 0 &\leq u_4^j \leq 12, \quad j = 1, \dots, 4.
 \end{aligned} \tag{21}$$

For the system (20), the initial states are given:

$$\begin{aligned}
 \mathbf{x}^{1,0} &= [0 \ 5 \ 0 \ 0 \ 0 \ 0]^T, & \mathbf{x}^{2,0} &= [10 \ 5 \ 0 \ 0 \ 0 \ 0]^T, \\
 \mathbf{x}^{3,0} &= [0 \ 5 \ 10 \ 0 \ 0 \ 0]^T, & \mathbf{x}^{1,0} &= [10 \ 5 \ 10 \ 0 \ 0 \ 0]^T.
 \end{aligned} \tag{22}$$

The terminal states are given

$$\begin{aligned}
 \mathbf{x}^{1,f} &= [10 \ 5 \ 10 \ 0 \ 0 \ 0]^T, & \mathbf{x}^{2,f} &= [0 \ 5 \ 10 \ 0 \ 0 \ 0]^T, \\
 \mathbf{x}^{3,f} &= [10 \ 5 \ 0 \ 0 \ 0 \ 0]^T, & \mathbf{x}^{1,f} &= [0 \ 5 \ 0 \ 0 \ 0 \ 0]^T.
 \end{aligned} \tag{23}$$

The static phase constraints are given

$$\varphi_i(\mathbf{x}^j) = r_i - \sqrt{(x_i - x_1^j)^2 + (z_i - x_3^j)^2} \leq 0, \tag{24}$$

where $i = 1, \dots, 4$, $j = 1, \dots, 4$, $r_1 = r_2 = r_3 = r_4 = 1.5$, $x_1 = 1.5$, $z_1 = 2.5$, $x_2 = 1.5$, $z_2 = 7.5$, $x_3 = 8.5$, $z_3 = 2.5$, $x_4 = 8.5$, $z_4 = 7.5$.

The dynamic phase constraints are given

$$\chi(\mathbf{x}^{j_1}, \mathbf{x}^{j_2}) = s - \sqrt{\sum_{i=1}^3 (x_i^{j_1} - x_i^{j_2})^2} \leq 0, \tag{25}$$

where $j_1, j_2 \in \{1, 2, 3, 4\}$, $j_1 \neq j_2$, $s = 1.5$.

Initially, the optimal control problem is solved by the direct approach. For this purpose, the time axis is divided into equal time intervals. In each interval, a control function is found as a piecewise-linear time function. It is necessary to find the values of the constant parameters on the boundaries of the time intervals. Taking into account the constraints on control, the control function is

$$u_i^j = \begin{cases} u_i^{j,+}, & \text{if } \tilde{u}_i^j > u_i^{j,+} \\ u_i^{j,-}, & \text{if } \tilde{u}_i^j < u_i^{j,-} \\ \tilde{u}_i^j, & \text{otherwise} \end{cases}, \tag{26}$$

where

$$\tilde{u}_i^j = q_{i+(k-1)m} + (q_{i+km} - q_{i+(k-1)m}) \frac{t - (k-1)\Delta t}{\Delta t}, \quad (k-1)\Delta t \leq t \leq k\Delta t, \quad (27)$$

$i = 1, \dots, m$, m is a dimension of the control vector, $m = 4$, $k = 1, \dots, N$, N is the number of intervals.

It is necessary to find the control functions for each control object as (26). The time limit of the control process is $t^+ = 5.6$. The time interval is $\Delta t = 0.4$. To solve the optimal control problem, it is necessary to find 240 parameters, $4 \cdot m \cdot (N + 1) = 4 \cdot 4 \cdot (\lfloor \frac{5.6}{0.4} \rfloor + 1) = 4 \cdot 4 \cdot 15 = 240$.

To solve the optimal control problem by direct approach, the evolutionary hybrid algorithm is used [6,12]. The algorithm includes evolutionary transformations of three evolutionary algorithms, the genetic algorithm (GA) [13], particle swarm optimization (PSO) algorithm [14] and grey wolf optimizer (GWO) algorithm [15].

Figure 2 shows projections of the found optimal trajectories on the horizontal plane. Here, the solid line is for the trajectory of the first robot, dashed line—the second robot, dot-dashed line—the third robot, dots—the fourth robot, and circles are the static phase constraints.

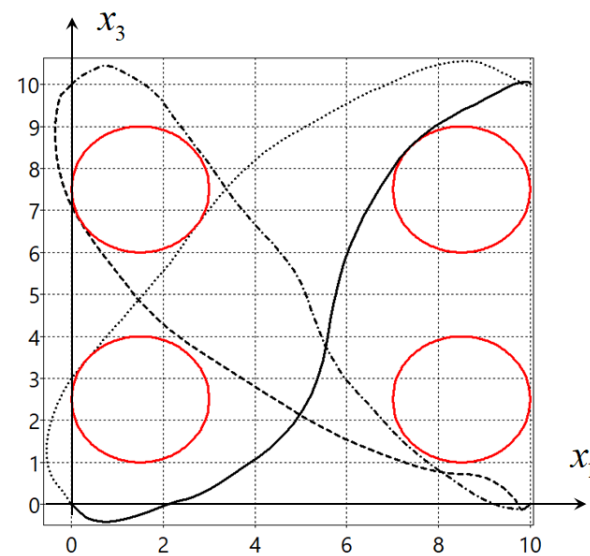


Figure 2. Optimal trajectories on the horizontal plane.

At the second stage, the tracking system synthesis problem is solved. To solve the synthesis problem, we used the network operator method.

The domain of initial states (9) was replaced by the set of points

$$x_i(0) = x_i^{j,0} \pm 0.2, \quad i = 1, 2, 3, \quad j = 1, \dots, 4. \quad (28)$$

In the quality criterion (11), an integral over domain was replaced by the sum of all $3^3 = 27$ initial states.

The following solution was found by the network operator method

$$u_i^j = \begin{cases} u_i^{j,+}, & \text{if } \hat{u}_i^j > u_i^{j,+} \\ u_i^{j,-}, & \text{if } \hat{u}_i^j < u_i^{j,-} \\ \hat{u}_i^j, & \text{otherwise} \end{cases} \quad (29)$$

where

$$\hat{u}_1^j = \mu(A), \quad (30)$$

$$\hat{u}_2^j = \mu(A) - \mu^3(A), \quad (31)$$

$$\hat{u}_3^j = \rho_{17}(\hat{u}_2^j) + \rho_{19}(B + \mu(A)) + \rho_{17}(C), \quad (32)$$

$$\begin{aligned} \hat{u}_4^j = & \hat{u}_3^j + \ln(\hat{u}_2^j) + \operatorname{sgn}(B + \mu(A))\sqrt{|B + \mu(A)|} + \rho_{19}(B) + \\ & \arctan(D) + \operatorname{sgn}(E) + \arctan(F) + \exp(q_2(x_2^{j*} - x_2^j)) + \sqrt{q_1}, \end{aligned} \quad (33)$$

$$A = q_3(x_3^{j*} - x_3^j) + q_6(x_6^{j*} - x_6^j),$$

$$B = G + \tanh(E + \sin(C)\sqrt[3]{F}) + \exp(H) - C,$$

$$C = q_1(x_1^{j*} - x_1^j) + q_4(x_4^{j*} - x_4^j)$$

$$D = E + \sin(C)\sqrt[3]{F} + \tanh(E) + \rho_{18}(V),$$

$$E = F + \arctan(A) + \arctan(H) - V,$$

$$F = H + \operatorname{sgn}(x_5^{j*} - x_5^j) + (x_2^{j*} - x_2^j)^3,$$

$$G = \exp(D) + \cos(q_6(x_6^{j*} - x_6^j)) + \operatorname{sgn}(E + \sin(C)\sqrt[3]{F})\sqrt{|E + \sin(C)\sqrt[3]{F}|} + E^{-1} - q_6,$$

$$H = \rho_{17}(A) + V^3 + C + \vartheta(q_5(x_5^{j*} - x_5^j)) + x_5^{j*} - x_5^j,$$

$$V = \sin(q_6(x_6^{j*} - x_6^j)) + q_5(x_5^{j*} - x_5^j) + \cos(q_1) + \vartheta(x_2^{j*} - x_2^j),$$

$$\vartheta(\alpha) = \begin{cases} 1, & \text{if } \alpha > 0 \\ 0, & \text{otherwise} \end{cases}, \quad \mu(\alpha) = \begin{cases} \alpha, & \text{if } |\alpha| < 1 \\ \operatorname{sgn}(\alpha), & \text{otherwise} \end{cases},$$

$$\rho_{17}(\alpha) = \operatorname{sgn}(\alpha) \ln(|\alpha| + 1), \quad \rho_{18}(\alpha) = \operatorname{sgn}(\alpha)(\exp(|\alpha|) - 1), \quad \rho_{19}(\alpha) = \operatorname{sgn}(\alpha) \exp(-|\alpha|),$$

$$q_1 = 11.52295, q_2 = 7.54395, q_3 = 15.45825, q_4 = 12.39771, q_5 = 15.94922, q_6 = 10.84229, \\ \mathbf{x}^{j*} = [x_1^{j*} \dots x_6^{j*}]^T \text{ is a program trajectory, generated by the reference model.}$$

Figure 3 shows the trajectories from eight random initial states of given domain (28). Here, blue lines mark trajectories from (3) and black lines mark trajectories from (28). As it can be seen from the experiment, the obtained control system (29) for optimal trajectory tracking guarantees the presence of an attracting neighborhood for the optimal trajectory in the extended optimal control problem.

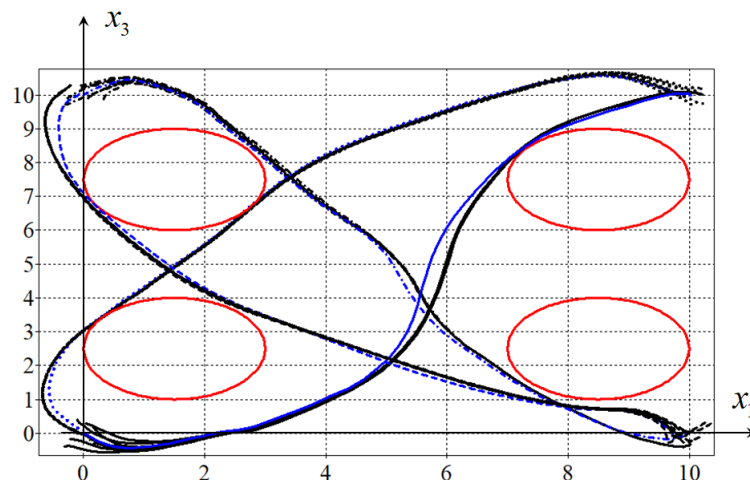


Figure 3. The optimal trajectory and disturbances trajectories from eight initial states.

5. Results

The paper presents the statements of the optimal control problem and the stabilization system synthesis with respect to the optimal program trajectory. To solve the synthesis problem, machine learning by the network operator method was used. A brief description of the network operator method is presented. A computational example is given for a group of four quadcopters. To analyze the quality of the obtained solutions, we simulated the synthesized stabilization system under perturbations of the initial state. The experiments showed the high quality of the stabilization system.

6. Discussion

The implementation of the control system is related to the solution of the control synthesis problem. As a result of solving the synthesis problem, we obtain a feasible feedback control. It is a complex problem that does not have universal computational methods. In this paper, we applied an approach based on the machine learning by the symbolic regression method. The method is universal, but requires lots of calculations. It is necessary to continue the study of this approach for other complex control problems.

Author Contributions: Conceptualization, A.D.; methodology, A.D.; software, A.D. and E.S.; validation, A.D., E.S., N.K. and A.B.; investigation, A.D. and E.S.; writing—original draft preparation, A.D., E.S. and N.K.; writing—review and editing, E.S.; visualization, A.B.; supervision, N.K.; project administration, N.K.; funding acquisition, N.K. 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 Education and Science of the Republic of Kazakhstan (Grant No. AP14869851).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Experimental data is available at: <https://cloud.mail.ru/public/L51J/oDPtP3Xhc> (accessed on 7 June 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; Volume 4, p. 360.
2. Egerstedt, M.; Hu, X.; Stotsky, A. Control of a Car-Like Robot Using a Virtual Vehicle Approach. *IEEE Trans. Robot. Autom.* **2001**, *17*, 947–951. [\[CrossRef\]](#)
3. Walsh, G.; Tilbury, D.; Sastry, S.; Murray, R.; Laumond, J.P. Stabilization of Trajectories for Systems with Nonholonomic Constraints. *IEEE Trans. Autom. Control* **1994**, *1*, 216–222. [\[CrossRef\]](#)
4. 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*; Gao, C., Ed.; Intech: Rijeka, Croatia, 2013; pp. 19–42.
5. Diveev, A.I.; Shmalko, E.Y. *Machine Learning Control by Symbolic Regression*; Springer: Cham, Switzerland, 2021; p. 155.
6. Diveev, A.; Sofronova, E. Synthesized Control for Optimal Control Problem of Motion Along the Program Trajectory. In Proceedings of the 8th International Conference on Control, Decision and Information Technologies (CoDIT), Istanbul, Turkey, 17–20 May 2022.
7. Udin, N. Trajectory Tracking Control System Design For Autonomous Two-Wheeled Robot. *Inform. Telecommun. Electron.* **2018**, *3*, 90–97. [\[CrossRef\]](#)
8. Samir, A.; Hammad, A.; Hafez, A.; Mansour, H. Quadcopter Trajectory Tracking Control using State-Feedback Control with Integral Action. *Int. J. Comput. Appl.* **2017**, *9*, 1–7. [\[CrossRef\]](#)
9. Siti, I.; Mjahed, M.; Ayad, H.; El Kari, A. New Trajectory Tracking Approach for a Quadcopter Using Genetic Algorithm and Reference Model Methods. *Appl. Sci.* **2019**, *9*, 1780. [\[CrossRef\]](#)
10. Dong, W.; Gu, G.-Y.; Zhu, X.; Ding, H. High-performance trajectory tracking control of a quadrotor with disturbance observer. *Sens. Actuators A Phys.* **2014**, *1*, 67–77. [\[CrossRef\]](#)
11. Diveev, A.I. Refined Optimal Control Problem and Its Solution Using Symbolic Regression. In *Lecture Notes in Networks and Systems*; Springer: Cham, Switzerland, 2022; Volume 507, pp. 294–305.
12. Diveev, A. Hybrid evolutionary algorithm for optimal control problem. In *Intelligent Systems and Applications*; Arai, K., Ed.; IntelliSys: Amsterdam, The Netherlands, 2022; Volume 543, pp. 726–738.

13. Davis, L. *Handbook of Genetic Algorithms*; Van Nostrand Reinhold: New York, NY, USA, 1991.
14. 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.
15. 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.