ISSN 0005-1179 (print), ISSN 1608-3032 (online), Automation and Remote Control, 2025, Vol. 86, No. 1, pp. 61–73.

On Guaranteed Estimate of Deviations from the Target Set in a Control Problem under Reinforcement Learning

I. A. Chistiakov

Lomonosov Moscow State University, Faculty of Computational Mathematics and Cybernetics, Moscow, Russia

e-mail: chistyakov.ivan@yahoo.com

Received August 29, 2024

Revised October 14, 2024

Accepted October 29, 2024

Abstract—We consider a target control problem of a special form, in which a system of differential equations includes nonlinear terms depending on state variables. We show that reinforcement learning algorithms such as Proximal Policy Optimization (PPO) can be used to find an inexact feedback solution. The chosen strategy is further approximated with a piecewise affine control. Based on the dynamic programming method, an inner estimate of the solvability set is calculated, as well as a corresponding a priori estimate of the distance between a final trajectory point and the target set. To do this, we examine an auxiliary problem for a piecewise linear system with noise and calculate a piecewise quadratic function as an approximate solution of the Hamilton–Jacobi–Bellman equation.

Keywords: nonlinear dynamics, dynamic programming, comparison principle, linearization, piecewise quadratic value function, reinforcement learning, PPO algorithm, solvability set

DOI: 10.31857/S0005117925010055

1. INTRODUCTION

We consider a target control problem for a nonlinear system of differential equations on a fixed finite time interval. This problem is closely related to construction of the solvability set containing all starting positions from which the control synthesis problem can be solved. To approximate this set, one may use various methods based on analysis of the corresponding differential inclusion [1–3] or depending on the Hamilton–Jacobi–Bellman (HJB) equation [4–7]. These approaches are applicable to a wide class of nonlinear systems, yet they require large computational costs. Recently, algorithms based on machine learning have also been developed, making it possible to approximate solution of the HJB equation with a neural network [8, 9] or to search for the control function directly [10]. However, the latter do not provide any guaranteed estimates.

This paper proposes to reduce computational complexity of solving the HJB equation by searching for an approximate solution in the class of piecewise quadratic functions defined on a set of simplices. We develop the ideas introduced in [11–13] and present a method based on piecewise linearization of the right-hand side of differential equations, considering an auxiliary control problem for a system with piecewise linear dynamics and bounded noise (linearization error). The comparison principle [14, 15] allows us to derive equations for the coefficients of the sought-for value function, whose zero sublevel set is an internal estimate of the solvability set of the original nonlinear system.

The search for an approximate HJB solution using the above-mentioned method is accompanied by construction of a suboptimal control strategy. Previously, a control strategy was proposed in the form of continuous piecewise affine function [13], determined by values at the vertices of the partition

simplices. In this case, the values at the vertices should be chosen in such a way as to minimize the derivative of the value function along the trajectory. However, since the constructed estimate of the value function is not smooth, it is required to use additional heuristics, which increase the error of the method. In this paper, we demonstrate that results of other algorithms can also be used as controls at the vertices. In particular, we propose using reinforcement learning [16, 17]. If control values are chosen based on the output of a neural network model, the resulting estimate of the value function can take smaller values at the initial time moment. Therefore, it would a priori guarantee reaching a smaller neighborhood of the target set.

Note that reinforcement learning algorithms also imply construction of a value function, which is an estimate of the resulting benefit from each possible position (in this case, we are talking about the distance to the target set at the final moment of time), or its analogues. But even with a well-chosen control, such an estimate is not reliable and may be inaccurate. At the same time, the approach indicated in this paper allows any predetermined control strategy to be approximated by a piecewise affine function, for which the resulting estimate will be guaranteed. This can be especially useful in case of additional interference, when calculation of trajectories for different initial points is not sufficient to estimate all possible variants of the system's behavior.

2. PROBLEM STATEMENT

We consider a system of nonlinear differential equations

$$\dot{x} = \mathbf{f}(t,x) + \mathbf{g}(t,x)u, \quad t \in [t_0, t_1],\ x \in \Omega, \tag{1}$$

where $\Omega \in \mathbb{R}^{n_x}$ is a compact set, large enough to contain all the trajectories of (1) for any $t \in [t_0, t_1]$; we assume that the boundary of Ω is a polyhedron. The nonlinear vector function $\mathbf{f}(t,x)$ and the matrix function $\mathbf{g}(t,x) \in \mathbb{R}^{n_x \times n_u}$ are continuous in t and twice continuously differentiable with respect to x. The interval $[t_0, t_1]$ is fixed. At every moment of time, the control vector u must belong to a compact convex set $\mathcal{P}$:

$$u \in \mathcal{P} \subset \mathbb{R}^{n_u}. \tag{2}$$

The main problem is to construct a continuous feedback control in the form $u = u(t,x)$, which steers the system (1) from a given point x_0 at time t_0 to the smallest possible neighborhood of a compact target set $\mathcal{X}_1 \subset \Omega$ at time t_1. Let $u(\cdot)$ denote feedback control. Thus,

$$x(t_1; t_0, x_0)|_{u(\cdot)} \in \mathcal{X}_1 + B_\varepsilon(0)$$

must hold, where $x(t_1; t_0, x_0)|_{u(\cdot)}$ is the final point of a trajectory that started at time t_0 from the point x_0, closed by control $u(\cdot)$; $B_\varepsilon(0)$ is a ball of radius ε centered at zero, and the value of $\varepsilon \geqslant 0$ must be minimized. We also assume that the target set is representable as $\mathcal{X}_1 = \{x \in \Omega \colon \phi_{\mathcal{X}_1}(x) \leqslant 0\}$, where $\phi_{\mathcal{X}_1}(x)$ is a twice differentiable function.

In addition, it is required to construct *the solvability set* $\mathcal{W}(t, t_1, \mathcal{X}_1)$ [15], that is, the set of all vectors $x \in \Omega$, for each of which there is a control $u(\cdot)$, satisfying the constraint (2) and transferring the system from position $\{t, x\}$ ($t \in [t_0, t_1]$) to the target set: $x(t_1; t, x)|_{u(\cdot)} \in \mathcal{X}_1$. However, since the task of constructing the exact solvability set is difficult, we limit ourselves to searching for internal estimates of this set.

3. SYSTEM WITH PIECEWISE LINEAR DYNAMICS

Let *n-dimensional simplex* [20] with vertices $x_1, x_2, \ldots, x_{n+1} \in \mathbb{R}^n$ be the set

$$S^n = \left\{ \alpha_1 x_1 + \alpha_2 x_2 + \ldots + \alpha_{n+1} x_{n+1} : \alpha_i \geqslant 0,\ \sum_{i=1}^{n+1} \alpha_i = 1 \right\},$$

where the vectors $x_2 - x_1, \ldots, x_{n+1} - x_1$ are linearly independent. In this case, a vector of *barycentric coordinates* $\alpha(x) = (\alpha_1, \ldots, \alpha_{n+1})^T$ uniquely defines the position of any point x inside the simplex. In addition, there is a matrix $\tilde{H}$ [11] such that the barycentric coordinates $\alpha(x)$ are linearly expressed in terms of x: $\alpha = \tilde{H} \times (x^T, 1)^T$.

Consider some partition of the Ω set into N simplices $\Omega^{(i)}$ such that any two simplices do not intersect or intersect only along any of their common faces of dimension smaller than n_x. In practice, having an arbitrary set of vertices, one may implement such partition using Delaunay triangulation [21, 22], which is efficiently computed by constructing a convex hull of points in $(n_x + 1)$-dimensional space [23].

The superscript (i) further denotes correspondence of a vector, matrix, or function to the simplex $\Omega^{(i)}$. In particular, vertices of a simplex are denoted as $g_1^{(i)}, \ldots, g_{n_x+1}^{(i)} \in \mathbb{R}^{n_x}$, $i = \overline{1, N}$. Note that each vertex can belong to several simplices.

In [11–13], a method was proposed to construct a continuous piecewise affine approximation of the system (1), which substantially uses a partition of the Ω set into simplices. The matrices $A^{(i)}$, $B^{(i)}$ and vectors $f^{(i)}$ are selected in such a way that the following representation is valid for all $u \in \mathcal{P}$:

$$\mathbf{f}(t,x) + \mathbf{g}(t,x)u = A^{(i)}(t)x + B^{(i)}(t)u + f^{(i)}(t) + v^{(i)}(t,x,u), \quad x \in \Omega^{(i)}, \tag{3}$$

where $v^{(i)}$ is a local linearization error. This error is bounded and there is an estimate for it based on decomposition of the functions $\mathbf{f}(t,x)$ and $\mathbf{g}(t,x)$ according to the Taylor's formula. Moreover, this estimate is independent of particular values $x \in \Omega^{(i)}$ and $u \in \mathcal{P}$. Thus, all possible values of $v^{(i)}$ can be bounded with some ellipsoid $\mathcal{Q}^{(i)}(t)$:

$$\mathcal{Q}^{(i)}(t) = \mathcal{E}(0, Q^{(i)}(t)) = \{x \in \mathbb{R}^{n_x} : \ \langle x, (Q^{(i)})^{-1}x\rangle \leqslant 1\}, \quad Q^{(i)} = (Q^{(i)})^T > 0. \tag{4}$$

Remark 1. If system (1) additionally contains an additive term in the form of unknown bounded function (interference), then it can also be taken into account during piecewise linearization by scaling the ellipsoids $\mathcal{Q}^{(i)}(t)$ and shifting their centers.

It is convenient to consider the extended variable space, where a vector $\tilde{x}$ is obtained by adding an auxiliary component with a fixed value equal to one: $\tilde{x} = (x^T, 1)^T$. Then, based on (3), we can write the following piecewise linear system of differential equations with autonomous switching [24, pp. 5–9] in the extended space of variables:

$$\dot{\tilde{x}} = \tilde{A}^{(i)}(t)\tilde{x} + \tilde{B}^{(i)}(t)u + \tilde{C}v^{(i)}, \quad \tilde{x} \in \Omega^{(i)} \times \{1\}, \ t \in [t_0, t_1], \tag{5}$$

$$\tilde{A}^{(i)}(t) = \begin{bmatrix} A^{(i)}(t) & f^{(i)}(t) \\ \mathbb{O}_{1\times n_x} & 0 \end{bmatrix}, \quad \tilde{B}^{(i)}(t) = \begin{bmatrix} B^{(i)}(t) \\ \mathbb{O}_{1\times n_u} \end{bmatrix}, \quad \tilde{C} = \begin{bmatrix} \mathbb{I}_{n_x\times n_x} \\ \mathbb{O}_{1\times n_x} \end{bmatrix},$$

where $v^{(i)}$ is interpreted as interference. We will call an interference acceptable if it is a measurable function of time and it satisfies the constraint $v^{(i)}(t) \in \mathcal{Q}^{(i)}(t)$ at each time moment. The index $i = i(x(t))$ in formula (5) is a function of system state at time t, however, for the sake of brevity, we omit the arguments of this function.

4. VALUE FUNCTION

4.1. General Background

Consider an auxiliary value function

$$\bar{V}(t,x) = \min_{u(\cdot)} \{\phi_{\mathcal{X}_1}(x(t_1)) : \ x(t) = x\}, \tag{6}$$

where $x(\cdot)$ is a trajectory of the nonlinear system (1), starting at the initial position and closed by a fixed feedback control $u(\cdot)$. Using the value function, the solvability set is constructed [15] as

$$\mathcal{W}(t,t_1,\mathcal{X}_1)=\left\{x\in\Omega:\ \bar{V}(t,x)\leqslant 0\right\}. \tag{7}$$

Along with (7), consider a formula for the neighborhood of the solvability set:

$$\mathcal{W}_\varepsilon(t,t_1,\mathcal{X}_1)=\left\{x\in\Omega:\ \bar{V}(t,x)\leqslant \varepsilon\right\},$$
$$\mathcal{W}_\varepsilon(t,t_1,\mathcal{X}_1)=\left\{x\in\Omega\mid \exists u(\cdot):\ \phi_{\mathcal{X}_1}(x(t_1;t,x)|_{u(\cdot)})\leqslant\varepsilon\right\}.$$

At any point of differentiability (t,x), where $t<t_1,\ x\in\Omega$, the function $\bar{V}(t,x)$ satisfies the backward Hamilton–Jacobi–Bellman equation

$$\min_{u\in\mathcal{P}}\bar{V}'(t,x;(1,(\mathbf{f}(t,x)+\mathbf{g}(t,x)u)^T)^T)=0, \tag{8}$$

where $\bar{V}'(t,x;\ell)$ is the derivative of the function $\bar{V}(t,x)$ at the point (t,x) in the direction $\ell\in\mathbb{R}^{n_x+1}$. At the final moment of time, the equality $\bar{V}(t_1,x)=\phi_{\mathcal{X}_1}(x)$ holds. The function $\bar{V}(t,x)$ can be non-differentiable, and hence the solution of (8) must be recognized in a generalized sense [25]. Nevertheless, we can replace the solution $\bar{V}(t,x)$ with such a piecewise quadratic function that equation (8) would be fulfilled approximately. This function will be further found based on consideration of piecewise linear system (5).

4.2. Piecewise Quadratic Function

At each vertex $g_l^{(i)}$ of each simplex $\Omega^{(i)}$, consider a function $\langle k_l^{(i)}(t),\tilde{x}\rangle$, which is affine in x. For each $t\in[t_0,t_1]$, the vector $k_l^{(i)}\in\mathbb{R}^{n_x+1}$ is a vector of unknown coefficients. Then, for each simplex $\Omega^{(i)}$, we can define a matrix of parameters, whose structure corresponds to the set of vertices $g_1^{(i)},\dots,g_{n_x+1}^{(i)}$:

$$K^{(i)}(t)=\left[k_1^{(i)}(t),\dots,k_{n_x+1}^{(i)}(t)\right]\in\mathbb{R}^{(n_x+1)\times(n_x+1)}.$$

Consider a piecewise quadratic function

$$V^{(i)}(t,\tilde{x})=\langle\tilde{x},K^{(i)}(t)\tilde{H}^{(i)}\tilde{x}\rangle,\quad \tilde{x}=(x^T,1)^T,\quad x\in\Omega^{(i)}. \tag{9}$$

Equation (9) corresponds to interpolation of the considered affine functions:

$$V^{(i)}(t,\tilde{x})=\langle\tilde{x},K^{(i)}(t)\tilde{H}^{(i)}\tilde{x}\rangle=\langle(K^{(i)}(t))^T\tilde{x},\alpha^{(i)}(x)\rangle=\sum_{l=1}^{n_x+1}\alpha_l^{(i)}(x)\langle k_l^{(i)}(t),\tilde{x}\rangle.$$

Since the function (9) is defined for the extended space of variables $\tilde{x}=(x^T,1)^T$, an arbitrary piecewise quadratic function defined on a set of simplices can be represented in such form.

We will use piecewise affine controls of the form

$$u(t,x)=Y^{(i)}(t)\tilde{H}^{(i)}\tilde{x}=\sum_{k=1}^{n_x+1}\alpha_k^{(i)}(x)y_k^{(i)}(t)\in\mathbb{R}^{n_u}, \tag{10}$$

where the matrix $Y^{(i)}(t)\in\mathbb{R}^{n_u\times(n_x+1)}$ consists of column vectors $y_k^{(i)}(t)\in\mathcal{P}$ that are the control values at the vertices of $\Omega^{(i)}$. Those values will be chosen later. Note that the values of $y_k^{(i)}(t)$ corresponding to the same vertex in different simplices will coincide, hence the control function $u(t,x)$ is continuous in x. Due to convexity of $\mathcal{P}$, the condition $u(t,x)\in\mathcal{P}$ is satisfied.

Consider the derivative of $V^{(i)}(t,\tilde{x})$ in the direction $\ell=(\ell_t,\ell_x)\in\mathbb{R}^{n_x+2}$:

$$\frac{dV^{(i)}}{d\ell}=\ell_t\langle\tilde{x},\dot{K}^{(i)}\tilde{H}^{(i)}\tilde{x}\rangle+\langle\ell_x,[K^{(i)}\tilde{H}^{(i)}+(\tilde{H}^{(i)})^T(K^{(i)})^T]\tilde{x}\rangle. \tag{11}$$

It was proved in [13] that the following estimate holds for $\ell=(\ell_t,\ell_x)^T$, where $\ell_t=1$, $\ell_x=\tilde{A}^{(i)}\tilde{x}+\tilde{B}^{(i)}u+\tilde{C}v^{(i)}$:

$$\frac{dV^{(i)}}{d\ell}(t,\tilde{x})\leqslant\langle\tilde{x},[\dot{K}^{(i)}+Z^{(i)}]\tilde{H}^{(i)}\tilde{x}\rangle. \tag{12}$$

The matrix $Z^{(i)}$ is known and can be expressed in terms of coefficients $K^{(i)}(t)$, coefficients $\tilde{A}^{(i)}(t)$, $\tilde{B}^{(i)}(t)$, $\tilde{C}$ of the piecewise linear system (5), and matrices $Y^{(i)}(t)$ that define control values at the vertices of $\Omega^{(i)}$. The obtained estimate is valid for any acceptable interference $v^{(i)}\in\mathcal{Q}^{(i)}(t)$.

Making $\dot{K}^{(i)}+Z^{(i)}$ to be zero matrix, we obtain the system of matrix differential equations which describes evolution of $V^{(i)}(t,\tilde{x})$ over time.

$$\dot{K}^{(i)}(t)+Z^{(i)}(t)=0,\quad t\in[t_0,t_1],\ i=\overline{1,N}. \tag{13}$$

Then it follows from (12)–(13) that along any trajectory of the system (5) the derivative of the function will not increase in each simplex $\Omega^{(i)}$. Next, we will show how to modify equations (13) so that the resulting function $V^{(i)}(t,\tilde{x})$ would be continuous, and therefore the derivative would not grow even when passing through the simplex boundaries. This can be used to construct a guaranteed a priori estimate for trajectory's end point deviation from the target set.

4.3. Boundary Conditions

To solve (13), one has to set boundary conditions at the final time moment $t=t_1$. In turn, it is necessary to construct a piecewise quadratic upper bound for the function $\phi_{\mathcal{X}_1}$. Based on representation (9), matrices $K^{(i)}(t_1)$ can be defined. In particular, if the boundary of the set $\mathcal{X}_1$ is a second-order hypersurface, then the representation $\phi_{\mathcal{X}_1}(x)=\langle\tilde{x},\hat{K}\tilde{x}\rangle$ is valid for some matrix $\hat{K}=\hat{K}^T$. Thus, let the parameter values of $V^{(i)}(t_1,\tilde{x})$ be equal to

$$K^{(i)}(t_1)=\hat{K}\times(\tilde{H}^{(i)})^{-1} \tag{14}$$

in each simplex. In general, for any twice differentiable function $\phi_{\mathcal{X}_1}$, it is possible to construct a piecewise affine upper bound [12], which is a special case of piecewise quadratic function and thus leads to conditions similar to (14). The function $V^{(i)}(t,\tilde{x})$ will be continuous in x over the entire set $\Omega\times\{1\}$ at time $t=t_1$.

4.4. Function Smoothing

Note that solution $V^{(i)}(t,x)$ (9) of the Cauchy problem (13)–(14) can be discontinuous at the boundaries of simplices. Each column of $K^{(i)}(t)$ defines the coefficients of the piecewise affine function $\langle k_l^{(i)}(t),\tilde{x}\rangle$ at some vertex g_l. However, generally speaking, each such point is a vertex of several simplices at once. Since matrices $Z^{(i)}$ in (11) are constructed independently for each simplex, the values of the derivatives $\dot{k}_l^{(i)}(t)$ are determined by several incompatible conditions.

Thus, the estimate (11) needs to be modified, so that the resulting function $V^{(i)}(t,x)$ would be continuous. We propose an alternative way to calculate the matrices $Z^{(i)}$ rather than in [13].

Instead of (13), consider a differential equation for each column of the matrix $K^{(i)}$:

$$\dot{k}_l^{(i)}(t)+z_l^{(i)}(t)=0,\quad t\in[t_0,t_1],\ i=\overline{1,N},\ l=\overline{1,n_x+1}, \tag{15}$$

where $z_l^{(i)}$ is the corresponding column of the matrix $Z^{(i)}$. Hence the estimate (12) can be rewritten in the form

$$\frac{dV^{(i)}}{d\ell}(t,\tilde{x}) \leqslant \langle \tilde{x}, [\dot{K}^{(i)} + Z^{(i)}]\tilde{H}^{(i)}\tilde{x}\rangle = \langle \tilde{x}, \dot{K}^{(i)}\tilde{H}^{(i)}\tilde{x}\rangle + \langle \tilde{x}, Z^{(i)}\tilde{H}^{(i)}\tilde{x}\rangle$$
$$\leqslant \langle \tilde{x}, \dot{K}^{(i)}\tilde{H}^{(i)}\tilde{x}\rangle + \langle \tilde{x}, Z^{(i)}\alpha^{(i)}(x)\rangle = \langle \tilde{x}, \dot{K}^{(i)}\tilde{H}^{(i)}\tilde{x}\rangle + \sum_{l=1}^{n_x+1} \alpha_l^{(i)}(x)\langle \tilde{x}, z_l^{(i)}\rangle. \tag{16}$$

For any fixed $t \in [t_0, t_1]$ and each vertex $g_l^{(i)}$, consider an auxiliary linear programming problem with respect to a new unknown vector $\hat{z}_l^{(i)}$:

$$\begin{cases} \langle \hat{z}_l^{(i)}, \tilde{g}_l^{(i)}\rangle \to \min \\ \langle \hat{z}_l^{(i)}, \tilde{g}_k^{(j)}\rangle \geqslant \langle z_{\nu(i,l,j)}^{(j)}, \tilde{g}_k^{(j)}\rangle \quad \forall j: \ g_l^{(i)} \in \Omega^{(i)} \cap \Omega^{(j)}, \ k = \overline{1, n_x+1}, \end{cases} \tag{17}$$

where $\nu(i,l,j)$ denotes a local index of a vertex $g_l^{(i)} \in \Omega^{(i)} \cap \Omega^{(j)}$ in the simplex $\Omega^{(j)}$.

Using solutions $\hat{z}_l^{(i)}$, we obtain matrices $\hat{Z}^{(i)}$ in a similar way. Given conditions of the problem (17) and linearity of the functions under consideration, we can continue the inequality (16):

$$\frac{dV^{(i)}}{d\ell}(t,\tilde{x}) \leqslant \langle \tilde{x}, [\dot{K}^{(i)} + Z^{(i)}]\tilde{H}^{(i)}\tilde{x}\rangle \leqslant \langle \tilde{x}, \dot{K}^{(i)}\tilde{H}^{(i)}\tilde{x}\rangle + \sum_{l=1}^{n_x+1} \alpha_l^{(i)}(x)\langle \tilde{x}, z_l^{(i)}\rangle$$
$$\leqslant \langle \tilde{x}, \dot{K}^{(i)}\tilde{H}^{(i)}\tilde{x}\rangle + \sum_{l=1}^{n_x+1} \alpha_l^{(i)}(x)\langle \tilde{x}, \hat{z}_l^{(i)}\rangle = \langle \tilde{x}, [\dot{K}^{(i)} + \hat{Z}^{(i)}]\tilde{H}^{(i)}\tilde{x}\rangle.$$

Note that solutions of the problems (17) corresponding to the same vertex in different simplices $\Omega^{(i)}$ will coincide (if the linear programming problem admits several solutions, they can be chosen the same). Hence, the piecewise-defined value function (9) obtained by solving the Cauchy problem

$$\begin{cases} \dot{K}^{(i)} + \hat{Z}^{(i)} = 0, & i = \overline{1, N}, \quad t \in [t_0, t_1] \\ K^{(i)}(t_1) = \hat{K} \times (\tilde{H}^{(i)})^{-1}, & i = \overline{1, N}, \end{cases} \tag{18}$$

will be continuous in $(t, \tilde{x})$ throughout the entire domain. The objective function in (17) corresponds to the values $V^{(i)}(t, \tilde{x})$ at the simplex vertices, and thus it tends to reduce function values at these points.

5. CONTROL SELECTION

Before solving the problem (18), we need to determine the controls $y_k^{(i)}$ from (10) at the vertices of simplices in order to construct matrices $\hat{Z}^{(i)}$ based on these values. In the previous works [11–13], they were chosen in such a way that the derivative (11) of $V^{(i)}(t, \tilde{x})$ is minimized in each simplex $\Omega^{(i)}$. However, taking into account the piecewise defined nature of this function, there was ambiguity in the choice of $y_k^{(i)}$. To eliminate it, the controls were additionally adjusted, and this negatively affected the resulting solution.

In this paper, we demonstrate that the method allows the use of controls obtained on the basis of alternative approaches, reinforcement learning in particular. As a result, the constructed approximation of the value function (6) can be more accurate.

Reinforcement learning [16] is a domain of machine learning, where the agent's behavior is adjusted by repeated interaction with the environment, depending on the rewards received from it

for each action performed. In our task, the agent implements the control strategy $u = u(t, x)$ and we choose

$$\mathcal{L}(t,x) = \begin{cases} 0, & t < t_1 \\ -d^2(x, \mathcal{X}_1), & t = t_1, \end{cases} \tag{19}$$

as a reward function. Here $d(x, \mathcal{X}_1)$ is the distance between a point x and a set $\mathcal{X}_1$.

Proximal Policy Optimization (PPO) is one of the reinforcement learning algorithms in which the control strategy is represented using a neural network, and its weights are updated by gradient descent while optimizing some objective function. The objective is to maximize the cumulative reward at the end of the experiment; however, this function is described by a more complex expression [17] to ensure a stable learning process.

The advantage of the PPO algorithm is possibility of its application to continuous dynamics, including the system (1). Some other algorithms also have this property, for example, DDPG [18] and SAC [19]. They can also be used in the proposed approach. However, they showed lower accuracy in the examples discussed below.

Let the set $\mathcal{P}$ admit finite-dimensional parametrization. In this case, a vector $u \in \mathcal{P}$ is defined by parameters $\theta \in \mathbb{R}^r$, where $\theta_i \in [\theta_i^{\min}, \theta_i^{\max}]$, $i = \overline{1,r}$. The goal is to determine these parameters for each fixed position (t, x). However, since the PPO algorithm is designed for stochastic strategies, θ is usually assumed to be a random vector with a multidimensional normal distribution $\theta \sim \mathcal{N}(\mu, \Sigma)$ with a diagonal covariance matrix. When using the algorithm, at first a neural network is trained, which predicts the parameters of this distribution, and then realizations of the corresponding random vector are generated during the calculation of values $u(t, x)$. Nevertheless, having a trained neural network, it is easy to obtain deterministic control. Instead of generating a random vector, one can take the corresponding expected value: $\theta = \mu$.

Note that the values of θ_i are subject to interval constraints, while the support of a normal random vector is the entire space $\mathbb{R}^r$. In order to meet the requirements, the parameter values are "truncated" [26]. New values are obtained using the formula $\tilde{\theta}_i = \min\{\theta_i^{\max}, \max\{\theta_i, \theta_i^{\min}\}\}$, although other transformations are allowed. In addition, distributions with a bounded support [27] can be used for the specified random variables.

Such deterministic controls, which are based on a neural network model and satisfy the constraint (2), are further denoted as $\hat{u}(t, x)$. The resulting piecewise affine control used in this work is determined by formula (10):

$$u(t,x) = \sum_{k=1}^{n_x+1} \alpha_k^{(i)}(x) \hat{u}(t, g_k^{(i)}), \quad x \in \Omega^{(i)}. \tag{20}$$

The function $\hat{u}(t, x)$ will be continuous in (t, x) due to the structure of a neural network. It also follows that as the diameter of the partition of Ω into simplices tends to zero, the resulting control (20) will converge pointwise to $\hat{u}(t, x)$.

6. MAIN RESULT

The constructions above allow us to prove the following theorem.

Theorem 1. *Let the matrix functions $K^{(i)}(t) \in \mathbb{R}^{(n_x+1)\times(n_x+1)}$ be a solution of the Cauchy problem* (18)*. Let $V(t, \tilde{x})$ be a continuous piecewise quadratic function defined on $[t_0, t_1] \times \Omega \times \{1\}$ and determined by the equation $V^{(i)}(t, \tilde{x}) = \langle \tilde{x}, K^{(i)}(t) \tilde{H}^{(i)} \tilde{x} \rangle$ in each simplex $\Omega^{(i)}$. Then the set $\mathcal{W}_\varepsilon^{int}(t_0) = \left\{ x \in \Omega \;\middle|\; V(t_0, \tilde{x}) \leqslant \varepsilon \right\}$ (assuming that it is not empty) is an internal estimate of the solvability set of the original nonlinear system* (1)*, i.e.*

$$\mathcal{W}_\varepsilon^{int}(t_0) \subseteq \mathcal{W}_\varepsilon(t_0, t_1, \mathcal{X}_1).$$

The proof is based on trajectory analysis of the nonlinear system (1), closed by control (10). However, it does not depend on the method of finding the vectors $y_k^{(i)}(t) \in \mathcal{P}$ at the vertices of the simplices. The proof follows the scheme presented in [13].

7. EXAMPLES

7.1. Nonlinear System

Consider the motion of a pendulum on a trolley, taking the frictional force into account [28]. It is described by the system of equations

$$\begin{cases} \dot{x}_1 = x_2 \\ \dot{x}_2 = -w^2 \sin(x_1) - 2\gamma x_2 - w^2 \cos(x_1) \times u, \end{cases} \tag{21}$$

where ω and γ are parameters, x_1 and x_2 are the angle of pendulum deflection and the angular velocity, respectively. The control u denotes acceleration of the cart. We consider $\omega = 1$, $\gamma = 0.1$. It is required to steer the system from the initial position $(-0.3,\, 0.6)^T$ at time $t_0 = 0$ to a small neighborhood of the origin at time $t_1 = 1$. The control is bounded by $u \in [-1, 1]$.

For neural networks used in the PPO algorithm, we chose a two-layer perceptron [29] with activation function $\tanh(x)$ due to its simplicity. During training, 10 000 test trajectories of the system (21) were generated, starting from various random points $x^0 \in \Omega$ at time t_0. The control strategy $\hat{u}(t, x)$ was updated based on the penalties (19). Figure 1 shows the trajectory obtained using the PPO algorithm without any additional modifications. The distance between the end point of the trajectory and the origin is 0.027.

To calculate the piecewise quadratic function (9), at first we fixed the vertices $g_k \in \mathbb{R}^2$ located on a rectangular grid with sides of length $\Delta = 0.1$. These vertices were used to partition the set $\Omega = [-1, 1] \times [-1, 1]$ into $N = 800$ equal simplices. Figure 2 shows the results obtained using the control selection algorithm described in [13]. The dotted line indicates the boundary of the set which a trajectory of the system is a priori guaranteed to hit. The distance between $x(t_1)$ and the target set is 0.043.

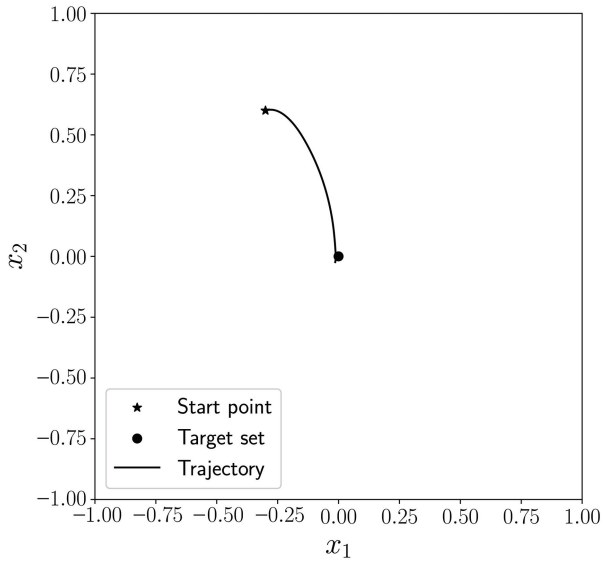

Fig. 1. The trajectory based on control $\hat{u}(t, x)$.

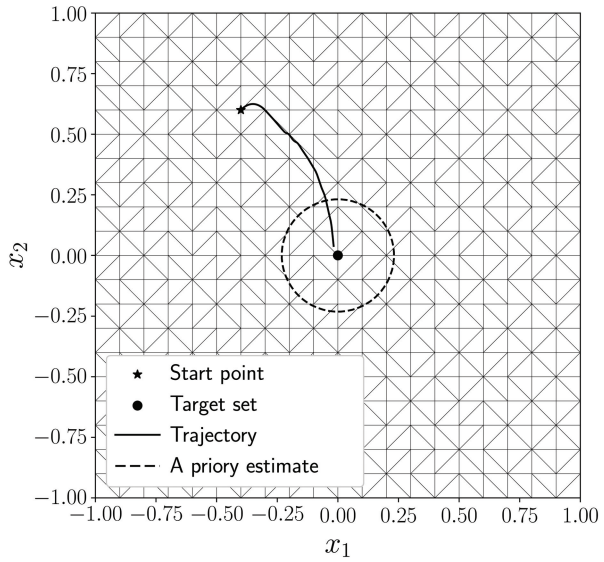

Fig. 2. The trajectory based on control strategy described in [13].

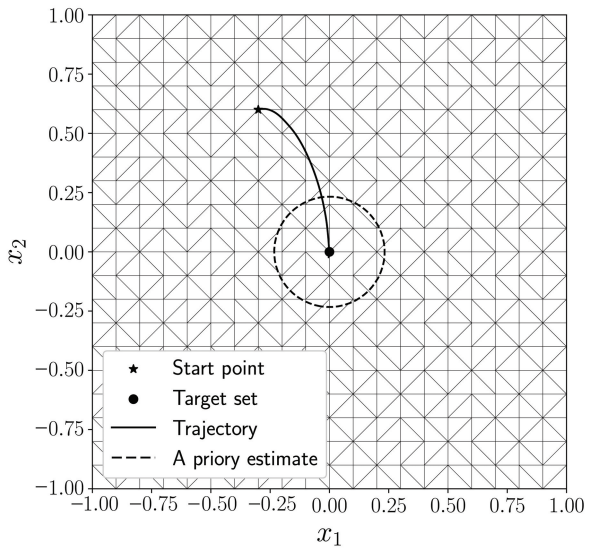

Fig. 3. The trajectory based on approximation (20) of neural network control $\hat{u}(t, x)$.

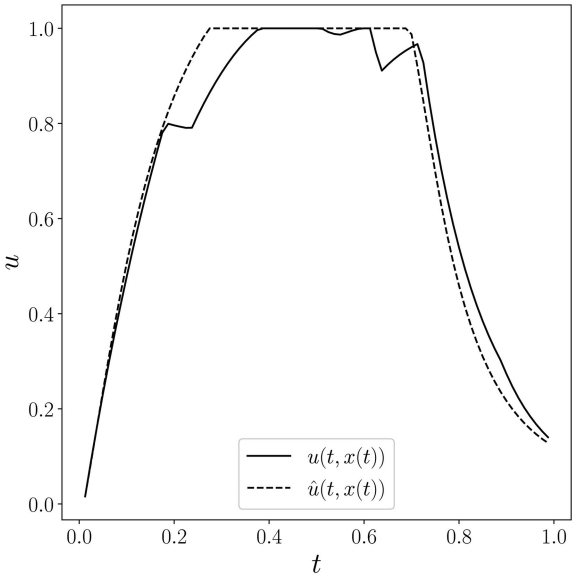

Fig. 4. Neural network control $\hat{u}(t, x(t))$ and the resulting control $u(t, x(t))$.

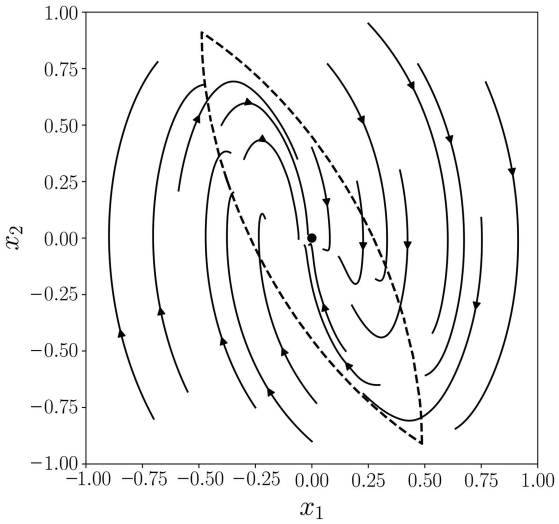

Fig. 5. The boundary of solvability set at $t = t_0$ and trajectories of the system (21) closed by the resulting control $u(t, x)$.

Figure 3 shows the trajectory obtained by the combination of methods described in the current work. The same partition into simplices is used. The distance from the origin in this case is 0.023. The change in error is explained by the difference between the original neural network control $\hat{u}(t, x)$ and its approximation (20). Figure 4 shows the controls corresponding to the trajectories shown in Figs. 1 and 3. The a priori error is less for the presented method than for the algorithm [13]. This example confirms that the a priori estimate obtained from the value function approximation (9) is guaranteed in each case.

In Fig. 5, continuous lines indicate the trajectories obtained by the proposed method when starting from various starting points. Arrows indicate the direction of movement along the trajectories. In addition, the dotted line stands for the boundary of the solvability set using the class of piecewise continuous program controls, calculated on the basis of the Pontryagin's maximum principle [30, pp. 336–344].

7.2. Linear System

To better understand accuracy of the proposed approach, consider a linear system

$$\dot{x}_1 = x_2, \quad \dot{x}_2 = u \tag{22}$$

on the segment $t \in [0, 1]$. In this case, the method of piecewise linearization described above is not needed, however, this system is well studied in literature [30]. Let the control satisfy $u \in [-2, 2]$. It is required to transfer the system to the origin at time $t = 1$. Then it can be proved that the point $x^0 = (0.5, 0)^T$ lies on the boundary of the solvability set at the moment $t = 0$, and it is achieved by piecewise constant control $u^*(t) = 2 \times \mathrm{sgn}(t - 0.5)$.

A neural network model of the same structure as in the previous example was chosen for numerical experiments. The model was trained on a personal computer for one hour, and then piecewise quadratic functions of the form (9) were constructed for different diameters of simplices $\Omega^{(i)}$. We consider the set $\Omega = [-1.5, 1.5] \times [-1.5, 1.5]$.

Figure 6 shows the resulting trajectory and the a priori estimate of hitting the origin from the point x^0 when step of the rectangular grid was $\Delta = 0.25$. This corresponds to the division into 288 simplices shown in the figure. Figure 7 shows the corresponding control $u(t, x(t))$ of the form (20). Figure 8 shows the solvability set calculated on the basis of the Pontryagin's maximum principle as well as the trajectories obtained by the proposed method when starting from various points.

Figure 9 shows dependencies of the a priori and a posteriori errors on the number of simplices in the partition $\Omega = \bigcup_{i=1}^{N} \Omega^{(i)}$ for the same starting point $x^0 = (0.5, 0)^T$. As the partition diameter decreases, the a posteriori error converges to 0.104, which corresponds to the accuracy of the original neural network control $\hat{u}(t, x)$. Note that this accuracy can be improved by considering other neural network models, which can require more parameters. In addition, it follows from Fig. 9 that the a priori error decreases at first, yet at some point it begins to increase again. This increase is explained by imperfection of auxiliary optimization problems (17): their solutions in neighboring vertices can differ significantly from each other, and it affects the stability of the method when using a small partition diameter. This problem can be eliminated by replacing the objective in

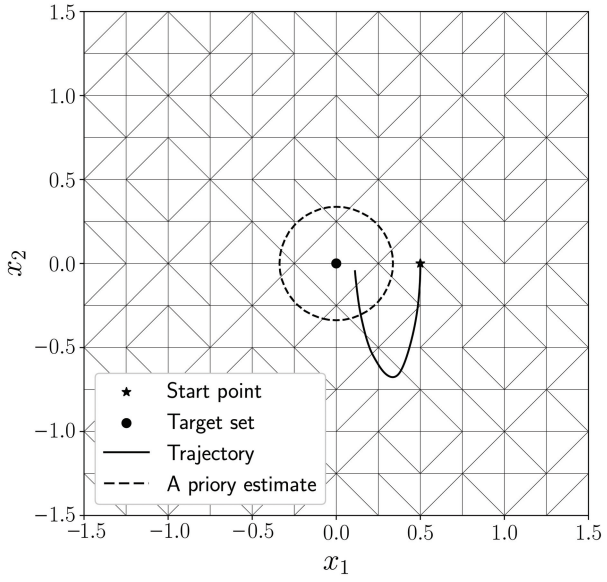

Fig. 6. Trajectory of the system (22) and a priori estimate of hitting the target point.

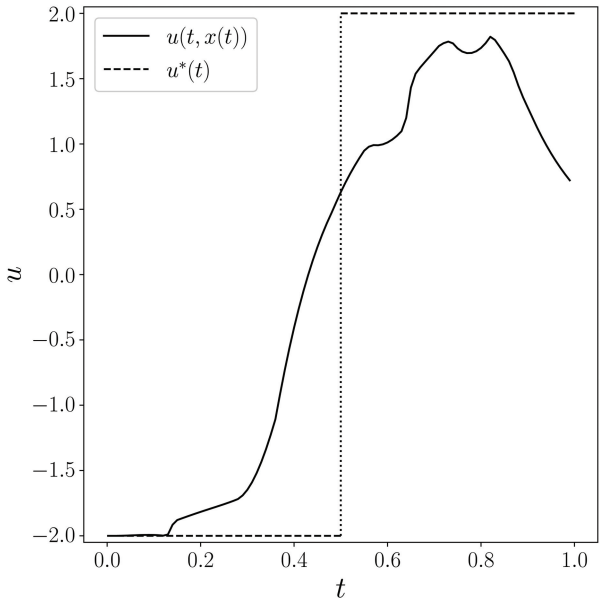

Fig. 7. The resulting control $u(t, x(t))$ for the system (22), and the optimal control $u^*(t)$.

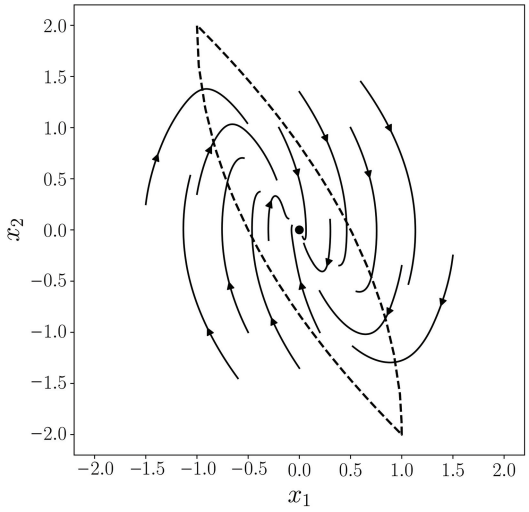

Fig. 8. Boundary of the solvability set at $t = t_0$ and trajectories of the system (22) when using the resulting control $u(t, x)$.

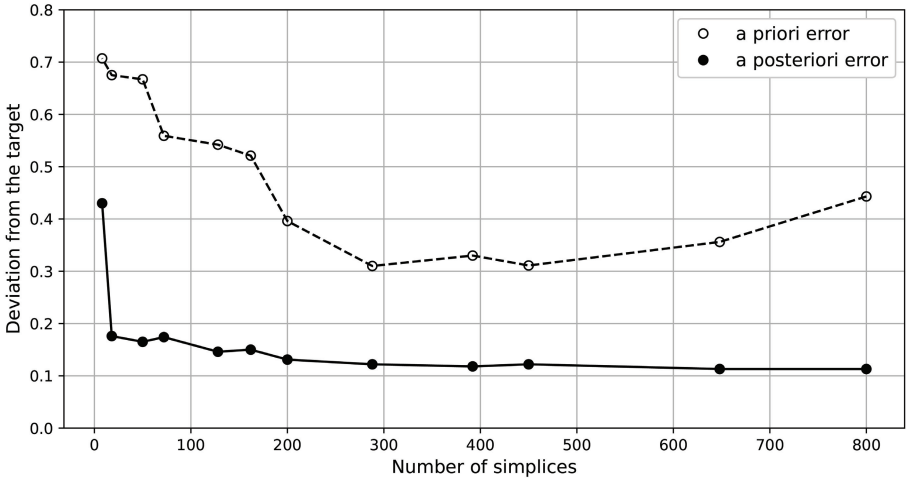

Fig. 9. Deviation from the target point $x^1 = (0, 0)^T$ depending on the number of partition simplices.

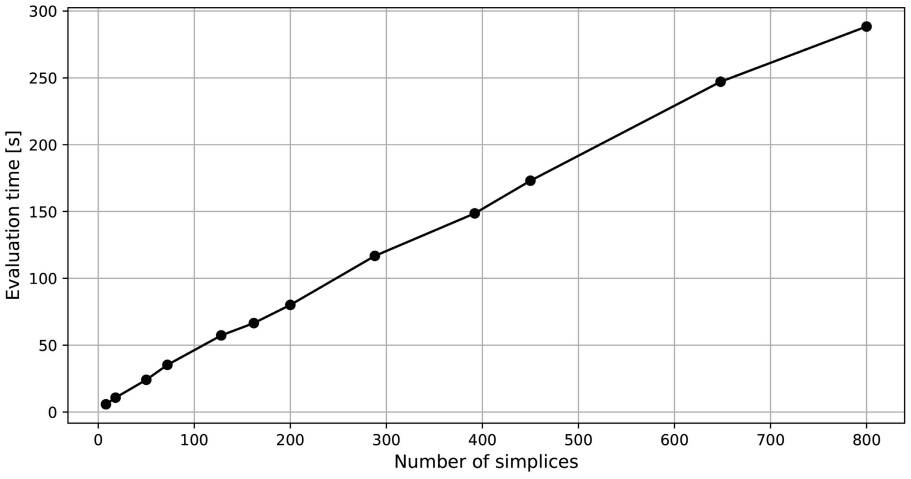

Fig. 10. Computation time of value function approximation for a fixed control strategy $\hat{u}(t, x)$.

problems (17) or by introducing additional "regularizing" terms into the system (18), which were proposed in [11].

Figure 10 shows computation times of the value function estimate depending on the number of simplices, using a fixed neural network strategy $\hat{u}(t,x)$. Time costs increase linearly as the number of simplices becomes greater. If this number is not too large, the computation time is small compared to the training time of a neural network.

8. CONCLUSION

The formulas presented in this paper make it possible to obtain a feedback control strategy that solves the problem approximately, as well as a piecewise affine approximation of this strategy on a set of simplices. The latter is used to construct a continuous piecewise quadratic function which defines an internal estimate of the solvability set in a target control problem. For the obtained piecewise affine control, a guaranteed a priori error estimate of hitting the target set is valid. The proposed approach can be used in solving control problems for nonlinear systems in case of a small state space dimension.

FUNDING

This work was carried out with financial support from the Ministry of Science and Higher Education of the Russian Federation within the framework of the program of the Moscow Center for Fundamental and Applied Mathematics under the agreement no. 075-15-2022-284.

REFERENCES

1. Neznakhin, A.A. and Ushakov, V.N., A discrete method for constructing an approximate viability kernel of a differential inclusion, *Zh. Vychisl. Mat. Mat. Fiz.*, 2001, vol. 41, no. 6, pp. 895–908.
2. Goubault, E. and Putot, S., Inner and Outer Reachability for the Verification of Control Systems, *Proceedings of the 22nd ACM International Conference on Hybrid Systems: Computation and Control*, 2019, pp. 11–22.
3. Shafa, T. and Ornik, M., *Reachability of Nonlinear Systems with Unknown Dynamics*, 2021.
4. Garrido, S., Moreno, L.E., Blanco, D., et al, Optimal control using the Fast Marching Method, *35th Annual Conference of IEEE Industrial Electronics*, 2009, pp. 1669–1674.
5. Subbotina, N.N. and Tokmantsev, T.B., Classical characteristics of the Bellman equation in constructions of grid optimal synthesis, *Proceedings of the Steklov institute of mathematics*, 2010, vol. 271, no. 1, pp. 246–264.
6. Xue, B., Fränzle, M., and Zhan, N., Inner-Approximating Reachable Sets for Polynomial Systems with Time-Varying Uncertainties, *IEEE Transactions on Automatic Control*, 2019, vol. 65, no. 4, pp. 1468–1483. https://doi.org/10.1109/TAC.2019.2923049
7. Lee, D. and Tomlin, C.J., Efficient Computation of State-Constrained Reachability Problems Using Hopf–Lax Formulae, *IEEE Transactions on Automatic Control*, 2023, pp. 1–15.
8. Cheng, T., Lewis, F.L., and Abu–Khalaf, M., Fixed-Final-Time-Constrained Optimal Control of Nonlinear Systems Using Neural Network HJB Approach, *IEEE Transactions on Neural Networks*, 2007, vol. 18, no. 6, pp. 1725–1737.
9. Onken, D., Nurbekyan, L., Li, X., et al., A Neural Network Approach for High-Dimensional Optimal Control Applied to Multiagent Path Finding, *IEEE Transactions on Control Systems Technology*, 2023, vol. 31, no. 1, pp. 235–251.
10. Sánchez–Sánchez, C., Izzo, D., and Hennes, D., Learning the optimal state-feedback using deep networks, *2016 IEEE Symposium Series on Computational Intelligence*, 2016, pp. 1–8.

11. Tochilin, P.A., Piecewise affine feedback control for approximate solution of the target control problem, *IFAC-PapersOnLine*, 2020, vol. 53, no. 2, pp. 6127–6132.

12. Tochilin, P.A., On the construction of a piecewise affine value function in an infinite-horizon optimal control problem, *Trudy Instituta Matematiki i Mekhaniki UrO RAN*, 2020, vol. 26, no. 1, pp. 223–238.

13. Chistyakov, I.A. and Tochilin, P.A., Application of Piecewise Quadratic Value Functions to the Approximate Solution of a Nonlinear Target Control Problem, *Differential Equations*, 2020, vol. 56, no. 11, pp. 1513–1523.

14. Kurzhanski, A.B., Comparison principle for equations of the Hamilton-Jacobi type in control theory, *Proceedings of the Steklov Institute of Mathematic*, 2006, vol. 253, pp. 185–195.

15. Kurzhanski, A.B. and Varaiya, P., *Dynamics and control of trajectory tubes. Theory and computation*, Birkhäuser, 2014.

16. Sutton, R.S. and Barto, A.G., *Reinforcement learning: An introduction*, MIT press, 2018.

17. Schulman, J., Wolski, F., Dhariwal, P., et al., Proximal policy optimization algorithms, 2017. https://doi.org/10.48550/arXiv.1707.06347

18. Lillicrap, T.P., Hunt, J.J., Pritzel, A., et al., Continuous control with deep reinforcement learning, 2019. https://doi.org/10.48550/arXiv.1509.02971

19. Haarnoja, T., Zhou, A., Abbeel, P., et al., Soft Actor-Critic: Off-Policy Maximum Entropy Deep Reinforcement Learning with a Stochastic Actor, 2018. https://doi.org/10.48550/arXiv.1801.01290

20. Pshenichnyi, B.N., *Vypuklyi analiz i ekstremal'nye zadachi* (Convex Analysis and Extremum Problems), Moscow: Nauka, 1980.

21. Skvortsov, A.V. and Mirza, N.S., *Algoritmy postroeniya i analiza trianguyatsii* (Algorithms for constructing and analyzing triangulation), Tomsk: Izd-vo Tom. un-ta, 2006.

22. Rajan, V.T., Optimality of the Delaunay triangulation in $\mathbb{R}^d$, *Discrete & Computational Geometry*, 1994, vol. 12, no. 2, pp. 189–202.

23. Brown, K.Q., Voronoi diagrams from convex hulls, *Information processing letters*, 1979, vol. 9, no. 5, pp. 223–228.

24. Liberzon, D., *Switching in Systems and Control*, Birkhäuser, 2003.

25. Bardi, M. and Capuzzo-Dolcetta, I., *Optimal control and viscosity solutions of Hamilton–Jacobi–Bellman equations. Ser. Systems & Control: Foundations & Applications*, Boston: Birkhäuser, 2008.

26. Raffin, A., Hill, A., Gleave, et al., Stable-Baselines3: Reliable Reinforcement Learning Implementations, *Journal of Machine Learning Research*, 2021, vol. 22, no. 268, pp. 1–8.

27. Petrazzini, I.G.B. and Antonelo, E.A., Proximal Policy Optimization with Continuous Bounded Action Space via the Beta Distribution, *2021 IEEE Symposium Series on Computational Intelligence (SSCI)*, 2022, pp. 1–8.

28. Reissig, G., Computing Abstractions of Nonlinear Systems, *IEEE Trans. Automatic Control*, 2011, vol. 56, no. 11, pp. 2583–2598.

29. Golubev, Yu.F., Neural networks in mechatronics, *Journal of Mathematical Sciences*, 2007, vol. 147, pp. 6607–6622.

30. Lee, E.B. and Markus, L., *Foundations of Optimal Control Theory*, Wiley, 1967.

This paper was recommended for publication by P.V. Pakshin, a member of the Editorial Board