Article

Adaptive Synthesized Control for Solving the Optimal Control Problem

Askhat Diveev [†] [ID] **and Elizaveta Shmalko** *,[†] [ID]

Federal Research Center "Computer Science and Control" of the Russian Academy of Sciences, 44/2, Vavilova Str., Moscow 119333, Russia; aidiveev@frccsc.ru

* Correspondence: e.shmalko@frccsc.ru
† These authors contributed equally to this work.

Abstract: The development of artificial intelligence systems assumes that a machine can independently generate an algorithm of actions or a control system to solve the tasks. To do this, the machine must have a formal description of the problem and possess computational methods for solving it. This article deals with the problem of optimal control, which is the main task in the development of control systems, insofar as all systems being developed must be optimal from the point of view of a certain criterion. However, there are certain difficulties in implementing the resulting optimal control modes. This paper considers an extended formulation of the optimal control problem, which implies the creation of such systems that would have the necessary properties for its practical implementation. To solve it, an adaptive synthesized optimal control approach based on the use of numerical methods of machine learning is proposed. Such control moves the control object, optimally changing the position of the stable equilibrium point in the presence of some initial position uncertainty. As a result, from all possible synthesized controls, one is chosen that is less sensitive to changes in the initial state. As an example, the optimal control problem of a quadcopter with complex phase constraints is considered. To solve this problem, according to the proposed approach, the control synthesis problem is firstly solved to obtain a stable equilibrium point in the state space using a machine learning method of symbolic regression. After that, optimal positions of the stable equilibrium point are searched using a particle swarm optimization algorithm using the source functional from the initial optimal control problem statement. It is shown that such an approach allows for generating the control system automatically by computer, basing this on the formal statement of the problem and then directly implementing it onboard as far as the stabilization system has already been introduced.

Keywords: stabilization; optimization; symbolic regression; synthesized control; evolutionary computations; quadcopter model; ordinary differential equations

MSC: 49M99

1. Introduction

Long ago, Leonard Euler spoke about the optimal arrangement of everything in the world: "For since the fabric of the universe is most perfect and the work of a most wise Creator, nothing at all takes place in the universe in which some rule of maximum or minimum does not appear". Striving for optimality is natural in every sphere.

In order to optimally move an autonomous robot to a certain target position, currently, as a standard, engineers first solve the problem of optimal control, obtain the optimal trajectory, and then solve the additional problem of moving the robot along the obtained optimal trajectory. In most cases, the following approach is used to move the robot along a path. Initially, the object is made stable relative to a certain point in the state space. Then, the stability points are positioned along the desired path and the object is moved along the trajectory by following these points from one point to another [1–7]. The difference between

the existing methods is in solving the control synthesis problem to ensure stability relatively to some equilibrium point in the state space and in the location of these stability points.

Often, to ensure stability, the model of the control object is linearized relative to a certain point in the state space. Then, for the linear model of the object, a linear feedback control is found to arrange the eigenvalues of the closed-loop control system matrix on the left side of the complex plane. Sometimes, to improve the quality of stabilization, control channels or components of the control vector are defined that affect the movement of an object along a specific coordinate system axis of the state space. Then controllers, as a rule PI controllers, are inserted into these channels with the coefficients that are adjusted according to the specified control quality criterion [3,4]. In some cases, analytical or semi-analytical methods are used to solve the control synthesis problem and build nonlinear stable control systems [5,7]. But the stability property of the nonlinear model of the control object, obtained from the linearization of this model, is generally preserved only in the vicinity of a stable equilibrium point.

The main drawback of the approach when the control object is moved along the stable points on the trajectory is that even if this trajectory is obtained as a solution of the optimal control problem [8], then the movement itself will never be optimal. To ensure optimality, it is necessary to move along the trajectory at a certain speed, but when approaching the stable equilibrium point, the speed of the control object tends to zero.

The optimal control problem generally does not require ensuring the stability of the control object. The construction of a stabilization system that provides the stability of the object relative to some equilibrium point in the state space is carried out by the researcher to achieve predictable behavior of the control object in the vicinity of a given trajectory.

The optimal control problem in the classical formulation is solved for a control object without any stabilization system; therefore, the resulting optimal control and the optimal trajectory will not be optimal for this object with a further introduced stabilization system. It follows that the classical formulation of the optimal control problem [9] is missing something as far as its solution cannot be directly implemented in the real object, since this leads to an open-loop control. The open-loop control system is very sensitive to small disturbances, but they are always possible in real conditions, since no model accurately describes the control object. In order to achieve optimal control in a real object, it is necessary to build a feedback control system, which should provide some additional properties, for example stability relative to the trajectory or points on this trajectory. The authors of [10] proposed an extended formulation statement of the optimal control problem, which has additional requirements established for the optimal trajectory. The optimal trajectory must have a non-empty neighborhood with a property of attraction. Performing these requirements provides implementation of the solution of the optimal control problem directly in the real control object.

In [11,12], an approach to solving the extended optimal control problem on the base of the synthesized control is presented. This approach ensures obtaining a solution of the optimal control problem in the class of practically implemented control functions. According to this approach, initially, the control synthesis problem is solved. So, the control object becomes stable in the state space relatively to some equilibrium point. In the second stage, the optimal control problem is solved by determination of optimal positions of the stable equilibrium point. Switching stable points after a constant time interval ensures moving the control object from initial state to the terminal one optimally according to the given quality criterion. Optimal positions of stable equilibrium points can be far from the optimal trajectory in the state space; therefore, a control object does not slow down its motion speed. Studies of synthesized control in various optimal control problems have shown that such control is not sensitive to perturbations and can be directly implemented in a real object [13,14].

In synthesized control, the optimal control problem is solved for a control object already with a stabilization system. Another advantage of synthesized control is that the position of the stable point does not change during the time interval; that is, an optimal

control function is solved using the class of piece-wise constant functions, which simplifies the search for the optimal solution.

It is possible that piece-wise constant control in the synthesized approach finds several optimal solutions with practically the same value of the quality criterion. This circumstance prompted in us the idea to find among all almost-optimal solutions one that is less sensitive to perturbations. This approach is called adaptive synthesized control.

In this work, a principle of adaptive synthesized control is proposed in Section 2, methods for solving it are discussed in Section 3 and further in the Section 4, a computational experiment to determine the solution of the optimal control problem for the spatial motion of quadcopter by adaptive synthesized control is considered.

2. Adaptive Synthesized Control

Consider the principle of adaptive synthesized control for solving the optimal control problem in its extended formulation [10].

Initially, the control synthesis problem is solved to provide stability of the control object relatively some point in the state space. In the problem, the mathematical model of the control object in the form of ordinary differential equation system is given.

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

where $\mathbf{x}$ is a state vector, $\mathbf{x} \in \mathbb{R}^n$, $\mathbf{u}$ is a control vector, $\mathbf{u} \in U \subseteq \mathbb{R}^m$, U is a compact set that determines restrictions on the control vector.

The domain of admissible initial states is given

$$X_0 \subseteq \mathbb{R}^n. \tag{2}$$

To solve the problem numerically, the initial domain (2) is taken in the form of the finite number of points in the state space:

$$\tilde{X}_0 = \{\mathbf{x}^{0,1}, \dots, \mathbf{x}^{0,K}\}. \tag{3}$$

Sometimes, it is convenient to set one initial state and deviations from it:

$$\mathbf{x}^{0,j} = \mathbf{x}^0 - \mathbf{\Delta}_0 + 2 \odot (j)_2 \mathbf{\Delta}_0, \; j = 1, \dots, 2^n - 1, \tag{4}$$

where $\mathbf{x}^0$ is a given initial state, $\mathbf{\Delta}_0$ is a deviations vector, $\mathbf{\Delta}_0 = [\Delta_1 \dots \Delta_n]^T$, $\odot$ is Hadamard product of vectors, $(j)_2$ is a binary code of the number j. In this case $K = 2^n - 1$.

The stabilization point as a terminal state is given by

$$\mathbf{x}^{f_1} \in \mathbb{R}^n. \tag{5}$$

It is necessary to find a control function in the form

$$\mathbf{u} = \mathbf{h}(\mathbf{x}^{f_1}\mathbf{x}) \in U, \tag{6}$$

where $\mathbf{h}(\mathbf{x}) \colon \mathbb{R}^n \to \mathbb{R}^m$, such that it minimizes the quality criterion

$$J_0 = \sum_{i=0}^{K} \left(t_{f_1,i} + p \| \mathbf{x}^{f_1} - \mathbf{x}(t_{f_1,i}, \mathbf{x}^{0,i}) \| \right) \to \min, \tag{7}$$

where $t_{f_1,i}$ is the time of achieving the terminal state (5) from the initial state $\mathbf{x}^{0,i}$, $t_{f_1,i}$ is determined by an equation

$$t_{f_1,i} = \begin{cases} t, \text{if } t < t^+ \text{and } \| \mathbf{x}^{f_1} - \mathbf{x}(t, \mathbf{x}^{0,i}) \| \leq \varepsilon_0 \\ t^+, \text{otherwise} \end{cases}, \tag{8}$$

$\mathbf{x}(t, \mathbf{x}^{0,i})$ is a particular solution from initial state $\mathbf{x}^{0,i}$, $i = 1, \dots, K$, of the differential Equation (1) with an inserted control function (6)

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{h}(\mathbf{x}^{f_1} - \mathbf{x})), \qquad (9)$$

ε_0 is a given accuracy for hitting to terminal state (5), t^+ is a given maximal time for control process, p is a weight coefficient.

Further, using the principles of synthesized optimal control the following optimal control problem is considered. The model of control object in the form (9) is used

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{h}(\mathbf{x}^* - \mathbf{x})), \qquad (10)$$

where the terminal state vector (5) is changed into the new unknown vector $\mathbf{x}^*$, which will be a control vector in the considered optimal control problem.

In accordance with the classical formulation of the optimal control problem, the initial state of the object (10) is given

$$\mathbf{x}^0 \in \mathbb{R}^n. \qquad (11)$$

In the engineering practice, there can be some deviations in the initial position; therefore, in adaptive synthesized control, instead of one initial state (11) the set of initial states used are defined by Equation (4). The vector of initial deviations $\mathbf{\Delta}_0$ is defined as a level of disturbances.

The goal of control is defined by achievement of the terminal state

$$\mathbf{x}^f \in \mathbb{R}^n. \qquad (12)$$

The quality criterion is given

$$J_1 = \int_0^{t_f} f_0(\mathbf{x}, \mathbf{h}(\mathbf{x}^* - \mathbf{x}))dt \to \min, \qquad (13)$$

where t_f is a terminal time, t_f is not given but is limited, $t_f \leq t^+$, t^+ is a given limit time of control process.

According to the principle of synthesized control, it is necessary to choose time interval Δt and to search for optimal constant values of the control vector $\mathbf{x}^{*,i}$ for each interval

$$\mathbf{x}^* = \mathbf{x}^{*,i}, \text{ if } (i-1)\Delta t \leq t < i\Delta t, \ i = 1 \dots, M, \qquad (14)$$

where M is a number of intervals

$$M = \left\lfloor \frac{t^+}{\Delta t} \right\rfloor. \qquad (15)$$

So the system (10) with the found optimal constant values of the control vector (14) in the right-hand side of differential equations has a particular solution which reaches the terminal state (12) from the given initial state (11) with an optimal value of the quality criterion (13).

Algorithmically, in the second stage of the adaptive synthesized control approach, the optimal values of the vector $\mathbf{x}^*$ are found as a result of the optimization task with the following quality criterion, which takes into account the given grid according to the initial conditions:

$$J_2 = \sum_{i=1}^{K} \left(\int_0^{t_{f,i}} f_0(\mathbf{x}, \mathbf{h}(\mathbf{x}^{*,i} - \mathbf{x}))dt + p\|\mathbf{x}^f - \mathbf{x}(t_{f,i}, \mathbf{x}^{0,i})\| \right) \to \min_{\mathbf{x}^*}, \qquad (16)$$

where K is number of initial states, $t_{f,i}$ is determined by Equation (8).

3. Methods of Solving

As described in the previous section, the approach based on the principle of adaptive synthesized optimal control consists of two stages.

To implement the first stage of the approach under consideration for solving the control synthesis problem (1)–(9), any known method can be used. For linear systems, for example, methods of modal control [15] can be applied, as well as such analytical methods such as backstepping [16,17] or synthesis based on the application of the Lyapunov function [18]. In practice, stability is ensured through linearization of the model (1) in the terminal state and setting PI or PID controllers in control channels [19,20]. All known analytical and technical methods have their limitations, which mostly depend on the type of the model used to describe the control object. The mathematical formulation of the stabilization problem as a control synthesis problem is needed to apply numerical methods and automatically obtain a feedback control function. Today, to solve the synthesis problem for nonlinear dynamic objects of varying complexity, modern numerical methods of machine learning can be applied [21]. Among different machine learning techniques, only symbolic regression allows searching both for the structure of the needed mathematical function and its parameters. In our case, the needed function is a control function. So, in the present paper machine learning by symbolic regression [22,23] is used.

Methods of symbolic regression search for the mathematical expression of the desired function in the encoded form. These methods differ in the form of this code. The search for solutions is performed in the space of codes by a special genetic algorithm.

Let us demonstrate the main features of symbolic regression on the example of the network operator method (NOP), which was used in this work in the computational experiment. To code a mathematical expression NOP uses an alphabet of elementary functions:

– Functions without arguments or parameters and variables of the mathematical expression

$$\mathrm{F}_0 = \{f_{0,1} = x_1, \dots, f_{0,n} = x_n, f_{0,n+1} = q_1, \dots, f_{0,n+r} = q_{n+r}\}; \tag{17}$$

– Functions with one argument

$$\mathrm{F}_1 = \{f_{1,1}(z) = z, f_{1,2}(z), \dots, f_{1,W}(z)\}; \tag{18}$$

– Functions with two arguments

$$\mathrm{F}_2 = \{f_{2,1}(z_1, z_2), \dots, f_{2,V}(z_1, z_2)\}. \tag{19}$$

Any elementary function is coded by two digits: the first one is the number of arguments, the second one is the function number in the corresponding set. These digits are written as indexes of elements in the introduced sets of the alphabet (17)–(19). The set of functions with one argument must include the identity function $f_{1,1}(z) = 1$. Functions with two arguments should be commutative, associative and have a unit element.

NOP encodes a mathematical expression in the form of an oriented graph. Source-nodes of the NOP-graph are connected with functions without arguments, while other nodes are connected with functions with two arguments. Arcs of the NOP-graph are connected with functions with one argument. If on the NOP-graph some node has one input arc, then the second argument is a unit element for the function with two arguments connected with this node.

Let us define the following alphabet of elementary functions:

$$
\begin{aligned}
\mathrm{F}_0 &= \{f_{0,1} = x_1, f_{0,2} = x_2, f_{0,3} = q_1, f_{0,4} = q_2\}; \\
\mathrm{F}_1 &= \{f_{1,1}(z) = z, f_{1,2}(z) = -z, f_{1,3}(z) = \cos(z), f_{1,4}(z) = \sin(z)\}; \\
\mathrm{F}_2 &= \{f_{2,1}(z_1, z_2) = z_1 + z_2, f_{2,2}(z_1, z_2) = z_1 z_2\}.
\end{aligned}
\tag{20}
$$

With this alphabet the following mathematical expressions can be encoded in the form of NOP:

$$
\begin{aligned}
y_1 &= \cos(x_1)\sin(x_2) - \sin(x_1)\cos(x_2); \\
y_2 &= \cos(q_1 x_1 + q_2 \sin(x_2)); \\
y_3 &= q_1 \sin(q_2 x_2) + q_2 \cos(q_1 x_1); \\
y_4 &= q_1 \sin(q_2 \cos(x_1)) + q_2 \cos(q_1 \sin(x_2)).
\end{aligned}
\tag{21}
$$

The NOP-graphs of these mathematical expressions are presented in Figure 1. The nodes of the graph are numbered. Inside each node there is either the number of a binary operation or an element of the set of variables and parameters $F0$, and the arcs of the graph indicate the numbers of unary operations.

In the computer memory, the NOP-graphs are presented in the form of integer matrices.

$$
\mathbf{\Psi} = [\psi_{i,j}], \quad i, j = 1, \dots, L.
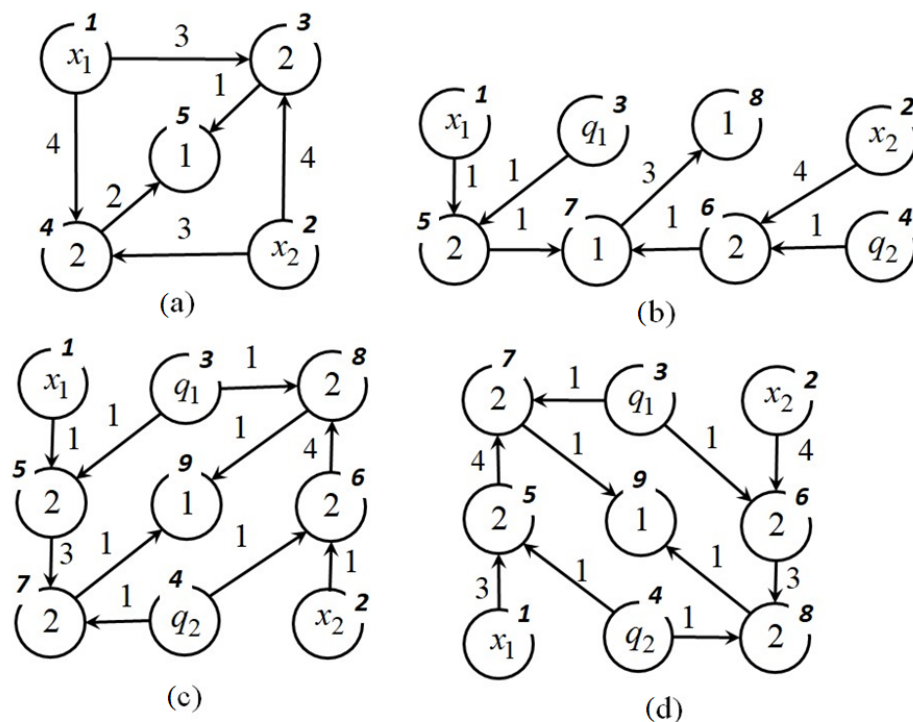\tag{22}
$$

Figure 1. NOP-graphs for mathematical expressions (21), (a) y_1, (b) y_2, (c) y_3, (d) y_4.

As the NOP-nodes are enumerated in such a way that the node number from which an arc comes out is less than the node number to which an arc enters, then the NOP-matrix has an upper triangular form. Every line of the matrix corresponds some node of the graph. Lines with zeros in the main diagonal corresponds to source-nodes of the graph. Other elements in the main diagonal are the function numbers with two arguments. Non-zero elements above the main diagonal are the function numbers with one argument.

NOP-matrices for the mathematical expressions (21) have the following forms:

$$
\mathbf{\Psi}_1 =
\begin{bmatrix}
0 & 0 & 3 & 4 & 0 \\
0 & 0 & 4 & 3 & 0 \\
0 & 0 & 2 & 0 & 2 \\
0 & 0 & 0 & 2 & 1 \\
0 & 0 & 0 & 0 & 1
\end{bmatrix},
\quad
\mathbf{\Psi}_2 =
\begin{bmatrix}
0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 4 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 2 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 2 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 3 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1
\end{bmatrix},
$$

$$\Psi_3 = \begin{bmatrix} 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 2 & 0 & 3 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 2 & 0 & 4 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 2 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}, \; \Psi_4 = \begin{bmatrix} 0 & 0 & 0 & 0 & 3 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 4 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 2 & 0 & 4 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 2 & 0 & 3 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 2 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}. \quad (23)$$

To calculate a mathematical expression by its NOP-matrix, initially, the vector of nodes is determined. The number of components of the vector of nodes equals to the number of nodes in a graph. The initial vector of nodes includes variables and parameters in positions that correspond to source nodes, as well as other components equal to the unit elements of the corresponding functions with two arguments. Further, every line of the matrix is checked. If element of the matrix does not equal zero, then corresponding element of the vector of nodes is changed. To calculate mathematical expression by the NOP-matrix, the following equation is used:

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

where

$$z_i^{(0)} = \begin{cases} f_{0,i}, \text{ if } \psi_{i,i} = 0 \\ e_{\psi_{i,i}}, \text{ otherwise} \end{cases}, \quad (25)$$

e_j is a unit element for function with two arguments $f_{2,j}(z_1, z_2)$,

$$f_{2,j}(e_j, z) = f(z, e_j) = z. \quad (26)$$

Consider an example of calculating the second mathematical expression in (21) on its NOP-matrix Ψ_2.

The initial vector of nodes is

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

Further, all strings in the matrix Ψ_2 are checked and non-zero elements are found.

$$\psi_{1,5} = 1, \; z_5^{(1)} = f_{2,2}(z_5^{(0)}, f_{1,1}(z_1^{(0)})) = 1 \cdot f_{1,1}(z_1^{(0)}) = 1 \cdot x_1 = x_1,$$

$$\psi_{2,6} = 4, \; z_6^{(2)} = f_{2,2}(z_6^{(1)}, f_{1,4}(z_1^{(1)})) = 1 \cdot f_{1,4}(z^{(1)}) = 1 \cdot \sin(x_2) = \sin(x_2),$$

$$\psi_{3,5} = 1, \; z_5^{(3)} = f_{2,2}(z_5^{(2)}, f_{1,1}(z_3^{(2)})) = x_1 \cdot q_1 = q_1 x_1,$$

$$\psi_{4,6} = 1, \; z_6^{(4)} = f_{2,2}(z_6^{(3)}, f_{1,1}(z_4^{(3)})) = \sin(x_2) \cdot q_2 = q_2 \sin(x_2),$$

$$\psi_{5,7} = 1, \; z_7^{(5)} = f_{2,1}(z_7^{(4)}, f_{1,1}(z_5^{(4)})) = 0 + q_1 x_1 = q_1 x_1,$$

$$\psi_{6,7} = 1, \; z_7^{(6)} = f_{2,1}(z_7^{(5)}, f_{1,1}(z_6^{(5)})) = q_1 x_1 + q_2 \sin(x_2),$$

$$\psi_{7,8} = 3, \; z_8^{(7)} = f_{2,1}(z_8^{(6)}, f_{1,3}(z_7^{(6)})) = 0 + \cos(q_1 x_1 + q_2 \sin(x_2)) = \cos(q_1 x_1 + q_2 \sin(x_2)).$$

The last mathematical expression coincides with the needed mathematical expression for y_2 (21).

So, we considered the way of coding in the NOP method. Then, to search for an optimal mathematical expression in some task, the NOP method applies a principle of

small variations of a basic solution. According to this principle, one possible solution is encoded in the form of the NOP-matrix $\mathbf{\Psi}_0$. This solution is the basic solution and it is set by a researcher as a good solution. Other possible solutions are presented in the form of sets of small-variation vectors. A small variation vector consists of four integer numbers

$$\mathbf{w} = [w_1 \; w_2 \; w_3 \; w_4]^T, \tag{27}$$

where w_1 is a type of small variation, w_2 is a line number of the NOP-matrix, w_3 is a column number of NOP-matrix, w_4 is a new value of an NOP-matrix element. There are four types of small variations: $w_1 = 0$ 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 $\psi_{w_2,w_3} \leftarrow 0$.

The initial population includes H possible solutions. Each possible solution $i \in \{1, \dots, H\}$ except the basic solution is encoded in the form of the set of small variation vectors

$$W_i = (\mathbf{w}^{i,1}, \dots, \mathbf{w}^{i,d}), \; i \in \{1, \dots, H\}, \tag{28}$$

where d is a depth of variations, which is set as a parameter of the algorithm.

The NOP-matrix of a possible solution is determined after application of all small variations to the basic solution

$$\mathbf{\Psi}_i = \mathbf{w}^{i,d} \circ \dots \circ \mathbf{w}^{i,1} \circ \mathbf{\Psi}_0, \; i \in \{1, \dots, H\}, \tag{29}$$

Here, the small variation vector is written as a mathematical operator changing matrix $\mathbf{\Psi}_0$.

During the search process sometimes the basic solution is replaced by the current best possible solution. This process is called a change of an epoch.

Consider an example of applying small variations to the NOP-matrix $\mathbf{\Psi}_3$. Let $d = 3$ and there are three following small variation vectors:

$$\mathbf{w}^1 = [0 \; 3 \; 5 \; 2]^T, \; \mathbf{w}^2 = [2 \; 5 \; 6 \; 3]^T, \; \mathbf{w}^3 = [2 \; 8 \; 9 \; 4]^T$$

After application of these small variation vectors to the NOP-matrix $\mathbf{\Psi}_3$, the following NOP-matrix is obtained:

$$\mathbf{\Psi}_5 = \begin{bmatrix} 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 2 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 2 & 3 & 3 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 2 & 0 & 4 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 2 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 & 4 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}.$$

This NOP-matrix corresponds to the following mathematical expression:

$$y_5 = \sin(q_1 \sin(q_2 x_2 \cos(-q_1 x_1))) + q_2 \cos(-q_1 x_1).$$

Similar to a search engine, a genetic algorithm is used. To perform the main genetic operation of crossover, two possible solutions are selected randomly

$$\begin{aligned} W_\alpha &= (\mathbf{w}^{\alpha,1}, \dots, \mathbf{w}^{\alpha,d}), \\ W_\beta &= (\mathbf{w}^{\beta,1}, \dots, \mathbf{w}^{\beta,d}). \end{aligned} \tag{30}$$

A crossover point is selected randomly $c \in \{1, \ldots, d\}$. Two new possible solutions are obtained as the result of exchanging elements of the selected possible solutions after the crossover point:

$$
\begin{aligned}
W_{H+1} &= (\mathbf{w}^{\alpha,1}, \ldots, \mathbf{w}^{\alpha,c}, \mathbf{w}^{\beta,c+1}, \ldots, \mathbf{w}^{\beta,d}), \\
W_{H+2} &= (\mathbf{w}^{\beta,1}, \ldots, \mathbf{w}^{\beta,c}, \mathbf{w}^{\alpha,c+1}, \ldots, \mathbf{w}^{\alpha,d}).
\end{aligned}
\tag{31}
$$

The second stage of the synthesized principle under consideration is to solve the problem of optimal control via determination of the optimal position of the equilibrium points. Studies have shown that for a complex optimal control problem with phase constraints, evolutionary algorithms allow the system to cope with such problems. Good results were demonstrated [24] by such algorithms as a genetic algorithm (GA) [25], a particle swarm optimization (PSO) algorithm [26–28], a grey wolf optimizer (GWO) algorithm [29] or a hybrid algorithm [24] involving one population of possible solutions and all three evolutionary transformations of GA, PSO and GWO selected randomly.

4. Computational Experiment

Consider the optimal control problem for the spatial motion of a quadcopter. In the problem, the quadcopter should move for a minimum time on a closed-loop circle from the given initial state to the same terminal state, avoiding collisions with obstacles and passing through the given areas.

4.1. Mathematical Model of Spatial Movement of Quadcopter

In the general case, the mathematical model of a quadcopter as a hard body has the following form:

$$
\begin{aligned}
\ddot{x} &= F(\cos(\gamma)\sin(\theta)\cos(\psi) + \sin(\gamma)\sin(\psi))/m; \\
\ddot{y} &= F\cos(\gamma)\cos(\theta)/m - g; \\
\ddot{z} &= F(\cos(\gamma)\sin(\theta)\sin(\psi) + \sin(\gamma)\cos(\psi))/m; \\
\ddot{\gamma} &= ((I_{yy} + I_{zz})\dot{\theta}\dot{\psi} + M_x)/I_{xx}; \\
\ddot{\psi} &= ((I_{zz} + I_{xx})\dot{\gamma}\dot{\theta} + M_y)/I_{yy}; \\
\ddot{\theta} &= ((I_{xx} + I_{yy})\dot{\gamma}\dot{\theta} + M_z)/I_{zz}.
\end{aligned}
\tag{32}
$$

where F is a summary thrust force of all drone screws, m is a mass of drone, g is acceleration of gravity, $g = 9.80665$, M_x, M_y, M_z are control moments around the respective axes.

Figure 2 shows how the angles of a quadcopter turn are linked with its axes.
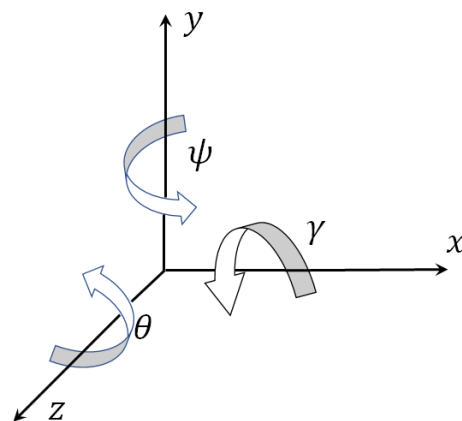

Figure 2. Inertial coordinate system for quadcopter.

To transform the model (32) to a vector record, the following designations are entered: $x = x_1, y = x_2, z = x_3, \dot{x}_1 = x_4, \dot{x}_2 = x_5, \dot{x}_3 = x_6, \gamma = x_7, \psi = x_8, \theta = x_9, \dot{\gamma} = x_{10}, \dot{\psi} = x_{11}, \dot{\theta} = x_{12}, M_1 = M_x, M_2 = M_y, M_3 = M_z$.

As a result the following mathematical model is received:

$$
\begin{aligned}
\dot{x}_1 &= x_4; \\
\dot{x}_2 &= x_5; \\
\dot{x}_3 &= x_6; \\
\dot{x}_4 &= F(\cos(x_7)\sin(x_9)\cos(x_8) + \sin(x_7)\sin(x_8))/m; \\
\dot{x}_5 &= F(\cos(x_7)\cos(x_9))/m - g; \\
\dot{x}_6 &= F(\cos(x_7)\sin(x_9)\sin(x_8) + \sin(x_7)\cos(x_8))/m; \\
\dot{x}_7 &= x_{10}; \\
\dot{x}_8 &= x_{11}; \\
\dot{x}_9 &= x_{12}; \\
\dot{x}_{10} &= ((I_{yy} + I_{zz})x_{11}x_{12} + M_x)/I_{xx}; \\
\dot{x}_{11} &= ((I_{zz} + I_{xx})x_{10}x_{12} + M_y)/I_{yy}; \\
\dot{x}_{12} &= ((I_{xx} + I_{yy})x_{10}x_{11} + M_z)/I_{zz};
\end{aligned}
\tag{33}
$$

where $\mathbf{x}$ is a state space vector, $\mathbf{x} = [x_1 \ldots x_n]^T$, $\mathbf{M}$ is a vector of control moments, $\mathbf{M} = [M_1\ M_2\ M_3]^T$.

As a rule, quadcopters are manufactured with some angle stabilization systems. This means that a drone can be stabilized at any angle for some interval. The system of angle stabilization provides a stable location of the drone relatively, given angles by control moments:

$$
M_i = w_i(x_7^* - x_7, x_8^* - x_8, x_9^* - x_9, x_{10}, x_{11}, x_{12}), \ i = 1, 2, 3.
\tag{34}
$$

Assume that the angular stabilization system works out the given angles of the quadcopter quickly enough, at least in comparison with spatial movement. In this case we can assume that the control of the spatial movement of the quadcopter is carried out using the angular position of the drone and the thrust force. Let us define components of the spatial control vector: $x_7 = u_1$, $x_8 = u_2$, $x_9 = u_3$, $F/m = u_4$.

As a result we receive the following model of spatial quadcopter movement:

$$
\begin{aligned}
\dot{x}_1 &= x_4; \\
\dot{x}_2 &= x_5; \\
\dot{x}_3 &= x_6; \\
\dot{x}_4 &= u_4(\cos(u_1)\sin(u_3)\cos(u_2) + \sin(u_1)\sin(u_2)); \\
\dot{x}_5 &= u_4\cos(u_1)\cos(u_3) - g; \\
\dot{x}_6 &= u_4(\cos(u_1)\sin(u_3)\sin(u_2) + \sin(u_1)\cos(u_2)).
\end{aligned}
\tag{35}
$$

In this work, this model is used to obtain optimal control for the spatial motion of the quadcopter.

4.2. The Optimal Control Problem for Spatial Motion of Quadcopter

The model (35) of the control object is given. Here, $\mathbf{x}$ is a state space vector, $\mathbf{x} \in \mathbb{R}^6$, $\mathbf{u}$ is a control vector $\in U \in \mathbb{R}^4$. U is a compact set that defines restrictions on values of control vector components,

$$
\begin{aligned}
u_1^- = -\pi/12 &\le u_1 \le \pi/12 = u_1^+, \\
u_2^- = -\pi &\le u_2 \le \pi = u_2^+, \\
u_3^- = -\pi/12 &\le u_3 \le \pi/12 = u_3^+, \\
u_4^- = 0 &\le u_4 \le 12 = u_4^+.
\end{aligned}
\tag{36}
$$

According to the principle of synthesized control, initially the control synthesized problem (1)–(9) is solved. The model (35) is used as a model of the control object. To construct the set of initial states (4), the following vector of deviations is used:

$$
\Delta_0 = [2\ 2\ 2\ 0\ 0\ 0]^T.
\tag{37}
$$

In the problem, initial state and terminal state were equal:

$$
\mathbf{x}^0 = \mathbf{x}^f = [0\ 5\ 0\ 0\ 0\ 0]^T.
\tag{38}
$$

For calculation of the quality criterion (7), the following parameters are used: $t^+ = 2$, $\varepsilon_0 = 0.1$, $p = 2$.

To solve the control synthesis problem, the network operator method [23] was used. NOP found the following solution:

$$
u_i = \begin{cases} u_i^+, \text{ if } \hat{u}_i > u_i^+ \\ u_i^-, \text{ if } \hat{u}_i < u_i^- \quad , i = 1, \dots, m = 4, \\ \hat{u}_i, \text{ otherwise} \end{cases} \tag{39}
$$

where

$$
\hat{u}_1 = \mu(C), \tag{40}
$$

$$
\hat{u}_2 = \hat{u}_1 - \hat{u}_1^3, \tag{41}
$$

$$
\hat{u}_3 = \hat{u}_2 + \rho_{19}(W + \mu(C)) + \rho_{17}(A), \tag{42}
$$

$$
\hat{u}_4 = \hat{u}_3 + \ln(|\hat{u}_2|) + \text{sgn}(W + \mu(C))\sqrt{|W + \mu(C)|} + \rho_{19}(W) +
$$
$$
\arctan(H) + \text{sgn}(F) + \arctan(E) + \exp(q_2(x_2^f - x_2)) + \sqrt{q_1}, \tag{43}
$$

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

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

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

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

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

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

$$
\mu(z) = \begin{cases} z, \text{ if } |z| < 1 \\ \text{sgn}(z), \text{ otherwise} \end{cases}, \ \rho_{17}(z) = \text{sgn}(z)\ln(|z| + 1),
$$

$$
\rho_{18}(z) = \text{sgn}(z)(\exp(|z|) - 1), \ \rho_{19}(z) = \text{sgn}(z)\exp(-|z|),
$$

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

In the second stage, the optimal control problem is considered. In the problem, the mathematical model (35) is given. The initial state coincides with the terminal state (38).

It is necessary to find a control in the form of points in the state space (14). For synthesized control it is necessary to minimize the following quality criterion:

$$
J_3 = t_f + p_1 \|\mathbf{x}^f - \mathbf{x} + (t_f)\| + p_2 \sum_{i=0}^{N} \int_0^{t_f} \vartheta(\varphi_i(\mathbf{x}))dt +
$$

$$
p_3 \sum_{j=1}^{S} p_3 \vartheta(\min_t |\delta_j(\mathbf{x})| - \varepsilon) \to \min_{\mathbf{x}^*}, \tag{44}
$$

where $p_1 = 2$, $p_2 = 3$, $p_3 = 3$,

$$
\varphi_i(\mathbf{x}) = r_i - \sqrt{(x_{i,1} - x_1)^2 + (x_{i,3} - x_3)^2}, \tag{45}
$$

$i = 1, \ldots, N = 4$, $r_1 = r_2 = r_3 = r_4 = 2$, $x_{1,1} = 5$, $x_{1,3} = 0$, $x_{2,1} = 10$, $x_{2,3} = 5$, $x_{3,1} = 5$, $x_{3,3} = 10$, $x_{4,1} = 0$, $x_{4,3} = 5$,

$$\delta_j(\mathbf{x}) = \sqrt{(y_{i,1} - x_1)^2 + (y_{j,3} - x_3)^2}, \tag{46}$$

$j = 1, \ldots, S = 7$, $y_{1,1} = 5$, $y_{1,3} = -2$, $y_{2,1} = 10$, $y_{2,3} = 0$, $y_{3,1} = 12$,+ $y_{3,3} = 5$, $y_{4,1} = 10$, $y_{4,3} = 10$, $y_{5,1} = 5$, $y_{5,3} = 12$, $y_{6,1} = 0$, $y_{6,3} = 10$, $y_{7,1} = 2$, $y_{7,3} = 5$, $\varepsilon = 0.6$.

In the optimal control problem, the terminal time t_f is determined by the Equation (8) with $t^+ = 14.4$, $\varepsilon_0 = 0.1$. It is necessary to find coordinates of control points on each time interval, $\Delta t = 0.8$. The desired vector includes $3M$ parameters, where

$$M = \left\lfloor \frac{t^+}{\Delta t} \right\rfloor = \left\lfloor \frac{14.4}{0.8} \right\rfloor = 18, \tag{47}$$

that is, $\mathbf{q}^* = [q_1 \ldots q_{54}]^T$. The hybrid evolutionary algorithm has found the following optimal solution:

$$
\begin{aligned}
\mathbf{x}^{*,1} &= [4.83910 \; 1.14025 \; -5.22899]^T, \quad \mathbf{x}^{*,2} = [11.07056 \; 6.79389 \; -2.48647]^T, \\
\mathbf{x}^{*,3} &= [9.19808 \; 1.54674 \; 15.87195]^T, \quad \mathbf{x}^{*,4} = [-0.12204 \; 0.12276 \; -1.82381]^T, \\
\mathbf{x}^{*,5} &= [-4.08347 \; 2.93658 \; 5.89553]^T, \quad \mathbf{x}^{*,6} = [\;16.72896 \; 2.18022 \; 2.27907]^T, \\
\mathbf{x}^{*,7} &= [1.18106 \; 2.56582 \; 14.41088]^T, \quad \mathbf{x}^{*,8} = [8.67198 \; 5.78737 \; -2.90409]^T, \\
\mathbf{x}^{*,9} &= [8.59478 \; 2.73948 \; 11.33252]^T, \quad \mathbf{x}^{*,10} = [-1.25924 \; [-1.97448 \; -1.42747]^T, \\
\mathbf{x}^{*,11} &= [2.45445 \; 7.42257 \; -0.38164]^T, \quad \mathbf{x}^{*,12} = [8.68306 \; -0.78496 \; 15.41667]^T, \\
\mathbf{x}^{*,13} &= [0.60972 \; 7.02724 \; 7.66403]^T, \quad \mathbf{x}^{*,141} = [-0.59975 \; 0.39324 \; -1.31307]^T, \\
\mathbf{x}^{*,15} &= [-2.39004 \; 7.95279 \; 3.02003]^T, \quad \mathbf{x}^{*,16} = [2.52642 \; 6.69332 \; 9.17356]^T \\
\mathbf{x}^{*,17} &= [-0.95896 \; 4.42529 \; -0.36318]^T, \quad \mathbf{x}^{*,18} = [-0.01193 \; 5.02821 \; 15.40007]^T.
\end{aligned} \tag{48}
$$

For the found solution (48), the value of the quality criterion is $J_3 = 14.7010$.

In Figure 3, projections of the optimal trajectory on the horizontal plane $\{x_1; x_3\}$ are presented. Here, red circles are phase constraints described by (45), small black circles are passing areas described by (46) and small black boxes are control points (48).
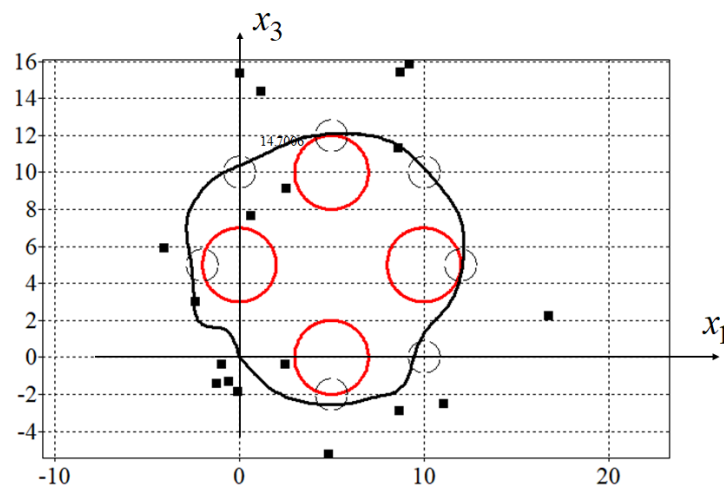

Figure 3. Optimal trajectory for synthesized control.

For the new adaptive synthesized control proposed in this paper, the set of initial states is determined by Equation (3) with deviation vector

$$\boldsymbol{\Delta}_0 = [0.2 \; 0.2 \; 0.2 \; 0 \; 0 \; 0]^T. \tag{49}$$

It is necessary to find the same number of control points according to the following quality criterion:

$$J_4 = \sum_{k=1}^{K} \left(t_{f,k} + p_1 \|\mathbf{x}^f - \mathbf{x}(t_{f,i}, \mathbf{x}^{0,i}) + p_2 \sum_{i=0}^{N} \int_0^{t_f} \vartheta(\varphi_i(\mathbf{x}))dt + \right.$$

$$\left. p_3 \sum_{j=1}^{S} p_3 \vartheta(\min_t |\delta_j(\mathbf{x})| - \varepsilon) \right) \to \min_{\mathbf{x}^*}, \tag{50}$$

where $K = 7$, $t_{f,k}$ is defined by Equation (8). Other parameters of the criterion are the same as for the criterion (44).

Again, the hybrid algorithm was applied and the following optimal solution has been found:

$$\mathbf{x}^{*,1} = [17.46361\ 1.14030\ -8.00000]^T,\ \mathbf{x}^{*,2} = [11.07060\ 6.79390\ -2.48650]^T,$$
$$\mathbf{x}^{*,3} = [9.19810\ 2.05890\ 15.87207]^T,\ \mathbf{x}^{*,4} = [-0.27800\ 2.51633\ -1.99493]^T,$$
$$\mathbf{x}^{*,5} = [-3.98430\ 2.27048\ 13.40976]^T,\ \mathbf{x}^{*,6} = [17.18235\ 0.26253\ 2.19246]^T,$$
$$\mathbf{x}^{*,7} = [-3.56784\ 3.44842\ 14.94369]^T,\ \mathbf{x}^{*,8} = [4.53881\ 2.20612\ -2.99328]^T,$$
$$\mathbf{x}^{*,9} = [9.06419\ 2.49928\ 11.30274]^T,\ \mathbf{x}^{*,10} = [-0.16333\ -1.88939\ -0.75766]^T. \tag{51}$$
$$\mathbf{x}^{*,11} = [2.17956\ 6.92983\ -1.06412]^T,\ \mathbf{x}^{*,12} = [10.24873\ -0.51465\ 5.82840]^T,$$
$$\mathbf{x}^{*,13} = [1.12164\ 2.84506\ 7.93804]^T,\ \mathbf{x}^{*,14} = [0.10678\ 3.23489\ -1.55778]^T,$$
$$\mathbf{x}^{*,15} = [-2.54374\ 0.99732\ 2.82005]^T,\ \mathbf{x}^{*,16} = [7.41006\ 6.49634\ 12.02799]^T,$$
$$\mathbf{x}^{*,17} = [-0.67510\ 4.03845\ -0.28527]^T,\ \mathbf{x}^{*,18} = [-0.18037\ 4.62980\ 6.89661]^T.$$

A value of the quality criterion (50) for one initial state $\mathbf{x}(0) = [0\ 5\ 0\ 0\ 0\ 0]^T$, is $J_4 = 15.6090$. In Figure 4, projections of the optimal trajectory on the horizontal plane $\{x_1; x_3\}$ found by the adaptive synthesized control (51) are presented.
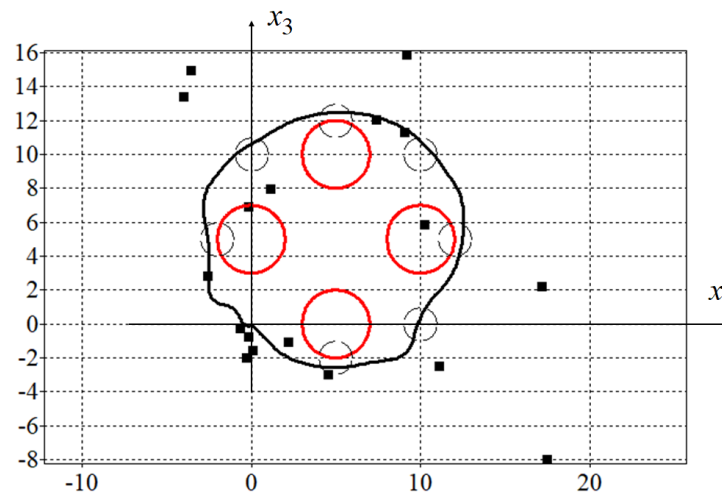

Figure 4. Optimal trajectory for adaptive synthesized control.

Since the initial state in the problem coincided with the terminal state, in order to force the control object to move along a closed path, mandatory conditions for passing through certain areas were added to the quality criterion. For trajectories that meet the criteria for passing through the specified areas, the value of the quality criterion will not change at $p_3 = 0$. This is seen in Figures 3 and 4 as both trajectories pass through all specified areas.

Let us check the sensitivity of the obtained solutions to random perturbations of the initial state

$$x_i(0) = x_i^0 + \beta_0(2\xi(t) - 1),\ i = 1, \dots, n = 6, \tag{52}$$

where $\xi(t)$ is a function generator of random noise, which returns a random number from interval $(0; 1)$ at every call, β_0 is a constant level of noise.

In Figures 5 and 6, the optimal (in blue) and eight perturbed trajectories (in black) for $\beta_0 = 0.1$ for the solutions obtained by synthesized (Figure 5) and adaptive synthesized (Figure 6) control are presented.
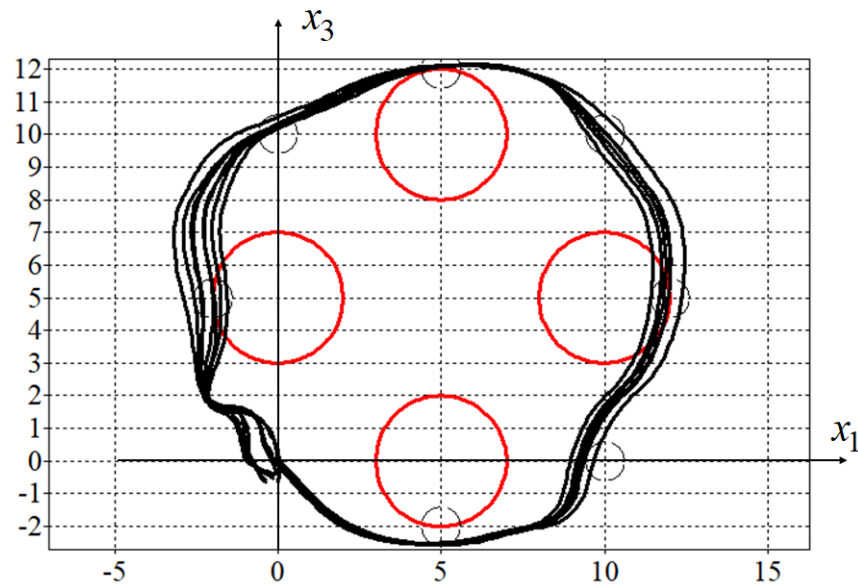

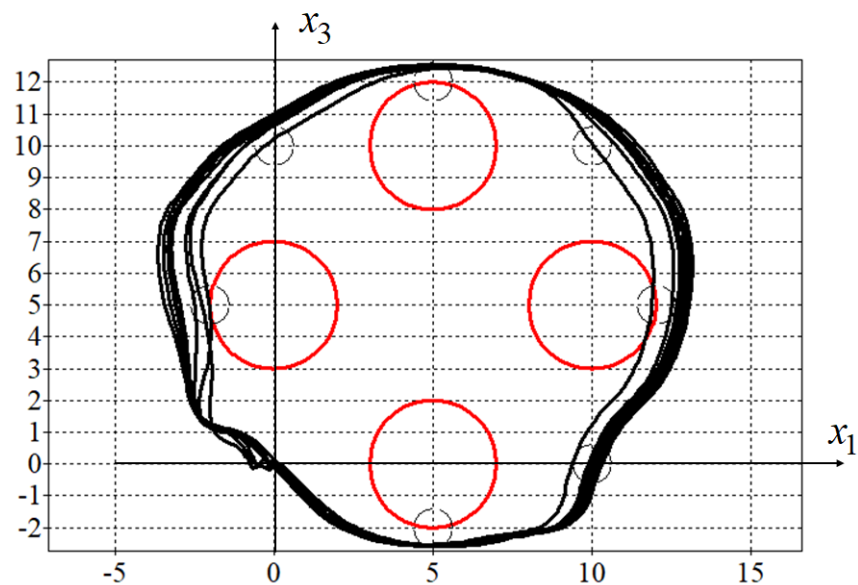**Figure 5.** Optimal and eight disturbance trajectories of synthesized control.

Figure 6. Optimal and eight disturbance trajectories of adaptive synthesized control.

For comparison, for a model (35) without stabilization systems (39), the problem of optimal control directly was solved, where control was sought in the form of a piece-wise linear function, taking into account restrictions (36).

$$u_i = \begin{cases} u_i^+, & \text{if } \tilde{u}_i > u_i^+ \\ u_i^-, & \text{if } \tilde{u}_i < u_i^- \\ \tilde{u}_i, & \text{otherwise} \end{cases}, i = 1, \ldots, m = 4, \tag{53}$$

where

$$\tilde{u}_i = (q_{i+jm} - q_{i+(j-1)m})\frac{t - (j-1)\Delta t}{\Delta t} + q_{i+(j-1)m}, \ i = 1, 2, 3, 4, \tag{54}$$

$(j-1)\Delta t \leq t < j\Delta t$, $j \in \{1, \ldots, K+1\}$, q_i is a component of desired parameters vector, $i = 1, \ldots, m(M+1)$,

$$\mathbf{q} = [q_1 \ldots q_{m(M+1)}]^T. \tag{55}$$

In this work, we set the same time interval $\Delta t = 0.8$; therefore, from (15) $M = 18$, and it is necessary to find $m(M+1) = 4 \cdot 19 = 76$ parameters, $\mathbf{q} = [q_1 \ldots q_{76}]^T$.

To solve the optimal control problem, the same hybrid algorithm was used. As a result, the following solution was obtained:

$$
\begin{aligned}
\mathbf{q} \ = \ & [12.57045 - 4.58471 - 2.74617\ 2.90422 - 9.26325 - 0.10990 \\
& -2.63222\ 18.36841\ 0.00816 - 17.35177\ 0.00165\ 4.28718 \\
& -11.81492\ 1.88489 - 8.01206\ 15.87943\ 10.84894\ 0.06505 \\
& 9.65475\ 19.51903\ 2.79860 - 4.06408 - 0.88992\ 10.50507 \\
& -19.19030\ 17.90240\ 12.52431\ 19.00010\ 4.76513 - 11.97648 \\
& 0.00010\ 8.85464\ 2.92334\ 0.14238\ 8.60919\ 7.83194 \\
& 5.74904 - 8.35383 - 3.42757\ 12.87671\ 18.58717\ 15.43057 \\
& 9.06137\ 12.55621 - 1.54628\ 1.47314\ 2.40706\ 8.67602 \\
& 0.00091 - 11.91236 - 19.94063\ 17.08304\ 19.92640 - 1.33145 \\
& -7.77258\ 15.54094 - 19.93278 - 17.37121 - 9.31290\ 5.03257 \\
& -0.90297 - 5.22021\ 0.62653\ 4.21368 - 2.04314 - 0.53192 \\
& 0.09353\ 14.25213 - 0.11587\ 9.05588 - 0.03270\ 11.23667 \\
& 0.03826 - 16.78047\ 0.18220\ 19.81652]^T.
\end{aligned}
\tag{56}
$$

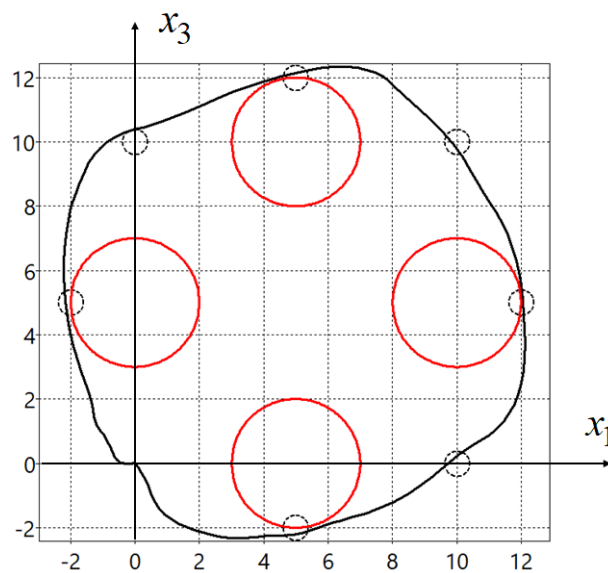In Figure 7, the projection of the optimal trajectory obtained by the direct method is presented.

Figure 7. Optimal trajectory of direct control.

In Table 1 there are values of the quality criterion (44) of ten experiments for perturbed solutions obtained by the synthesized (column Synthesized), the adaptive synthesized (column Adaptive) control and the direct solution (Direct). In two last strings of the table, average values of the functionals and standard deviations for all experiments are presented.

As can be seen from Figures 5 and 6 and Table 1, the solutions obtained by adaptive synthesized control are less sensitive to perturbations of initial states than the solutions obtained by simple synthesized control or, especially, by the direct approach.

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

No	Synthesized	Adaptive	Direct
1	14.7651	15.4892	19.2082
2	20.7377	15.4829	19.8854
3	15.2888	15.6947	16.7706
4	16.9743	15.4935	16.2334
5	18.6159	16.0397	19.2815
6	19.5227	15.7950	19.3866
7	20.0937	15.4178	16.8263
8	17.5416	16.1424	23.3437
9	20.1225	17.0695	19.6251
10	19.9257	15.3893	20.8163
Av	18.3588	15.8014	19.1377
SD	2.1234	0.5167	2.1285

5. Conclusions

A new method for solving the problem of optimal control in the class of implemented functions, an adaptive synthesized control principle is presented. Unlike synthesized control, the new method takes into account the perturbations of the initial state when solving the optimal control problem. Therefore, the value of the quality criterion is calculated as the sum of the quality criterion values for the different initial states. As a result of this approach, a solution is chosen in such a way that for the origin initial state it may not give the best quality criterion value, but in the case of disturbances of the initial state, the quality criterion value changes slightly.

6. Discussion

Obtaining a solution based on replacing the optimal solution is less optimal, but also less sensitive to disturbances. At first glance, this seems obvious and can be applied to any method of solving the optimal control problem. However, this is not the case. A direct solution to the optimal control problem results in control in the form of a time function and an open-loop control system. Perturbation of the initial conditions for such a system gives large variations in quality criterion values, which cannot be reliably estimated from the average value due to the large variance.

The synthesized control method firstly makes the control object stable relative to some equilibrium point in the state space. This means that the perturbed and unperturbed trajectories at each point in time move towards a stable equilibrium point. The adaptive synthesized control method sets the positions of the equilibrium points so that all disturbed trajectories are located in some tube that does not violate phase constraints whenever possible.

In the future, when using the adaptive synthesized control method, it is necessary to assess the required size of the initial state region and reduce the number of initial state points, since this significantly increases the time for finding the optimal solution.

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

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

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

References

1. Egerstedt, M. Motion Planning and Control of Mobile Robots. Ph.D. Thesis, Royal Institute of Technology, Stockholm, Sweden, 2000.
2. 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]
3. 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]
4. Allagui, N.Y.; Abid, D.B.; Derbel, N. Autonomous navigation of mobile robot with combined fractional order PI and fuzzy logic controllers. In Proceedings of the 2019 16th International Multi-Conference on Systems, Signals & Devices (SSD), Istanbul, Turkey, 21–24 March 2019; pp. 78–83. [CrossRef]
5. Chen, B.; Cao, Y.; Feng, Y. Research on Trajectory Tracking Control of Non-holonomic Wheeled Robot Using Backstepping Adaptive PI Controller. In Proceedings of the 2022 7th Asia-Pacific Conference on Intelligent Robot Systems (ACIRS), Tianjin, China, 1–3 July 2022; pp. 7–12. [CrossRef]
6. 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]
7. Nguyen, A.T.; Nguyen, X.-M.; Hong, S.-K. Quadcopter Adaptive Trajectory Tracking Control: A New Approach via Backstepping Technique. *Appl. Sci.* **2019**, *3873*, 3873. [CrossRef]
8. Lee, E.B.; Marcus, L. *Foundations of Optimal Control Theory*; Robert & Krieger Publishing Company: Malabar, Florida, 1967; 576p.
9. Pontryagin, L.S.; Boltyanskii, V.G.; Gamkrelidze, R.V.; Mishchenko, E.F. *The Mathematical Theory of Optimal Process. L. S. Pontryagin, Selected Works*; Gordon and Breach Science Publishers: New York, NY, USA; London, UK; Paris, France; Montreux, Switzerland; Tokyo, Japan, 1985; Volume 4, 360p.
10. Shmalko, E., Diveev, A. Extended Statement of the Optimal Control Problem and Machine Learning Approach to Its Solution. *Math. Probl. Eng.* **2022**, *2022*, 1932520. [CrossRef]
11. Diveev, A.; Shmalko, E.Y.; Serebrenny, V.V.; Zentay, P. Fundamentals of synthesized optimal control. *Mathematics* **2020**, *9*, 21. [CrossRef]
12. Shmalko, E.Y. Feasibility of Synthesized Optimal Control Approach on Model of Robotic System with Uncertainties. In *Electromechanics and Robotics*; Smart Innovation, Systems and Technologies; Ronzhin, A., Shishlakov, V., Eds.; Springer: Singapore, 2022; Volume 232.
13. Shmalko, E. Computational Approach to Optimal Control in Applied Robotics. In *Frontiers in Robotics and Electromechanics. Smart Innovation, Systems and Technologies*; Ronzhin, A., Pshikhopov, V., Eds.; Springer: Singapore, 2023; Volume 329.
14. Diveev, A.; Shmalko, E.Y. Stability of the Optimal Control Problem Solution. In Proceedings of the 8th International Conference on Control, Decision and Information Technologies, CoDIT, Istanbul, Turkey, 17–20 May 2022; pp. 33–38.
15. Simon J.D.; Mitter S.K. A theory of modal control. *Inf. Control* **1968**, *13*, 316–353. [CrossRef]
16. Jouffroy, J.; Lottin, J. Integrator backstepping using contraction theory: A brief methodological note. In Proceedings of the 15th IFAC World Congress, Barcelona, Spain, 21–26 July 2002.
17. Zhou H.; Liu, Z. Vehicle Yaw Stability-Control System Design Based on Sliding Mode and Backstepping Contol Approach. *IEEE Trans. Veh. Technol.* **2010**, *59*, 3674–3678. [CrossRef]
18. Febsya, M.R.; Ardhi, R.; Widyotriatmo, A.; Nazaruddin, Y.Y. Design Control of Forward Motion of an Autonomous Truck-Trailer using Lyapunov Stability Approach. In Proceedings of the 2019 6th International Conference on Instrumentation, Control, and Automation (ICA), Bandung, Indonesia, 31 July–2 August 2019; pp. 65–70. [CrossRef]
19. Zihao, S.; Bin, W.; Ting, Z. Trajectory Tracking Control of a Spherical Robot Based on Adaptive PID Algorithm. In Proceedings of the 2019 Chinese Control And Decision Conference (CCDC), Nanchang, China, 3–5 June 2019; pp. 5171–5175. [CrossRef]
20. Liu, J.; Song, X.; Gao, S.; Chen, C.; Liu K.; Li, T. Research on Horizontal Path Tracking Control of a Biomimetic Robotic Fish. In Proceedings of the 2022 International Conference on Mechanical and Electronics Engineering (ICMEE), Xi'an, China, 21–23 October 2022; pp. 100–105. [CrossRef]
21. Duriez, T.; Brunton, S.L.; Noack, B.R. *Machine Learning Control–Taming Nonlinear Dynamics and Turbulence*; Springer International Publishing: Cham, Switzerland, 2017.
22. Awange, J.L.; Paláncz, B.; Lewis, R.H.; Völgyesi, L. Symbolic Regression. In *Mathematical Geosciences*; Springer: Cham, Switzerland, 2018.
23. Diveev, A.I.; Shmalko, E.Y. *Machine Learning Control by Symbolic Regression*; Springer: Cham, Switzerland, 2021; 155p.
24. Diveev, A.; Shmalko, E. Machine Learning Feedback Control Approach Based on Symbolic Regression for Robotic Systems. *Mathematics* **2022**, *10*, 4100. [CrossRef]
25. Davis, L. *Handbook of Genetic Algorithms*; Van Nostrand Reinhold: New York, NY, USA, 1991.
26. Eberhardt, R.C.; Kennedy, J.A. Particle Swarm Optimization. In Proceedings of the IEEE International Conference on Neural Networks, Piscataway, NJ, USA, 27 November–1 December 1995; pp. 1942–1948.
27. Eltamaly, A. A Novel Strategy for Optimal PSO Control Parameters Determination for PV Energy Systems. *Sustainability* **2021**, *13*, 1008. [CrossRef]

28.   Salehpour, E.; Vahidi, J.; Hosseinzadeh H. Solving optimal control problems by PSO-SVM. *Comput. Methods Differ. Equ.* **2018**, *6*, 312–325

29.   Mirjalili, S.; Mirjalil, S.M.; Lewis, A. Grey Wolf Optimizer. *Adv. Eng. Softw.* **2014**, *69*, 46–61. [CrossRef]