

**Volume 86, Number 9,
September 2025**

**ISSN 0005-1179
CODEN: AURCAT**



AUTOMATION AND REMOTE CONTROL

**Editor-in-Chief
Andrey A. Galyaev**

<http://ait.mtas.ru>

Automation and Remote Control

Vol. 86, No. 9, September 2025

Available via license: CC BY 4.0

Automation and Remote Control

ISSN 0005-1179

Editor-in-Chief

Andrey A. Galyaev

Deputy Editors-in-Chief M.V. Khlebnikov and E.Ya. Rubinovich

Coordinating Editor A.S. Samokhin

Editorial Board

F.T. Aleskerov, A.V. Arutyunov, N.N. Bakhtadze, A.A. Bobtsov, P.Yu. Chebotarev, A.G. Chkhartshvili, L.Yu. Filimonyuk, A.L. Fradkov, O.N. Granichin, M.F. Karavai, E.M. Khorov, M.M. Khrustalev, A.I. Kibzun, S.A. Krasnova, A.P. Krishchenko, A.G. Kushner, N.V. Kuznetsov, A.A. Lazarev, A.I. Lyakhov, A.I. Matasov, S.M. Meerkov (USA), R.V. Mescheryakov, A.I. Mikhali'skii, B.M. Miller, O.V. Morzhin, R.A. Munasypov, A.V. Nazin, A.S. Nemirovskii (USA), D.A. Novikov, A.Ya. Oleinikov, P.V. Pakshin, D.E. Pal'chunov, A.E. Polyakov (France), V.Yu. Protasov, L.B. Rapoport, I.V. Rodionov, N.I. Selvesyuk, P.S. Shcherbakov, A.N. Sobolevski, O.A. Stepanov, A.B. Tsybakov (France), D.V. Vinogradov, V.M. Vishnevskii, K.V. Vorontsov, and L.Yu. Zhilyakova

Staff Editor E.A. Martekhina

SCOPE

Automation and Remote Control is one of the first journals on control theory. The scope of the journal is control theory problems and applications. The journal publishes reviews, original articles, and short communications (deterministic, stochastic, adaptive, and robust formulations) and its applications (computer control, components and instruments, process control, social and economy control, etc.).

Automation and Remote Control is abstracted and/or indexed in *ACM Digital Library*, *BFI List*, *CLOCKSS*, *CNKI*, *CNPIEC Current Contents/Engineering, Computing and Technology*, *DBLP*, *Dimensions*, *EBSCO Academic Search*, *EBSCO Advanced Placement Source*, *EBSCO Applied Science & Technology Source*, *EBSCO Computer Science Index*, *EBSCO Computers & Applied Sciences Complete*, *EBSCO Discovery Service*, *EBSCO Engineering Source*, *EBSCO STM Source*, *EI Compendex*, *Google Scholar*, *INSPEC*, *Japanese Science and Technology Agency (JST)*, *Journal Citation Reports/Science Edition*, *Mathematical Reviews*, *Naver*, *OCLC WorldCat Discovery Service*, *Portico*, *ProQuest Advanced Technologies & Aerospace Database*, *ProQuest-ExLibris Primo*, *ProQuest-ExLibris Summon*, *SCImago*, *SCOPUS*, *Science Citation Index*, *Science Citation Index Expanded (Sci-Search)*, *TD Net Discovery Service*, *UGC-CARE List (India)*, *WTI Frankfurt eG*, *zbMATH*.

Journal website: <http://ait.mtas.ru>

© The Author(s), 2025 published by Trapeznikov Institute of Control Sciences, Russian Academy of Sciences.

Automation and Remote Control participates in the Copyright Clearance Center (CCC) Transactional Reporting Service.

Available via license: CC BY 4.0

0005-1179/25. *Automation and Remote Control* (ISSN: 0005-1179 print version, ISSN: 1608-3032 electronic version) is published monthly by Trapeznikov Institute of Control Sciences, Russian Academy of Sciences, 65 Profsoyuznaya street, Moscow 117997, Russia.

Volume 86 (12 issues) is published in 2025.

Publisher: Trapeznikov Institute of Control Sciences, Russian Academy of Sciences.

65 Profsoyuznaya street, Moscow 117997, Russia; e-mail: redacsia@ipu.rssi.ru; <http://ait.mtas.ru>, <http://ait-arc.ru>

Contents

Automation and Remote Control

Vol. 86, No. 9, 2025

Special Issue¹

A Gradient Method with Inexact Gradient Information: A Study on Some Classes
of (L_0, L_1) -Smooth Nonconvex Problems

*S. S. Ablaev, F. S. Stonyakin, M. N. Fedotov, M. S. Alkousa, O. S. Savchuk,
and A. V. Gasnikov*

797

Gradient-Free Stochastic Optimization for Additive Models

A. Akhavan and A.B. Tsybakov

817

Robust Control Design Based on Noisy Current Data

R. S. Biryukov and M. M. Kogan

831

Reviews

An Overview of Conflict Prevention Methods in Air Traffic Control Using Deep Reinforcement
Learning

E. L. Kulida and V. G. Lebedev

845

Control in Technical Systems

Development of the Design and Trajectory Planning of a Snake-Like Manipulator Robot
Based Empirical Approach

*Xinbin Zhang, Zhuo Chen, Haoxiang Su, Liguo Tan, Hongwei Liu, Manlu Liu, Jianwen Huo,
and S. V. Novikova*

867

¹ Articles published in this heading, are the termination of special issue dedicated to the 90th anniversary of Boris Polyak (see Autom. Remote Control, 2025, no. 8).

A Gradient Method with Inexact Gradient Information: A Study on Some Classes of (L_0, L_1) -Smooth Nonconvex Problems

S. S. Ablaev^{*, **, a}, F. S. Stonyakin^{*, **, b}, M. N. Fedotov^{***, c}, M. S. Alkousa^{****, d},
 O. S. Savchuk^{*, ****, e}, and A. V. Gasnikov^{****, **, f}

^{*} Vernadsky Crimean Federal University, Simferopol, Russia

^{**} Moscow Institute of Physics and Technology, Dolgoprudny, Russia

^{***} National Research University Higher School of Economics, Moscow, Russia

^{****} Innopolis University, Innopolis, Russia

^{*****} Adyghe State University, Maykop, Russia

e-mail: ^aseydamet.ablaev@yandex.ru, ^bfedyor@mail.ru, ^cMaximFd-Nk@yandex.ru,

^dm.alkousa@innopolis.ru, ^eoleg.savchuk19@mail.ru, ^fgasnikov@yandex.ru

Received March 3, 2025

Revised May 20, 2025

Accepted June 27, 2025

Abstract—In this paper, a gradient method is studied on the class of (L_0, L_1) -smooth objective functions under the condition that only approximate values of the gradient are available at the iterations of the method. This setting arises when using noisy data. Two classes of problems are considered, namely, quasar-convex functions with respect to any solution that satisfy the Polyak–Łojasiewicz gradient dominance condition (the PL condition) and quasar-convex functions without the PL condition but with an additional constraint on the quasar-convexity parameter. For quasar-convex functions with the PL condition, a nearly linear convergence rate is established in the neighborhood of the exact solution. If inexact gradient values are sufficiently small (which is achieved in a finite number of iterations), the method will converge with a nearly linear rate on the class of problems with the PL condition without the additional assumption of quasar-convexity. For $(0, M)$ -smooth quasar-convex functions, an adaptive gradient method is proposed, and its convergence rate is estimated. As shown, when using exact gradient values, the method converges with a linear rate.

Keywords: gradient method, Δ -inexact gradient, (L_0, L_1) -smooth function, ρ -quasar-convex function, Polyak–Łojasiewicz condition, logistic regression

DOI: 10.31857/S0005117925090012

1. INTRODUCTION

With the rapid advancement of machine learning, it is increasingly topical to develop and analyze numerical optimization methods with effective convergence guarantees. To achieve the optimal convergence rate of conventional methods, such as gradient descent and its modifications, a key assumption is the smoothness of the objective function (i.e., the Lipschitz continuity of its gradient). In the context of modern machine learning problems, this assumption may turn out to be too stringent [1]. As a rule, quite natural and simple-to-implement methods perform poorly for deep learning problems, where the global smoothness condition often fails. For example, variance reduction methods [2–8] are known to be theoretically faster (for minimizing finite sums of smooth functions), but in practice may be inferior to methods that do not theoretically reduce variance [9]. According to the experiments presented in [1], when training neural networks, the norm of the

Hessian may correlate with the norm of the gradient of the loss function. Thus, the above factors motivate researchers to find and propose alternative assumptions relaxing the standard smoothness requirement for the objective function.

One such assumption is the so-called generalized (L_0, L_1) -smoothness, pioneered several years ago [1] for twice differentiable functions. This assumption allows the norm of the Hessian of the objective function to increase linearly with the growth of the gradient norm. In particular, (L_0, L_1) -smoothness can hold even for functions with polynomially growing gradients, which is a typical situation for deep learning problems. Moreover, the concept of (L_0, L_1) -smoothness covers a wide class of problems and can be generalized to differentiable, but not necessarily twice differentiable, functions [10, 11].

In recent years, many research works have considered various methods for solving (L_0, L_1) -smooth optimization problems. However, despite the interest from the scientific community, the existing results on the convergence of methods remain nonoptimal in several important cases, and the theoretical analysis is often insufficiently complete. While the most general class of nonconvex problems, admitting only the convergence of metrics like the gradient norm to zero, has been the focus of [1] and the later publications [10–16], the class of (L_0, L_1) -smooth convex functions has been investigated much less until very recent time (the late 2024–early 2025). In particular, when applied to (L_0, L_1) -smooth convex problems, known results on the convergence of methods, such as gradient descent with clipping [17] and gradient descent with Polyak’s step size [18], either additionally assume L -smoothness [19, 20] or restrict the step size to make the method’s trajectory bounded in a domain where the gradient satisfies the Lipschitz condition due to the (L_0, L_1) -smoothness of the objective function [21].

Let us review several recent works devoted to the optimization of (L_0, L_1) -smooth functions.

In [22], the class of strongly convex (L_0, L_1) -smooth functions was studied, and new convergence rate guarantees for several existing methods were obtained. In particular, better convergence rate estimates were derived for gradient descent with (smoothed) clipping and for gradient descent with Polyak’s step size. Unlike the existing results, these estimates neither involve the standard smoothness assumption nor have an exponential dependence on the distance from the initial point to the solution.

Parallel to [22], similar results for the gradient method with various step size strategies, including Polyak’s step, were independently obtained in [23], along with new complexity estimates for the gradient method with normalized step on the class of (L_0, L_1) -smooth problems.

A more detailed analysis of the convergence of gradient descent and its modifications under generalized (L_0, L_1) -smoothness was presented in [24]. In the convex case, it was shown that the convergence rate of the gradient method changes depending on the gradient norm: for $\|\nabla f(x_k)\| \geq \frac{L_0}{L_1}$, a linear rate is achieved, and a standard sublinear convergence rate is observed if $\|\nabla f(x_k)\| < \frac{L_0}{L_1}$.

Unlike the above-mentioned works, here we analyze the gradient method for (L_0, L_1) -smooth problems under the condition that only approximate (additively inexact) gradient values are available at each iteration of the method. That is, this research further develops, in a new direction, the paper [25], where similar problems were investigated for smooth problems in the classical sense. Note that several directions are considered. First, we study the convergence of the method for ρ -quasar-convex, with respect to any exact solution [26–28], (L_0, L_1) -smooth functions satisfying the Polyak–Łojasiewicz (PL) condition

$$f(x) - f(x_*) \leq \frac{1}{2\mu} \|\nabla f(x)\|^2 \quad \forall x \in \mathbb{R}^n, \mu > 0, \quad (1)$$

where $\|\cdot\|$ denotes the Euclidean norm and x_* is a minimizer of the function f . As is shown below, under some assumptions, the method converges with a linear rate to some neighborhood of the exact

solution. It is also proven that starting from some iteration, the method converges on the class of problems with the PL condition (without requiring quasar-convexity). Second, we separately consider the class of ρ -quasar-convex $(0, M)$ -smooth functions, representing interest in the context of machine learning problems (particularly when training logistic regression models). For this class of problems, an adaptive modification of the gradient method using inexact gradients at iterations is proposed, and convergence rate estimates are obtained under some constraint on the quasar-convexity parameter ρ . The efficiency of the proposed methods is confirmed by computational experiments for logistic regression problems (which satisfy the PL condition on any compact set [29]) and a certain nonconvex quasar-convex problem [27].

2. PROBLEM STATEMENT AND NECESSARY BACKGROUND

Consider minimization problems of the form

$$\min_{x \in \mathbb{R}^n} f(x), \quad (2)$$

where $f: \mathbb{R}^n \rightarrow \mathbb{R}$ is a convex and (L_0, L_1) -smooth function. Throughout this paper, x_* and $f^* = f(x_*)$ denote a global minimizer and the global minimum value, respectively, of the function f ; the norm $\|\cdot\|$ is Euclidean.

As already noted, the (L_0, L_1) -smoothness condition ($L_0, L_1 > 0$) was initially introduced in [1] for twice differentiable functions $f: \mathbb{R}^n \rightarrow \mathbb{R}$ as follows:

$$\|\nabla^2 f(x)\| \leq L_0 + L_1 \|\nabla f(x)\| \quad \forall x \in \mathbb{R}^n, \quad (3)$$

where $\|\cdot\|$ indicates the Euclidean norm for vectors and the spectral norm for matrices.

In [10], this concept was extended to the class of differentiable, but not necessarily twice differentiable, functions.

Definition 1. A function $f: \mathbb{R}^n \rightarrow \mathbb{R}$ is said to be (L_0, L_1) -smooth if, for any $x, y \in \mathbb{R}^n$ such that $\|y - x\| \leq \frac{1}{L_1}$,

$$\|\nabla f(x) - \nabla f(y)\| \leq (L_0 + L_1 \|\nabla f(y)\|) \|x - y\|. \quad (4)$$

Consider several examples of functions satisfying condition (4).

Example 1. The function $f(x) = \|x\|^{2m}$ is $(2m, 2m-1)$ -smooth. At the same time, for all $L \geq 0$, f is not L -smooth for $m \geq 2$; for details, see [22]. (Recall that an L -smooth function is one whose gradient satisfies the Lipschitz condition.)

Example 2. The function $f(x) = e^{ax}$, $a \in \mathbb{R}^n$, is $(0, \|a\|)$ -smooth. At the same time, for all $L \geq 0$, $f(x)$ is not L -smooth for $a \neq 0$ [22].

Example 3. The logistic function $f(x) = \log(1 + \exp(-a^T x))$, where $a \in \mathbb{R}^n$, is (L_0, L_1) -smooth with $L_0 = 0$ and $L_1 = \|a\|$ [22].

The concept of (L_0, L_1) -smoothness was generalized in [11] as follows: a function $f: \mathbb{R}^n \rightarrow \mathbb{R}$ is said to be α -symmetrically generalized smooth if, for some $\alpha \in [0, 1]$ and any $x, y \in \mathbb{R}^n$,

$$\|\nabla f(y) - \nabla f(x)\| \leq \left(L_0 + L_1 \max_{\theta \in [0, 1]} \|\nabla f(\theta y + (1 - \theta)x)\|^\alpha \right) \|y - x\|. \quad (5)$$

As demonstrated in [11], condition (5) is more general than (3) and (4): the validity of (3) or (4) implies the validity of (5) (with $\alpha = 1$). Moreover, for twice differentiable functions, conditions (3) and (5) are equivalent for $\alpha = 1$ [11].

Now we give the definition of a quasar-convex function, a key in this work.

Definition 2. Let $\rho \in (0, 1]$ and $\mu > 0$. A function $f : \mathbb{R}^n \rightarrow \mathbb{R}$ is said to be (ρ, μ) -quasar strongly convex if, for all $x \in \mathbb{R}^n$,

$$f(x_*) \geq f(x) + \frac{1}{\rho} \langle \nabla f(x), x_* - x \rangle + \frac{\mu}{2} \|x_* - x\|^2, \quad (6)$$

where x_* is a global minimizer of the function f . For $\mu = 0$ the function f is called a ρ -quasar convex function.

The PL condition can be treated as a relaxation of ordinary strong convexity and, moreover, of strong quasar-convexity [27]: if $f : \mathbb{R}^n \rightarrow \mathbb{R}$ is a (ρ, μ) -quasar strongly convex function, then it satisfies the PL condition (1) with the constant $\mu\rho^2$.

Example 4 [27]. Let $f : \mathbb{R}^n \rightarrow \mathbb{R}$ be defined as

$$f(x) = \|x\|^2 g\left(\frac{x}{\|x\|}\right), \quad \forall x \in \mathbb{R}^n,$$

where

$$g(x) = 1 + \sum_{i=1}^n a_i \sin(b_i x_i)^2, \quad \forall x = (x_1, \dots, x_n) \in \mathbb{R}^n.$$

Then the function f is $(1, 2)$ -quasar strongly convex, i.e., satisfies the PL-condition (1) with constant 2.

The issues of convergence of the gradient method for convex problems with the generalized smoothness condition (4) have been actively investigated recently; for example, see [22–24]. The cited research works presented results on the convergence of the normalized gradient method, the gradient method with Polyak's step, and the accelerated gradient method based on the assumption of generalized smoothness. It was also shown that the gradient method converges with a linear rate as long as the gradient norms at the iterations remain sufficiently large. This paper is a continuation and further development of the above works.

Here, we focus on studying gradient descent for the minimization problems (2) under the condition that, at each iteration of the method, only a gradient with an additive error is available instead of exact gradient values. Recall that for $\Delta > 0$, a vector $\tilde{\nabla}f(x)$ is called the inexact gradient of a function f at a point x if

$$\|\nabla f(x) - \tilde{\nabla}f(x)\| \leq \Delta;$$

and $\Delta > 0$ is called the gradient inexactness parameter.

In the case of small gradient norms at the method's iterations, it becomes challenging to derive theoretical estimates of the convergence rate. The reason is that, unlike the case of using exact gradients, it becomes impossible to ensure a monotonic decrease of the gradient norm at each iteration. This problem can be circumvented by additionally requiring that the objective function f satisfies the PL condition. According to [25], for problems with inexact gradients, the PL condition implies the inequality

$$\|\tilde{\nabla}f(x)\|^2 \geq \mu(f(x) - f^*) - \Delta^2 \quad \forall x \in \mathbb{R}^n, \quad \mu > 0. \quad (7)$$

This section provides the problem statement and necessary background. In Section 3, we study the gradient method for (L_0, L_1) -smooth problems under an additive error in the gradient at each iteration. The method's convergence rate on the class of ρ -quasar-convex functions satisfying the PL condition is estimated. As established, the convergence rate is linear up to a certain level of gradient values. Section 4 deals with an adaptive modification of the gradient method for $(0, M)$ -smooth ρ -quasar-convex problems. In Section 5, we present the results of computational experiments for the logistic regression problem (which satisfies the PL condition on any compact set [29]) and a certain nonconvex quasar-convex problem [27].

3. GRADIENT METHOD FOR (L_0, L_1) -SMOOTH ρ -QUASAR CONVEX FUNCTIONS

Let $f : \mathbb{R}^n \rightarrow \mathbb{R}$ be an (L_0, L_1) -smooth (4) and ρ -quasar convex function with respect to any solution x_* (see Definition 2, $\mu = 0$):

$$f(x_*) \geq f(x) + \frac{1}{\rho} \langle \nabla f(x), x_* - x \rangle, \quad \rho \in (0, 1]. \quad (8)$$

If f is strongly quasar-convex (see Definition 2, $\mu > 0$), then the solution x_* is unique and f is strongly quasar-convex with respect to any solution. Note that problems with nonconvex smooth (hence, (L_0, L_1) -smooth) ρ -quasar-convex functions are often encountered in applications (see Example 4) [27].

To solve problem (2), we investigate the gradient method of the form

$$x_{k+1} = x_k - \eta_k \tilde{\nabla} f(x_k) \quad (9)$$

with the step size

$$\eta_k = \frac{\alpha}{L_0 + L_1 \left(\|\tilde{\nabla} f(x_k)\| + \Delta \right)}, \quad (10)$$

where $\alpha \in (0, 1]$ and $\tilde{\nabla} f(x_k)$ denotes the inexact gradient of the function f at a point x_k .

Note that, despite the use of approximate gradient values in the algorithm, the theoretical analysis assumes the existence of the exact gradient.

Let us prove several lemmas necessary for deriving convergence rate estimates of the method.

Lemma 1 [10, Lemma A3]. *Let a function $f(x)$ be (L_0, L_1) -smooth (4). Then, for all $x, y \in \mathbb{R}^n$ such that $\|x - y\| \leq \frac{1}{L_1}$,*

$$f(y) \leq f(x) + \langle \nabla f(x), y - x \rangle + \frac{L_0 + L_1 \|\nabla f(x)\|}{2} \|y - x\|^2. \quad (11)$$

Proof. Let $g(t) = f(x + t(y - x))$ for $t \in [0, 1]$. Then

$$f(y) - f(x) = g(1) - g(0) = \int_0^1 g'(t) dt = \int_0^1 \langle \nabla f(x + t(y - x)), y - x \rangle dt.$$

We add and subtract $\langle \nabla f(x), y - x \rangle$ to/from the integrand:

$$f(y) - f(x) = \int_0^1 \langle \nabla f(x), y - x \rangle dt + \int_0^1 \langle \nabla f(x + t(y - x)) - \nabla f(x), y - x \rangle dt.$$

Computing the first integral and applying the Cauchy–Bunyakovsky–Schwarz inequality to the expression under the second integral, we obtain

$$f(y) - f(x) \leq \langle \nabla f(x), y - x \rangle + \int_0^1 \|\nabla f(x + t(y - x)) - \nabla f(x)\| \|y - x\| dt.$$

In view of

$$\|\nabla f(y) - \nabla f(x)\| \leq (L_0 + L_1 \|\nabla f(x)\|) \|y - x\|,$$

we have the inequality

$$f(y) - f(x) \leq \langle \nabla f(x), y - x \rangle + \|y - x\| \int_0^1 (L_0 + L_1 \|\nabla f(x)\|) \|y - x\| t dt$$

and, consequently,

$$f(y) \leq f(x) + \langle \nabla f(x), y - x \rangle + \frac{L_0 + L_1 \|\nabla f(x)\|}{2} \|y - x\|^2.$$

Remark 1. Under an additive error in the gradient, inequality (11) takes the form

$$f(y) \leq f(x) + \langle \tilde{\nabla} f(x), y - x \rangle + \frac{L_0 + L_1 (\|\tilde{\nabla} f(x)\| + \Delta)}{2} \|y - x\|^2 + \Delta \|y - x\|. \quad (12)$$

Lemma 2. Let $f(x)$ be an (L_0, L_1) -smooth function (4). Assume that $\|\nabla f(x_k)\| > \Delta$. Then at the iterations of algorithm (9) with the step size (10), the values of this function decrease monotonically, $f(x_{k+1}) \leq f(x_k)$, and

$$f(x_k) - f(x_{k+1}) \geq \frac{\alpha(2 - \alpha)}{2} \frac{(\|\nabla f(x_k)\| - \Delta)^2}{L_0 + L_1 \|\nabla f(x_k)\| + 2L_1 \Delta}. \quad (13)$$

Proof. Consider inequality (11) with $y = x_{k+1}$ and $x = x_k$. Note that $\|x_{k+1} - x_k\| \leq \frac{1}{L_1}$. Then we have

$$\begin{aligned} & f(x_{k+1}) - f(x_k) \\ & \leq \frac{-\alpha \langle \nabla f(x_k), \tilde{\nabla} f(x_k) \rangle}{L_0 + L_1 (\|\tilde{\nabla} f(x_k)\| + \Delta)} + \frac{L_0 + L_1 (\|\tilde{\nabla} f(x_k)\| + \Delta)}{2 (L_0 + L_1 (\|\tilde{\nabla} f(x_k)\| + \Delta))^2} \alpha^2 \|\tilde{\nabla} f(x_k)\|^2 \\ & = \frac{\alpha^2 \|\tilde{\nabla} f(x_k)\|^2 - 2\alpha \langle \nabla f(x_k), \tilde{\nabla} f(x_k) \rangle}{2 (L_0 + L_1 (\|\tilde{\nabla} f(x_k)\| + \Delta))} \\ & = \frac{\alpha^2 \|\tilde{\nabla} f(x_k)\|^2 - 2\alpha \langle \nabla f(x_k), \tilde{\nabla} f(x_k) \rangle + \|\nabla f(x_k)\|^2 - \|\nabla f(x_k)\|^2}{2 (L_0 + L_1 (\|\tilde{\nabla} f(x_k)\| + \Delta))} \\ & = \frac{\|\alpha \tilde{\nabla} f(x_k) - \nabla f(x_k)\|^2 - \|\nabla f(x_k)\|^2}{2 (L_0 + L_1 (\|\tilde{\nabla} f(x_k)\| + \Delta))}. \end{aligned}$$

Therefore,

$$f(x_k) - f(x_{k+1}) \geq \frac{\|\nabla f(x_k)\|^2 - \|\alpha \tilde{\nabla} f(x_k) - \nabla f(x_k)\|^2}{2 (L_0 + L_1 (\|\tilde{\nabla} f(x_k)\| + \Delta))}.$$

We utilize the following relation for $\alpha \in (0, 1]$ and $a, b \in \mathbb{R}^n$:

$$\|\alpha a - b\| = \|\alpha(a - b) + (\alpha - 1)b\| \leq \alpha \|a - b\| + (1 - \alpha) \|b\|.$$

As a result,

$$\begin{aligned}
& f(x_k) - f(x_{k+1}) \\
& \geq \frac{\|\nabla f(x_k)\|^2 - \left(\alpha\|\tilde{\nabla} f(x_k) - \nabla f(x_k)\| + (1-\alpha)\|\nabla f(x_k)\|\right)^2}{2\left(L_0 + L_1\left(\|\tilde{\nabla} f(x_k)\| + \Delta\right)\right)} \\
& \geq \frac{\|\nabla f(x_k)\|^2 - (\alpha\Delta + (1-\alpha)\|\nabla f(x_k)\|)^2}{2\left(L_0 + L_1\left(\|\tilde{\nabla} f(x_k)\| + \Delta\right)\right)} \\
& = \frac{\alpha(\|\nabla f(x_k)\| - \Delta)((2-\alpha)(\|\nabla f(x_k)\| - \Delta) + 2\Delta)}{2\left(L_0 + L_1\left(\|\tilde{\nabla} f(x_k)\| + \Delta\right)\right)} \\
& > \frac{\alpha(2-\alpha)}{2} \frac{(\|\nabla f(x_k)\| - \Delta)^2}{L_0 + L_1\left(\|\tilde{\nabla} f(x_k)\| + \Delta\right)} \\
& \geq \frac{\alpha(2-\alpha)}{2} \frac{(\|\nabla f(x_k)\| - \Delta)^2}{L_0 + L_1\|\nabla f(x_k)\| + 2L_1\Delta}.
\end{aligned}$$

Lemma 3. Let f be an (L_0, L_1) -smooth function (4) satisfying the PL condition (1), and let x_* be the exact solution of problem (2) closest to x_{k+1} . Then the following inequality holds for the gradient method (9) with the step size (10):

$$\|x_{k+1} - x_*\| \leq \sqrt{\frac{2}{\mu}(f(x_0) - f^*)}. \quad (14)$$

Proof. According to [29, Appendix A], for the closest exact solution x_* , the PL condition implies the quadratic growth of the function:

$$\frac{\mu}{2}\|x_{k+1} - x_*\|^2 \leq f(x_{k+1}) - f^*.$$

Due to the monotonicity of the function f , the inequality becomes

$$\frac{\mu}{2}\|x_{k+1} - x_*\|^2 \leq f(x_0) - f^*,$$

and the desired inequality is immediate:

$$\|x_{k+1} - x_*\| \leq \sqrt{\frac{2}{\mu}(f(x_0) - f^*)}.$$

Let us denote $g_k = \|\nabla f(x_k)\| - \Delta$. From Lemma 2 it follows that

$$f(x_k) - f(x_{k+1}) > \frac{\alpha(2-\alpha)}{2} \frac{g_k^2}{L_0 + L_1g_k + 3L_1\Delta}. \quad (15)$$

Assume that $g_k \geq 3\Delta$. We will investigate the convergence of the method depending on the value of g_k . Two cases are possible: $g_k > \frac{L_0}{L_1}$ and $g_k \leq \frac{L_0}{L_1}$.

For $g_k > \frac{L_0}{L_1}$, from (15) we have

$$f(x_k) - f(x_{k+1}) > \frac{\alpha(2-\alpha)}{2} \frac{g_k^2}{L_1g_k + L_1g_k + L_1g_k} > \frac{\alpha(2-\alpha)}{2} \frac{g_k}{3L_1} = \frac{\alpha(2-\alpha)}{6L_1} g_k. \quad (16)$$

If $g_k \leq \frac{L_0}{L_1}$, then inequality (15) gives

$$f(x_k) - f(x_{k+1}) > \frac{\alpha(2-\alpha)}{2} \frac{g_k^2}{L_0 + L_0 + L_0} > \frac{\alpha(2-\alpha)}{2} \frac{g_k^2}{3L_0} = \frac{\alpha(2-\alpha)}{6L_0} g_k^2. \quad (17)$$

The following result is true for this class of problems.

Theorem 1. *Let f be a ρ -quasar-convex function with respect to each exact solution and, simultaneously, an (L_0, L_1) -smooth function satisfying the PL condition (1). Also, let $\|\tilde{\nabla} f(x_k)\| \geq 5\Delta$ and $\mu < \min \left\{ \frac{6L_0}{\alpha(2-\alpha)}; \frac{72L_1^2(f(x_0)-f^*)}{\alpha^2(2-\alpha)^2\rho^2} \right\}$. Then the following inequality holds for algorithm (9) with the step size (10):*

$$\begin{aligned} f(x_{k+1}) - f^* &\leq \max \left\{ 1 - \frac{\alpha(2-\alpha)\rho}{6L_1\sqrt{\frac{2}{\mu}(f(x_0)-f^*)}}; 1 - \frac{\alpha(2-\alpha)\mu}{6L_0} \right\}^{k+1} (f(x_0) - f^*) \\ &\quad + \Delta \max \left\{ \frac{1}{L_1}; \frac{\Delta}{L_0} \right\} \max \left\{ \frac{L_1\sqrt{\frac{2}{\mu}(f(x_0)-f^*)}}{\rho}; \frac{L_0}{\mu} \right\}. \end{aligned} \quad (18)$$

Proof. Consider first the case $g_k > \frac{L_0}{L_1}$. By the quasar-convexity of the function $f(x)$ at $x = x_k$, we have the inequality

$$\begin{aligned} f(x_k) - f^* &\leq \frac{1}{\rho} \langle \nabla f(x_k), x_k - x_* \rangle \leq \frac{1}{\rho} \|\nabla f(x_k)\| \|x_k - x_*\| \\ &= \frac{1}{\rho} (g_k + \Delta) \|x_k - x_*\|. \end{aligned} \quad (19)$$

Applying Lemma 3 to (19) yields

$$f(x_k) - f^* \leq \frac{1}{\rho} \sqrt{\frac{2}{\mu} (f(x_0) - f^*)} (g_k + \Delta)$$

and, consequently,

$$g_k \geq \frac{\rho(f(x_k) - f^*)}{\sqrt{\frac{2}{\mu} (f(x_0) - f^*)}} - \Delta. \quad (20)$$

Inequalities (16) and (20) imply the estimate

$$\begin{aligned} f(x_{k+1}) - f^* &\leq f(x_k) - f^* - \frac{\alpha(2-\alpha)}{6L_1} \left(\frac{\rho(f(x_k) - f^*)}{\sqrt{\frac{2}{\mu} (f(x_0) - f^*)}} - \Delta \right) \\ &= \left(1 - \frac{\alpha(2-\alpha)\rho}{6L_1\sqrt{\frac{2}{\mu} (f(x_0) - f^*)}} \right) (f(x_k) - f^*) + \frac{\alpha(2-\alpha)\Delta}{6L_1}. \end{aligned} \quad (21)$$

For $g_k \leq \frac{L_0}{L_1}$, from (17) it follows that

$$\begin{aligned} f(x_{k+1}) - f(x_k) &\leq -\frac{\alpha(2-\alpha)}{6L_0} g_k^2 = -\frac{\alpha(2-\alpha)}{6L_0} (\|\nabla f(x_k)\| - \Delta)^2 \\ &\leq -\frac{\alpha(2-\alpha)}{6L_0} \left(\frac{1}{2} \|\nabla f(x_k)\|^2 - \Delta^2 \right) \\ &= -\frac{\alpha(2-\alpha)}{12L_0} \|\nabla f(x_k)\|^2 + \frac{\alpha(2-\alpha)\Delta^2}{6L_0} \\ &= -\frac{\alpha(2-\alpha)\mu}{6L_0} (f(x_k) - f^*) + \frac{\alpha(2-\alpha)\Delta^2}{6L_0}. \end{aligned}$$

Thus,

$$f(x_{k+1}) - f^* \leq \left(1 - \frac{\alpha(2-\alpha)\mu}{6L_0}\right) (f(x_k) - f^*) + \frac{\alpha(2-\alpha)\Delta^2}{6L_0}. \quad (22)$$

Due to inequalities (21) and (22), we obtain

$$\begin{aligned} f(x_{k+1}) - f^* &\leq \max \left\{ 1 - \frac{\alpha(2-\alpha)\rho}{6L_1\sqrt{\frac{2}{\mu}}(f(x_0) - f^*)}; 1 - \frac{\alpha(2-\alpha)\mu}{6L_0} \right\} (f(x_k) - f^*) \\ &\quad + \max \left\{ \frac{\alpha(2-\alpha)\Delta}{6L_1}; \frac{\alpha(2-\alpha)\Delta^2}{6L_0} \right\}, \end{aligned}$$

or

$$\begin{aligned} f(x_{k+1}) - f^* &\leq \max \left\{ 1 - \frac{\alpha(2-\alpha)\rho}{6L_1\sqrt{\frac{2}{\mu}}(f(x_0) - f^*)}; 1 - \frac{\alpha(2-\alpha)\mu}{6L_0} \right\} (f(x_k) - f^*) \\ &\quad + \frac{\alpha(2-\alpha)\Delta}{6} \max \left\{ \frac{1}{L_1}; \frac{\Delta}{L_0} \right\} \\ &\leq \max \left\{ 1 - \frac{\alpha(2-\alpha)\rho}{6L_1\sqrt{\frac{2}{\mu}}(f(x_0) - f^*)}; 1 - \frac{\alpha(2-\alpha)\mu}{6L_0} \right\}^{k+1} (f(x_0) - f^*) \\ &\quad + \frac{\alpha(2-\alpha)\Delta}{6} \max \left\{ \frac{1}{L_1}; \frac{\Delta}{L_0} \right\} \sum_{i=0}^k \max \left\{ 1 - \frac{\alpha(2-\alpha)\rho}{6L_1\sqrt{\frac{2}{\mu}}(f(x_0) - f^*)}; 1 - \frac{\alpha(2-\alpha)\mu}{6L_0} \right\}^i \\ &\leq \max \left\{ 1 - \frac{\alpha(2-\alpha)\rho}{6L_1\sqrt{\frac{2}{\mu}}(f(x_0) - f^*)}; 1 - \frac{\alpha(2-\alpha)\mu}{6L_0} \right\}^{k+1} (f(x_0) - f^*) \\ &\quad + \frac{\alpha(2-\alpha)\Delta}{6} \max \left\{ \frac{1}{L_1}; \frac{\Delta}{L_0} \right\} \frac{1}{1 - \max \left\{ 1 - \frac{\alpha(2-\alpha)\rho}{6L_1\sqrt{\frac{2}{\mu}}(f(x_0) - f^*)}; 1 - \frac{\alpha(2-\alpha)\mu}{6L_0} \right\}}. \end{aligned}$$

The following chain of inequalities is true for positive numbers a and b :

$$\frac{1}{1 - \max\{1-a; 1-b\}} = \frac{1}{1 - (1 - \min\{a; b\})} = \frac{1}{\min\{a; b\}} = \max \left\{ \frac{1}{a}; \frac{1}{b} \right\}.$$

Hence,

$$\begin{aligned} f(x_{k+1}) - f^* &\leq \max \left\{ 1 - \frac{\alpha(2-\alpha)\rho}{6L_1\sqrt{\frac{2}{\mu}}(f(x_0) - f^*)}; 1 - \frac{\alpha(2-\alpha)\mu}{6L_0} \right\}^{k+1} (f(x_0) - f^*) \\ &\quad + \frac{\alpha(2-\alpha)\Delta}{6} \max \left\{ \frac{1}{L_1}; \frac{\Delta}{L_0} \right\} \frac{6}{\alpha(2-\alpha)} \max \left\{ \frac{L_1\sqrt{\frac{2}{\mu}}(f(x_0) - f^*)}{\rho}; \frac{L_0}{\mu} \right\} \\ &\leq \max \left\{ 1 - \frac{\alpha(2-\alpha)\rho}{6L_1\sqrt{\frac{2}{\mu}}(f(x_0) - f^*)}; 1 - \frac{\alpha(2-\alpha)\mu}{6L_0} \right\}^{k+1} (f(x_0) - f^*) \\ &\quad + \Delta \max \left\{ \frac{1}{L_1}; \frac{\Delta}{L_0} \right\} \max \left\{ \frac{L_1\sqrt{\frac{2}{\mu}}(f(x_0) - f^*)}{\rho}; \frac{L_0}{\mu} \right\}. \end{aligned}$$

Remark 2. For algorithm (9), the number of iterations with $\|\tilde{\nabla}f(x_k)\| \geq 5\Delta$ can be estimated as

$$N \leq \left\lceil \frac{6L_1^2(f(x_0) - f^*)}{\alpha(2 - \alpha)L_0} \right\rceil + \left\lceil \frac{2L_0(f(x_0) - f^*)}{3\alpha(2 - \alpha)\Delta^2} \right\rceil.$$

Proof. Let us estimate the number of iterations where $g_k > \frac{L_0}{L_1}$. From (16) it follows that

$$f(x_k) - f(x_{k+1}) > \frac{\alpha(2 - \alpha)L_0}{6L_1^2}.$$

Let $i_1, \dots, i_m, p = \overline{1, m}$, be the iterations for which $g_{i_p} > \frac{L_0}{L_1}$. Then summation yields the inequality

$$f(x_0) - f^* \geq f(x_{i_1}) - f(x_{i_m}) \geq \frac{\alpha(2 - \alpha)L_0}{6L_1^2}m$$

and, consequently,

$$m \leq \left\lceil \frac{6L_1^2(f(x_0) - f^*)}{\alpha(2 - \alpha)L_0} \right\rceil, \quad L_0 \neq 0.$$

Now we estimate the number of iterations for which $g_k \leq \frac{L_0}{L_1}$. In view of $g_k^2 \geq 9\Delta^2$, inequality (17) leads to the estimate $f(x_k) - f(x_{k+1}) > \frac{9\alpha(2 - \alpha)\Delta^2}{6L_0}$. Let $j_1, \dots, j_n, l = \overline{1, n}$, be the iterations for which $g_{j_l} \leq \frac{L_0}{L_1}$. Then summation yields the inequality

$$f(x_0) - f^* \geq f(x_{j_1}) - f(x_{j_n}) \geq \frac{9\alpha(2 - \alpha)\Delta^2}{6L_0}n,$$

and the number of such iterations satisfies the upper bound

$$n \leq \left\lceil \frac{2L_0(f(x_0) - f^*)}{3\alpha(2 - \alpha)\Delta^2} \right\rceil.$$

Remark 3. According to inequality (21), the method (9) with the step size (10) converges with a rate close to that of a geometric progression for sufficiently large gradient norms (i.e., $g_k > \frac{L_0}{L_1}$).

Remark 4. Suppose that an acceptable solution quality is achieved under the condition $\|\tilde{\nabla}f(x_k)\| < 5\Delta$. Indeed, if $\|\tilde{\nabla}f(x_k)\| < 5\Delta$, then $\|\nabla f(x_k)\| < 6\Delta$, and the PL condition (1) gives

$$f(x_k) - f^* \leq \frac{1}{2\mu}\|\nabla f(x_k)\|^2 < \frac{18\Delta^2}{\mu}.$$

Remark 5. Since the number of iterations with $g_k > \frac{L_0}{L_1}$ is limited, starting from some iteration N the gradient norm will be less than $\frac{L_0}{L_1}$. Then it can be stated that the method (9) with the step size (10) converges on the class of problems with the PL condition (without requiring ρ -quasar-convexity) starting from this number. Thus, starting from some iteration, we obtain:

1) either

$$f(x_{k+1}) - f^* \leq \left(1 - \frac{\alpha(2 - \alpha)\mu}{6L_0}\right)^k (f(x_N) - f^*) + \frac{\Delta^2}{\mu}$$

from inequality (22) by recursion,

2) or an acceptable accuracy has already been achieved.

4. ADAPTIVE GRADIENT METHOD FOR QUASAR-CONVEX (0, M)-SMOOTH FUNCTIONS

Consider the case $L_0 = 0$, corresponding, e.g., to the logistic regression problem. Denoting $M = L_1$ for convenience, we write condition (4) as

$$\|\nabla f(y) - \nabla f(x)\| \leq M \|\nabla f(x)\| \|y - x\| \quad \forall x, y \in \mathbb{R}^n \text{ such that } \|y - x\| \leq \frac{1}{M}. \quad (23)$$

In this section, an adaptive modification of the gradient method using inexact gradient information will be investigated on the class of ρ -quasar convex functions.

Let f be a $(0, M)$ -smooth and ρ -quasar-convex function with $\rho \in (\frac{1}{2}, 1]$. (Unlike the previous section, this condition holds not necessarily with respect to each exact solution.) Consider the following adaptive modification of the gradient method using inexact gradient values at the iterations.

Algorithm 1 (the adaptive gradient method for $(0, M)$ -smooth problems).

1. Input: $x_0 \in \mathbb{R}^n$, $\Delta > 0$, $0 < M_0 < 2M$, and $\alpha > 0$.
2. $k = 0$.
3. Repeat:
4. $M_{k+1} = \max \{M_0/2, M_k/2\}$,
5. $\eta_k = \frac{\alpha}{M_{k+1}(\|\tilde{\nabla} f(x_k)\| + \Delta)}$,
6. $x_{k+1} = x_k - \eta_k \tilde{\nabla} f(x_k)$,
7. If $f(x_k) - f(x_{k+1}) \geq \frac{\alpha(2-\alpha)}{4M_{k+1}} (\|\tilde{\nabla} f(x_k)\| - 2\Delta)$ and

$$f(x_k) - f(x_{k+1}) \geq \alpha \left(1 - \frac{\alpha}{2}\right) \frac{\|\tilde{\nabla} f(x_k)\|^2}{M_{k+1}(\|\tilde{\nabla} f(x_k)\| + \Delta)} - \frac{\Delta\alpha}{M_{k+1}},$$
8. then $k := k + 1$;
9. otherwise,
10. $M_{k+1} = 2M_{k+1}$ and get back to line 5.
11. Output: $\hat{x} = x_{k+1}$.

Now we prove several auxiliary statements for $(0, M)$ -smooth functions. In particular, they will justify the correctness of Algorithm 1, i.e., the validity of the iteration exit criterion (see line 7 of Algorithm 1) for $(0, M)$ -smooth functions for sufficiently large M_{k+1} .

Lemma 4. *Let $f(x)$ be a $(0, M)$ -smooth function. Assume that $\|\tilde{\nabla} f(x_k)\| > 5\Delta$ and $\{M_i\}_{i=0}^k$ is a sequence such that $M_{i+1} \geq M$. Then at the iterations of the algorithm*

$$x_{k+1} = x_k - \frac{\alpha}{M_{k+1}(\|\tilde{\nabla} f(x_k)\| + \Delta)} \tilde{\nabla} f(x_k), \quad \alpha \in (0, 1],$$

the values of this function are monotonically nonincreasing, $f(x_{k+1}) \leq f(x_k)$, and

$$f(x_k) - f(x_{k+1}) \geq \frac{\alpha(2-\alpha)}{2} \frac{(\|\tilde{\nabla} f(x_k)\| - 2\Delta)^2}{M_{k+1}(\|\tilde{\nabla} f(x_k)\| + \Delta)}. \quad (24)$$

Proof. For $L_0 = 0$ and $L_1 = M$, inequality (11) implies

$$f(y) \leq f(x) + \langle \nabla f(x), y - x \rangle + \frac{M \|\nabla f(x)\|}{2} \|y - x\|^2.$$

With $y = x_{k+1}$ and $x = x_k$, we obtain

$$\begin{aligned}
 & f(x_{k+1}) - f(x_k) \\
 & \leq \frac{-\alpha \langle \nabla f(x_k), \tilde{\nabla} f(x_k) \rangle}{M_{k+1} (\|\tilde{\nabla} f(x_k)\| + \Delta)} + \frac{M \|\nabla f(x_k)\|}{2M_{k+1}^2 (\|\tilde{\nabla} f(x_k)\| + \Delta)^2} \alpha^2 \|\tilde{\nabla} f(x_k)\|^2 \\
 & \leq \frac{-\alpha \langle \nabla f(x_k), \tilde{\nabla} f(x_k) \rangle}{M_{k+1} (\|\tilde{\nabla} f(x_k)\| + \Delta)} + \frac{M (\|\tilde{\nabla} f(x_k)\| + \Delta)}{2M (\|\tilde{\nabla} f(x_k)\| + \Delta)} \frac{\alpha^2 \|\tilde{\nabla} f(x_k)\|^2}{M_{k+1} (\|\tilde{\nabla} f(x_k)\| + \Delta)} \\
 & = \frac{\alpha^2 \|\tilde{\nabla} f(x_k)\|^2 - 2\alpha \langle \nabla f(x_k), \tilde{\nabla} f(x_k) \rangle}{2M_{k+1} (\|\tilde{\nabla} f(x_k)\| + \Delta)} \\
 & = \frac{\alpha^2 \|\tilde{\nabla} f(x_k)\|^2 - 2\alpha \langle \nabla f(x_k), \tilde{\nabla} f(x_k) \rangle + \|\nabla f(x_k)\|^2 - \|\nabla f(x_k)\|^2}{2M_{k+1} (\|\tilde{\nabla} f(x_k)\| + \Delta)} \\
 & = \frac{\|\alpha \tilde{\nabla} f(x_k) - \nabla f(x_k)\|^2 - \|\nabla f(x_k)\|^2}{2M_{k+1} (\|\tilde{\nabla} f(x_k)\| + \Delta)}.
 \end{aligned}$$

Therefore,

$$f(x_k) - f(x_{k+1}) \geq \frac{\|\nabla f(x_k)\|^2 - \|\alpha \tilde{\nabla} f(x_k) - \nabla f(x_k)\|^2}{2M_{k+1} (\|\tilde{\nabla} f(x_k)\| + \Delta)}.$$

We take advantage of the following relation for $\alpha \in (0, 1]$ and $a, b \in \mathbb{R}^n$:

$$\|\alpha a - b\| = \|\alpha(a - b) + (\alpha - 1)b\| \leq \alpha \|a - b\| + (1 - \alpha) \|b\|.$$

As a result,

$$\begin{aligned}
 & f(x_k) - f(x_{k+1}) \\
 & \geq \frac{\|\nabla f(x_k)\|^2 - (\alpha \|\tilde{\nabla} f(x_k) - \nabla f(x_k)\| + (1 - \alpha) \|\nabla f(x_k)\|)^2}{2M_{k+1} (\|\tilde{\nabla} f(x_k)\| + \Delta)} \\
 & \geq \frac{\|\nabla f(x_k)\|^2 - (\alpha \Delta + (1 - \alpha) \|\nabla f(x_k)\|)^2}{2M_{k+1} (\|\tilde{\nabla} f(x_k)\| + \Delta)} \\
 & = \frac{\alpha (\|\nabla f(x_k)\| - \Delta) ((2 - \alpha) (\|\nabla f(x_k)\| - \Delta) + 2\Delta)}{2M_{k+1} (\|\tilde{\nabla} f(x_k)\| + \Delta)} \\
 & > \frac{\alpha(2 - \alpha)}{2} \frac{(\|\nabla f(x_k)\| - \Delta)^2}{M_{k+1} (\|\tilde{\nabla} f(x_k)\| + \Delta)} \\
 & \geq \frac{\alpha(2 - \alpha)}{2} \frac{(\|\tilde{\nabla} f(x_k)\| - 2\Delta)^2}{M_{k+1} (\|\tilde{\nabla} f(x_k)\| + \Delta)}.
 \end{aligned}$$

Let $\{M_i\}_{i=0}^k$ be a sequence of positive numbers satisfying the assumptions of the previous theorem. Given $\|\tilde{\nabla}f(x_k)\| \geq 5\Delta$, from (24) it follows that

$$\begin{aligned} f(x_k) - f(x_{k+1}) &\geq \frac{\alpha(2-\alpha)(\|\tilde{\nabla}f(x_k)\| - 2\Delta)^2}{2(M_{k+1}(\|\tilde{\nabla}f(x_k)\| - 2\Delta) + 3M_{k+1}\Delta)} \\ &\geq \frac{\alpha(2-\alpha)}{4M_{k+1}} (\|\tilde{\nabla}f(x_k)\| - 2\Delta). \end{aligned} \quad (25)$$

Lemma 5. Let $f(x)$ be a $(0, M)$ -smooth function, and let $\{M_i\}_{i=0}^k$ be some sequence such that $M_{i+1} \geq M$. Then for the algorithm

$$x_{k+1} = x_k - \frac{\alpha}{M_{k+1}(\|\tilde{\nabla}f(x_k)\| + \Delta)} \tilde{\nabla}f(x_k), \quad \alpha \in (0, 1],$$

the following inequality holds:

$$\frac{\|\tilde{\nabla}f(x_k)\|^2}{M_{k+1}(\|\tilde{\nabla}f(x_k)\| + \Delta)} \leq \frac{2}{\alpha(2-\alpha)}(f(x_k) - f(x_{k+1})) + \frac{2\Delta}{M_{k+1}(2-\alpha)}. \quad (26)$$

Proof. For $L_0 = 0$ and $L_1 = M$, inequality (12) gives

$$f(y) \leq f(x) + \langle \tilde{\nabla}f(x), y - x \rangle + \frac{M(\|\tilde{\nabla}f(x)\| + \Delta)}{2} \|y - x\|^2 + \Delta \|y - x\|.$$

With $y = x_{k+1}$ and $x = x_k$, we have

$$\begin{aligned} f(x_{k+1}) &\leq f(x_k) + \langle \tilde{\nabla}f(x_k), x_{k+1} - x_k \rangle \\ &\quad + \frac{M(\|\tilde{\nabla}f(x_k)\| + \Delta)}{2} \|x_{k+1} - x_k\|^2 + \Delta \|x_{k+1} - x_k\| \\ &\leq f(x_k) - \frac{\alpha\|\tilde{\nabla}f(x_k)\|^2}{M_{k+1}(\|\tilde{\nabla}f(x_k)\| + \Delta)} + \frac{\alpha^2\|\tilde{\nabla}f(x_k)\|^2}{2M_{k+1}(\|\tilde{\nabla}f(x_k)\| + \Delta)} \\ &\quad + \frac{\Delta\alpha\|\tilde{\nabla}f(x_k)\|}{M_{k+1}(\|\tilde{\nabla}f(x_k)\| + \Delta)} \\ &= f(x_k) - \alpha \left(1 - \frac{\alpha}{2}\right) \frac{\|\tilde{\nabla}f(x_k)\|^2}{M_{k+1}(\|\tilde{\nabla}f(x_k)\| + \Delta)} + \frac{\Delta\alpha\|\tilde{\nabla}f(x_k)\|}{M_{k+1}(\|\tilde{\nabla}f(x_k)\| + \Delta)}. \end{aligned}$$

Let us estimate the term containing Δ :

$$\frac{\Delta\alpha\|\tilde{\nabla}f(x_k)\|}{M_{k+1}(\|\tilde{\nabla}f(x_k)\| + \Delta)} \leq \frac{\frac{\Delta\alpha}{M_{k+1}} M_{k+1}(\|\tilde{\nabla}f(x_k)\| + \Delta)}{M_{k+1}(\|\tilde{\nabla}f(x_k)\| + \Delta)} = \frac{\Delta\alpha}{M_{k+1}}.$$

Thus,

$$f(x_k) - f(x_{k+1}) \geq \alpha \left(1 - \frac{\alpha}{2}\right) \frac{\|\tilde{\nabla}f(x_k)\|^2}{M_{k+1}(\|\tilde{\nabla}f(x_k)\| + \Delta)} - \frac{\Delta\alpha}{M_{k+1}},$$

which can be equivalently written as

$$\frac{\|\tilde{\nabla}f(x_k)\|^2}{M_{k+1}(\|\tilde{\nabla}f(x_k)\| + \Delta)} \leq \frac{2}{\alpha(2-\alpha)}(f(x_k) - f(x_{k+1})) + \frac{2\Delta}{M_{k+1}(2-\alpha)}.$$

Inequalities (25) and (26) mean that for $M_{k+1} \geq M$, the iteration exit criterion of Algorithm 1 (see line 7) is automatically satisfied.

Lemma 6. *Let $f(x)$ be a ρ -quasar-convex function with $\rho \in (\frac{1}{2}, 1]$, and let $\{M_i\}_{i=0}^k$ be some sequence of positive numbers from Algorithm 1. If $\alpha \leq \frac{2\rho-1}{\rho}$, then*

$$\|x_{k+1} - x_*\| \leq \|x_k - x_*\| + \Delta\eta_k + \frac{1}{M_{k+1}}. \quad (27)$$

Proof. Since $x_{k+1} = x_k - \eta_k \tilde{\nabla} f(x_k)$, the following equalities are valid:

$$\begin{aligned} \|x_{k+1} - x_*\|^2 &= \|x_k - x_* - \eta_k \tilde{\nabla} f(x_k)\|^2 \\ &= \|x_k - x_*\|^2 - 2\eta_k \langle \tilde{\nabla} f(x_k), x_k - x_* \rangle + \eta_k^2 \|\tilde{\nabla} f(x_k)\|^2. \end{aligned}$$

By the quasar-convexity of the function f , we obtain

$$\begin{aligned} &\|x_{k+1} - x_*\|^2 \\ &\leq \|x_k - x_*\|^2 - 2\eta_k(\rho(f(x_k) - f^*) - \Delta\|x_k - x_*\|) + \eta_k^2 \|\tilde{\nabla} f(x_k)\|^2 \\ &\stackrel{(26)}{\leq} \|x_k - x_*\|^2 - 2\eta_k(\rho(f(x_k) - f^*) - \Delta\|x_k - x_*\|) \\ &\quad + \alpha\eta_k \left(\frac{2}{\alpha(2-\alpha)}(f(x_k) - f^*) + \frac{2\Delta}{M_{k+1}(2-\alpha)} \right) \\ &= \|x_k - x_*\|^2 + \eta_k \left(\frac{2}{2-\alpha} - 2\rho \right) (f(x_k) - f^*) \\ &\quad + 2\Delta\eta_k \left(\|x_k - x_*\| + \frac{\alpha}{M_{k+1}(2-\alpha)} \right). \end{aligned}$$

Let the parameter α be chosen so that $\frac{2}{2-\alpha} - 2\rho \leq 0$. As $\alpha \in (0, 1]$, this is possible if and only if $\rho \in (\frac{1}{2}, 1]$. In this case, we have

$$\begin{aligned} \|x_{k+1} - x_*\|^2 &\leq \|x_k - x_*\|^2 + 2\Delta\eta_k\|x_k - x_*\| + \Delta^2\eta_k^2 + \frac{2\alpha\Delta\eta_k}{M_{k+1}(2-\alpha)} - \Delta^2\eta_k^2 \\ &= (\|x_k - x_*\| + \Delta\eta_k)^2 + \Delta\eta_k \left(\frac{2\alpha}{M_{k+1}(2-\alpha)} - \Delta\eta_k \right). \end{aligned}$$

Consider the function $y(t) = -t^2 + \frac{2\alpha}{M_{k+1}(2-\alpha)}t$. At the point $t = \frac{\alpha}{M_{k+1}(2-\alpha)}$ it reaches the maximum value $\frac{\alpha^2}{M_{k+1}^2(2-\alpha)^2}$, and $\frac{\alpha^2}{M_{k+1}^2(2-\alpha)^2} \leq \frac{1}{M_{k+1}^2}$ for $\alpha \leq 1$. Thus,

$$\|x_{k+1} - x_*\|^2 \leq (\|x_k - x_*\| + \Delta\eta_k)^2 + \frac{1}{M_{k+1}^2}.$$

Note that for $a, b, c > 0$, the inequality $a^2 \leq b^2 + c^2$ implies $a \leq (b+c)$. Hence, $a \leq b+c$, and, consequently,

$$\|x_{k+1} - x_*\| \leq \|x_k - x_*\| + \Delta\eta_k + \frac{1}{M_{k+1}}.$$

Note that inequalities (25) and (26) surely hold under $M_{k+1} \geq M$. If $\|\tilde{\nabla} f(x_k)\| \geq 5\Delta$ and $M_i \geq \frac{M_0}{2} \forall i = \overline{1, k+1}$ with some $M_0 > 0$, then

$$\eta_k = \frac{\alpha}{M_{k+1}(\|\tilde{\nabla} f(x_k)\| + \Delta)} \leq \frac{\alpha}{6M_{k+1}\Delta} \leq \frac{\alpha}{3M_0\Delta}.$$

Theorem 2. Let $f(x)$ be a ρ -quasar-convex and $(0, M)$ -smooth function, where $\rho \in (\frac{1}{2}, 1]$, $\alpha \in (0, 1]$, and $\alpha \leq \frac{2\rho-1}{\rho}$. Then after $\left\lceil \frac{8M(f(x_0)-f^*)}{3\alpha(2-\alpha)\Delta} \right\rceil$ iterations of Algorithm 1, there exists an iteration k for which either $\|\tilde{\nabla}f(x_k)\| < 5\Delta$, or the exact solution will be reached.

Proof. In view of $\|\tilde{\nabla}f(x_k)\| \geq 5\Delta$, summing over k from 0 to $N-1$ in inequality (25) yields the relations

$$f(x_0) - f^* \geq f(x_0) - f(x_N) \geq \frac{3\alpha(2-\alpha)\Delta}{4} \sum_{k=0}^{N-1} \frac{1}{M_{k+1}},$$

which can be written as

$$\sum_{k=0}^{N-1} \frac{1}{M_{k+1}} \leq \frac{4(f(x_0) - f^*)}{3\alpha(2-\alpha)\Delta}.$$

Considering that $M_i \leq 2M$, we estimate the number of iterations for which $\|\tilde{\nabla}f(x_k)\| \geq 5\Delta$. We have $\frac{N}{2M} \leq \frac{4(f(x_0)-f^*)}{3\alpha(2-\alpha)\Delta}$, or, in the final form,

$$N \leq \left\lceil \frac{8M(f(x_0) - f^*)}{3\alpha(2-\alpha)\Delta} \right\rceil.$$

Remark 6. Suppose that an acceptable solution quality is achieved under the condition $\|\tilde{\nabla}f(x_k)\| < 5\Delta$. Indeed, the ρ -quasar-convexity of the function f and inequality (27) imply

$$f(x_k) - f^* \leq \frac{1}{\rho} (\|\tilde{\nabla}f(x_k)\| + \Delta) \|x_k - x_*\| \leq \frac{6\Delta}{\rho} \left(\|x_0 - x_*\| + \frac{(\alpha+6)N}{3M_0} \right).$$

Hence, with a small number of iterations, one can guarantee a solution quality within $O(\Delta)$. In the case of large values of N , this cannot be stated for sure. Note that if the objective function satisfies the PL condition, an acceptable solution quality will be achieved under a small norm of the inexact gradient (see Remark 4).

In the special case $\Delta = 0$, we have the following result.

Theorem 3. Let $f(x)$ be a ρ -quasar-convex and $(0, M)$ -smooth function, where $\rho \in (\frac{1}{2}, 1]$, $\alpha \in (0, 1]$, and $\alpha \leq \frac{2\rho-1}{\rho}$. Then Algorithm 1 with $\Delta = 0$ converges at the rate of a geometric progression.

Proof. Consider the case $\Delta = 0$, i.e., $\tilde{\nabla}f(x) = \nabla f(x)$ for each x . Then from line 7 of Algorithm 1 it follows that

$$f(x_k) - f(x_{k+1}) \geq \frac{\alpha(2-\alpha)}{2M_{k+1}} \frac{\|\nabla f(x_k)\|^2}{\|\nabla f(x_k)\|} = \frac{(2-\alpha)\eta_k}{2} \|\nabla f(x_k)\|^2. \quad (28)$$

In this case, we also have the relations

$$\begin{aligned} \|x_{k+1} - x_*\|^2 &= \|x_k - x_*\|^2 - 2\eta_k \langle \nabla f(x_k), x_k - x_* \rangle + \eta_k^2 \|\nabla f(x_k)\|^2 \\ &\leq \|x_k - x_*\|^2 - 2\eta_k \rho (f(x_k) - f^*) + \eta_k^2 \|\nabla f(x_k)\|^2 \\ &\leq \|x_k - x_*\|^2 - 2\eta_k \rho (f(x_k) - f^*) + \frac{2\eta_k}{(2-\alpha)} (f(x_k) - f^*) \\ &= \|x_k - x_*\|^2 + \eta_k (f(x_k) - f^*) \left(\frac{2}{2-\alpha} - 2\rho \right) \\ &\leq \|x_k - x_*\|^2 \leq \|x_0 - x_*\|^2 = R^2, \end{aligned}$$

where $\alpha \in (0, 1]$ and $\rho \in \left(\frac{1}{2}, 1\right]$. Further, by the quasar-convexity of the function f ,

$$f(x_k) - f^* \leq \frac{1}{\rho} \|\nabla f(x_k)\| \|x_k - x_*\| \leq \frac{1}{\rho} \|\nabla f(x_k)\| \|x_0 - x_*\| = \frac{R}{\rho} \|\nabla f(x_k)\|.$$

Recall that, due to Lemmas 4 and 5, for $M_{k+1} \geq M$ the iteration exit criterion of Algorithm 1 (see line 7) is surely satisfied. Then for each $k \geq 0$ we have $M_{k+1} \leq 2M$, and from (28) it follows that

$$\begin{aligned} f(x_{k+1}) - f^* &\leq f(x_k) - f^* - \frac{\alpha(2-\alpha)}{2M_{k+1}} \|\nabla f(x_k)\| \\ &\leq f(x_k) - f^* - \frac{\alpha(2-\alpha)}{2M_{k+1}} \frac{\rho}{R} (f(x_k) - f^*) \\ &\leq \left(1 - \frac{\alpha(2-\alpha)\rho}{2M_{k+1}R}\right) (f(x_k) - f^*) \\ &\leq \left(1 - \frac{\alpha(2-\alpha)\rho}{4MR}\right) (f(x_k) - f^*) \\ &\leq \left(1 - \frac{\alpha(2-\alpha)\rho}{4MR}\right)^{k+1} (f(x_0) - f^*). \end{aligned}$$

The proof of Theorem 3 is complete.

5. COMPUTATIONAL EXPERIMENTS

This section presents the results of computational experiments conducted to assess the efficiency of the proposed algorithms when minimizing the logistic function (29) and the quasar-convex function (30).

5.1. Testing Results for Adaptive Algorithm 1

Let us compare Algorithm 1 with the adaptive universal gradient method [30] for the problem of minimizing the logistic function from Example 3:

$$f(x) = \frac{1}{m} \sum_{i=1}^m \log(1 + \exp(-a_i^\top x)), \quad a_i \in \mathbb{R}^n. \quad (29)$$

As is well known, the logistic function (29) is convex and L -smooth with the constant $L = \frac{1}{m} \sum_{i=1}^m \|a_i\|^2$, as well as (L_0, L_1) -smooth with the parameters $L_0 = 0$ and $L_1 = \max_{1 \leq i \leq m} \|a_i\|$.

The algorithms were run with the parameters $n = 1000$ and $m = 100$ and the initial point $x_0 = (1, \dots, 1) \in \mathbb{R}^n$. The gradient inexactness parameter Δ took values from the set $\{0.5; 0.1; 0.05; 0.01\}$. The vectors $\{a_i\}_{i=1}^m$ were generated randomly from the standard Gaussian distribution. Figure 1 shows the iteration-to-iteration dynamics of the objective function values for the algorithms compared. According to the numerical results, Algorithm 1, which uses (L_0, L_1) -smoothness, demonstrates higher efficiency compared to the adaptive universal gradient method [30].

5.2. A Nonconvex Problem Example from the Class of Quasar-Convex Functions

This subsection presents a comparison of the proposed method (9) with the step size (10) and the gradient method with a constant step size of 0.001 on the example of a quasar-convex function from [27]:

$$f(x) = h(\|x\|_2) g\left(\frac{x}{\|x\|_2}\right), \quad (30)$$

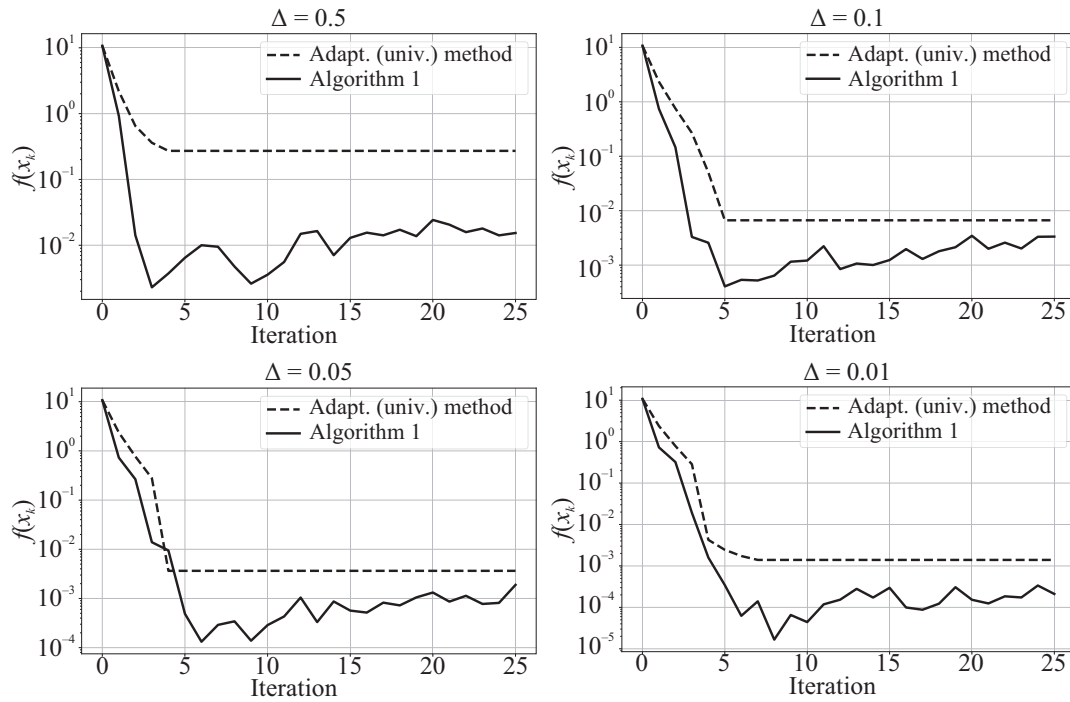


Fig. 1. Numerical solution of the logistic regression problem with (29), $n = 10^3$, and $m = 100$: Algorithm 1 vs. the adaptive universal gradient method [30].

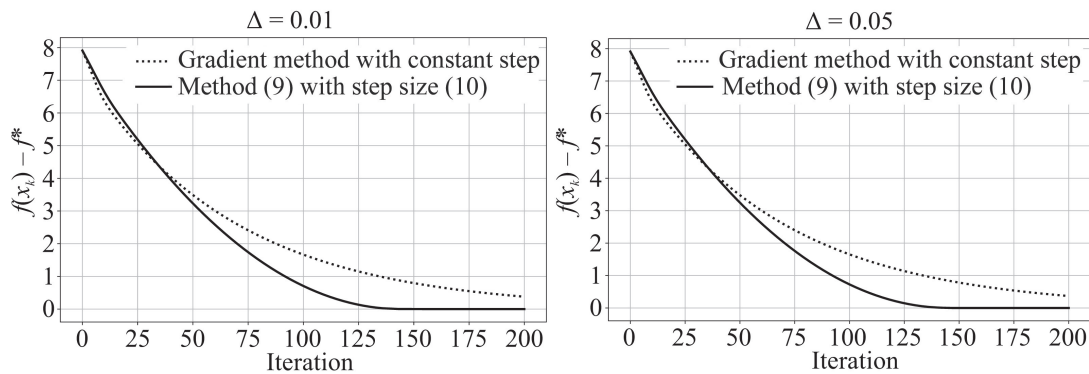


Fig. 2. The results of minimizing the objective function (30): algorithm (9) with the step size (10) vs. the gradient method with constant step size.

where $h(t) = t^2$ and

$$g(x_1, x_2) = 1 + \frac{1}{4N} \sum_{i=1}^N a_i \sin^2(b_i x_1) + c_i \cos^2(d_i x_2).$$

In the computational experiments, the elements of the sequences $\{a_i\}_i$ and $\{c_i\}_i$ were assigned independent uniform distributions on the interval $[0, 20)$. Similarly, the elements of the sequences $\{b_i\}_i$ and $\{d_i\}_i$ were independently and uniformly distributed on the interval $[-25, 25)$. All methods were initialized from the starting point $(1, 1)$ with the parameters $N = 10$ and $\alpha = 0.01$. Figure 2 shows the iteration-to-iteration dynamics of the objective function values for the algorithms compared. According to the experimental results, the proposed algorithm (9) with the step size (10) chosen based on the (L_0, L_1) -smoothness of the objective function has higher efficiency than the gradient method with a constant step size.

6. CONCLUSIONS

In this paper, we have proposed and analyzed several gradient methods for minimizing (L_0, L_1) -smooth objective functions under the additive inexactness in the gradient values at the iterations. Attention has been focused on investigating the convergence of these methods for some classes of nonconvex functions, such as ρ -quasar-convex ones and those satisfying the Polyak–Łojasiewicz (PL) condition.

The state-of-the-art research in this field has been reviewed. All key definitions and auxiliary results have been adapted to the case of a Δ -inexact gradient. Some lemmas characterizing the impact of the additive gradient error on the trajectory of the gradient method have been formulated and proven. The convergence rate of the gradient method on the class of quasar-convex functions satisfying the PL condition has been estimated theoretically. In particular, the method has been shown to converge at a nearly linear rate (up to some parameters related to the errors). It has also been proven that starting from some iteration, the method converges to a neighborhood of the exact solution on the class of functions with the PL condition. Separate consideration has been given to a detailed study of the case $L_0 = 0$ (the class of $(0, M)$ -smooth functions), which is of significant interest for machine learning problems, such as training logistic regression models. An adaptive modification of the gradient method using inexact gradient values has been developed, and a theoretical estimate of its convergence rate has been derived. In the case of exact gradient values (for $\Delta = 0$), the method has been shown to converge at the rate of a geometric progression. The efficiency of the proposed methods has been confirmed by the results of computational experiments conducted both on the logistic regression problem and on a certain nonconvex quasar-convex problem.

Concerning directions for further research, it seems interesting to study accelerated methods for (L_0, L_1) -smooth problems with inexact gradient information and develop methods with adaptive tuning to the parameters L_0 and L_1 .

FUNDING

This work was supported by the Ministry of Science and Higher Education of the Russian Federation within state assignment no. 075-03-2024-074 under the project “Study of Asymptotic Characteristics of Fluctuations of Differential Equations and Systems, and Optimization Methods.”

REFERENCES

1. Zhang, J., He, T., Sra, S., and Jadbabaie, A., Why Gradient Clipping Accelerates Training: A Theoretical Justification for Adaptivity, *Proceedings of the International Conference on Learning Representations*, 2020, pp. 1–14.
2. Schmidt, M., Le Roux, N., and Bach, F., Minimizing Finite Sums with the Stochastic Average Gradient, *Mathematical Programming*, 2017, vol. 162, pp. 83–112.
3. Johnson, R. and Zhang, T., Accelerating Stochastic Gradient Descent Using Predictive Variance Reduction, *Advances in Neural Information Processing Systems*, 2013, no. 26, pp. 315–323.
4. Defazio, A., Bach, F., and Lacoste-Julien, S., SAGA: A Fast Incremental Gradient Method with Support for Non-Strongly Convex Composite Objectives, *Advances in Neural Information Processing Systems*, 2014, no. 2, pp. 1646–1654.
5. Nguyen, L., Liu, J., Scheinberg, K., and Takač, M., Sarah: A Novel Method for Machine Learning Problems Using Stochastic Recursive Gradient, *Proceedings of the 34th International Conference on Machine Learning (ICML)*, 2014, vol. 6, pp. 4009–4023.
6. Nguyen, L., Scheinberg, K., and Takač, M., Inexact Sarah Algorithm for Stochastic Optimization, *Optimization Methods and Software*, 2021, vol. 36, no. 1, pp. 237–258.

7. Beznosikov, A. and Takač, M., Random-Reshuffled SARAH Does Not Need Full Gradient Computations, *Optim. Lett.*, 2024, vol. 18, pp. 727–749.
8. Shi, Z., Sadiev, A., Loizou, N., et al., Ai-Sarah: Adaptive and Implicit Stochastic Recursive Gradient Methods, *Transactions on Machine Learning Research*, 2023, pp. 1–40.
9. Defazio, A. and Bottou, L., On the Ineffectiveness of Variance Reduced Optimization for Deep Learning, *Advances in Neural Information Processing Systems*, 2019, vol. 32, pp. 1753–1763.
10. Zhang, B., Jin, J., Fang, C., and Wang, L., Improved Analysis of Clipping Algorithms for Non-Convex Optimization, *Advances in Neural Information Processing Systems*, 2020, vol. 19, pp. 15511–15522.
11. Chen, Z., Zhou, Y., Liang, Y., and Lu, Z., Generalized-Smooth Nonconvex Optimization Is as Efficient as Smooth Nonconvex Optimization, *Proceedings of the International Conference on Machine Learning (PMLR)*, 2023, pp. 5396–5427.
12. Zhao, S.-Y., Xie, Y.-P., and Li, W.-J., On the Convergence and Improvement of Stochastic Normalized Gradient Descent, *Science China Information Sciences*, 2021, vol. 64, pp. 1–13.
13. Faw, M., Rout, L., Caramanis, C., and Shakkottai, S., Beyond Uniform Smoothness: A Stopped Analysis of Adaptive SGD, *Proceedings of the 36th Annual Conference on Learning Theory (PMLR)*, 2023, pp. 89–160.
14. Wang, B., Zhang, H., Ma, Z., and Chen, W., Convergence of Adagrad for Non-Convex Objectives: Simple Proofs and Relaxed Assumptions, *Proceedings of the 36th Annual Conference on Learning Theory (PMLR)*, 2023, pp. 161–190.
15. Li, H., Rakhlin, A., and Jadbabaie, A., Convergence of Adam under Relaxed Assumptions, *Advances in Neural Information Processing Systems*, 2024, pp. 1792–1804.
16. Hubler, F., Yang, J., Li, X., and He, N., Parameter-Agnostic Optimization under Relaxed Smoothness, *Proceedings of the International Conference on Artificial Intelligence and Statistics (PMLR)*, 2024, pp. 4861–4869.
17. Pascanu, R., Mikolov, T., and Bengio, Y., On the Difficulty of Training Recurrent Neural Networks, *Proceedings of the International Conference on Machine Learning*, 2013, vol. 28, pp. 1310–1318.
18. Polyak, B., *Introduction to Optimization*, New York: Optimization Software, 1987.
19. Koloskova, A., Hendrikx, H., and Stich, S., Revisiting Gradient Clipping: Stochastic Bias and Tight Convergence Guarantees, *Proceedings of the International Conference on Machine Learning*, 2023, pp. 17343–17363.
20. Takezawa, Y., Bao, H., Sato, R., et al., Polyak Meets Parameter-Free Clipped Gradient Descent, *arXiv:2405.15010*, 2024.
21. Li, H., Qian, J., Tian, Y., Rakhlin, A., and Jadbabaie, A., Convex and Non-Convex Optimization under Generalized Smoothness, *Advances in Neural Information Processing Systems*, 2024, vol. 36, pp. 2675–2686.
22. Gorbunov, E., Tupitsa, N., Choudhury, S., et al., Methods for Convex (L_0, L_1) -Smooth Optimization: Clipping, Acceleration, and Adaptivity, *arXiv:2409.14989*, 2024.
23. Vankov, D., Rodomanov, A., Nedich, A., et al., Optimizing (L_0, L_1) -Smooth Functions by Gradient Methods, *arXiv:2410.10800*, 2024.
24. Lobanov, A., Gasnikov, A., Gorbunov, E., and Takač, M., Linear Convergence Rate in Convex Setup is Possible! Gradient Descent Method Variants under (L_0, L_1) -Smoothness, *arXiv:2412.17050*, 2024.
25. Stonyakin, F., Kuruzov, I., and Polyak, B., Stopping Rules for Gradient Methods for Non-Convex Problems with Additive Noise in Gradient, *Journal of Optimization Theory and Applications*, 2022, vol. 198, pp. 531–551.
26. Wang, J. and Wibisono, A., Continuized Acceleration for Quasar Convex Functions in Non-Convex Optimization, *arXiv.2302.07851*, 2023.

27. Hermant, J., Aujol, J.F., Dossal, C., and Rondepierre, A., Study of the Behaviour of Nesterov Accelerated Gradient in a Non Convex Setting: The Strongly Quasar Convex Case, *arXiv:2405.19809*, 2024.
28. Hinder, O., Sidford, A., and Sohoni, N., Near-Optimal Methods for Minimizing Star-Convex Functions and Beyond, *arXiv.1906.11985*, 2019.
29. Karimi, H., Nutini, J., and Schmidt, M., Linear Convergence of Gradient and Proximal-Gradient Methods under the Polyak–Łojasiewicz Condition, *Proceedings of the Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, 2016, pp. 795–811.
30. Stonyakin, F., Tyurin, A., Gasnikov, A., et al., Inexact Relative Smoothness and Strong Convexity for Optimization and Variational Inequalities by Inexact Model, *arXiv:2402.06319*, 2024.

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

Gradient-Free Stochastic Optimization for Additive Models

A. Akhavan^{*,a} and A.B. Tsybakov^{**,b}

^{*}University of Oxford, United Kingdom

^{**}CREST, ENSAE, IP Paris, France

e-mail: ^aarya.akhavan@stats.ox.ac.uk, ^balexandre.tsybakov@ensae.fr

Received March 3, 2025

Revised May 25, 2025

Accepted June 27, 2025

Abstract—We address the problem of zero-order optimization from noisy observations for an objective function satisfying the Polyak–Lojasiewicz or the strong convexity condition. Additionally, we assume that the objective function has an additive structure and satisfies a higher-order smoothness property, characterized by the Hölder family of functions. The additive model for Hölder classes of functions is well-studied in the literature on nonparametric function estimation, where it is shown that such a model benefits from a substantial improvement of the estimation accuracy compared to the Hölder model without additive structure. We study this established framework in the context of gradient-free optimization. We propose a randomized gradient estimator that, when plugged into a gradient descent algorithm, allows one to achieve minimax optimal optimization error of the order $dT^{-(\beta-1)/\beta}$, where d is the dimension of the problem, T is the number of queries and $\beta \geq 2$ is the Hölder degree of smoothness. We conclude that, in contrast to nonparametric estimation problems, no substantial gain of accuracy can be achieved when using additive models in gradient-free optimization.

Keywords: additive model, gradient-free optimization, minimax optimality, Polyak–Lojasiewicz condition

DOI: 10.31857/S0005117925090026

1. INTRODUCTION

Additive modeling is a popular approach to dimension reduction in nonparametric estimation problems [12, 28, 30]. It consists of considering that the unknown function $f : \mathbb{R}^d \rightarrow \mathbb{R}$ to be estimated from the data has the form $f(\mathbf{x}) = \sum_{j=1}^d f_j(x_j)$, where x_j 's are the coordinates of $\mathbf{x} \in \mathbb{R}^d$ and f_j 's are unknown functions of one variable. The main property proved in the literature on additive models in nonparametric regression can be summarized as follows. If each of the functions f_j is β -Hölder (see Definition 1 below) then the minimax rate of estimation of f , pointwise or in L_2 -norm, is of the order $n^{-\beta/(2\beta+1)}$, where n is the number of observations [28]. This is in contrast with the problem of estimating β -Hölder functions on $\mathbb{R}^d$ without any additive structure, since for such functions the minimax rate is known to be $n^{-\beta/(2\beta+d)}$ [13, 14, 26, 27]. Thus, there is a substantial improvement in the rate of estimation when passing from general to additive models in nonparametric regression setting.

In the present paper, we show that such a dimension reduction property fails to hold in the context of gradient-free optimization. We consider additive modeling in the problem of minimizing an unknown function $f : \mathbb{R}^d \rightarrow \mathbb{R}$ when only sequential evaluations of values of f are available, cor-

rupted with noise. We assume that f is either strongly convex or satisfies the Polyak–Łojasiewicz (PL) condition [18, 22] and admits an additive representation as described above, where the components f_j are β -Hölder.

The setting that we consider belongs to the family of gradient-free (or zero-order) stochastic optimization problems, for which a rich literature is now available, see [1, 2, 4–6, 8, 10, 15, 17, 19–21, 23, 25] and the references therein. These papers did not assume any additive structure of f . It was proved in [23] that the minimax optimal rate of the optimization error, when f is β -Hölder with $\beta \geq 2$ and satisfies the quadratic growth condition, is of the order $T^{-(\beta-1)/\beta}$ as function of the number of sequential queries T , to within an unspecified factor depending on the dimension d . Further developments were devoted to exploring the dependency of the minimax rate on d assuming that f is β -Hölder with $\beta \geq 2$ and is either strongly convex [1, 4, 15, 21, 25] or satisfies the PL condition [1, 9]. In the PL case, unconstrained minimization was studied while the strongly convex case was analyzed both in constrained and unconstrained settings. A considerable progress was achieved though the complete solution valid for all $\beta \geq 2$ is not yet available. There exists a minimax lower bound for the class of β -Hölder and strongly convex functions, which scales as $dT^{-(\beta-1)/\beta}$ (proved in [25] for $\beta = 2$ and Gaussian noise and in [2] for all $\beta \geq 2$ and more general noise; see also [1] for a yet more general lower bound). Moreover, the same rate is attained for $\beta = 2$ under general conditions on the noise (no independence or zero-mean assumption), see [2]. Thus, for $\beta = 2$ we know that the minimax rate is of the order $d\sqrt{T}$. For $\beta > 2$, the literature provides different dependencies of the upper bounds on d determined by the geometry of the β -Hölder condition. Thus, for the Hölder classes defined by pointwise Taylor approximation the best known upper bound is of the order $d^{2-1/\beta}T^{-(\beta-1)/\beta}$ when strongly convex functions are considered [21]. On the other hand, for the Hölder classes defined by tensor-type conditions one can achieve the rate $d^{2-2/\beta}T^{-(\beta-1)/\beta}$ both for strongly convex functions and for PL functions [1]. Finally, the recent paper [31], assuming again strong convexity, deals with the class of functions that admit a Lipschitz Hessian. This represents a type of Hölder condition for $\beta = 3$ and the paper [31] derives the upper rate $dT^{-2/3}$. We note that the lower bound of [2] with the rate $dT^{-(\beta-1)/\beta}$ is valid for all the above mentioned Hölder classes since this lower bound is obtained on additive functions that belong to all of them. Thus, the rate $dT^{-(\beta-1)/\beta}$ appears to be minimax optimal not only for $\beta = 2$ but also for $\beta = 3$ under a suitable definition of 3-Hölder class of strongly convex functions.

The main result of the present paper is to establish an upper bound with the rate $dT^{-(\beta-1)/\beta}$ for the optimization error in the noisy gradient-free setting over the class of β -Hölder functions satisfying the additive model and either the PL condition or the strong convexity condition. Together with the lower bound proved in [2], it implies that $dT^{-(\beta-1)/\beta}$ is the minimax optimal rate of the optimization error in this setting for all $\beta \geq 2$. This conclusion is quite surprising as it goes against the intuition acquired from the classical results on nonparametric estimation mentioned above. Indeed, it means that, at least for $\beta \in \{2, 3\}$, there is no improvement in the rate neither in T nor in d when passing from the general β -Hölder model to the additive β -Hölder model. If any, an improvement can be obtained only in a factor independent of T and d . Such a property may be explained by the fact that the optimization setting is easier than nonparametric function estimation in the sense that it aims at estimating a specific functional of the unknown f (namely, its minimizer) rather than f as a whole object. We also refer to another somewhat similar fact in the gradient-free stochastic optimization setting. Specifically, there is no dramatic difference between the complexity of minimizing a strongly convex function with Lipschitz gradient, which corresponds to the case $\beta = 2$ discussed above with the minimax rate $d\sqrt{T}$, and the complexity of minimizing a convex function with no additional properties, for which one can construct an algorithm converging with the rate between $d^{1.5}\sqrt{T}$ and $d^{1.75}\sqrt{T}$ (up to a logarithmic factor) as recently shown in [7].

2. PROBLEM SETUP

Let Θ be a closed convex subset of $\mathbb{R}^d$. We consider the problem of minimizing an unknown function $f : \mathbb{R}^d \rightarrow \mathbb{R}$ over the set Θ based on noisy evaluations of f at query points that can be chosen sequentially depending on the past observations. Specifically, we assume that at round $t \in \{1, \dots, T\}$ we can observe two noisy evaluations of f at points $\mathbf{z}_t, \mathbf{z}'_t \in \mathbb{R}^d$, i.e.,

$$y_t = f(\mathbf{z}_t) + \xi_t, \quad y'_t = f(\mathbf{z}'_t) + \xi'_t,$$

where ξ_t, ξ'_t are scalar noise variables and the query points $\mathbf{z}_t, \mathbf{z}'_t$ can be chosen depending on $\{\mathbf{z}_i, \mathbf{z}'_i, y_i, y'_i\}_{i=1}^{t-1}$ and on a suitable randomization.

Throughout the paper we assume that f follows the additive model

$$f(\mathbf{x}) = \sum_{j=1}^d f_j(x_j),$$

where x_j 's are the coordinates of $\mathbf{x} \in \mathbb{R}^d$ and f_j 's are unknown functions of one variable.

We assume that each of the functions $f_j : \mathbb{R} \rightarrow \mathbb{R}$, $j = 1, \dots, d$, belongs to the class of β -Hölder functions specified by the following definition, with $\beta \geq 2$.

Definition 1. For $\beta > 0$, $L > 0$, denote by $\mathcal{F}_\beta(L)$ the set of all functions $f : \mathbb{R} \rightarrow \mathbb{R}$ that are $\ell = \lfloor \beta \rfloor$ times differentiable and satisfy, for all $x, z \in \mathbb{R}$, the condition

$$\left| f(z) - \sum_{m=0}^{\ell} \frac{1}{m!} f^{(m)}(x)(z-x)^m \right| \leq L|z-x|^\beta, \quad (1)$$

where $f^{(m)}$ is the m th derivative of f and $\lfloor \beta \rfloor$ is the largest integer less than β . Elements of the class $\mathcal{F}_\beta(L)$ are referred to as **β -Hölder functions**.

If $\beta > 2$ the fact that $f_j \in \mathcal{F}_\beta(L)$ does not imply that $f_j \in \mathcal{F}_2(L)$, however we will need the latter condition as well. It will be convenient to use it in a slightly different form given by the next definition.

Definition 2. The function $f : \mathbb{R} \rightarrow \mathbb{R}$ is called $\bar{L}$ -smooth if it is differentiable on $\mathbb{R}$ and there exists $\bar{L} > 0$ such that, for every $x, x' \in \mathbb{R}$, it holds that

$$|f'(x) - f'(x')| \leq \bar{L}|x - x'|.$$

The class of all $\bar{L}$ -smooth functions will be denoted by $\mathcal{F}'_2(\bar{L})$.

We also assume that f is either an α -strongly convex function or an α -PL function as stated in the next two definitions.

Definition 3. Let $\alpha > 0$. The function $f : \mathbb{R}^d \rightarrow \mathbb{R}$ is called an α -Polyak–Łojasiewicz function (shortly, α -PL function) if f is differentiable on $\mathbb{R}^d$ and

$$2\alpha \left(f(\mathbf{x}) - \min_{\mathbf{z} \in \mathbb{R}^d} f(\mathbf{z}) \right) \leq \|\nabla f(\mathbf{x})\|^2 \quad \text{for all } \mathbf{x} \in \mathbb{R}^d,$$

where $\|\cdot\|$ denotes the Euclidean norm.

Functions satisfying the PL condition are not necessarily convex. The PL condition is a useful tool in optimization problems since it leads to linear convergence of the gradient descent algorithm without convexity as shown by Polyak [22]. For more details and discussion on the PL condition see [16].

Algorithm 1: Zero-Order Stochastic Projected Gradient

Input: Constraint set Θ , kernel function $K : [-1, 1] \rightarrow \mathbb{R}$, step size $\eta_t > 0$ and perturbation parameter $h_t > 0$, for $t = 1, \dots, T$

Initialization: Generate vectors $\mathbf{r}_t = (r_{t,1}, \dots, r_{t,d}) \in \mathbb{R}^d$ for $t = 1, \dots, T$, where the components $r_{t,i}$ are independent and distributed uniformly from the interval $[-1, 1]$, and choose $\mathbf{x}_1 \in \Theta$

for $t = 1, \dots, T$ **do**

Observe $y_t = f(\mathbf{x}_t + h_t \mathbf{r}_t) + \xi_t$ and $y' = f(\mathbf{x}_t - h_t \mathbf{r}_t) + \xi'_t$; // query

for $j = 1, \dots, d$ **do**

$g_{t,j} = \frac{d}{2h_t}(y_t - y'_t)K(r_{t,j})$

Let $\mathbf{g}_t = (g_{t,1}, \dots, g_{t,d})$; // gradient estimator

$\mathbf{x}_{t+1} = \text{Proj}_\Theta(\mathbf{x}_t - \eta_t \mathbf{g}_t)$; // update

Definition 4. Let $\alpha > 0$. The function $f : \mathbb{R}^d \rightarrow \mathbb{R}$ is called α -strongly convex if f is differentiable on $\mathbb{R}^d$ and

$$f(\mathbf{x}) - f(\mathbf{x}') \leq \langle \nabla f(\mathbf{x}), \mathbf{x} - \mathbf{x}' \rangle - \frac{\alpha}{2} \|\mathbf{x} - \mathbf{x}'\|^2 \quad \text{for all } \mathbf{x}, \mathbf{x}' \in \mathbb{R}^d.$$

In order to minimize f , we apply a version of projected gradient descent with estimated gradient presented in Algorithm 1. Let $\{\eta_t\}_{t=1}^T$ be a sequence of positive numbers and let $\{\mathbf{g}_t\}_{t=1}^T$ be a sequence of random vectors. Consider any fixed $\mathbf{x}_1 \in \mathbb{R}^d$ and let the vectors $\mathbf{x}_t$ for $t = 2, \dots, T$ be defined by the recursion

$$\mathbf{x}_{t+1} = \text{Proj}_\Theta(\mathbf{x}_t - \eta_t \mathbf{g}_t), \quad (2)$$

where $\text{Proj}_\Theta(\cdot)$ is the Euclidean projection over Θ .

In this paper, the gradient estimator $\mathbf{g}_t = (g_{t,1}, \dots, g_{t,d}) \in \mathbb{R}^d$ at round $t \in \{1, \dots, T\}$ of the algorithm is defined as follows. For given $\beta \geq 2$ and $\ell = \lfloor \beta \rfloor$, let $K : [-1, 1] \rightarrow \mathbb{R}$ be a function such that

$$\int u K(u) du = 1, \quad \int u^j K(u) du = 0, \quad j = 0, 2, 3, \dots, \ell, \quad \text{and} \quad \kappa_\beta \equiv \int |u|^\beta |K(u)| du < \infty. \quad (3)$$

Assume that $\kappa := \int K^2(r) dr$ is finite. Functions K satisfying these conditions are not hard to construct. In particular, a construction based on Legendre polynomials can be used, see, for example, [4, 23, 29].

At each round t of the algorithm, we generate a random vector $\mathbf{r}_t = (r_{t,1}, \dots, r_{t,d}) \in \mathbb{R}^d$, where the components $r_{t,j}$ are independent and distributed uniformly on $[-1, 1]$. For $h_t > 0$, we draw two noisy evaluations

$$y_t = f(\mathbf{x}_t + h_t \mathbf{r}_t) + \xi_t, \quad y'_t = f(\mathbf{x}_t - h_t \mathbf{r}_t) + \xi'_t,$$

and, for $j \in \{1, \dots, d\}$, we define

$$g_{t,j} = \frac{1}{2h_t}(y_t - y'_t)K(r_{t,j}). \quad (4)$$

We consider the gradient estimator $\mathbf{g}_t = (g_{t,1}, \dots, g_{t,d})$. Note that other choices of gradient estimator can lead to similar results as those that we obtain below, namely estimators based on finite difference approximations taking into account higher order smoothness. In contrast to (4), such higher order finite difference schemes have a complicated form and require many queries per step of the algorithm.

We assume that the noise variables ξ_t, ξ'_t and the randomizing variables $r_{t,j}$ satisfy the following.

Assumption 1. There exists $\sigma^2 > 0$ such that for all $t \in \{1, \dots, T\}$ the following holds.

- (i) The random variables $r_{t,j} \sim U[-1, 1]$, $j = 1, \dots, d$, are independent of $\mathbf{x}_t$ and conditionally independent of ξ_t, ξ'_t given $\mathbf{x}_t$,
- (ii) $\mathbb{E}[\xi_t^2] \leq \sigma^2$, $\mathbb{E}[(\xi'_t)^2] \leq \sigma^2$.

Assumption 1 (i) can be considered not as a restriction but as a part of the definition of the algorithm dealing with the choice of the randomizing variables $r_{t,j}$. Randomizations are naturally chosen independent of all other sources of randomness. For the proofs we need even a weaker property that we state here as an assumption in order to refer to it in what follows. Note also that Assumption 1 does not require the noises ξ_t, ξ'_t to have zero mean. Moreover, they can be non-random and we do not assume independence between these noises on different rounds of the algorithm. The fact that convergence of gradient-free algorithms can be achieved under general conditions on the noise of similar type, not requiring independence and zero means, dates back to [11, 24].

3. STATEMENT OF THE RESULTS

In this section, we provide upper bounds on the optimization error of the algorithm defined in Section 2. First, we assume that function f represented by the additive model satisfies the α -PL condition. Note that imposing this condition on the sum f implies that all the components f_j are PL functions. When dealing with PL functions we consider the problem of unconstrained minimization and we introduce the notation

$$f^* = \min_{\mathbf{x} \in \mathbb{R}^d} f(\mathbf{x}).$$

In what follows, for any positive integer n we denote by $[n]$ the set of positive integers less than or equal to n .

Theorem 1. Let $\alpha > 0$, $\beta \geq 2$ and $\bar{L}, L > 0$. For $j \in [d]$, let $f_j : \mathbb{R} \rightarrow \mathbb{R}$ be such that $f_j \in \mathcal{F}'_2(\bar{L}) \cap \mathcal{F}_\beta(L)$. Assume that $f : \mathbb{R}^d \rightarrow \mathbb{R}$ has the form $f(\mathbf{x}) = \sum_{j=1}^d f_j(x_j)$, and that f is an α -PL function. Let Assumption 1 hold, and let $\{\mathbf{x}_t\}_{t=1}^T$ be the updates of Algorithm 1 with $\Theta = \mathbb{R}^d$, and

$$\eta_t = \min\left(\frac{4}{\alpha t}, \frac{1}{18\bar{L}d\kappa}\right),$$

$$h_t = \left(\frac{3\bar{L}}{\alpha} \cdot \frac{\kappa\sigma^2}{L^2\kappa_\beta^2}\right)^{\frac{1}{2\beta}} \cdot \begin{cases} t^{-\frac{1}{2\beta}} & \text{if } \eta_t = \frac{4}{\alpha t}, \\ T^{-\frac{1}{2\beta}} & \text{if } \eta_t = \frac{1}{18\bar{L}d\kappa}. \end{cases}$$

Then,

$$\mathbf{E}[f(\mathbf{x}_T) - f^*] \leq \frac{144\bar{L}d\kappa}{\alpha T}(f(\mathbf{x}_1) - f^*) + \frac{d}{\alpha} \left(A_1 \left(\frac{\bar{L}}{\alpha T} \right)^{\frac{\beta-1}{\beta}} + A_2 \frac{\bar{L}^3 d}{T} \left(\frac{\bar{L}}{\alpha T L^2} \right)^{\frac{1}{\beta}} \right),$$

where $A_1, A_2 > 0$ depend only on β and σ^2 .

Corollary 1. Under the conditions of Theorem 1, if $T \geq (\bar{L}/\alpha)((\bar{L}^2 d \alpha)^{\frac{\beta}{2}}/L)$, then

$$\mathbf{E}[f(\mathbf{x}_T) - f^*] \leq \frac{144\bar{L}d\kappa}{\alpha T}(f(\mathbf{x}_1) - f^*) + A \frac{d}{\alpha} \left(\frac{\bar{L}}{\alpha T} \right)^{\frac{\beta-1}{\beta}},$$

where $A > 0$ depends only on β and σ^2 .

The bounds of Theorem 1 and Corollary 1 show how the rate of convergence depends on the parameters T, d and α but also on the aspect ratio $\bar{L}/\alpha$. Moreover, in Corollary 1 the condition $T \geq Cd^{\beta/2}$, where $C > 0$ is a constant, does not bring an additional restriction when $2 \leq \beta \leq 3$ since the condition is weaker than giving the range of T , for which the bound makes sense. Indeed, for $2 \leq \beta \leq 3$ the bound is greater than a constant independent of T, d if $T \leq Cd^{\beta/2}$.

Next, we study the optimization error of the algorithm defined in Section 2 under the assumption that the objective function is α -strongly convex. Unlike the case of PL functions, we consider here the problem of constrained minimization

$$\min_{\mathbf{x} \in \Theta} f(\mathbf{x}),$$

where Θ is a compact convex subset of $\mathbb{R}^d$.

Theorem 2. *Let $\alpha > 0$, $\beta \geq 2$ and $\bar{L}, L > 0$. For $j \in [d]$, let $f_j : \mathbb{R} \rightarrow \mathbb{R}$ be such that $f_j \in \mathcal{F}'_2(\bar{L}) \cap \mathcal{F}_\beta(L)$. Assume that $f : \mathbb{R}^d \rightarrow \mathbb{R}$ has the form $f(\mathbf{x}) = \sum_{j=1}^d f_j(x_j)$, and that f is an α -strongly convex function on a compact convex subset Θ of $\mathbb{R}^d$ such that $\max_{\mathbf{x} \in \Theta} \|\nabla f(\mathbf{x})\| \leq G$. Let Assumption 1 hold, and let $\{\mathbf{x}_t\}_{t=1}^T$ be the updates of Algorithm 1 with*

$$\eta_t = \frac{4}{\alpha(T+1)}, \quad h_t = \left(\frac{3}{2t} \frac{\kappa \sigma^2}{\kappa_\beta^2 L^2} \right)^{\frac{1}{2\beta}}.$$

Consider the weighted estimator

$$\bar{\mathbf{x}}_T = \frac{2}{T(T+1)} \sum_{t=1}^T t \mathbf{x}_t.$$

Then, for any $\mathbf{x} \in \Theta$ we have

$$\mathbf{E}[f(\bar{\mathbf{x}}_T) - f(\mathbf{x})] \leq \frac{18G^2d}{\alpha T} + \frac{d}{\alpha} \left(A_1 L^{\frac{2}{\beta}} + A_2 d \bar{L}^2 (LT)^{-\frac{2}{\beta}} \right) T^{-\frac{\beta-1}{\beta}},$$

where $A_1, A_2 > 0$ depend only on β and σ^2 .

Corollary 2. *Under the conditions of Theorem 2, if $T \geq (d\bar{L})^{\beta/2} L^{-2}$ then*

$$\mathbf{E}[f(\bar{\mathbf{x}}_T) - \min_{\mathbf{x} \in \Theta} f(\mathbf{x})] \leq A \frac{d}{\alpha} T^{-\frac{\beta-1}{\beta}},$$

where $A > 0$ does not depend on T, d and α .

Similarly to Corollary 1, we may note that for $2 \leq \beta \leq 3$ the condition $T \geq Cd^{\beta/2}$ in Corollary 2, where $C > 0$ is a constant, does not bring an additional restriction but indicates the meaningful range of T since the bound is greater than a constant when $T \leq Cd^{\beta/2}$.

Remark 1. In view of strong convexity, Theorem 2 and Corollary 2 immediately imply the corresponding bounds on the estimation error $\mathbf{E}[\|\bar{\mathbf{x}}_T - \mathbf{x}^*\|^2]$, where $\mathbf{x}^*$ is the minimizer of f on Θ . Thus, under the assumptions of Corollary 2 we have

$$\mathbf{E}[\|\bar{\mathbf{x}}_T - \mathbf{x}^*\|^2] \leq 2A \frac{d}{\alpha^2} T^{-\frac{\beta-1}{\beta}},$$

where $A > 0$ is the constant from Corollary 2.

It follows from Corollary 2 and the proofs of the lower bounds in [1, 2] that, under non-restrictive conditions on the parameters of the problem, the rate $\frac{d}{\alpha} T^{-\frac{\beta-1}{\beta}}$ in Corollary 2 is minimax optimal on the class of additive functions f satisfying the assumptions of Theorem 2. The lower bound that we need is not explicitly stated in [1, 2] but follows immediately from the proofs in those papers since the lower bounds in [1, 2] are obtained on additive functions. For completeness, we provide here the statement of the lower bound for additive functions based on [1].

Consider all strategies of choosing the query points as $\mathbf{z}_t = \Phi_t((\mathbf{z}_i, y_i)_{i=1}^{t-1}, (\mathbf{z}'_i, y'_i)_{i=1}^{t-1}, \tau_t)$ and $\mathbf{z}'_t = \Phi'_t((\mathbf{z}_i, y_i)_{i=1}^{t-1}, (\mathbf{z}'_i, y'_i)_{i=1}^{t-1}, \tau_t)$ for $t \geq 2$, where Φ_t 's and Φ'_t 's are measurable functions, $\mathbf{z}_1, \mathbf{z}'_1 \in \mathbb{R}^d$ are any random variables, and $\{\tau_t\}$ is a sequence of random variables with values in a measurable space $(\mathcal{Z}, \mathcal{U})$, such that τ_t is independent of $((\mathbf{z}_i, y_i)_{i=1}^{t-1}, (\mathbf{z}'_i, y'_i)_{i=1}^{t-1})$. We denote by Π_T the set of all such strategies of choosing query points up to $t = T$. The class Π_T includes the sequential strategy of the algorithm of Section 2 with the gradient estimator (4). In this case, $\tau_t = \mathbf{r}_t$, $\mathbf{z}_t = \mathbf{x}_t + h_t \mathbf{r}_t$ and $\mathbf{z}'_t = \mathbf{x}_t - h_t \mathbf{r}_t$.

The lower bound of [1] that we are using here is proved under the following assumption on the noises (ξ, ξ'_t) . Let $H^2(\cdot, \cdot)$ be the squared Hellinger distance defined, for two probability measures $\mathbf{P}, \mathbf{P}'$ on a measurable space $(\Omega, \mathcal{A})$, as

$$H^2(\mathbf{P}, \mathbf{P}') \triangleq \int (\sqrt{d\mathbf{P}} - \sqrt{d\mathbf{P}'})^2.$$

Assumption 2. For every $t \geq 1$, the following holds:

- The cumulative distribution function $F_t : \mathbb{R}^2 \rightarrow \mathbb{R}$ of random variable (ξ_t, ξ'_t) is such that

$$H^2(P_{F_t(\cdot, \cdot)}, P_{F_t(\cdot + v, \cdot + v)}) \leq I_0 v^2, \quad |v| \leq v_0, \quad (5)$$

for some $0 < I_0 < \infty$, $0 < v_0 \leq \infty$. Here, $P_{F(\cdot, \cdot)}$ denotes the probability measure corresponding to the c.d.f. $F(\cdot, \cdot)$.

- The random variable (ξ_t, ξ'_t) is independent of $((\mathbf{z}_i, y_i)_{i=1}^{t-1}, (\mathbf{z}'_i, y'_i)_{i=1}^{t-1}, \tau_t)$.

Let $\Theta = \{\mathbf{x} \in \mathbb{R}^d : \|\mathbf{x}\| \leq 1\}$. Fix $\alpha, L, \bar{L} > 0, G > \alpha$, $\beta \geq 2$, and denote by $\mathcal{F}$ the set of all functions f that satisfy the assumptions of Theorem 2 and attain their minimum over $\mathbb{R}^d$ in Θ .

Theorem 3. Let $\Theta = \{\mathbf{x} \in \mathbb{R}^d : \|\mathbf{x}\| \leq 1\}$ and let Assumption 2 hold. Assume that $\alpha > T^{-1/2+1/\beta}$ and $T \geq d^\beta$. Then, for any estimator $\tilde{\mathbf{x}}_T$ based on the observations $((\mathbf{z}_t, y_t), (\mathbf{z}'_t, y'_t), t = 1, \dots, T)$, where $((\mathbf{z}_t, \mathbf{z}'_t), t = 1, \dots, T)$ are obtained by any strategy in the class Π_T we have

$$\sup_{f \in \mathcal{F}} \mathbf{E}[f(\tilde{\mathbf{x}}_T) - \min_{\mathbf{x} \in \Theta} f(\mathbf{x})] \geq C \frac{d}{\alpha} T^{-\frac{\beta-1}{\beta}}, \quad (6)$$

where $C > 0$ is a constant that does not depend of T, d , and α .

Theorem 3 follows immediately from the proof of Theorem 22 in [1] since the family of functions used there belongs to the class $\mathcal{F}$. The lower bound of Theorem 22 in [1] has the form

$$C \min \left(\max(\alpha, T^{-1/2+1/\beta}), \frac{d}{\sqrt{T}}, \frac{d}{\alpha} T^{-\frac{\beta-1}{\beta}} \right),$$

which reduces to $C \frac{d}{\alpha} T^{-\frac{\beta-1}{\beta}}$ under the assumptions on T, d and α used in Theorem 3.

Remark 2. Since the strong convexity and the PL property hold for each additive component of f , another possible approach would be to run the procedure component-wise (minimize separately each component f_j). However, it leads to a worse result. Indeed, in this case we need to make at each step $2d$ queries in parallel (two queries for each component) and thus can make only $\sim T/d$

steps if the total budget of queries is T . At the end, applying Theorems 1 or 2 in the one-dimensional case, for each component we obtain the error in T and d of the order $(T/d)^{-(\beta-1)/\beta}$. This rate cannot be improved as follows from the one-dimensional instance of Theorem 3. Summing up over the d components, the overall error will be of the order $d(T/d)^{-(\beta-1)/\beta} = d^{2-1/\beta}T^{-(\beta-1)/\beta}$, that is, the error will depend on d in a sub-optimal way.

4. PROOFS

We start by proving some auxiliary lemmas.

Lemma 1. *Let $f : \mathbb{R}^d \rightarrow \mathbb{R}$ be a differentiable function such that $\|\nabla f(\mathbf{x}) - \nabla f(\mathbf{x}')\| \leq \bar{L}\|\mathbf{x} - \mathbf{x}'\|$ for all $\mathbf{x}, \mathbf{x}' \in \mathbb{R}^d$, where $\bar{L} > 0$. Let Assumption 1 hold and let $\{\mathbf{x}_t\}_{t=1}^T$ be the updates of Algorithm 1 with $\Theta = \mathbb{R}^d$. Then, for all $t \in [T]$ we have*

$$\mathbf{E}[f(\mathbf{x}_{t+1})|\mathbf{x}_t] \leq f(\mathbf{x}_t) - \frac{\eta_t}{2}\|\nabla f(\mathbf{x}_t)\|^2 + \frac{\eta_t}{2}\|\mathbf{E}[\mathbf{g}_t|\mathbf{x}_t] - \nabla f(\mathbf{x}_t)\|^2 + \frac{\bar{L}\eta_t^2}{2}\mathbf{E}[\|\mathbf{g}_t\|^2|\mathbf{x}_t]. \quad (7)$$

Proof. The assumption on f implies that

$$f(\mathbf{x}_{t+1}) = f(\mathbf{x}_t - \eta_t \mathbf{g}_t) \leq f(\mathbf{x}_t) - \eta_t \langle \nabla f(\mathbf{x}_t), \mathbf{g}_t \rangle + \frac{\bar{L}\eta_t^2}{2}\|\mathbf{g}_t\|^2.$$

By adding and subtracting $\eta_t \|\nabla f(\mathbf{x}_t)\|^2$ we obtain

$$f(\mathbf{x}_{t+1}) \leq f(\mathbf{x}_t) - \eta_t \|\nabla f(\mathbf{x}_t)\|^2 - \eta_t \langle \nabla f(\mathbf{x}_t), \mathbf{g}_t - \nabla f(\mathbf{x}_t) \rangle + \frac{\bar{L}\eta_t^2}{2}\|\mathbf{g}_t\|^2.$$

Taking the conditional expectation gives

$$\begin{aligned} \mathbf{E}[f(\mathbf{x}_{t+1})|\mathbf{x}_t] &\leq f(\mathbf{x}_t) - \eta_t \|\nabla f(\mathbf{x}_t)\|^2 - \eta_t \langle \nabla f(\mathbf{x}_t), \mathbf{E}[\mathbf{g}_t|\mathbf{x}_t] - \nabla f(\mathbf{x}_t) \rangle + \frac{\bar{L}\eta_t^2}{2}\mathbf{E}[\|\mathbf{g}_t\|^2|\mathbf{x}_t] \\ &\leq f(\mathbf{x}_t) - \eta_t \|\nabla f(\mathbf{x}_t)\|^2 + \eta_t \|\nabla f(\mathbf{x}_t)\| \|\mathbf{E}[\mathbf{g}_t|\mathbf{x}_t] - \nabla f(\mathbf{x}_t)\| + \frac{\bar{L}\eta_t^2}{2}\mathbf{E}[\|\mathbf{g}_t\|^2|\mathbf{x}_t]. \end{aligned}$$

The lemma follows by using the inequality $2ab \leq a^2 + b^2$, $\forall a, b \in \mathbb{R}$.

Lemma 2. *For $j \in [d]$, let $f_j : \mathbb{R} \rightarrow \mathbb{R}$ be such that $f_j \in \mathcal{F}_\beta(L)$, where $\beta \geq 2$ and $L > 0$. Assume that $f : \mathbb{R}^d \rightarrow \mathbb{R}$ satisfies the additive model $f(\mathbf{x}) = \sum_{j=1}^d f_j(x_j)$. Let Assumption 1(i) hold. Then, for all $t \in [T]$ we have*

$$\|\mathbf{E}[\mathbf{g}_t|\mathbf{x}_t] - \nabla f(\mathbf{x}_t)\| \leq \kappa_\beta L \sqrt{d} h_t^{\beta-1}.$$

Proof. Using Assumption 1(i) for any $j \in [d]$ and $t \in [T]$ we have $\mathbf{E}(K(r_{t,j})) = 0$, $\mathbf{E}[\xi_t K(r_{t,j})|\mathbf{x}_t] = 0$ and $\mathbf{E}[\xi'_t K(r_{t,j})|\mathbf{x}_t] = 0$. Thus,

$$\begin{aligned} \mathbf{E}[g_{t,j}|\mathbf{x}_t] &= \frac{1}{2h_t} \mathbf{E}[(f_j(x_{t,j} + h_t r_{t,j}) - f_j(x_{t,j} - h_t r_{t,j}))K(r_{t,j})|\mathbf{x}_t] \\ &\quad + \frac{1}{2h_t} \sum_{m \neq j} \mathbf{E}[(f_m(x_{t,m} + h_t r_{t,m}) - f_m(x_{t,m} - h_t r_{t,m}))K(r_{t,j})|\mathbf{x}_t] \\ &= \frac{1}{2h_t} \mathbf{E}[(f_j(x_{t,j} + h_t r_{t,j}) - f_j(x_{t,j} - h_t r_{t,j}))K(r_{t,j})|\mathbf{x}_t], \end{aligned}$$

where we have used the fact that $\mathbf{E}(K(r_{t,j})) = 0$ and $r_{t,j}$ is independent of $r_{t,m}$, for $m \neq j$, and of $\mathbf{x}_t$. From Taylor expansion we obtain that

$$\begin{aligned} \frac{1}{2h_t} (f_j(x_{t,j} + h_t r_{t,j}) - f_j(x_{t,j} - h_t r_{t,j})) &= f'_j(x_{t,j})r_{t,j} + \frac{1}{h_t} \sum_{1 \leq m \leq \ell, m \text{ odd}} \frac{h_t^m}{m!} f_j^{(m)}(x_{t,j})r_{t,j}^m \\ &\quad + \frac{R(h_t r_{t,j}) - R(-h_t r_{t,j})}{2h_t}, \end{aligned}$$

where $|R(-h_t r_{t,j})|, |R(h_t r_{t,j})| \leq L|r_{t,j}|^\beta h_t^\beta$. Multiplying both sides of this inequality by $K(r_{t,j})$ and taking the conditional expectation implies

$$|\mathbf{E}[g_{t,j}|\mathbf{x}_t] - f'_j(x_{t,j})| \leq L\mathbf{E}\left[|r_{t,j}|^\beta K(r_{t,j})\right] h_t^{\beta-1} = \kappa_\beta L h_t^{\beta-1}.$$

The result of the lemma follows from this inequality and the fact that

$$\|\mathbf{E}[\mathbf{g}_t|\mathbf{x}_t] - \nabla f(\mathbf{x}_t)\| \leq \sqrt{d} \max_{j=1,\dots,d} |\mathbf{E}[g_{t,j}|\mathbf{x}_t] - f'_j(x_{t,j})|.$$

Lemma 3. For $j \in [d]$, let $f_j : \mathbb{R} \rightarrow \mathbb{R}$ be such that $f_j \in \mathcal{F}'_2(\bar{L})$, where $\bar{L} > 0$. Assume that $f : \mathbb{R}^d \rightarrow \mathbb{R}$ satisfies the additive model $f(\mathbf{x}) = \sum_{j=1}^d f_j(x_j)$. Let Assumption 1 hold. Then, for all $t \in [T]$ we have

$$\mathbf{E}\left[\|\mathbf{g}_t\|^2|\mathbf{x}_t\right] \leq \frac{3}{2}\kappa d \left(\frac{3}{4}\left(d\bar{L}^2 h_t^2 + 8\|\nabla f(\mathbf{x}_t)\|^2\right) + \frac{\sigma^2}{h_t^2}\right).$$

Proof. For $i \in [d]$ we define

$$G_i = f_i(x_{t,i} + h_t r_{t,i}) - f_i(x_{t,i} - h_t r_{t,i}).$$

We have

$$\mathbf{E}\left[g_{t,j}^2|\mathbf{x}_t\right] = \frac{1}{4h_t^2}\mathbf{E}\left[\left(\sum_{i=1}^d G_i + \xi_t - \xi'_t\right)^2 K^2(r_{t,j})|\mathbf{x}_t\right].$$

Note that $r_{t,i}$ and $-r_{t,i}$ have the same distribution. Therefore, $\mathbf{E}[G_i|\mathbf{x}_t] = 0$ and we can write

$$\begin{aligned} \mathbf{E}\left[g_{t,j}^2|\mathbf{x}_t\right] &\leq \frac{3}{4h_t^2}\mathbf{E}\left[\left(\left(\sum_{i=1}^d G_i\right)^2 + \xi_t^2 + (\xi'_t)^2\right) K^2(r_{t,j})|\mathbf{x}_t\right] \\ &\leq \frac{3}{4h_t^2}\mathbf{E}\left[\sum_{i=1}^d G_i^2 K^2(r_{t,j}) + \sum_{i,k=1, i \neq k}^d G_i G_k K^2(r_{t,j})|\mathbf{x}_t\right] + \frac{3\sigma^2\kappa}{2h_t^2} \\ &= \frac{3}{4h_t^2}\mathbf{E}\left[\sum_{i=1}^d G_i^2 K^2(r_{t,j})|\mathbf{x}_t\right] + \frac{3\sigma^2\kappa}{2h_t^2}. \end{aligned}$$

Since $f_i \in \mathcal{F}'_2(\bar{L})$ we have that, for all $i \in [d]$,

$$\begin{aligned} G_i^2 &= ((f_i(x_{t,i} + h_t r_{t,i}) - f(x_{t,i}) - f'_i(x_{t,i})h_t r_{t,i}) - (f_i(x_{t,i} - h_t r_{t,i}) - f(x_{t,i}) + f'_i(x_{t,i})h_t r_{t,i})) \\ &\quad + 2f'_i(x_{t,i})h_t r_{t,i})^2 \\ &\leq 3\left((f_i(x_{t,i} + h_t r_{t,i}) - f(x_{t,i}) - f'_i(x_{t,i})h_t r_{t,i})^2 + (f_i(x_{t,i} - h_t r_{t,i}) - f(x_{t,i}) + f'_i(x_{t,i})h_t r_{t,i})^2\right. \\ &\quad \left.+ 4(f'_i(x_{t,i})h_t r_{t,i})^2\right) \\ &\leq 3\left(\frac{\bar{L}^2}{2}h_t^4 + 4(f'_i(x_{t,i}))^2 h_t^2\right). \end{aligned}$$

Thus,

$$\mathbf{E}\left[g_{t,j}^2|\mathbf{x}_t\right] \leq \frac{9}{4}\kappa \left(\frac{d\bar{L}^2}{2}h_t^2 + 4\sum_{i=1}^d (f'_i(x_{t,i}))^2\right) + \frac{3\sigma^2\kappa}{2h_t^2},$$

which implies the lemma.

Lemma 4. For $j \in [d]$, let $f_j : \mathbb{R} \rightarrow \mathbb{R}$ be such that $f_j \in \mathcal{F}'_2(\bar{L}) \cap \mathcal{F}_\beta(L)$, where $\beta \geq 2$ and $\bar{L}, L > 0$. Assume that $f : \mathbb{R}^d \rightarrow \mathbb{R}$ satisfies the additive model $f(\mathbf{x}) = \sum_{j=1}^d f_j(x_j)$. Let Assumption 1 hold. Assume that $\{\mathbf{x}_t\}_{t=1}^T$ are updates of Algorithm 1 with $\Theta = \mathbb{R}^d$ and with the gradient estimators defined by (4). Then, for all $t \in [T]$ we have

$$\begin{aligned} \mathbf{E}[f(\mathbf{x}_{t+1})|\mathbf{x}_t] &\leq f(\mathbf{x}_t) - \frac{\eta_t}{2} (1 - 9\bar{L}d\kappa\eta_t) \|\nabla f(\mathbf{x}_t)\|^2 \\ &\quad + \frac{\eta_t}{2} d \left(L\kappa_\beta h_t^{\beta-1} \right)^2 + \frac{3\bar{L}\eta_t^2}{4} \kappa d \left(\frac{\sigma^2}{h_t^2} + \frac{3\bar{L}^2 d}{4} h_t^2 \right). \end{aligned}$$

Proof. The result follows by combining Lemmas 1, 2 and 3.

Lemma 5. For $\alpha > 0$ assume that f is an α -strongly convex function. Assume that $\{\mathbf{x}_t\}_{t=1}^T$ are the updates of Algorithm 1. Then, for all $t \in [T]$ and $\mathbf{x} \in \Theta$, we have

$$\begin{aligned} f(\mathbf{x}_t) - f(\mathbf{x}) &\leq (2\eta_t)^{-1} \left(\|\mathbf{x}_t - \mathbf{x}\|^2 - \mathbf{E}[\|\mathbf{x}_{t+1} - \mathbf{x}\|^2|\mathbf{x}_t] \right) + \frac{1}{\alpha} \|\nabla f(\mathbf{x}_t) - \mathbf{E}[\mathbf{g}_t|\mathbf{x}_t]\|^2 \\ &\quad + \frac{\eta_t}{2} \mathbf{E}[\|\mathbf{g}_t\|^2|\mathbf{x}_t] - \frac{\alpha}{4} \|\mathbf{x}_t - \mathbf{x}\|^2. \end{aligned}$$

Proof. Fix $\mathbf{x} \in \Theta$. Then, by the contraction property of the Euclidean projection we have $\|\mathbf{x}_{t+1} - \mathbf{x}\|^2 \leq \|\mathbf{x}_t - \eta_t \mathbf{g}_t - \mathbf{x}\|^2$. Equivalently,

$$\langle \mathbf{g}_t, \mathbf{x}_t - \mathbf{x} \rangle \leq (2\eta_t)^{-1} (\|\mathbf{x}_t - \mathbf{x}\|^2 - \|\mathbf{x}_{t+1} - \mathbf{x}\|^2) + \frac{\eta_t}{2} \|\mathbf{g}_t\|^2.$$

Let $a_t = \|\mathbf{x}_t - \mathbf{x}\|^2$. Since f is an α -strongly convex function we have

$$\begin{aligned} f(\mathbf{x}_t) - f(\mathbf{x}) &\leq \langle \nabla f(\mathbf{x}_t), \mathbf{x}_t - \mathbf{x} \rangle - \frac{\alpha}{2} a_t \\ &= \langle \mathbf{g}_t, \mathbf{x}_t - \mathbf{x} \rangle + \langle \nabla f(\mathbf{x}_t) - \mathbf{g}_t, \mathbf{x}_t - \mathbf{x} \rangle - \frac{\alpha}{2} a_t \\ &\leq (2\eta_t)^{-1} (a_t - a_{t+1}) + \langle \nabla f(\mathbf{x}_t) - \mathbf{g}_t, \mathbf{x}_t - \mathbf{x} \rangle + \frac{\eta_t}{2} \|\mathbf{g}_t\|^2 - \frac{\alpha}{2} a_t. \end{aligned}$$

Taking the conditional expectation given $\mathbf{x}_t$ and using the inequality $ab \leq a^2/\lambda + \lambda b^2/4$ valid for all $a, b \in \mathbb{R}$ and $\lambda > 0$ we deduce that

$$\begin{aligned} f(\mathbf{x}_t) - f(\mathbf{x}) &\leq (2\eta_t)^{-1} (a_t - \mathbf{E}[a_{t+1}|\mathbf{x}_t]) + \langle \nabla f(\mathbf{x}_t) - \mathbf{E}[\mathbf{g}_t|\mathbf{x}_t], \mathbf{x}_t - \mathbf{x} \rangle + \frac{\eta_t}{2} \mathbf{E}[\|\mathbf{g}_t\|^2|\mathbf{x}_t] - \frac{\alpha}{2} a_t \\ &\leq (2\eta_t)^{-1} (a_t - \mathbf{E}[a_{t+1}|\mathbf{x}_t]) + \|\nabla f(\mathbf{x}_t) - \mathbf{E}[\mathbf{g}_t|\mathbf{x}_t]\| \|\mathbf{x}_t - \mathbf{x}\| + \frac{\eta_t}{2} \mathbf{E}[\|\mathbf{g}_t\|^2|\mathbf{x}_t] - \frac{\alpha}{2} a_t \\ &\leq (2\eta_t)^{-1} (a_t - \mathbf{E}[a_{t+1}|\mathbf{x}_t]) + \frac{1}{\alpha} \|\nabla f(\mathbf{x}_t) - \mathbf{E}[\mathbf{g}_t|\mathbf{x}_t]\|^2 + \frac{\eta_t}{2} \mathbf{E}[\|\mathbf{g}_t\|^2|\mathbf{x}_t] - \frac{\alpha}{4} a_t. \end{aligned}$$

Proof of Theorem 1. Since $\eta_t \leq 1/(18\bar{L}d\kappa)$ from Lemma 4 we have

$$\mathbf{E}[f(\mathbf{x}_{t+1})|\mathbf{x}_t] \leq f(\mathbf{x}_t) - \frac{\eta_t}{4} \|\nabla f(\mathbf{x}_t)\|^2 + \frac{\eta_t}{2} d \left(L\kappa_\beta h_t^{\beta-1} \right)^2 + \frac{3\bar{L}\eta_t^2}{4} \kappa d \left(\frac{\sigma^2}{h_t^2} + \frac{3\bar{L}^2 d}{4} h_t^2 \right).$$

Taking the expectation from both sides of this inequality and using the fact that f is an α -PL function we obtain

$$\delta_{t+1} \leq \delta_t \left(1 - \frac{\eta_t \alpha}{2} \right) + \frac{\eta_t}{2} d \left(L\kappa_\beta h_t^{\beta-1} \right)^2 + \frac{3\bar{L}\eta_t^2}{4} \kappa d \left(\frac{\sigma^2}{h_t^2} + \frac{3\bar{L}^2 d}{4} h_t^2 \right), \quad (8)$$

where $\delta_t = \mathbf{E}[f(\mathbf{x}_t) - f^*]$. Let $T_0 = \lceil 72\bar{L}d\kappa/\alpha \rceil$. Note that

$$\eta_t = \begin{cases} \frac{1}{18\bar{L}d\kappa} & \text{if } t < T_0, \\ \frac{4}{\alpha t} & \text{if } t \geq T_0. \end{cases}$$

We consider separately the cases $T > T_0$ and $T \leq T_0$.

If $T > T_0$, then for $t \geq T_0$ we have $\eta_t = 4/(\alpha t)$ and we can write

$$\delta_{t+1} \leq \delta_t \left(1 - \frac{2}{t}\right) + \frac{d}{\alpha t} \left(2 \left(L\kappa_\beta h_t^{\beta-1}\right)^2 + \frac{3\bar{L}}{\alpha t} \kappa \left(\frac{\sigma^2}{h_t^2} + \frac{3\bar{L}^2 d}{4} h_t^2\right)\right).$$

Substituting here $h_t = (\frac{3\bar{L}}{\alpha t} \cdot \frac{\kappa\sigma^2}{2L^2\kappa_\beta^2})^{\frac{1}{2\beta}}$ gives

$$\delta_{t+1} \leq \delta_t \left(1 - \frac{2}{t}\right) + \frac{d}{\alpha t} \left(\mathbf{A}_3 \left(\frac{\bar{L}}{\alpha t}\right)^{\frac{\beta-1}{\beta}} + \mathbf{A}_4 \frac{\bar{L}^3 d}{t} \left(\frac{\bar{L}}{\alpha t L^2}\right)^{\frac{1}{\beta}}\right),$$

where $\mathbf{A}_3 = 6 \left((2\kappa_\beta^2/3)(\kappa\sigma^2)^{\beta-1}\right)^{\frac{1}{\beta}}$ and $\mathbf{A}_4 = 9 \left(3\kappa\sigma^2/(2\kappa_\beta^2)\right)^{\frac{1}{\beta}}/4$. Since $T \geq T_0$, applying [1, Lemma 32] leads to the bound

$$\delta_T \leq \frac{2T_0}{T} \delta_{T_0+1} + \frac{d}{\alpha} \left(\frac{\beta\mathbf{A}_1}{\beta+1} \left(\frac{\bar{L}}{\alpha T}\right)^{\frac{\beta-1}{\beta}} + \frac{\beta\mathbf{A}_2}{2\beta+1} \frac{\bar{L}^3 d}{T} \left(\frac{\bar{L}}{\alpha T L^2}\right)^{\frac{1}{\beta}}\right).$$

If $T_0 = 0$ then the proof is complete for the case $T > T_0$. Otherwise, for any $t \leq T_0$, we have $\eta_t = 1/(18\bar{L}d\kappa)$ and $h_t = (\frac{3\bar{L}}{\alpha t} \cdot \frac{\kappa\sigma^2}{2L^2\kappa_\beta^2})^{\frac{1}{2\beta}}$. Note that

$$\frac{4}{(T_0+1)\alpha} \leq \eta_t \leq \frac{4}{T_0\alpha},$$

and from (8) we deduce that, for all $t \leq T_0$,

$$\delta_{t+1} \leq \delta_t \left(1 - \frac{2}{T_0+1}\right) + \frac{2d}{\alpha T_0} \left(\mathbf{A}_3 \left(\frac{\bar{L}}{\alpha}\right)^{\frac{\beta-1}{\beta}} \left(T^{-\frac{\beta-1}{\beta}} + \frac{T^{\frac{1}{\beta}}}{T_0}\right) + \mathbf{A}_4 \frac{\bar{L}^3 d}{T_0} \left(\frac{\bar{L}}{\alpha T L^2}\right)^{\frac{1}{\beta}}\right).$$

Note that $1 - 2/(T_0+1) \leq 1$. Thus, by taking into account the definition of T_0 and the fact that $T > T_0$ we get

$$\frac{2T_0}{T} \delta_{t+1} \leq \frac{144\bar{L}d\kappa}{\alpha T} \delta_1 + \frac{4d}{\alpha} \left(2\mathbf{A}_3 \left(\frac{\bar{L}}{\alpha T}\right)^{\frac{\beta-1}{\beta}} + \mathbf{A}_4 \frac{\bar{L}^3 d}{T} \left(\frac{\bar{L}}{\alpha T L^2}\right)^{\frac{1}{\beta}}\right).$$

Therefore,

$$\delta_T \leq \frac{144\bar{L}d\kappa}{\alpha T} \delta_1 + \frac{d}{\alpha} \left(\mathbf{A}_1 \left(\frac{\bar{L}}{\alpha T}\right)^{\frac{\beta-1}{\beta}} + \mathbf{A}_2 \frac{\bar{L}^3 d}{T} \left(\frac{\bar{L}}{\alpha T L^2}\right)^{\frac{1}{\beta}}\right),$$

where $\mathbf{A}_1 = \mathbf{A}_3 (\beta/(\beta+1) + 8)$ and $\mathbf{A}_2 = \mathbf{A}_4 (\beta/(2\beta+1) + 4)$.

Consider now the case $T \leq T_0$. Then from (8) we get that, for all $t \leq T$,

$$\delta_{t+1} \leq \delta_t \left(1 - \frac{2}{T_0 + 1}\right) + \frac{2d}{\alpha T_0} \left(\mathbf{A}_3 \left(\frac{\bar{L}}{\alpha} \right)^{\frac{\beta-1}{\beta}} \left(T^{-\frac{\beta-1}{\beta}} + \frac{T^{\frac{1}{\beta}}}{T_0} \right) + \mathbf{A}_4 \frac{\bar{L}^3 d}{T_0} \left(\frac{\bar{L}}{\alpha T L^2} \right)^{\frac{1}{\beta}} \right).$$

Thus,

$$\delta_{T+1} \leq \delta_1 \left(1 - \frac{2}{T_0 + 1}\right)^T + \frac{2d}{\alpha} \left(2\mathbf{A}_3 \left(\frac{\bar{L}}{\alpha T} \right)^{\frac{\beta-1}{\beta}} + \mathbf{A}_4 \frac{\bar{L}^3 d}{T} \left(\frac{\bar{L}}{\alpha T L^2} \right)^{\frac{1}{\beta}} \right).$$

Since $(1 - \lambda)^T \leq \exp(-\lambda T) \leq 1/(\lambda T)$ for all $T, \lambda > 0$, we deduce from the previous display that

$$\delta_{T+1} \leq \frac{T_0 + 1}{2T} \delta_1 + \frac{2d}{\alpha} \left(2\mathbf{A}_3 \left(\frac{\bar{L}}{\alpha T} \right)^{\frac{\beta-1}{\beta}} + \mathbf{A}_4 \frac{\bar{L}^3 d}{T} \left(\frac{\bar{L}}{\alpha T L^2} \right)^{\frac{1}{\beta}} \right).$$

By using the fact that $\lambda + 1 \leq 2\lambda$ for all $\lambda \geq 1$ we finally get

$$\begin{aligned} \delta_{T+1} &\leq \frac{2T_0}{T+1} \delta_1 + \frac{2d}{\alpha} \left(2\mathbf{A}_3 \left(\frac{2\bar{L}}{\alpha(T+1)} \right)^{\frac{\beta-1}{\beta}} + \mathbf{A}_4 \frac{2\bar{L}^3 d}{T+1} \left(\frac{2\bar{L}}{\alpha(T+1)L^2} \right)^{\frac{1}{\beta}} \right) \\ &\leq \frac{144\bar{L}d\kappa}{\alpha(T+1)} \delta_1 + \frac{d}{\alpha} \left(\mathbf{A}_1 \left(\frac{\bar{L}}{\alpha(T+1)} \right)^{\frac{\beta-1}{\beta}} + \mathbf{A}_2 \frac{\bar{L}^3 d}{T+1} \left(\frac{\bar{L}}{\alpha(T+1)L^2} \right)^{\frac{1}{\beta}} \right), \end{aligned}$$

where $\mathbf{A}_1 = 2^{\frac{3\beta-1}{\beta}} \mathbf{A}_3$ and $\mathbf{A}_2 = 2^{\frac{2\beta+1}{\beta}} \mathbf{A}_4$.

Proof of Theorem 2. Fix $\mathbf{x} \in \Theta$. By Lemma 5 we have

$$f(\mathbf{x}_t) - f(\mathbf{x}) \leq \frac{a_t - \mathbf{E}[a_{t+1}|\mathbf{x}_t]}{2\eta_t} + \frac{1}{\alpha} \|\nabla f(\mathbf{x}_t) - \mathbf{E}[\mathbf{g}_t|\mathbf{x}_t]\|^2 + \frac{\eta_t}{2} \mathbf{E}[\|\mathbf{g}_t\|^2|\mathbf{x}_t] - \frac{\alpha}{4} a_t,$$

where $a_t = \|\mathbf{x}_t - \mathbf{x}\|^2$. Using Lemmas 2 and 3 and the assumption that $\max_{\mathbf{x} \in \Theta} \|\nabla f(\mathbf{x})\| \leq G$ we obtain

$$f(\mathbf{x}_t) - f(\mathbf{x}) \leq \frac{a_t - \mathbf{E}[a_{t+1}|\mathbf{x}_t]}{2\eta_t} + \frac{d}{\alpha} (\kappa_\beta L h_t^{\beta-1})^2 + \frac{3\eta_t}{4} \kappa d \left(\frac{3}{4} (d\bar{L}^2 h_t^2 + 8G^2) + \frac{\sigma^2}{h_t^2} \right) - \frac{\alpha}{4} a_t.$$

Let $b_t = \mathbf{E}[\|\mathbf{x}_t - \mathbf{x}\|^2]$. Using the definition $\eta_t = 4/(\alpha(t+1))$ and taking the expectation we obtain

$$\mathbf{E}[f(\mathbf{x}_t) - f(\mathbf{x})] \leq \frac{\alpha}{4} (t(b_t - b_{t+1}) - b_t) + \frac{d}{\alpha} (\kappa_\beta L h_t^{\beta-1})^2 + \frac{3}{2\alpha(t+1)} \kappa d \left(\frac{3}{4} (d\bar{L}^2 h_t^2 + 8G^2) + \frac{\sigma^2}{h_t^2} \right).$$

Summing up both sides of this inequality from 1 to T and using the fact that

$$\sum_{t=1}^T t((t+1)(b_t - b_{t+1}) - b_t) \leq 0$$

we find

$$\sum_{t=1}^T t \mathbf{E}[f(\mathbf{x}_t) - f(\mathbf{x})] \leq \frac{d}{\alpha} \sum_{t=1}^T \left(t(\kappa_\beta L h_t^{\beta-1})^2 + \frac{3t}{2(t+1)} \kappa d \left(\frac{3}{4} (d\bar{L}^2 h_t^2 + 8G^2) + \frac{\sigma^2}{h_t^2} \right) \right).$$

Substituting here $h_t = (\frac{3}{2t} \frac{\kappa \sigma^2}{\kappa_\beta^2 L^2})^{\frac{1}{2\beta}}$ yields

$$\sum_{t=1}^T t \mathbf{E} [f(\mathbf{x}_t) - f(\mathbf{x})] \leq \frac{9G^2 dT}{\alpha} + \frac{d}{\alpha} \sum_{t=1}^T \left(A_3 (L^2 t)^{\frac{1}{\beta}} + A_4 d \bar{L}^2 (L^2 t)^{-\frac{1}{\beta}} \right),$$

where $A_3 = 2(3\kappa\sigma^2/2)^{\frac{\beta-1}{\beta}} \kappa_\beta^{\frac{2}{\beta}}$ and $A_4 = (1/2)(3/2)^{\frac{2\beta+1}{\beta}} \kappa^{\frac{\beta+1}{\beta}} (\sigma/\kappa_\beta)^{\frac{2}{\beta}}$. Using the inequality $\sum_{t=1}^T t^{-\frac{1}{\beta}} \leq (\beta/(\beta-1)) T^{\frac{\beta-1}{\beta}}$ we find

$$\sum_{t=1}^T t \mathbf{E} [f(\mathbf{x}_t) - f(\mathbf{x})] \leq \frac{9G^2 dT}{\alpha} + \frac{d}{\alpha} \left(A_3 L^{\frac{2}{\beta}} + \frac{\beta A_4}{\beta-1} d \bar{L}^2 (LT)^{-\frac{2}{\beta}} \right) T^{\frac{\beta+1}{\beta}}.$$

Dividing both sides of this inequality by $T(T+1)/2$ and applying Jensen's inequality we finally get

$$\mathbf{E} [f(\bar{\mathbf{x}}_T) - f(\mathbf{x})] \leq \frac{18G^2 d}{\alpha T} + \frac{d}{\alpha} \left(A_1 L^{\frac{2}{\beta}} + A_2 d \bar{L}^2 (LT)^{-\frac{2}{\beta}} \right) T^{-\frac{\beta-1}{\beta}},$$

where $A_1 = 2A_3$ and $A_2 = 2\beta A_4/(\beta-1)$.

FUNDING

The research of Arya Akhavan was funded by UK Research and Innovation (UKRI) under the UK government's Horizon Europe funding guarantee [grant number EP/Y028333/1].

REFERENCES

1. Akhavan, A., Chzhen, E., Pontil, M., and Tsybakov, A.B., Gradient-free optimization of highly smooth functions: improved analysis and a new algorithm, *J. Mach. Learn. Res.*, 2024, vol. 25, no. 370, pp. 1–50.
2. Akhavan, A., Pontil, M., and Tsybakov, A., Exploiting higher order smoothness in derivative-free optimization and continuous bandits, *Adv. Neural Inf. Process. Syst.*, 2020, vol. 33, pp. 9017–9027.
3. Akhavan, A., Pontil, M., and Tsybakov, A., Distributed zero-order optimization under adversarial noise, *Adv. Neural Inf. Process. Syst.*, 2021, vol. 34, pp. 10209–10220.
4. Bach, F. and Perchet, V., Highly-smooth zero-th order online optimization, *Proc. 29th Annu. Conf. Learn. Theory*, 2016, pp. 1–27.
5. Balasubramanian, K. and Ghadimi, S., Zeroth-order nonconvex stochastic optimization: Handling constraints, high dimensionality, and saddle points, *Found. Comput. Math.*, 2021, pp. 1–42.
6. Fabian, V., Stochastic approximation of minima with improved asymptotic speed, *Ann. Math. Stat.*, 1967, vol. 38, no. 1, pp. 191–200.
7. Fokkema, H., van der Hoeven, D., Lattimore, T., and Mayo, J.J., Online Newton method for bandit convex optimisation, 2024, <https://arxiv.org/abs/2406.06506>.
8. Gasnikov, A., Lagunovskaya, A., Usmanova, I., and Fedorenko, F., Gradient-free proximal methods with inexact oracle for convex stochastic nonsmooth optimization problems on the simplex, *Autom. Remote Control*, 2016, vol. 77, no. 11, pp. 2018–2034.
9. Gasnikov, A., Lobanov, A.V., and Stonyakin, F.S., Highly smooth zeroth-order methods for solving optimization problems under the PL condition, *Comput. Math. Math. Phys.*, 2024, vol. 64, pp. 739–770.
10. Ghadimi, S. and Lan, G., Stochastic first- and zeroth-order methods for nonconvex stochastic programming, *SIAM J. Optim.*, 2013, vol. 23, no. 4, pp. 2341–2368.
11. Granichin, O.N., A stochastic approximation procedure with disturbance at the input, *Autom. Remote Control*, 1992, vol. 53, no. 2, pp. 232–237.

12. Hastie, T. and Tibshirani, R., Generalized additive models, *Stat. Sci.*, 1986, vol. 1, no. 3, pp. 297–310.
13. Ibragimov, I.A. and Khas'minskii, R.Z., Bounds for the risks of non-parametric regression estimates, *Theory Probab. Appl.*, 1982, vol. 27, pp. 84–99.
14. Ibragimov, I.A. and Khas'minskii, R.Z., *Statistical Estimation: Asymptotic Theory*, Springer, 1981.
15. Jamieson, K.G., Nowak, R., and Recht, B., Query complexity of derivative-free optimization, *Adv. Neural Inf. Process. Syst.*, 2012, vol. 26, pp. 2672–2680.
16. Karimi, H., Nutini, J., and Schmidt, M., Linear convergence of gradient and proximal-gradient methods under the Polyak–Łojasiewicz condition, *Mach. Learn. Knowl. Discov. Databases*, 2016, pp. 795–811.
17. Kiefer, J. and Wolfowitz, J., Stochastic estimation of the maximum of a regression function, *Ann. Math. Stat.*, 1952, vol. 23, pp. 462–466.
18. Łojasiewicz, S., A topological property of real analytic subsets, *Coll. CNRS, Équ. Dériv. Partielles*, 1963, vol. 117, pp. 87–89.
19. Nemirovsky, A.S. and Yudin, D.B., *Problem Complexity and Method Efficiency in Optimization*, Wiley, 1983.
20. Nesterov, Yu. and Spokoiny, V., Random gradient-free minimization of convex functions, *Found. Comput. Math.*, 2017, vol. 17, no. 2, pp. 527–566.
21. Novitskii, V. and Gasnikov, A., Improved exploitation of higher order smoothness in derivative-free optimization, *Optim. Lett.*, 2022, vol. 16, pp. 2059–2071.
22. Polyak, B.T., Gradient methods for minimizing functionals, *Comput. Math. Math. Phys.*, 1963, vol. 3, no. 4, pp. 864–878.
23. Polyak, B.T. and Tsybakov, A.B., Optimal order of accuracy of search algorithms in stochastic optimization, *Probl. Inf. Transm.*, 1990, vol. 26, no. 2, pp. 45–53.
24. Polyak, B.T. and Tsybakov, A.B., On stochastic approximation with arbitrary noise (the KW-case), *Adv. Sov. Math.*, Amer. Math. Soc., 1992, vol. 12, pp. 107–113.
25. Shamir, O., On the complexity of bandit and derivative-free stochastic convex optimization, *Proc. 30th Annu. Conf. Learn. Theory*, 2013, pp. 1–22.
26. Stone, C.J., Optimal rates of convergence for nonparametric estimators, *Ann. Stat.*, 1980, vol. 8, pp. 1348–1360.
27. Stone, C.J., Optimal global rates of convergence for nonparametric regression, *Ann. Stat.*, 1982, vol. 10, pp. 1040–1053.
28. Stone, C.J., Additive regression and other nonparametric models, *Ann. Stat.*, 1985, vol. 13, pp. 689–705.
29. Tsybakov, A.B., *Introduction to Nonparametric Estimation*, Springer, 2009.
30. Wood, S.N., *Generalized Additive Models*, Boca Raton, Chapman and Hall/CRC, 2017.
31. Yu, Q., Wang, Y., Huang, B., Lei, Q., and Lee, J.D., Stochastic zeroth-order optimization under strongly convexity and Lipschitz Hessian: Minimax sample complexity, *Adv. Neural Inf. Process. Syst.*, 2024, vol. 37, pp. 99564–99600.

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

Robust Control Design Based on Noisy Current Data

R. S. Biryukov^{*,a} and M. M. Kogan^{**,b}

^{*}Lobachevsky State University of Nizhny Novgorod, Nizhny Novgorod, Russia

^{**}Sirius University of Science and Technology, Sochi, Russia

e-mail: ^abiryukovrs@gmail.com, ^bmkogan@nngasu.ru

Received March 3, 2025

Revised May 20, 2025

Accepted June 27, 2025

Abstract—This paper presents a new approach to designing robust controllers for an uncertain dynamic system using the generalized H_∞ norm as a criterion. The controller parameters are tuned during the real-time operation of the system as current measurements of the state and control, obtained with error, become available. The numerical simulation results of the active vibration suppression of buildings under seismic loads illustrate the effectiveness of the approach proposed.

Keywords: current data, robust control, generalized H_∞ norm, dual systems, linear matrix inequalities (LMIs)

DOI: 10.31857/S0005117925090038

1. INTRODUCTION

In this paper, we develop a new approach to designing optimal controllers for linear dynamic systems with unknown mathematical models, unmeasurable disturbances, and uncertain initial conditions, utilizing a priori information and current data in an online mode. As is known, one of the most common robust control methods is based on the minimax approach: design a controller ensuring the best estimate of an optimized functional over the entire set of uncertain parameter values, determined based on a priori information (for example, see [1, 2]). Despite all its advantages, this approach has a significant drawback: the guaranteed upper bound of the functional is quite rough due to the large size of the uncertainty set, and in the absence of accurate a priori information, the uncertainty set can become so large that a robust controller ceases to exist.

Recently, there has been significant interest in the development of control methods using data obtained preliminarily during an experiment, often referred to as “data-driven control” [3, 4]. Initially, these methods were based on the fundamental result of [5], showing that a single trajectory obtained under the so-called persistent excitation condition can be used for the complete characterization of a linear time-invariant dynamic system without disturbances. According to [6], the parameterization of a closed-loop control system based on experimental data allows designing controllers using linear matrix inequalities (LMIs). As established in [7], for designing controllers from experimental data, it suffices to satisfy the less restrictive condition of data informativity with respect to the property under study. Subsequently, these methods were extended to systems with disturbances [8–11] and linear time-varying systems [12]; a stabilizing controller based on measurements obtained with an error was designed in [13].

The mathematical modeling results presented in the above works demonstrate that even for relatively small disturbance amplitudes, the guaranteed estimate of the control performance index turns out to be significantly overvalued. This happens because, as the noise amplitude increases, the set of systems consistent with the experimental data expands significantly. In most of these works,

experimental data must also satisfy certain requirements: the matrix composed of state and control measurements along the system trajectory must be of full row rank. To satisfy this rank condition, the input signals in the experiments must provide persistent excitation in the system, necessary for the identifiability of the unknown parameters. Although persistent excitation in the system is not formally required to ensure data informativity, the uncertainty set can become unbounded even for small noise amplitudes if the rank condition fails for the experimental data.

These circumstances suggest the idea of designing a controller based on both a priori information and experimental data. In [14–20], the parameters of state-feedback controllers for linear time-invariant systems on an infinite horizon, time-varying systems on a finite horizon, as well as some classes of nonlinear systems, were found with the joint use of a priori information and experimental data by solving a minimax problem on the set of unknown parameters consistent with the a priori information and experimental data. Such controllers, obtained by solving LMIs, provide a certainly smaller value of the guaranteed functional estimate than, first, classical robust controllers and, second, a controller designed only from experimental data.

During the experimental stage, which precedes the controller design, input signals are chosen based on the informativity requirements for the output data. However, this stage is not always feasible due to, e.g., the instability of the system. Furthermore, the parameters of some control systems depend on the state of the environment, e.g., temperature or pressure. In these cases, control designed from preliminary stage data may be just partially suitable under other conditions. Instead, it would be desirable to retune the controller parameters at each step of the real-time system operation as new current data arrive, based on solving a minimax problem for the set of models consistent with the information available by that time. Since this set narrows (or, at worst, does not increase) at each step, the feedback controllers determined in this way will provide monotonically decreasing or non-increasing guaranteed values of the functional for the unknown system. The parameter tuning procedure stops when the guaranteed functional value becomes less than a desired one. This approach is developed below. Note that the informativity conditions are not required, and the significant question about the necessary number of measurements is eliminated as well. An illustrative example of active vibration suppression of buildings under seismic loads demonstrates the effectiveness of the controller design.

2. PROBLEM STATEMENT

Consider an uncertain system of the form

$$\begin{aligned}x_{t+1} &= A_{real}x_t + B_{real}u_t + B_w w_t, \quad x(0) = x_0, \\z_t &= Cx_t + Du_t, \quad t = 0, 1, \dots,\end{aligned}\tag{2.1}$$

with the following notation: $x_t \in \mathbb{R}^{n_x}$ is the state vector, $u_t \in \mathbb{R}^{n_u}$ is the control vector (input), $w_t \in \mathbb{R}^{n_w}$ is a disturbance, and $z_t \in \mathbb{R}^{n_z}$ is the performance (controlled) output. By assumption, the initial state x_0 can be arbitrary, the disturbance $\{w_t, t \geq 0\}$ has a bounded l_2 norm, i.e., $\|w\| = (\sum_{t=0}^{\infty} |w_t|^2)^{1/2} < \infty$, where $|\cdot|$ stands for the Euclidean norm of an appropriate vector, and the state is available for control in the feedback loop. The system matrices, collected in the matrix $\Omega_{real} = (A_{real} \ B_{real})$, are unknown, but the pair (A_{real}, B_{real}) is supposed to be stabilizable; finally, the matrix B_w (in particular, an identity matrix) is known.

Since the system matrices are unknown, standard optimal control methods turn out to be inapplicable. In this regard, we will involve a priori information, as is customary in robust control, and current data, characteristic of adaptive control, for control design. Assume that in the system, the disturbance w_t has a bounded Euclidean norm and the state and control are measured with some bounded noise:

$$\hat{x}_t = x_t + \xi_t, \quad \hat{u}_t = u_t + \eta_t,\tag{2.2}$$

where $\hat{x}_t$ and $\hat{u}_t$ are the state and control measurements, respectively, at a time instant t , and

$$|\xi_t| \leq \varepsilon_\xi, \quad |\eta_t| \leq \varepsilon_\eta, \quad |w_t| \leq \varepsilon_w \quad \forall t \geq 0. \quad (2.3)$$

Note that the state $\hat{x}_t$ and control $\hat{u}_t$ measurements, which will be included in the current data, contain the specified bounded errors, and the state x_t is available for control.

Generally, it is required to design linear state-feedback controllers with tunable parameters based on a priori information and current data during the real-time operation of the system in order to reach, after some time, steady-state values that minimize the effect of the initial conditions and exogenous disturbance on the performance output of the closed-loop time-invariant system.

3. CHARACTERIZATION OF THE NORM IN DUAL SYSTEM TERMS

Before proceeding to the problem solution, let us recall the main results regarding the generalized H_∞ norm from the input w_t to output z_t of a given closed-loop system with a controller $u_t = \Theta x_t$, described by the equations

$$\begin{aligned} x_{t+1} &= (A + B\Theta)x_t + B_w w_t, \\ z_t &= (C + D\Theta)x_t. \end{aligned} \quad (3.1)$$

The generalized H_∞ norm characterizes the effect of initial conditions and an exogenous disturbance on the l_2 norm of the performance output (i.e., on the transients) and is defined as

$$\|H\|_{g\infty}(\Omega, \Theta) = \sup_{\{x_0, w_t \in l_2\}} \frac{\|z\|}{(x_0^T R^{-1} x_0 + \|w\|^2)^{1/2}}, \quad (3.2)$$

where $\Omega = (A \ B)$ and $R = R^T \succ 0$ is a weight matrix measuring the comparative significance of the uncertainty in the initial conditions and exogenous disturbance for the performance output; $S \succ T$ ($S \succeq T$) means that the matrix $S - T$ is positive definite (positive semidefinite, respectively). In this definition, by assumption, the denominator on the right-hand side of (3.2) does not vanish and

$$H(s) = (C + D\Theta)[sI - (A + B\Theta)]^{-1} B_w$$

is the transfer matrix of the closed-loop system relating the disturbance to the performance output. In particular, the generalized H_∞ norm of the system with zero initial conditions coincides with the standard H_∞ norm; in the absence of an exogenous disturbance, it characterizes the maximum value of the quadratic functional of the performance output for an initial state belonging to the ellipsoid $x^T R^{-1} x \leq 1$. The following results on the computation of these norms will be applied below.

Lemma 3.1 [21]. *The generalized H_∞ norm of system (3.1) satisfies the condition $\|H\|_{g\infty} < \gamma$ if and only if there exists a positive definite quadratic form $V(x) = x^T Y x$, $Y \prec \gamma^2 R^{-1}$ such that*

$$\Delta V(x_t) + |z_t|^2 - \gamma^2 |w_t|^2 < 0, \quad (3.3)$$

for all nonzero x_t and w_t , where $\Delta V(x_t) = V(x_{t+1}) - V(x_t)$ denotes the increment of the function $V(x)$ along the trajectory of the corresponding system.

Lemma 3.2 [17]. *The generalized H_∞ norm of system (3.1) satisfies the condition $\|H\|_{g\infty} < \gamma$ if and only if there exists a positive definite quadratic form $V_d(x) = x^T P x$, $P \succ R$, whose increment along the trajectory of the dual system*

$$\begin{aligned} x_{t+1}^{(d)} &= (A + B\Theta)^T x_t^{(d)} + (C + D\Theta)^T w_t^{(d)}, \\ z_t^{(d)} &= B_w^T x_t^{(d)} \end{aligned} \quad (3.4)$$

is such that

$$\Delta V_d(x_t^{(d)}) + |z_t^{(d)}|^2 - \gamma^2 |w_t^{(d)}|^2 < 0 \quad (3.5)$$

for all nonzero $x_t^{(d)}$ and $w_t^{(d)}$.

Remark 3.1. For the generalized H_∞ norm, the matrices of the quadratic forms $V(x) = x^T Y x$ and $V_d(x^{(d)}) = x^{(d)T} P x^{(d)}$ of the primal and dual systems are related by $P = \gamma^2 Y^{-1}$.

Remark 3.2. Under the conditions of each of the lemmas above, the corresponding systems are asymptotically stable in the absence of disturbances.

4. A PRIORI INFORMATION AND CURRENT DATA

Following conventional robust control methods, let there be a priori information that the real matrix $\Omega_{real} = (A_{real} \ B_{real})$ lies strictly inside the domain defined by the inequality

$$(\Omega - \Omega_*)(\Omega - \Omega_*)^T \preceq \rho^2 I, \quad \Omega_* = (A_* \ B_*), \quad (4.1)$$

where Ω_* and ρ characterize the nominal system and the size of the uncertainty domain, respectively. We write inequality (4.1) as

$$F_a(\Omega) := (\Omega \ I) \Psi_a (\Omega \ I)^T \preceq 0, \quad (4.2)$$

where

$$\Psi_a = \left(\begin{array}{c|c} I & \star \\ \hline -\Omega_* & \Omega_* \Omega_*^T - \rho^2 I \end{array} \right) \quad (4.3)$$

and $\star$ replaces the corresponding block of the symmetric matrix. The set $\Delta^{(a)} := \{\Omega \in \mathbb{R}^{n_x \times (n_x + n_u)} : F_a(\Omega) \preceq 0\}$ will be called the set of all matrices consistent with the a priori information. By assumption, $\Omega_{real} \in \Delta^{(a)}$.

Due to (2.1) and (2.2), the current data obtained at the time instant $t + 1$ satisfy the equation

$$\hat{x}_{t+1} = \Omega_{real} \hat{\varphi}_t + (I \ \Omega_{real} \ B_w) \begin{pmatrix} \xi_{t+1} \\ \zeta_t \\ w_t \end{pmatrix}, \quad (4.4)$$

where

$$\hat{\varphi}_t := \begin{pmatrix} \hat{x}_t \\ \hat{u}_t \end{pmatrix}, \quad \zeta_t := - \begin{pmatrix} \xi_t \\ \eta_t \end{pmatrix}.$$

The set of all matrices Ω satisfying equation (4.4) for some disturbance and noises under the constraints (2.3) will be called the set of all matrices consistent with the data $\hat{x}_t, \hat{x}_{t+1}$, and $\hat{u}_t$, and denoted by $\Delta_t^{(p)}$. Clearly, $\Omega_{real} \in \Delta_t^{(p)}$, and the set $\Delta_t^{(p)}$ corresponds to all systems that could have generated these data. This set is characterized as follows.

Lemma 4.1. All matrices $\Omega \in \Delta_t^{(p)}$ consistent with the data $\hat{x}_t, \hat{x}_{t+1}$, and $\hat{u}_t$ satisfy the inequality

$$(\Omega \ I) \begin{pmatrix} \hat{\varphi}_t \hat{\varphi}_t^T - d^2 I & \star \\ -\hat{x}_{t+1} \hat{\varphi}_t^T & \hat{x}_{t+1} \hat{x}_{t+1}^T - (I + B_w B_w^T) d^2 \end{pmatrix} (\Omega \ I)^T \preceq 0, \quad (4.5)$$

where $d^2 = 2\varepsilon_\xi^2 + \varepsilon_\eta^2 + \varepsilon_w^2$.

The proof of this and all subsequent results is given in the Appendix. Let some measurements of the states $\hat{x}_0, \dots, \hat{x}_k$ and previously chosen controls $\hat{u}_0, \dots, \hat{u}_{k-1}$ of system (2.1) under some unknown disturbances $w_0, \dots, w_{k-1}$ be obtained by a fixed time instant k . The matrices Ω consistent with all the data $\hat{x}_0, \dots, \hat{x}_k$ and $\hat{u}_0, \dots, \hat{u}_{k-1}$ obtained by a time instant $k \geq 1$ belong to the set $\Delta_{[0, k-1]}^{(p)} = \bigcap_{t=0}^{k-1} \Delta_t^{(p)}$ and are defined by inequalities (4.5) for $t = 0, \dots, k-1$. We denote by $\Delta_0 = \Delta^{(a)}$ and $\Delta_k = \Delta^{(a)} \cap \Delta_{[0, k-1]}^{(p)}$, $k \geq 1$, the set of all matrices Ω consistent with the a priori information and the data $\hat{x}_0, \dots, \hat{x}_k$ and $\hat{u}_0, \dots, \hat{u}_{k-1}$ obtained by a time instant k for $k \geq 1$. Note that $\Delta_0 \supseteq \Delta_1 \supseteq \dots$ and $\Omega_{real} \in \Delta_k$ for any $k \geq 0$.

Checking whether the intersection of ellipsoids belongs to a given ellipsoid cannot be represented as a semidefinite programming problem [22]. Therefore, we replace the set Δ_k with its outer ellipsoidal approximation. With this approach, it is possible to find a controller minimizing the upper bound of the considered norm of the closed-loop system for all models consistent with the available data.

5. OUTER APPROXIMATION OF THE SET Δ_k

Summing inequalities (4.2) and (4.5) with some nonnegative multipliers $\mu \geq 0$ and $\tau_0 \geq 0, \dots, \tau_{k-1} \geq 0$, we establish that all matrices Ω consistent with the a priori information and current data by the time instant k satisfy the inequality

$$\hat{F}_{\mu, \tau_0^{k-1}}(\Omega) := (\Omega \quad I) \Psi(\mu, \tau_0^{k-1}) (\Omega \quad I)^T \preceq 0, \quad k \geq 1, \quad (5.1)$$

where

$$\Psi(\mu, \tau_0^{k-1}) = \begin{pmatrix} \mu I + \sum_{t=0}^{k-1} \tau_t (\hat{\varphi}_t \hat{\varphi}_t^T - d^2 I) & \star \\ -\mu \Omega_* - \sum_{t=0}^{k-1} \tau_t \hat{x}_{t+1} \hat{\varphi}_t^T & \mu (\Omega_* \Omega_*^T - \rho^2 I) + \sum_{t=0}^{k-1} \tau_t \hat{x}_{t+1} \hat{x}_{t+1}^T - M_k \end{pmatrix}, \quad (5.2)$$

$$\tau_0^{k-1} = (\tau_0, \dots, \tau_{k-1}), \quad M_k := \sum_{t=0}^{k-1} \tau_t (I + B_w B_w^T) d^2.$$

We define $\widehat{\Delta}_k(\mu, \tau_0^{k-1}) = \{\Omega \in \mathbb{R}^{n_x \times (n_x + n_u)} : \hat{F}_{\mu, \tau_0^{k-1}}(\Omega) \preceq 0\}$ as the set of all matrices Ω satisfying inequality (5.1) for fixed $\mu \geq 0$ and $\tau_0^{k-1} \geq 0$. Thus, $\Delta_k \subseteq \widehat{\Delta}_k(\mu, \tau_0^{k-1})$ for any $\mu \geq 0$, $\tau_0^{k-1} \geq 0$ and $\Omega_{real} \in \widehat{\Delta}_k(\mu, \tau_0^{k-1})$. This set, with certain values of $\mu \geq 0$ and $\tau_0^{k-1} \geq 0$, will serve as an outer approximation of the set Δ_k .

Let us find boundedness conditions for the set $\widehat{\Delta}_k(\mu, \tau_0^{k-1})$. We denote by $\Psi_{ij}(\mu, \tau_0^{k-1})$, $i, j = 1, 2$, the corresponding blocks of the matrix (5.2), whose arguments will sometimes be omitted.

Lemma 5.1. *If*

$$\Psi_{11} = \mu I + \sum_{t=0}^{k-1} \tau_t (\hat{\varphi}_t \hat{\varphi}_t^T - d^2 I) \succ 0, \quad (5.3)$$

then the set $\widehat{\Delta}_k(\mu, \tau_0^{k-1})$ is a bounded matrix ellipsoid given by the inequality

$$(\Omega + \Psi_{12}^T \Psi_{11}^{-1}) \Psi_{11} (\Omega + \Psi_{12}^T \Psi_{11}^{-1})^T \preceq \Gamma_k, \quad (5.4)$$

where $\Gamma_k = \Psi_{12}^T \Psi_{11}^{-1} \Psi_{12} - \Psi_{22} \succeq 0$.

Remark 5.1. Condition (5.3) means that based on the a priori information, the “energy” of the measured signal on the entire interval must exceed the total “energy” of the disturbance and measurement noise. If this condition fails, the set $\widehat{\Delta}_k(\mu, \tau_0^{k-1})$ will be unbounded; see [19, Lemma 2.1].

6. ROBUST CONTROLLER BASED ON CURRENT DATA

To design a robust controller with a gain matrix tuned from current data, first it is necessary to solve the following auxiliary problem: determine the matrix Θ for which the upper bound of the generalized H_∞ norm of the uncertain closed-loop stable system (3.1) will be minimal for all $\Omega \in \widehat{\Delta}_k(\mu, \tau_0^{k-1})$ with some $\mu \geq 0$ and $\tau_0^{k-1} \geq 0$. Generally speaking, a particular Lyapunov function may exist for each model with the matrix Ω ; according to Lemma 3.1, it characterizes the bound of the norm considered. However, conditions for the existence of a common Lyapunov function for all $\Omega \in \widehat{\Delta}_k(\mu, \tau_0^{k-1})$ will be obtained below. This introduces some conservatism but allows deriving necessary and sufficient conditions for the existence of such a function and finding the corresponding gain matrices. Note that a similar approach was used in robust control when determining the so-called quadratic stability and performing stabilization under uncertainty (for example, see [23]).

Definition 6.1. A controller $u_t = \Theta x_t$ is called the guaranteed generalized H_∞ -controller with a level γ for the uncertain system (3.1) based on a priori information and current data by a time instant k if there exists a function $V_k(x) = x^T Y_k x$, $0 \prec Y_k \prec \gamma^2 R^{-1}$, and numbers $\mu \geq 0$ and $\tau_0^{k-1} \geq 0$ such that inequality (3.3) holds for all $\Omega \in \widehat{\Delta}_k(\mu, \tau_0^{k-1})$.

According to Lemma 3.1 and Remark 3.2, the closed-loop system with the guaranteed generalized H_∞ -controller with a level γ is asymptotically stable and $\|H\|_{g\infty} < \gamma$ for all models consistent with the a priori information and current data by a time instant k .

Theorem 6.1. A controller $u_t = \Theta x_t$ is the guaranteed generalized H_∞ -controller with a level γ for the uncertain system (3.1) based on current data by a time instant k if and only if $\Theta = Q_k P_k^{-1}$ and the following LMI is solvable for $P_k = P_k^T \succ R$, Q_k , $\mu \geq 0$, and $\tau_0^{k-1} \geq 0$:

$$\begin{pmatrix} -P_k & \star & \star & \star \\ \begin{pmatrix} P_k \\ Q_k \end{pmatrix} & -\Psi_{11} & \star & \star \\ 0 & -\Psi_{21} & -\Psi_{22} - P_k + B_w B_w^T & \star \\ (C \ D) \begin{pmatrix} P_k \\ Q_k \end{pmatrix} & 0 & 0 & -\gamma^2 I \end{pmatrix} \prec 0, \quad (6.1)$$

where Ψ_{ij} , $i, j = 1, 2$, denote the corresponding blocks of the matrix $\Psi(\mu, \tau_0^{k-1})$ (5.2).

Remark 6.1. Letting $\mu = 0$ in inequality (6.1) eliminates the a priori information from the control design; with $\tau_0^{k-1} = 0$, the current data will not be used.

Remark 6.2. If the matrix $\Psi(\mu, \tau_0^{k-1})$ consists of experimental data obtained on the interval $[0, k-1]$ with the open-loop uncertain system under experimenter's controls, then Theorem 6.1 remains valid. Note that in this case, with $\mu = 0$ and $\tau_0 = \dots = \tau_{k-1} = 1$, the corresponding set $\widehat{\Delta}_k(\mu, \tau_0^{k-1})$ coincides with the set of all matrices consistent with the obtained data, as established in [6, 7, 16]. This means that the approach considered here yields less conservative results than in the mentioned works.

The minimum value of $\gamma > 0$ for which inequality (6.1) is solvable will be called the guaranteed generalized H_∞ norm of the uncertain system based on data by a time instant k . This value will be denoted by γ_k^* whereas the corresponding gain matrix by Θ_k . Since inequality (6.1) for the time instant k with $\tau_{k-1} = 0$ turns into inequality (6.1) for the time instant $k-1$, the guaranteed generalized H_∞ norms form a nonincreasing sequence as current data are received:

$$\gamma_0^* \geq \gamma_1^* \geq \dots \quad (6.2)$$

Let us proceed to the robust control design. As the gain matrix at the initial time instant, we take the matrix obtained based on only the a priori information, i.e., $\Theta_0 = Q_0 P_0^{-1}$, where P_0 and Q_0 are the solution of inequality (6.1) with $\tau_0^{k-1} = 0$ and $\gamma = \gamma_0^*$. For $t = 1, \dots, N-1$, the matrices Θ_t are computed as in Theorem 6.1 for the minimum value $\gamma = \gamma_t^*$. For $t \geq N$, we set $\Theta_t = \Theta_{N-1}$.

Theorem 6.2. *For the uncertain system (2.1) under the controller $u_t = \Theta_t x_t$, where Θ_t , $t = 0, \dots, N-1$, are computed as in Theorem 6.1 for the minimum value of $\gamma > 0$ and $\Theta_t = \Theta_{N-1}$ for $t \geq N$, the following condition holds after the parameter tuning procedure:*

$$\sum_{t=N-1}^{\infty} |z_t|^2 < \gamma_{N-1}^{*2} \left(x_{N-1}^T R^{-1} x_{N-1} + \sum_{t=N-1}^{\infty} |w_t|^2 \right) \quad \forall w_t. \quad (6.3)$$

Due to (6.2) and (6.3), in the current data-based control design, it is necessary to trade off between the quality of the closed-loop system obtained after the tuning procedure and the duration of the tuning interval (i.e., the costs): the larger N is, the smaller γ_{N-1}^* will be, and vice versa.

In the case of a sufficiently large uncertainty radius (i.e., rough a priori information), inequality (6.1) with $\tau_0^{k-1} = 0$ may have no solutions. This means the absence of a common stabilizing controller for all models from the initial set Δ_0 . In this case, a random vector is chosen as the initial control action. If at the next step, still no common controller exists for the new set of models $\widehat{\Delta}_1(\mu, \tau_0) \subseteq \Delta_0$, then a random control action is chosen again, and this process continues until a time instant t_* when inequality (6.1) will have a solution. The guaranteed estimate of the control performance (see Theorem 6.2) remains valid in this case if the initial time is taken as $t = t_*$.

7. AN ILLUSTRATIVE EXAMPLE

Consider a model of vibrations of a three-story building under seismic load on its foundation. Figure 1 shows the general scheme of this process. The building stories are represented as material points, serially connected to each other and to the foundation by linear elastic and dissipative elements. Vibrations generated by the seismic load (foundation movement) occur in the horizontal plane. Assume that the building is homogeneous, i.e., the masses of the material points, as well as

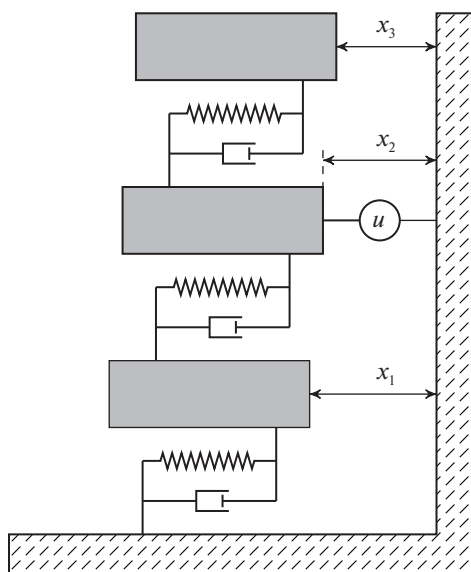


Fig. 1. The scheme of a building as a multimass elastic system.

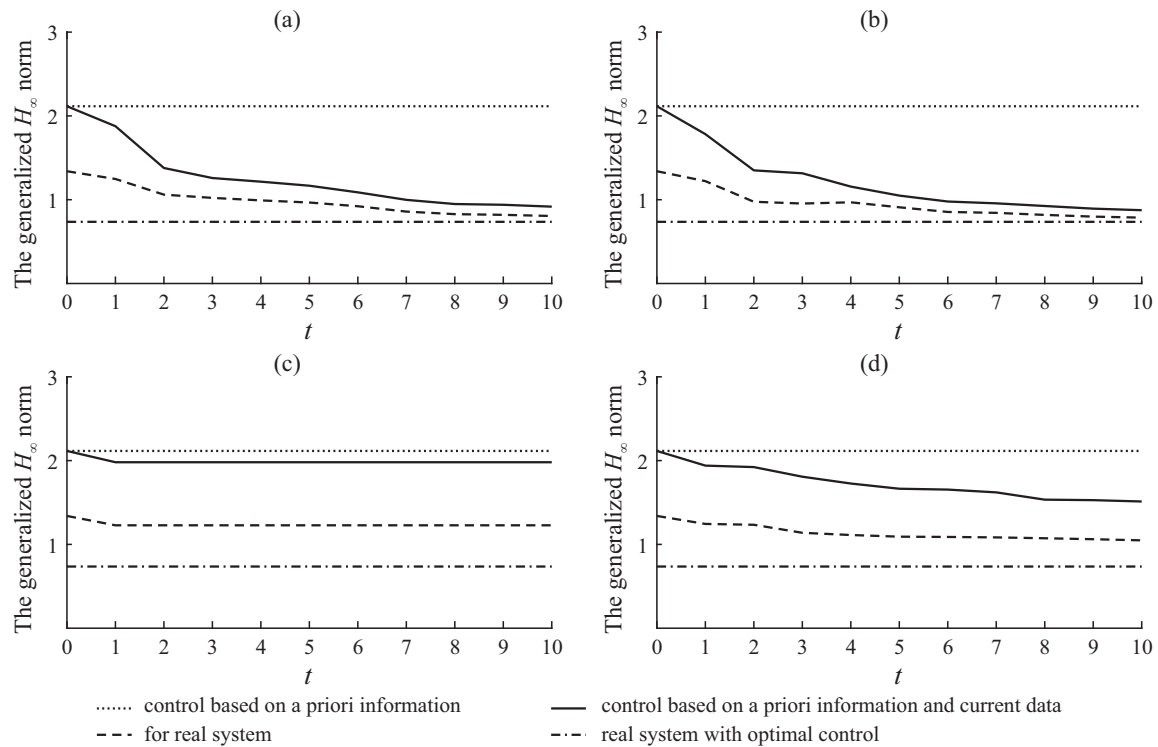


Fig. 2. The evolution of the guaranteed generalized H_∞ norms using different information with $\varepsilon_\xi = 0.01$ in the following cases: (a) $\varepsilon_w = 0.01$ and $\varepsilon_0 = 2$, (b) $\varepsilon_w = 0.01$ and $\varepsilon_0 = 5$, (c) $\varepsilon_w = 0.1$ and $\varepsilon_0 = 2$, and (d) $\varepsilon_w = 0.1$ and $\varepsilon_0 = 5$.

the elasticity and damping coefficients of the elastic and dissipative elements, are the same. Under this assumption, the dynamics of the structure under consideration (in dimensionless variables and parameters) are described by the system of differential equations

$$\begin{aligned}\ddot{x}_1 &= -\beta(2\dot{x}_1 - \dot{x}_2) - (2x_1 - x_2) + w, \\ \ddot{x}_2 &= -\beta(2\dot{x}_2 - \dot{x}_1 - \dot{x}_3) - (2x_2 - x_1 - x_3) + u + w, \\ \ddot{x}_3 &= -\beta(\dot{x}_3 - \dot{x}_2) - (x_3 - x_2) + w, \\ z &= -x_1 - \beta\dot{x}_1 + \alpha u\end{aligned}\tag{7.1}$$

with the following notation: $x_i, i = 1, 2, 3$, are the coordinates of the material points relative to the moving foundation; u is the damping force generated by the active vibration protection system installed in the building; w is the acceleration of the foundation, specified up to the sign; z is the scalar performance output defining the maximum force counteracting the displacement of the elastic system relative to the base; finally, β is the damping parameter. The vibration isolation problem is to find a controller minimizing z in some sense.

Let us pass from the continuous-time system to a discrete one with a discretization step of $h = 0.5$. We set the following numerical values of the parameters: $\alpha = 0.1$, $\beta = 0.1$, and $R = 0.01I$. The center of the matrix sphere $\Delta^{(a)}$ consists of the discrete model matrices of system (7.1) with $\beta = 0.2$ and inequality (4.1) holds for $\rho = 0.07$. The state measurement noise, disturbance, and initial state were chosen randomly on the spheres of radii ε_ξ , ε_w , and ε_0 , respectively; the control measurements were noiseless, i.e., $\varepsilon_\eta = 0$. In total, the uncertain system (7.1) in canonical form contains 49 unknown parameters.

Figure 2 presents the following evolutions in the case $\varepsilon_\xi = 0.01$: the guaranteed generalized H_∞ norm γ_t^* of the uncertain system (7.1) under the controller $u_t = \Theta_t x_t$ based on the a priori informa-

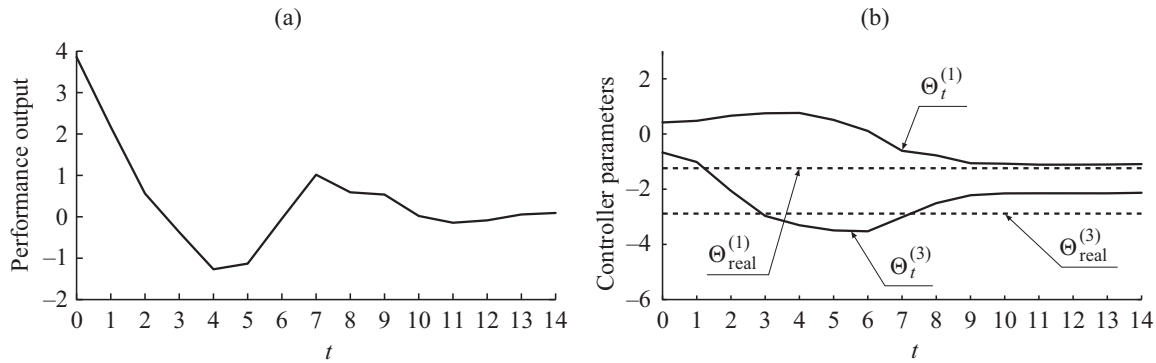


Fig. 3. The evolution of (a) the performance output and (b) two tunable parameters for the real system under $u_t = \Theta_t x_t$.

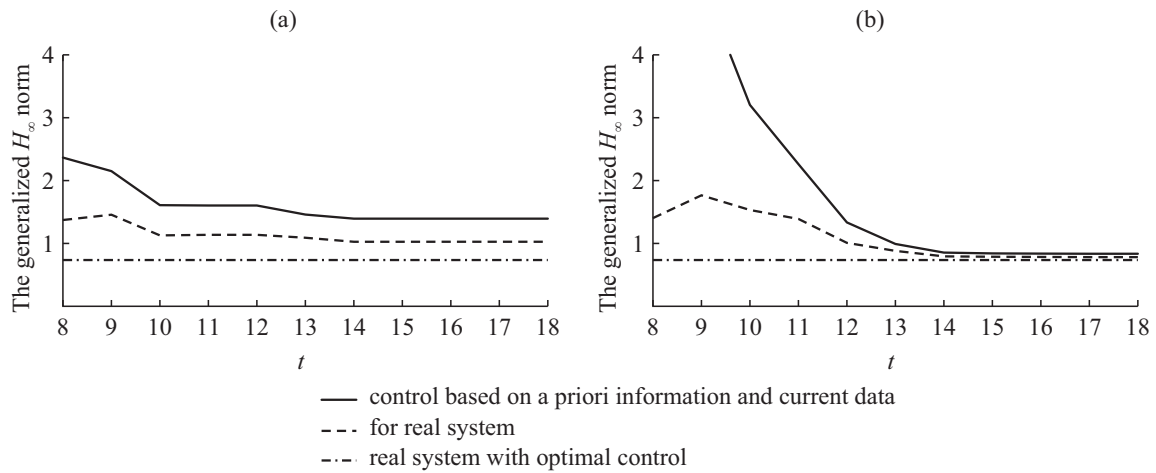


Fig. 4. The evolution of the guaranteed generalized H_∞ norm for $\varepsilon_\xi = 0.01$ and $\varepsilon_w = 0.1$ in the case of no common robust controller based on a priori information: (a) $\varepsilon_0 = 3$ and (b) $\varepsilon_0 = 5$.

tion and current data (solid curve); the generalized H_∞ norm $\|H\|_{g\infty}(\Omega_{real}, \Theta)$ for the closed-loop system consisting of the real system and the time-invariant state-feedback controller with $\Theta = \Theta_t$ (the dotted curve); the guaranteed generalized H_∞ norm of the uncertain system (7.1) with the robust controller $u_t = \Theta_0 x_t$ obtained based on only the a priori information (the dashed line); finally, the optimal H_∞ norm under the optimal controller for the real closed-loop system (the dash-and-dot line). Direct comparison of cases (a) and (b) or (c) and (d) in this figure shows that an increase in the initial state “increases” the block Ψ_{11} and, accordingly, decreases the matrix ellipsoid $\widehat{\Delta}_k(\mu, \tau_0^{k-1})$, ultimately accelerating the convergence of the generalized H_∞ norm to the minimum value. Comparing cases (a) and (b) with (c) and (d), we observe that a higher value of the disturbance amplitude slows down the convergence rate of the index, up to a possible stop. The reason lies in expanding the set of models consistent with the obtained data. Next, Fig. 3 illustrates typical trajectories of the performance output and two of the six control parameters based on the a priori information and current data for the real system.

According to Fig. 4, even when the a priori information with $\rho = 0.2$ is inaccurate and the guaranteed generalized H_∞ norm of the uncertain system (7.1) with the robust controller $u_t = \Theta_0 x_t$ obtained based on only the a priori information cannot be determined, the current information turns out to be sufficient for inequality (6.1) to become solvable, by the eighth step for $\varepsilon_0 = 3$ and by the tenth step for $\varepsilon_0 = 5$, and the controller is quickly retuned.

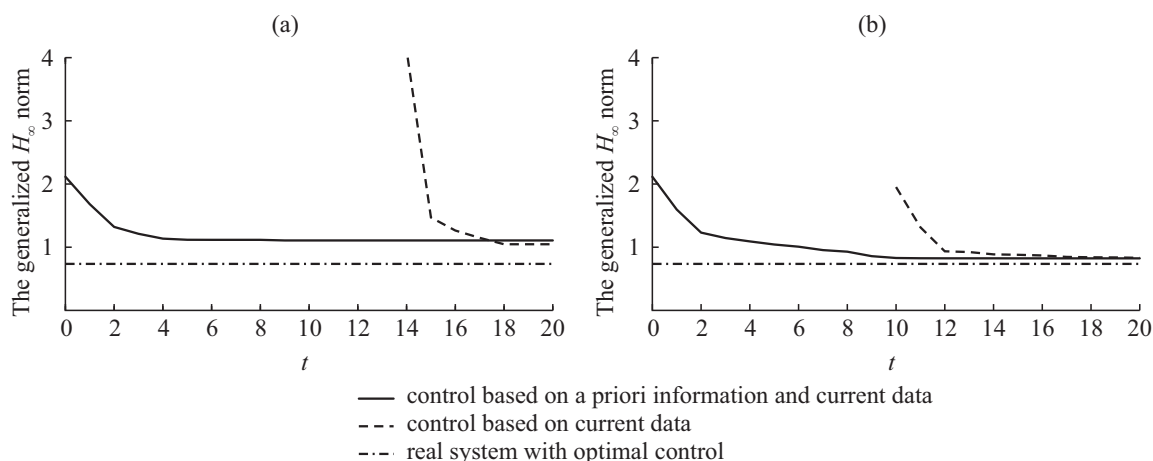


Fig. 5. The evolution of the guaranteed generalized H_∞ norms using both the a priori information and current data and only current data for $\varepsilon_\xi = 0.01$ and $\varepsilon_w = 0.1$: (a) $\varepsilon_0 = 3$ and (b) $\varepsilon_0 = 5$.

For the sake of comparison, Fig. 5 shows the evolution of the guaranteed generalized H_∞ norms when using both the a priori information and current data (the solid curve) and only the current data (the dotted curve). Clearly, the a priori information plays a positive role in the control design: the controller based on only the current data “turns on” merely at steps 14 or 10, whereas the controller based on the combined information provides an acceptable system performance level from the first step.

8. CONCLUSIONS

This paper has developed a new control design method for an unknown dynamic system based on a priori information and noisy current data obtained during the real-time operation of the system. At each step, the gain matrix is found by solving linear matrix inequalities to minimize the upper bound of the generalized H_∞ norm of the closed-loop system for all models consistent with the data obtained by this step. As has been proven, the sequence of the guaranteed values of the corresponding norm obtained in this way is monotonically nonincreasing and, when the tuning procedure stops, the resulting controller ensures a certain performance level of the closed-loop system. A distinctive feature of this approach is that the persistent excitation condition is not required for system identification. Furthermore, unlike adaptive control (where the main issue is to establish the convergence of the tunable parameters to their true values, and the performance of the resulting system cannot be assessed), it is possible to track the value of the optimized functional in real time. The numerical simulation of the active protection of buildings against seismic loads has demonstrated the effectiveness of the proposed approach.

FUNDING

This work was supported by the Russian Science Foundation, project no. 24-11-20023.

APPENDIX

Proof of Lemma 4.1. For matrices Ω consistent with the data $\hat{x}_t, \hat{x}_{t+1}$ and $\hat{u}_t$, equality (4.4) holds for some disturbances and noises satisfying the constraints (2.3). It follows that

$$\begin{aligned}
 & (\hat{x}_{t+1} - \Omega \hat{\varphi}_t)(\hat{x}_{t+1} - \Omega \hat{\varphi}_t)^T \\
 &= (I \ \Omega \ B_w) \begin{pmatrix} \xi_{t+1} \\ \zeta_t \\ w_t \end{pmatrix} \begin{pmatrix} \xi_{t+1} \\ \zeta_t \\ w_t \end{pmatrix}^T (I \ \Omega \ B_w)^T \preceq d^2 (I \ \Omega \ B_w) (I \ \Omega \ B_w)^T.
 \end{aligned} \tag{A.1}$$

Writing this condition equivalently as

$$\Omega(\hat{\varphi}_t \hat{\varphi}_t^T - d^2 I) \Omega^T - \Omega \hat{\varphi}_t \hat{x}_{t+1}^T - \hat{x}_{t+1} \hat{\varphi}_t^T \Omega^T + \hat{x}_{t+1} \hat{x}_{t+1}^T - (I + B_w B_w^T) d^2 \preceq 0,$$

we obtain (4.5). In turn, if the inequality in (A.1) is valid, equality (4.4) will hold [13] for some disturbance and noises satisfying the constraints (2.3).

Proof of Lemma 5.1. Writing inequality (5.1) as

$$\Omega \Psi_{11} \Omega^T + \Omega \Psi_{12} + \Psi_{12}^T \Omega^T + \Psi_{22} \preceq 0$$

and completing the square, we bring it to the form (5.4). Since $\Omega_{real} \in \widehat{\Delta}_k(\mu, \tau_0^{k-1})$, this set is non-empty and, consequently, $\Gamma_k \succeq 0$. Multiplying (5.4) by a^T and a on the left and right, respectively, and considering $\Psi_{11} > 0$, we arrive at

$$|(\Omega + \Psi_{12}^T \Psi_{11}^{-1})^T a|^2 \leq \lambda_{\min}^{-1}(\Psi_{11}) \lambda_{\max}(\Gamma_k) \quad \forall a : |a| = 1$$

$$\Rightarrow \|(\Omega + \Psi_{12}^T \Psi_{11}^{-1})\|^2 \leq \lambda_{\min}^{-1}(\Psi_{11}) \lambda_{\max}(\Gamma_k),$$

i.e., the set $\widehat{\Delta}_k(\mu, \tau_0^{k-1})$ is bounded.

Proof of Theorem 6.1. Let inequality (6.1) be solvable for some $P_k = P_k^T \succ R$, Q_k , $\mu \geq 0$, and $\tau_0^{k-1} \geq 0$. We replace $Q_k = \Theta P_k$ there and apply the Schur complement lemma to the negative definite block in the first row and first column to get the equivalent inequality

$$\begin{pmatrix} \Upsilon - \Psi_{11} & \star & \star \\ -\Psi_{21} & -\Psi_{22} - P_k + B_w B_w^T & \star \\ (C \ D) \Upsilon & 0 & (C \ D) \Upsilon (C \ D)^T - \gamma^2 I \end{pmatrix} \prec 0,$$

$$\Upsilon = \begin{pmatrix} I \\ \Theta \end{pmatrix} P_k \begin{pmatrix} I \\ \Theta \end{pmatrix}^T.$$

Direct substitution shows that the last inequality is true if and only if, for all nonzero $x_t^{(a)}$, $w_t^{(a)}$, and w_t^Δ , the function $V_k(x^{(a)}) = x^{(a)T} P_k x^{(a)}$ with $P_k \succ R$ satisfies the inequality

$$\Delta V_k(x_t^{(a)}) + |z_t^{(a)}|^2 - \gamma^2 |w_t^{(a)}|^2 - \begin{pmatrix} w_t^\Delta \\ x_t^{(a)} \end{pmatrix}^T \Psi(\mu, \tau_0^{k-1}) \begin{pmatrix} w_t^\Delta \\ x_t^{(a)} \end{pmatrix} < 0; \quad (\text{A.2})$$

here, $\Delta V_k(x_t^{(a)}) = V_k(x_{t+1}^{(a)}) - V_k(x_t^{(a)})$ denotes the increment of the function $V_k(x^{(a)})$ along the trajectory of the system

$$x_{t+1}^{(a)} = \begin{pmatrix} I \\ \Theta \end{pmatrix}^T w_t^\Delta + \begin{pmatrix} I \\ \Theta \end{pmatrix}^T (C \ D)^T w_t^{(a)},$$

$$z_t^{(a)} = B_w^T x_t^{(a)}.$$
(A.3)

Further, letting $w_t^\Delta = \Omega^T x_t^{(a)}$ in (A.3) yields

$$x_{t+1}^{(d)} = \begin{pmatrix} I \\ \Theta \end{pmatrix}^T \Omega^T x_t^{(d)} + \begin{pmatrix} I \\ \Theta \end{pmatrix}^T (C \ D)^T w_t^{(d)},$$

$$z_t^{(d)} = B_w^T x_t^{(d)},$$
(A.4)

and these equations coincide with those of the dual system for the original one

$$\begin{aligned} x_{t+1} &= \Omega \begin{pmatrix} I \\ \Theta \end{pmatrix} x_t + B_w w_t, \\ z_t &= (C \ D) \begin{pmatrix} I \\ \Theta \end{pmatrix} x_t. \end{aligned} \quad (\text{A.5})$$

Hence, for all nonzero variables $x_t^{(d)}$ and $w_t^{(d)}$ of system (A.4), the function $V_k(x^{(d)}) = x^{(d)\top} P_k x^{(d)}$ with $P_k \succ R$ satisfies the inequalities

$$\Delta V_k(x_t^{(d)}) + |z_t^{(d)}|^2 - \gamma^2 |w_t^{(d)}|^2 - x_t^{(d)\top} \begin{pmatrix} \Omega I \\ I \end{pmatrix}^\top \Psi(\mu, \tau_0^{k-1}) \begin{pmatrix} \Omega I \\ I \end{pmatrix} x_t^{(d)} < 0. \quad (\text{A.6})$$

Since we have $(\Omega I) \Psi(\mu, \tau_0^{k-1}) (\Omega I)^\top \preceq 0$ for all $\Omega \in \widehat{\Delta}_k(\mu, \tau_0^{k-1})$, the function $V_k(x^{(d)}) = x^{(d)\top} P_k x^{(d)}$ with $P_k \succ R$ satisfies the inequalities

$$\Delta V_k(x_t^{(d)}) + |z_t^{(d)}|^2 - \gamma^2 |w_t^{(d)}|^2 < 0 \quad (\text{A.7})$$

for all nonzero variables $x_t^{(d)}$ and $w_t^{(d)}$ and all $\Omega \in \widehat{\Delta}_k(\mu, \tau_0^{k-1})$. Due to Lemma 3.2, it follows that $u_t = \Theta x_t$ is the guaranteed generalized H_∞ -controller with the level γ .

Conversely, let $u_t = \Theta x_t$ be the guaranteed generalized H_∞ -controller with a level γ . By the definition and Lemma 3.2, for the dual system (A.4), there exist a function $V_k(x^{(d)}) = x^{(d)\top} P_k x^{(d)}$ with $P_k \succ R$ and numbers $\widehat{\mu} \geq 0$ and $\widehat{\tau}_0^{k-1} \geq 0$ such that inequality (A.7) holds for all $\Omega \in \widehat{\Delta}_k(\widehat{\mu}, \widehat{\tau}_0^{k-1})$. We show that along the trajectory of system (A.3), the function $V_k(x^{(a)}) = x^{(a)\top} P_k x^{(a)}$ with $P_k \succ R$ satisfies the inequality

$$\Delta V_k(x_t^{(a)}) + |z_t^{(a)}|^2 - \gamma^2 |w_t^{(a)}|^2 < 0 \quad (\text{A.8})$$

for all nonzero $x_t^{(a)}$, $w_t^{(a)}$, and w_t^Δ such that

$$\begin{pmatrix} w_t^\Delta \\ x_t^{(a)} \end{pmatrix}^\top \Psi(\widehat{\mu}, \widehat{\tau}_0^{k-1}) \begin{pmatrix} w_t^\Delta \\ x_t^{(a)} \end{pmatrix} \leq 0. \quad (\text{A.9})$$

Indeed, we choose $x_t^{(d)} = x_t^{(a)}$ and $w_t^{(d)} = w_t^{(a)}$ and, for each w_t^Δ satisfying (A.9), define Ω as the solution of the linear matrix equation $x_t^{(a)\top} \Omega = w_t^{\Delta\top}$, containing one equation for each column of the matrix Ω . In this case, from (A.9) it follows that $x_t^{(a)\top} (\Omega I) \Psi(\widehat{\mu}, \widehat{\tau}_0^{k-1}) (\Omega I)^\top x_t^{(a)} \leq 0$, i.e., $\Omega \in \widehat{\Delta}_k(\widehat{\mu}, \widehat{\tau}_0^{k-1})$. Thus, for such Ω , equations (A.3) coincide with equations (A.4), and inequality (A.8) coincides with inequality (A.7).

By the losslessness of the S -procedure with one constraint, inequality (A.8) under the constraint (A.9) is equivalent to the inequality

$$\Delta V_k(x_t^{(a)}) + |z_t^{(a)}|^2 - \gamma^2 |w_t^{(a)}|^2 - \nu \begin{pmatrix} w_t^\Delta \\ x_t^{(a)} \end{pmatrix}^\top \Psi(\widehat{\mu}, \widehat{\tau}_0^{k-1}) \begin{pmatrix} w_t^\Delta \\ x_t^{(a)} \end{pmatrix} < 0$$

for some $\nu \geq 0$. Note that by assumption, Ω_{real} lies strictly inside $\widehat{\Delta}_k(\widehat{\mu}, \widehat{\tau}_0^{k-1})$, so inequality (A.9) is strict for $w_t^\Delta = \Omega_{real}^\top x_t^{(a)}$. Since $\Psi(\mu, \tau_0^{k-1})$ linearly depends on μ and τ_0^{k-1} , denoting $\nu \widehat{\mu} = \mu$ and $\nu \widehat{\tau}_0^{k-1} = \tau_0^{k-1}$, we obtain inequality (A.2), which is equivalent to the LMI (6.1) for $Q_k = \Theta P_k$.

Proof of Theorem 6.2. Similar to the proof of Theorem 6.1 for $k = N - 1$, we obtain that along the trajectory of system (A.4), the function $V_{N-1}(x_t^{(d)}) = x_t^{(d)T} P_{N-1} x_t^{(d)}$ with $P_{N-1} > R$ satisfies inequality (A.6) for all $\Omega \in \widehat{\Delta}_{N-1}(\mu, \tau_0^{N-1})$. Consequently, by Lemmas 3.1 and 3.2 and Remark 3.1, along the trajectory of system (2.1) with $u_t = \Theta_{N-1} x_t$, the function $V_{N-1}(x_t) = x_t^T Y_{N-1} x_t$, where $Y_{N-1} = \gamma_{N-1}^{*2} P_{N-1}^{-1}$, satisfies the inequalities

$$\Delta V_{N-1}(x_t) + |z_t|^2 - \gamma_{N-1}^{*2} |w_t|^2 < 0$$

for all $\Omega \in \widehat{\Delta}_{N-1}(\mu, \tau_0^{N-1})$ and $t \geq N - 1$. Summing these inequalities starting from $t = N - 1$ and considering the relations $\lim_{t \rightarrow \infty} V_{N-1}(x_t) = 0$ and $Y_{N-1} < \gamma_{N-1}^{*2} R^{-1}$, we finally arrive at (6.3).

REFERENCES

1. Polyak, B.T. and Shcherbakov, P.S., *Robustnaya ustoychivost' i upravlenie* (Robust Stability and Control), Moscow: Nauka, 2002.
2. Petersen, I.R. and Tempo, R., Robust Control of Uncertain Systems: Classical Results and Recent Developments, *Automatica*, 2014, vol. 50, no. 5, pp. 1315–1335.
3. Markovsky, I., Huang, L., and Dörfler, F., Data-Driven Control Based on the Behavioral Approach: From Theory to Applications in Power Systems, *IEEE Control Syst. Mag.*, 2023, vol. 43, no. 5, pp. 28–68.
4. van Waarde, H.J., Eising, J., Camlibel, M.K., and Trentelman, H.L., The Informativity Approach: To Data-Driven Analysis and Control, *IEEE Control Syst. Mag.*, 2023, vol. 43, no. 6, pp. 32–66.
5. Willems, J.C., Rapisarda, P., Markovsky, I., and De Moor, B., A Note on Persistency of Excitation, *Syst. Control Lett.*, 2005, vol. 54, pp. 325–329.
6. De Persis, C. and Tesi, P., Formulas for Data-Driven Control: Stabilization, Optimality and Robustness, *IEEE Trans. Automat. Control*, 2020, vol. 65, no. 3, pp. 909–924.
7. van Waarde, H.J., Eising, J., Trentelman, H.L., and Camlibel, M.K., Data Informativity: A New Perspective on Data-Driven Analysis and Control, *IEEE Trans. Automat. Control*, 2020, vol. 65, no. 11, pp. 4753–4768.
8. Berberich, J., Koch, A., Scherer, C.W., and Allgower, F., Robust Data-Driven State-Feedback Design, *Proc. Amer. Control Conf.*, 2020, pp. 1532–1538.
9. Bisoffi, A., De Persis, C., and Tesi, P., Trade-Offs in Learning Controllers from Noisy Data, *Syst. Control Lett.*, 2021, vol. 154, art. no. 104985.
10. van Waarde, H.J., Camlibel, M.K., and Mesbahi, M., From Noisy Data to Feedback Controllers: Nonconservative Design via a Matrix S-Lemma, *IEEE Trans. Automat. Control*, 2022, vol. 67, no. 1, pp. 162–175.
11. Bisoffi, A., De Persis, C., and Tesi, P., Data-Driven Control via Petersen's Lemma, *Automatica*, 2022, vol. 145, art. no. 110537.
12. Nortmann, B. and Mylvaganam, T., Direct Data-Driven Control of LTV Systems, *IEEE Trans. Automat. Control*, 2023, vol. 68, pp. 4888–4895.
13. Bisoffi, A., De Persis, C., and Monshizadeh, N., Controller Synthesis for Input-State Data with Measurement Errors, *IEEE Control Syst. Lett.*, 2024, vol. 8, pp. 1571–1576.
14. Berberich, J., Scherer, C.W., and Allgower, F., Combining Prior Knowledge and Data for Robust Controller Design, *IEEE Trans. Automat. Control*, 2023, vol. 68, no. 8, pp. 4618–4633.
15. Kogan, M.M. and Stepanov, A.V., Design of Suboptimal Robust Controllers Based on A Priori and Experimental Data, *Autom. Remote Control*, 2023, vol. 84, no. 8, pp. 918–932.
16. Li, L., De Persis, C., Tesi, P., and Monshizadeh, N., Data-Based Transfer Stabilization in Linear Systems, *IEEE Trans. Automat. Control*, 2024, vol. 69, pp. 1866–1873.

17. Kogan, M.M. and Stepanov, A.V., Design of Generalized H_∞ -Suboptimal Controllers Based on Experimental and A Priori Data, *Autom. Remote Control*, 2024, vol. 85, no. 1, pp. 1–14.
18. Niknejad, N. and Modares, H., Physics-Informed Data-Driven Safe and Optimal Control Design, *IEEE Control Syst. Lett.*, 2024, vol. 8, pp. 285–290.
19. Kogan, M.M. and Stepanov, A.V., How to Improve Robust Control of a Linear Time-Varying System by Using Experimental Data, *Autom. Remote Control*, 2024, vol. 85, no. 6, pp. 636–654.
20. Kogan, M.M. and Stepanov, A.V., Robust H_∞ Control with Transients for LTV Systems Based on Prior Knowledge and Data, *Automatica*, 2025, vol. 173, art. no. 112083.
21. Balandin, D.V., Biryukov, R.S., and Kogan, M.M., Pareto Suboptimal H_∞ Controls with Transients, *Proc. Eur. Control Conf.*, 2021, Rotterdam, pp. 542–547.
22. Boyd, S., El Ghaoui, L., Feron, E., and Balakrishnan, V., *Linear Matrix Inequalities in System and Control Theory*, Philadelphia: SIAM, 1994.
23. Petersen, I.R. and McFarlane, D.C., Optimal Guaranteed Cost Control and Filtering for Uncertain Linear Systems, *IEEE Trans. Automat. Control*, 1994, vol. 30, no. 9, pp. 1971–1977.
24. Yakubovich, V.A., S -Procedure in Nonlinear Control Theory, *Vestn. Leningrad. Univ. Mat.*, 1977, no. 4, pp. 73–93.

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

An Overview of Conflict Prevention Methods in Air Traffic Control Using Deep Reinforcement Learning

E. L. Kulida^{*,a} and V. G. Lebedev^{*,b}

**Trapeznikov Institute of Control Sciences, Russian Academy of Sciences, Moscow, Russia*
e-mail: ^aelena-kulida@yandex.ru, ^blebedev-valentin@yandex.ru

Received March 4, 2025

Revised March 17, 2025

Accepted March 25, 2025

Abstract—An overview of the development of modern approaches to conflict prevention between aircraft based on deep reinforcement learning is given. The basic concept of reinforcement learning and some fundamental algorithms used for aircraft conflict prevention are reviewed. Models with discrete and continuous actions for conflict prevention in two-dimensional and three-dimensional airspaces during flight along fixed trajectories or in free flight are presented. Various approaches to representing information about the state of the airspace (using a state vector and as a graph) and different types of interaction between aircraft (based on information about the state of surrounding aircraft or through message exchange) are considered.

Keywords: air traffic management, conflict prevention, aircraft maneuver, deep reinforcement learning

DOI: 10.31857/S0005117925090041

1. INTRODUCTION

Maintaining a safe separation distance, both vertically and horizontally, between any two aircraft at any time is the most important function of an air traffic control system [1]. Loss of separation between aircraft is called a conflict. The growing density of air traffic leads to an increase in the number of potential conflicts, so prevention methods are important to reduce the risk of collisions. Strategic prevention of potential conflicts based on the global optimization model that generates conflict-free four-dimensional trajectories for all aircraft in advance fails to manage uncertainty arising in the dynamics of flights in real time [2]. Tactical real-time prevention of potential conflicts is critical for ensuring safe air traffic control since it allows for better management of uncertainty that arises in flight dynamics.

At present, aircraft mainly move along fixed trajectories, with conflict prevention being the responsibility of air traffic controllers. Tactical decisions are still made by air traffic controllers with very few changes as compared to decisions made 50 years ago [3]. With the increasing intensity of air traffic, the workload of air traffic controllers is constantly growing and may exceed human capabilities. The development of existing methods for assessing the dynamic air situation by an air traffic controller in order to reduce risk factors, including information overload and lack of time for functional operations, is considered in [4]. In [5–7], a conception is proposed for the development of automation tools to increase the capacity and safety of airspace operation in order to effectively and intelligently support air traffic controllers' decision-making to prevent conflicts. The prospective air traffic management involves using free flights when aircraft will move along arbitrary trajectories and conflict prevention will be ensured by an autonomous air traffic control

system. Theoretical studies confirm that free flight can potentially increase safety [8] and reduce fuel consumption [9]. To implement the free flight concept, one needs to “ensure control over aircraft separation using on-board systems in addition to ground systems. The higher reliability of such a structure will make it possible to clarify a number of scenarios for simulating the risk of aircraft collision and will contribute to the creation of safer, more flexible, and more capacious conditions for air traffic management” [10]. The challenge to develop decentralized autonomous conflict prevention tools is of fundamental importance for the implementation of the free flight concept [11]. Methods and algorithms for detecting and preventing dangerous approaches in the air within the framework of a promising air traffic management system, taking into account flight safety and efficiency requirements, were studied in [12, 13].

Methods have been developed to provide air traffic controllers with recommendations on conflict resolution based on optimal control [14] and mathematical programming [15–18], geometric optimization [19–21], evolutionary algorithms [22], and the Monte Carlo tree search algorithm [23]. These methods work with the existing air traffic density; however, when it comes to higher density, their insufficient computational efficiency becomes an issue. It takes them tens or even hundreds of seconds to come up with a decision. Taking into account the uncertainty inherent in air traffic significantly increases the computation time, making the methods less capable of quickly generating decisions. Most conventional approaches to ensuring separation fail to cope with stochastic environments and high air traffic density [24]. New approaches are needed that can effectively respond to the dynamics of the external environment in real time, for example, based on neural networks and machine learning [25–27].

Recently, deep reinforcement learning has been widely used in various fields of aviation since it can solve decision-making problems unavailable previously due to a combination of nonlinearity and high dimensionality [28]. The use of deep reinforcement learning will automatically provide safe and effective conflict prevention decisions to support air traffic controllers’ decision-making and reduce their workload [29]. In the future, a fully automated control system will become the ultimate solution for handling high-density, complicated, and dynamic air traffic [30].

Reinforcement learning methods are applied in two stages, viz. the stage of training the model and the stage of applying the trained model in practice. While it can take long to train a model, the trained model helps generate decisions very quickly. The decision-making speed is an indicator of the efficiency and advantage of deep reinforcement learning as compared to conventional algorithms. Faster decision-making means earlier detection of conflicts and formulation of instructions to reduce the workload of air traffic controllers and pilots. In [31], the following figures are given (with the proviso that the data were obtained under different computation conditions)—it takes a mixed-integer linear programming algorithm 49 seconds to generate a decision, the average decision time using a genetic algorithm is 37.6 seconds, and a trained deep reinforcement learning agent requires less than 0.2 seconds. Reinforcement learning methods have a clear advantage over conventional methods in the speed of computing decisions and the ability to adapt to dynamics of the external environment, which is critically important when resolving air traffic conflicts.

Research on conflict resolution in air traffic using deep reinforcement learning has been continuously conducted since 2018, with many models and algorithms proposed during this period [31]. Conflict resolution models for both air route movement and free flight as well as models with both discrete and continuous actions are considered. At present, in most cases, two-dimensional models are proposed for conflict resolution using horizontal maneuvers, three-dimensional models for horizontal and vertical maneuvers being proposed much less frequently [32]. Two-dimensional models ignore vertical maneuvers due to the potential instability they can cause in air traffic [33]. The number of conflicting aircraft varies—conflicts between two aircraft and conflicts in groups of aircraft with a fixed or variable number of aircraft are considered. In [34–36], a hybrid approach

has been developed that combines the strengths of geometric and reinforcement learning methods to resolve conflicts. It is argued that the wide range of different optimal decisions found by the reinforcement learning method shows that the rules of the geometric method should be expanded to take into account different conflict geometries. In Russia, the problem of aircraft traffic control based on reinforcement learning is currently being actively studied by State Scientific Research Institute of Aviation Systems (GosNIIAS) experts, with a significant scientific and technical reserve already made [37].

Given a wide variety of conflict resolution problem statements, when constructing a reinforcement learning model, in many cases, one first develops an interactive environment for an agent to learn various strategies. An artificial intelligence agent learns using deep reinforcement learning algorithms through trial and error, applying various possible actions and receiving feedback from the environment in the form of rewards. The agent's goal is to study an action selection strategy that will maximize the mathematical expectation of the total discounted rewards over a long period of time. The convergence of the reinforcement learning model to the desired outcome is determined by the choice of the reward function, through which the agent learns to optimize the strategy for choosing actions in various situations. The reward function has an impact on the learning rate, convergence, and performance of agents. The principal thing that should be considered in the reward function is the success or failure of conflict resolution. Moreover, to increase the efficiency of the model, the reward should take into account the number of maneuvers and the time required to resolve the conflict. Using the reward function, it is possible to bring the agent's behavior closer to the existing rules for conflict resolution by air traffic controllers [38, 39].

Despite the success of reinforcement learning in research on solving air traffic control problems, two significant challenges remain for this method to be applied in real conditions. The first problem is the vulnerability of deep neural networks to adversarial attacks, the second is the problem of the explainability of "black box" models, viz. pilots and air traffic controllers do not understand how the models make certain decisions [40, 41]. In [42], approaches to resolving these issues in the autonomous resolution of aircraft conflicts are proposed.

2. REINFORCEMENT LEARNING

2.1. Basic Conception [43]

Machine learning is divided into supervised learning, unsupervised learning, and reinforcement learning. Supervised learning allows approximating any function; however, it requires labeled data sets to be available, which is not always so. Unsupervised learning relies on sets of unlabeled data. Reinforcement learning allows implementing a consistent decision-making process through trial and error, with the learning data synthesized as the agent interacts with the environment. When using reinforcement learning, each aircraft is simulated as an interactive agent whose actions are conflict prevention maneuvers.

Reinforcement learning is based on the Markov model of the decision-making process, in which the state of the system and the actions of the agent do not depend on how the system came to this state. The Bellman equations hold for the Markov decision-making process. The basis of the Markov model of the decision-making process is the environment and the agent operating in it. The environment is characterized by a set of parameters, and the state of the environment s is a specific set of values of these parameters. An agent is a program that can analyze the state of the environment and perform a specific set of actions $a \in A(s)$ in each state. As a result of the agent's action, the environment switches from the state s to the new state s' and receives feedback from the environment in the form of the reward $r = R(s, a, s')$. In the case of multistep interaction of the agent with the environment, starting from the state s at the step t to the end of the episode at

the step T , the total discounted benefit

$$G_t = r_{t+1} + \gamma r_{t+2} + \gamma^2 r_{t+3} + \cdots + \gamma^{T-1} r_T = r_{t+1} + \gamma G_{t+1}$$

is determined, and $\gamma \in [0, 1]$ is the discount coefficient, which specifies the decrease of the value of the reward received at later steps.

The transition function $p(s'|s, a)$ specifies the probability of transition to the state s' at the step t given that the action a was chosen in the state s at the step $t - 1$

$$p(s'|s, a) = P(S_t = s' | S_{t-1} = s, A_{t-1} = a), \quad \sum_{s' \in S} p(s'|s, a) = 1, \quad \forall s \in S, \quad \forall a \in A(s).$$

A strategy (or a policy) is a function $\pi(a|s)$ that matches the action of the agent with each non-terminal state of the environment.

The expected benefit when an agent follows the strategy π in the state s is called the state value function

$$V_\pi(s) = E_\pi[r_{t+1} + \gamma G_{t+1} | S_t = s].$$

However, the state value function does not allow us to know the expected benefit from the agent performing the action a in the state s when it follows the strategy π ; it is specified by the action value function

$$Q_\pi(s, a) = E_\pi[r_{t+1} + \gamma G_{t+1} | S_t = s, A_t = a].$$

The basic concept of reinforcement learning—generalized iteration with respect to strategies—is an iterative procedure. The step of this procedure involves two processes, viz. evaluating the current strategy to refine the current approximation of the value function followed by improving the strategy in accordance with the changed value function.

The value function can be represented as

$$V_\pi(s) = \sum_a \pi(a|s) \sum_{s', r} p(s', r|s, a) [r + \gamma V_\pi(s')], \quad \forall s \in S.$$

The iteration step to refine the value function is to calculate it for the action with the highest value

$$V_{k+1}(s) = \max_a \sum_{s', r} p(s', r|s, a) [r + \gamma V_k(s')].$$

The action that delivers the maximum value to the Q -function is called greedy. The strategy can be improved by using a strategy optimization algorithm, which consists in choosing a greedy action with respect to the Q -function

$$\pi'(s) = \operatorname{argmax}_a \sum_{s', r} p(s', r|s, a) [r + \gamma V_\pi(s')].$$

When studying the value function, it is very important to maintain a balance between choosing a greedy action and choosing a random action for exploration. There are many different approaches to solving this problem such as the epsilon-greedy strategy with a random action selected with the epsilon probability, the epsilon greedy strategy with decay with epsilon decreased as the agent learns, action selection strategies using knowledge gained up to the current step about the value and exploration degree of actions, etc.

The two processes considered stabilize when the value function matches the strategy, and the strategy is greedy with respect to the value function. The strategy and the value function are optimal if

$$V^*(s) = \max_\pi V_\pi(s).$$

2.2. Reinforcement Deep Learning Algorithms [44, 45]

For simple examples, the action value function is represented as a table. For important practical tasks, it is impossible to implement a table representation of value functions due to the large number

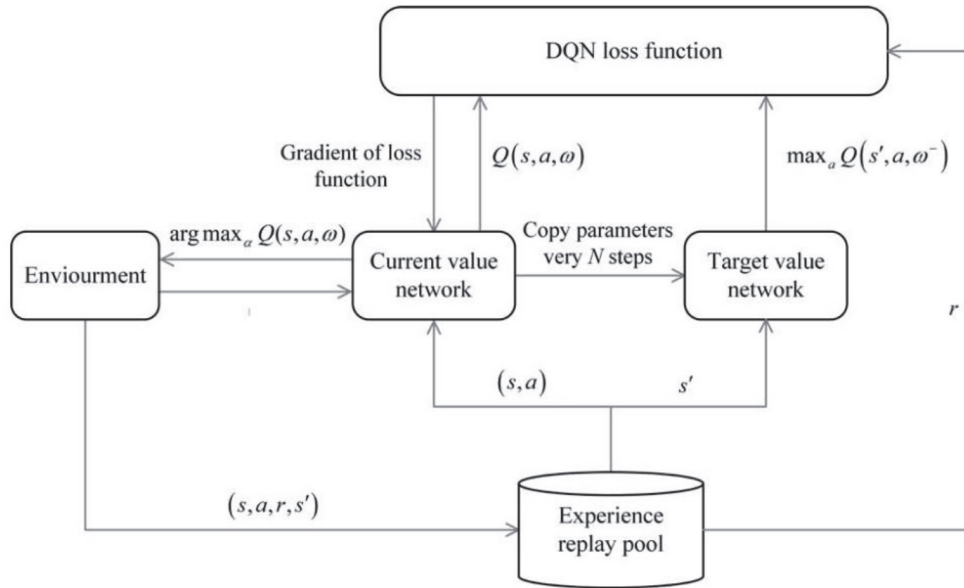


Fig. 1. The operation principle of the deep Q -learning algorithm.

of states, continuous variables or actions. This is the case when deep reinforcement learning is applied with deep learning algorithms with a teacher used to approximate value functions based on the samples (s, a, r, s') formed as the agent interacts with the environment.

At present, this approach is being rapidly developed in the field of aviation conflict prevention, and various algorithms for the value function approximation and strategy optimization are being proposed and studied. The deep Q -learning (deep Q -network, DQN) algorithm is widely used, which leverages two key technologies, viz. experience replay and the dual network structure. Figure 1 shows the operation principle of the DQN algorithm [46].

The experience replay consists in creating the replay buffer D that accumulates a large number of samples. The mini-sets $U(D)$ for network training are selected from the accumulated replay buffer uniformly and randomly and thus correspond to different trajectories and policies, increasing the network training stability. The dual network structure is based on using the same network with different sets of parameters. A dynamic network is used to approximate the current value $Q(s, a; \omega_i)$, and the parameters of this network ω_i are updated at each time step i . The target network is used to obtain a more stable target value Q , and the parameters of the target network ω^- are updated in N time steps. The loss function has the form

$$L_i(\omega_i) = E_{(s,a,r,s') \sim U(D)} \left[\left(r + \gamma \max_{a'} Q(s', a'; \omega^-) - Q(s, a; \omega_i) \right)^2 \right].$$

The loss function is optimized by the gradient descent method

$$\nabla_{\omega_i} L_i(\omega_i) = E_{(s,a,r,s') \sim U(D)} \left[\left(r + \gamma \max_{a'} Q(s', a'; \omega^-) - Q(s, a; \omega_i) \right) \nabla_{\omega_i} Q(s, a; \omega_i) \right].$$

The difference between the double deep Q -network (DDQN) method [47] and the DQN method is that the dynamic network parameters are used instead of the target network parameters to select an action in the equation.

$$\nabla_{\omega_i} L_i(\omega_i) = E_{(s,a,r,s') \sim U(D)} \left[\left(r + \gamma Q(s', \arg\max_{a'} Q(s', a'; \omega_i); \omega^-) - Q(s, a; \omega_i) \right) \nabla_{\omega_i} Q(s, a; \omega_i) \right].$$

The DQN and DDQN algorithms are used in conflict prevention models with a discrete action space [48–52].

Methods that use a continuous action space to resolve conflicts between aircraft are also proposed [34, 53–55]. In this case, strategies are represented by parameterized stochastic functions $\pi_\theta(a, s)$ that are optimized using actor-critic algorithms. In these algorithms, in addition to the neural network used to evaluate the strategy (the critic), a second neural network is used to form a strategy based on optimizing the value function (the actor).

For the trajectory $\tau = S_0, A_0, R_1, S_1, \dots, S_{T-1}, A_{T-1}, R_T, S_T$, the function $G(\tau)$ is the total discounted benefit, and $\pi_\theta(A_t|S_t)$ is the probability of choosing the action A_t in the state S_t at the step t . The actor's network is updated according to the gradient of the value function

$$\nabla_\theta E_{\tau \sim \pi_\theta}[G(\tau)] = E_{\tau \sim \pi_\theta} \left[\sum_{t=0}^T G_t(\tau) \nabla_\theta \log \pi_\theta(A_t|S_t) \right].$$

3. AIRCRAFT CONFLICT PREVENTION MODELS

3.1. Conflict Prevention Models for Two Aircraft

A conflict resolution strategy can include both two-dimensional (2D) maneuvers such as changing the course and speed in a plane airspace and three-dimensional (3D) maneuvers when there is also a change in altitude. Although a 2D model is not as effective in conflict resolution as a 3D model is, maneuvers in the two-dimensional airspace cause less discomfort for passengers and do not distort the vertically stratified structure of the airspace [56]. If they are too large, neural networks can be harmful to agent training due to an excessively big number of parameters. Therefore, smaller neural networks for 2D models have greater potential for the model to be expanded further to take into account more real-world factors [30].

The first studies on aircraft conflict prevention using reinforcement learning considered conflict resolution between two aircraft in a two-dimensional airspace. In one of the first works on aircraft conflict prevention, flights along routes were considered, with a hierarchical structure of deep reinforcement learning proposed [48]. The training environment used was the NASA Sector 33 software containing 35 air traffic control problems involving two to five aircraft. The hierarchical structure includes a parent agent designed to solve the problem of choosing aircraft routes at the beginning of the episode and the child agent that controls actions to change speeds on the selected routes. The hierarchical structure allows separating the route selection actions performed at the beginning of the episode and the speed control actions during the episode. A reinforcement learning algorithm based on a dual deep Q -network is used to train the agents. The first neural network (the target one) is used to select actions that are greedy with respect to the current Q -function, and the second neural network (the dynamic one) is used to adjust the Q -function based on evaluating the success of the actions performed. It was shown in [48] that a hierarchical deep reinforcement learning agent can choose optimal combinations of routes and speeds in order to avoid a conflict between two aircraft flying along routes.

One of the first works on reinforcement learning for conflict resolution in free flight deals with the case of two aircraft, with uncertainty taken into account [53]. An environment has been developed for simulating potential conflicts for agent training and testing. Figure 2 shows a conflict between two aircraft in a circular airspace with the radius R of 50 nautical miles and a maneuver to prevent it. The trajectory of the own aircraft is A_1B_1 , the trajectory of the intruder is A_2B_2 , and QP is the closest distance between the two aircraft at which they lose safe separation if none of them makes a maneuver. A single maneuver to change the course in the continuous two-dimensional space is used as a conflict prevention action. The maneuver A_1MNB_1 in Fig. 2 represents a series of actions performed by the own aircraft—deviating from the original trajectory at the point M by changing the course by the angle α , then moving along the vector MN , and turning to the point B_1 at the point N .

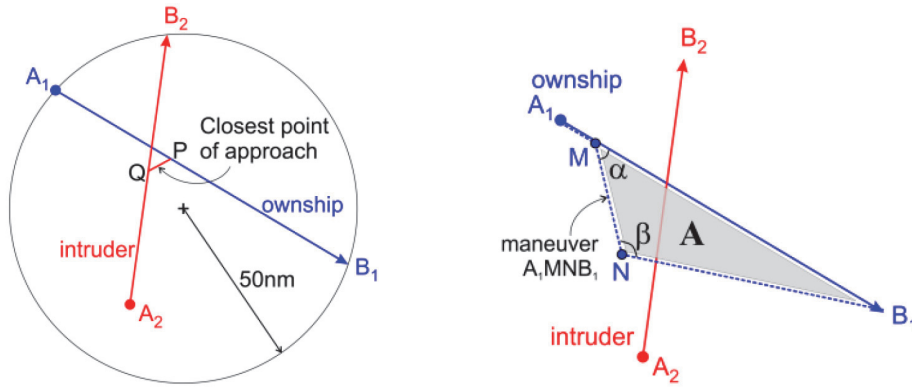


Fig. 2. The scenario of a conflict between two aircraft and a maneuver to prevent it.

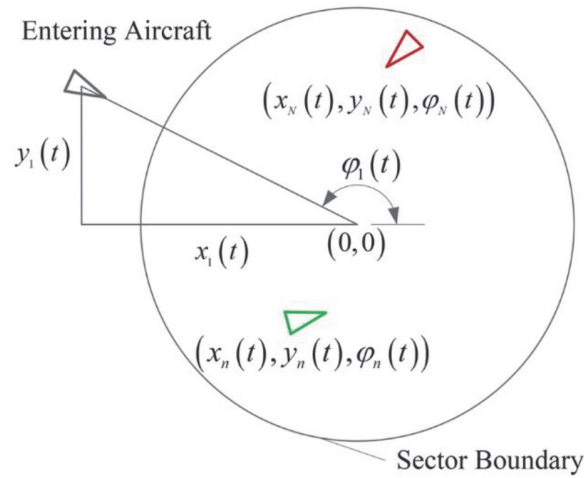


Fig. 3. Conflict scenario with several aircraft.

The reward of the agent being trained is calculated as the sum

$$R_{final} = 5 + R_{conflict} + R_{maneuver} + R_{deviation},$$

i.e., negative rewards are added to reward 5, viz. $R_{conflict} = -3$ if the maneuver fails to resolve the conflict, $R_{maneuver} = -2$ if it goes beyond the boundaries of the area or if the angle $\beta > 120^\circ$, $R_{deviation} = -S_{\Delta MNB_1}$, the deviation from the initial trajectory is estimated by the area between the trajectory of the maneuver and the initial trajectory.

The deep deterministic policy gradient (DDPG) method, which is one of the advanced methods of deep reinforcement learning for continuous action space control problems, is used [44]. The algorithm uses two neural networks, viz. the critic's network to study the utility function of state-action pairs $Q(s, a)$ and the actor's network to map the state into a deterministic action based on the policy gradient. The performance of the DDPG algorithm for preventing air traffic conflicts is close to the performance of conventional methods, yet the calculation time is significantly reduced [57].

In [54, 56, 58, 59], the conflict prevention problem statement is generalized to the case when there are several other aircraft in the area in addition to the own aircraft and the intruder. In this case, secondary potential conflicts (the domino effect) may arise when maneuvering to prevent the conflict [60].

In [54], a two-dimensional environment for simulating free flights was developed, which can be applied to several aircraft in a sector (no more than 5) (Fig. 3).

It is assumed that there are no conflicts between these aircraft, and conflicts can be caused by an aircraft entering the sector that aim to fly from the starting point to the end one in minimal time without collisions with other aircraft. An actor-critic algorithm with the fixed number K of actions (control cycles) is proposed. To avoid conflicts, actions are generated to change the heading angle of the entering aircraft. At each time step, the agent selects the action $A = \{\rho, \varphi | \rho \in [0, L], \varphi \in [-\pi, \pi]\}$ described by a two-dimensional polar coordinate, where ρ, φ are the polar radius and the angle.

The reward function

$$R_t = \begin{cases} -1 & \text{if there is a conflict,} \\ 1 - \frac{1}{K} \times \frac{|\Delta\varphi_t|}{\pi}, & \text{otherwise.} \end{cases}$$

The value function is approximated using the neural network $\hat{V}(S_t, \omega) \approx V_\pi(S)$, where ω are the weight of neurons.

For the critic's network, δ is specified

$$\delta_t = R_t + \gamma \hat{V}(S_{t+1}, \omega) - \hat{V}(S_t, \omega),$$

where R_t is the immediate reward, $\hat{V}(S_{t+1}, \omega)$ is the value of the value function of the next state, and $\hat{V}(S_t, \omega)$ is the value of the value function of the current state. The least squares method is used to update the parameters $\omega \leftarrow \omega + \alpha \nabla \delta^2$; α is the learning rate.

The policy gradient method is used for the actor's network. The equation

$$\ln \pi(\rho_t, \varphi_t | S_t, \theta) = \ln \pi(\rho_t | S_t, \theta) + \ln \pi(\varphi_t | S_t, \theta)$$

is used for the action (ρ, φ) , where $\pi(\rho_t, \varphi_t | S_t, \theta)$ is the probability of choosing ρ and φ in the state S_t with the parameters θ , $\pi(\rho_t | S_t, \theta)$ is the probability of choosing ρ in the state S_t with the parameters θ , and $\pi(\varphi_t | S_t, \theta)$ is the probability of choosing φ in the state S_t with the parameters θ . The parameters θ are updated by the formula

$$\theta \leftarrow \theta + \alpha^\theta \delta_t \nabla \ln \pi(\rho_t, \varphi_t | S_t, \theta).$$

The variable number of control cycles is not reflected in the reward function, and it cannot be adjusted dynamically. Since the computational efficiency of a well-trained agent is very high, several agents with different values K , for example $K = 1, 2, 3$, are proposed to be trained. In the real-world air traffic control process, one can quickly calculate several control options and select the best decision from several options. The simulation results confirmed that deep reinforcement learning can be used for conflict resolution and has advantage in computational efficiency as compared to known methods.

In [61], a model for two-aircraft conflict resolution was proposed and analyzed, taking into account wind-related uncertainty. The proposed conflict prevention method is applicable in the case when uncertainty associated with the direction and speed of the wind is unstable (stochastic) throughout the simulation.

In [46], a strategy for resolving two-aircraft conflicts in the three-dimensional space based on deep reinforcement learning is considered. One conflict between two aircraft is selected for resolution from an air traffic scenario that may contain several conflicts. The conflict resolution model is simulated as a discrete-time Markov decision-making process. The agent uses altitude adjustment, speed adjustment, or course correction commands to resolve the conflict. The preferences of the air traffic controller for choosing conflict resolution maneuvers are transmitted to the agent by adjusting the reward function.

3.2. Conflict Prevention Models for a Fixed Number of Aircraft

Multi-agent reinforcement learning considers a set of agents that interact with the same environment [62]. Each agent tries to achieve its goals, which are unknown to other agents. One of the strategies for solving problems in a multi-agent environment is represented by independent Q -learning with no communication between agents and other agents considered as part of the environment [63]. However, when an agent changes its policy, it affects the policies of other agents, resulting in training instability [64].

In [3, 65, 66], to ensure communication between agents, the state for each agent is proposed to include information about the state of N closest agents. The state space for an agent has a constant dimension since it depends only on N closest agents and does not scale to fit the increased number of agents in the environment. At the same time, it is highlighted that determining which N closest agents should be considered is very important for a good result to be obtained since adding irrelevant information to the state space complicates training [3].

In [67], a reinforcement learning algorithm combined with the Monte Carlo tree search (MCTS-UCT) is presented to solve the problem of self-maintenance of aircraft safe separation given high air traffic flows in the sector. All agents (aircraft) are at the same flight level, and their strategies include changing the course and the cruising speed.

The value of the reward for an agent, depending on the state s , is determined as

$$r(s) = \begin{cases} 1 & \text{if } s \text{ is the target state,} \\ 0 & \text{if } s \text{ is } LOS \text{ or out of the sector boundary,} \\ 1 - \frac{d(o, g)}{\max d(o, g)}, & \text{otherwise,} \end{cases}$$

where the state LOS is the loss of separation, $d(o, g)$ is the distance between the agent's current position and its target, and $\max d(o, g)$ is the greatest distance between the agent and its target.

Each agent state is considered as a node in the tree, and tree deployment is performed based on the state values calculated using the formula

$$UCT(S_j) = \bar{r}_j + 2C\sqrt{\frac{2 \ln N}{n_j}},$$

where $\bar{r}_j$ is the average value of the reward of the action j for the current agent, N is the counter of node visits, n_j is the action selection counter j , and $C = 1/\sqrt{2}$.

In the process of joint decision-making, all n agents $\{A_1, \dots, A_n\}$ must share their intention when choosing each individual action. One iteration of the algorithm of the multi-agent Markov decision-making process is as follows. First, n agents $\{A_1, \dots, A_n\}$ are initialized at the level $L - 1$. All agents continue to follow the default cooperative action policy $a_{-j} = \{a_i \text{ from the default cooperative action policy } | i = 1, \dots, n, i \neq j\}$. The agent A_j with the minimum index at the level $L - 1$ uses the MCTS-UCT algorithm to select its optimal strategy of actions a_j^* using the following equation

$$a_j^* = \operatorname{argmax}_{a_j} r_j^*(s, a_j, a_{-j}), \quad j = 1, \dots, n,$$

where $r_j^*(s, a_j, a_{-j})$ is the value of the reward of the agent A_j in the state s when performing the action a_j while the action strategies of other agents are represented as a_{-j} . When calculating a_j^* , other agents will continue to follow the default action strategy set as $a_{-j} = \{a_i \text{ from the default cooperative action policy } | i = 1, \dots, n, i \neq j\}$. When the agent A_j receives its optimal action strategy a_j^* , A_j is upgraded to the level L and preserves the action strategy a_j^* to update the default

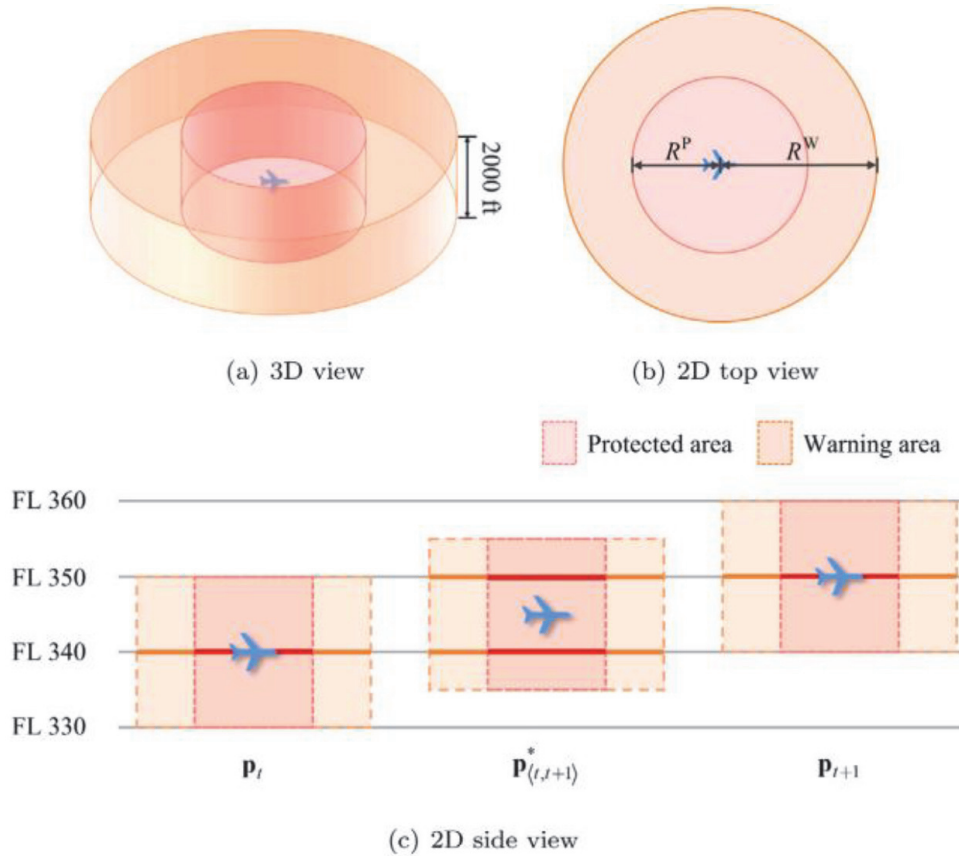


Fig. 4. Conflict buffer model.

joint decision-making strategy. Then, the next agent, with the minimum index at the level $L - 1$ chooses its optimal action strategy. This process continues until all agents have their optimal action strategy $\{a_i^*, i = 1, \dots, n\}$. The resulting joint decision-making strategies are used for the next time step Δt for all agents. Iterations are repeated until all agents reach the target state.

In [68], a method of deep ensemble multi-agent reinforcement learning was proposed for dynamic adjustment of the aircraft speed in real time. The authors claim that the extensive empirical results obtained using an open-source air traffic control model developed by Eurocontrol and based on real-world data involving thousands of aircraft demonstrate that the proposed method is significantly superior to other reference approaches.

In [69], an approach to multi-agent reinforcement learning is proposed for three-dimensional conflict resolution in a free route space, where agents use a common neural network. The trained network is deployed on each aircraft to form a distributed real-time decision-making system. In this case, communication between agents is reduced to informing other agents about the selected actions. The introduction of the three-dimensional space leads to an explosive increase in the scale of the neural network and, as a result, to an increase in the training complexity. To overcome this problem, it is proposed to consider conflicts in three planes rather than the three-dimensional space in the case of three-dimensional maneuvers, viz. the plane of the current flight level and two adjacent planes at the levels above and below it. This significantly reduces the agent training complexity. A conflict buffer model is proposed in which each aircraft is assigned a protective zone and a warning zone. Figure 4 [69] illustrates the conflict buffer model, where R^W and R^P stand for the radii of the warning zone and the protective zone, respectively. Intruders found inside the protective zone always receive a big fine while intruders found in the warning zone receive a small fine.

The number of conflicting aircraft that an agent can monitor is fixed in the neural network structure and should not change. To solve this problem, a partial surveillance model has been developed, which considers only a fixed number of aircraft that pose the greatest threat, for example, those closest to the agent. The DQN algorithm with improvements, called Rainbow, is used to implement reinforcement learning [70]. To train and evaluate the proposed approach, a simulation environment has been created that takes into account flight uncertainty (resulting, for example, from mechanical and navigational errors and wind). Experimental results show [69] that the proposed method can resolve conflicts in scenarios with a much higher traffic density than that in today's real-world situations. 2D and 3D models are compared. The training time of the 2D model is less than 1% of the training time of the 3D model; however, this does not affect the performance of the model during the decision-making due to the nature of reinforcement learning methods. In some scenarios, the 3D model can resolve conflicts more easily by changing the flight level. The 3D model is shown to be superior to the 2D model in terms of success rate and the rate of reduction of additional flight range.

With centralized air traffic control, when the air traffic controller transmits directives to the pilots, actions should be quite rare. However, in models with a continuous action space, agents can make small adjustments to their trajectories at each step. In [71], a multi-agent deep reinforcement learning method with a continuous action space is proposed, in which the number of actions is significantly reduced using a priority mechanism. At each time step, a maximum of one aircraft with the highest priority can perform actions. This approach significantly reduces the number of actions taken while maintaining a high level of conflict prevention performance. The resulting decisions are suitable for centralized air traffic control, where the number of directives that can be transmitted to the pilot is limited. In [72], the priority mechanism based on a dynamic assessment of the proximity of conflicts between aircraft is used in a model with a discrete action space.

3.3. Conflict Prevention Models for an Arbitrary Number of Aircraft

Above, we considered the models, in which the agent has access to information about the state of N nearest aircraft, where N is the hyperparameter selected during the experiments; however, this limits the adaptability of the model. Using the parameter N is a disadvantage since a small change in the location of the aircraft can change the set of N nearest aircraft and, thus, change the input data for the neural network. The neural network should understand that it is almost the same state of the airspace despite the aircraft rearrangements—however, this can be challenging [49]. The solution to this problem is to use more advanced neural network architectures, search for other ways to represent data, use data improvement algorithms, and select the most relevant neighboring aircraft [71]. One of the ways to solve the problem of a variable number of aircraft is to graphically encode information into fixed-size images and use convolutional neural networks (CNN) to extract useful information, similar to air traffic controllers' screens [73]. One can also use recurrent neural networks (RNN) with long short-term memory (LSTM) cells [74] or controlled recurrent blocks (GRU) [75] that work with the entire set of aircraft in the environment, encoding the relevant information into a hidden state of the fixed size.

In [30, 76], multi-agent reinforcement learning is considered to resolve conflicts on routes and intersections in a structured two-dimensional airspace between a variable number of aircraft. The state information is encoded using the LSTM neural network into a fixed-length vector. The agent has access to the encoded information on all aircraft in the sector; in this case, there is no need to determine the value N for each new environment. The BlueSky air traffic control simulator is used as a learning environment [77]. Centralized training and a decentralized execution scheme are used, in which one neural network is trained. This network is used by all agents to receive speed recommendations, and depending on the state, the actions of the agents may vary. The

environment is stochastic because of the uncertainty in the actions of other agents; therefore, the “actor-critic” algorithm called “proximal policy optimization” is used [78].

To ensure the safe separation requirements, an identical reward function is introduced for all agents

$$r_t = \begin{cases} -1 & \text{if } d_o^c < d^{LOS}, \\ -\alpha + \delta d_o^c & \text{if } d_o^c < 10 \text{ \& } d_o^c \geq d^{LOS}, \\ 0, & \text{otherwise,} \end{cases}$$

where d^{LOS} is the minimum safe separation distance in nautical miles ($d^{LOS} = 3$), d_o^c is the distance from the own aircraft to the nearest aircraft in nautical miles, α and δ are small positive constants to fine agents as they approach losing the safe separation distance. Three practical air traffic scenarios demonstrate the ability to solve decision-making problems with a variable number of agents and uncertainty [30].

Recurrent neural networks process input data sequentially, and the output data depends on this sequence. This can lead to undesirable results in situations when the input sequence does not matter. Transformers were introduced as an alternative to recurrent neural networks for sequential processing of input data in order to ensure parallel learning [79]. Transformers calculate the relative importance of so-called tokens containing information about the states of aircraft using the attention mechanism. In [80], absolute states are used for observation—the coordinates and velocities of the aircraft in the reference system associated with the environment. In [81], relative states are used—the coordinates and velocities of the aircraft in the reference system of the own aircraft with the positive direction of the abscissa axis in the direction of the flight. However, studies have not yet shown the superior performance of the transformer network architecture as compared to feed-forward neural networks and recurrent neural networks to ensure safe aircraft navigation [81].

4. COOPERATIVE CONFLICT PREVENTION STRATEGIES BASED ON NEURAL COMMUNICATION NETWORKS

Communication is a key ability of cooperative multi-agent systems, in which agents can significantly benefit from the information exchange prior to performing joint actions [82]. A model based on neural communication networks, which allows agents to exchange information through a communication protocol, can allow agents to develop cooperative strategies for joint actions to prevent conflicts [83].

In [80], aircraft in the airspace are simulated as agents of a cooperative multi-agent system. The state $s_i = [x_i, y_i, v_i, \chi_i]$ of each agent $i \in N$ consists of coordinates in the Euclidean space x_i, y_i , the velocity v_i , and the course χ_i . The state changes according to the formulas

$$\begin{aligned} x_i(t+1) &= x_i(t) + v_i(t) \sin \chi_i(t) \Delta t, \\ y_i(t+1) &= y_i(t) + v_i(t) \cos \chi_i(t) \Delta t, \\ v_i(t+1) &= v_i(t) + \Delta v_i, \\ \chi_i(t+1) &= \chi_i(t) + \Delta \chi_i, \end{aligned}$$

where Δv_i and $\Delta \chi_i$ are the increments of the velocity and the course, and Δt is the simulation step.

The interaction between agents is represented as the graph $G = (V, E)$, each node corresponds to one agent $i \in N$, and agents that can communicate are connected by the edges e_{ij} . The observation

vector for the current state of the i th agent consists of five elements

$$o_i = \begin{bmatrix} d_i/D \\ \cos(\chi_i - \psi_i) \\ \sin(\chi_i - \psi_i) \\ \bar{v}_i \\ \bar{v}_{ei} \end{bmatrix}^T,$$

where d_i is the shortest distance to the exit point from the observation area, D is the normalizing coefficient, ψ_i is the bearing angle to the exit point, and the normalized velocity and the velocity deviation are determined as

$$\bar{v}_i = \frac{v_i - v_{\min_i}}{v_{\max_i} - v_{\min_i}}; \quad \bar{v}_{ei} = \frac{v_i - v_{\text{opt}_i}}{v_{\max_i} - v_{\min_i}}.$$

At each step, the agent encodes its state o_i into a hidden state using the neural network

$$h_i^{(0)} = f_h(o_i).$$

Then, the communication phase begins, consisting of C communication rounds. At each round $c = 0, 1, \dots, C-1$, each agent's message is calculated as a weighted sum of the edges that connect it to its neighbors based on the attention mechanism,

$$m_i^{(c+1)} = \sum_{j \in N_i} a_{ij}^{(c+1)} e_{ij}^{(c+1)},$$

where N_i is the set of nodes connected by the edges with the node i .

The edge values are calculated using a neural network, taking into account the hidden states of the agents

$$e_{ij}^{(c+1)} = f_e^{(c)}([h_i^{(c)}, h_j^{(c)}, e_{ij}^{(c)}]).$$

The attention weights are calculated by the formula

$$a_{ij}^{(c+1)} = \frac{\exp(v_a^{(c)} f_a^{(c)}([h_i^{(c)}, h_j^{(c)}, e_{ij}^{(c)}]))}{\sum_{j \in N_i} \exp(v_a^{(c)} f_a^{(c)}([h_i^{(c)}, h_j^{(c)}, e_{ij}^{(c)}]))},$$

where $v_a^{(c)}$ is the vector of parameters.

Each node then updates its state using the updating function

$$h_i^{(c+1)} = U^{(c)}(h_i^{(c)}, m_i^{(c+1)}).$$

After C rounds of communication between the nodes, a probability distribution is generated over all possible actions for each agent

$$a_i = f_a([h_i^{(0)}, h_i^{(C)}]).$$

The expected reward, which is the same for all agents, is calculated using the read function

$$V^\pi = f_v\left(\sum_{i \in N} f_y([h_i^{(0)}, h_i^{(C)}])\right),$$

$f_h, f_e^{(c)}, f_a^{(c)}, f_a, f_v, f_y$ are feed-forward neural networks.

At each time step, each agent chooses an action a_i , followed by the environment giving a collective reward to the team.

The experiments on training the presented model showed that the total reward per episode increases, and the number of expected conflicts decreases as the number of episodes grows, i.e., agents can improve their policies based on their interaction with the environment.

5. USING GRAPH-BASED DEEP REINFORCEMENT LEARNING TO PREVENT CONFLICTS

In many studies, a multi-agent statement combines the state vectors of several aircraft into a multidimensional vector of the aggregate state using the concatenation operation [30, 84]. However, aggregating the states of all nearest neighboring aircraft, regardless of whether there are potential conflicts between them, can lead to processing redundant data and reduce the efficiency of the model. Such vectors cannot encapsulate spatiotemporal dynamics and distinguish between different levels of risk and urgency in conflict scenarios. Graph reinforcement learning (Graph RL) is designed to process data structured in the form of graphs [85]. Using graph-inherent properties, one can improve scalability, efficiency, and adaptability when working with multidimensional and dynamic environments [86]. Graph reinforcement learning allows using graph properties to represent relationships between planes [87–89]. Graph deep reinforcement learning methods are invariant to the order and number of planes.

In [49], a graph deep reinforcement learning method is proposed for air traffic control in the three-dimensional airspace. To prevent conflicts, the altitude, course, and speed of the aircraft are selected. Planes are represented by the graph vertices; nodes in this graph are connected if the distance between a pair of planes is below a certain threshold. Two approaches are compared, viz. graph neural networks with convolutional layers (GCN) [90] and graph neural networks with the attention mechanism (GAT, Graph Attention Network) [91] used to efficiently aggregate information from the neighboring nodes in the graph. With a normal traffic density, a model with the attention mechanism can prevent 100% of potential collisions and 89.8% of potential conflicts. However, performance deteriorates as the traffic density grows. With the increasing traffic flow density, both methods have difficulty overcoming congested airspace.

In [92], a graph convolutional network with LSTM cells is used to collect the space-time dependencies of flight data, and a graph neural network with increased attention is used to focus on the information characteristics of the key nodes.

In [93], a conflict graph that develops over time was proposed to be used, in which aircraft are represented by nodes, and the connections between them indicate the urgency of the conflict. The urgency of the conflict is determined by the time before the conflict if the current course and speed of the aircraft located at the points A and B (Fig. 5) are maintained. The observation is conducted from the point A . The point B is the center of a circle with the radius R (the radius of the protective zone). Tangents to this circle are drawn from the point A , forming an obstacle

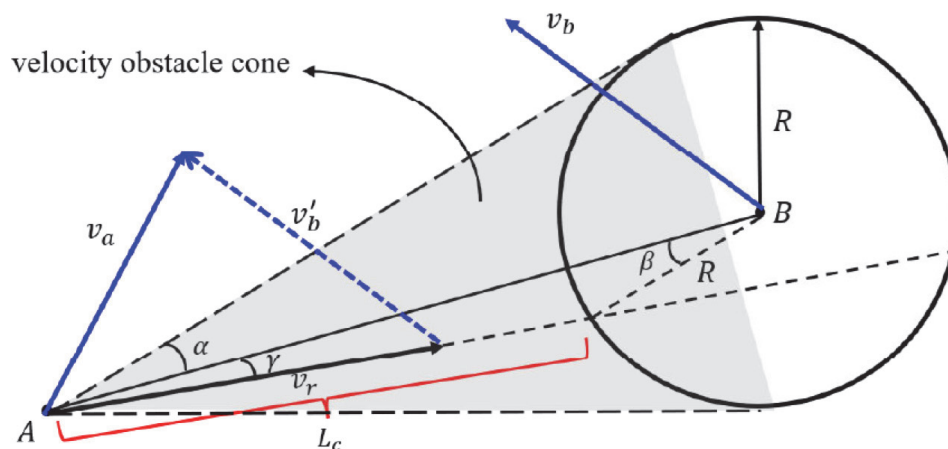


Fig. 5. Conflict geometry.

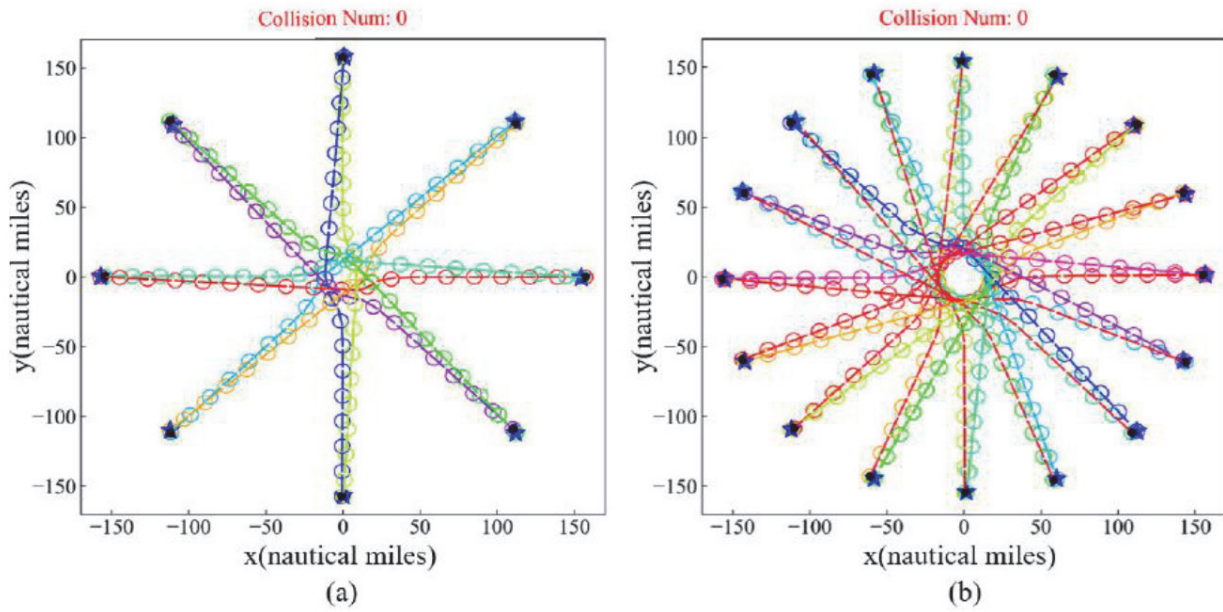


Fig. 6. Circular conflict scenario.

cone; v_A and v_B are the velocities of the aircraft A and B . A potential conflict exists if the relative velocity vector $v_r = v_A - v_B$ is inside the obstacle cone. The time before the conflict is determined based on a geometric model of the relative position and speeds of the pair of the aircraft A and B

$$t_c = \frac{L_c}{|v_r|},$$

where L_c is a straight line segment from the point A along the relative velocity v_r to the point of intersection with the protective zone of the aircraft B . The edge weight of the conflict graph ω_{AB} is normalized in the range $[0, 1]$ as follows

$$\omega_{AB} = e^{-t_c}.$$

If there is no potential conflict, then $\omega_{AB} = 0$; if the planes collided, then $\omega_{AB} = 1$.

Further, based on the conflict graph, information is aggregated using a multi-head neural attention network. The time regularization mechanism is used to increase the training stability. The efficiency of the proposed algorithm is demonstrated, among other things, by two visual scenarios [93]. Circular conflict scenario: In this setup, planes start flying at points in a circle with a radius of 160 nautical miles and fly in opposite directions. This configuration leads to the fact that each plane conflicts with all the others at the center of the circle. The experimental results for scenarios involving 8 and 16 aircraft are shown in Figs. 6a and 6b, respectively. These results demonstrate the ability of the proposed method to manage characteristic circular potential conflicts and prevent all potential collisions. Intersection conflict scenario: Aircraft are divided into two groups, each containing an equal number of them. These groups fly along the intersecting trajectories, causing conflicts at each intersection. Testing was carried out with 20 and 30 aircraft. As we can see in Figs. 7a and 7b, for 20 and 30 aircraft, respectively, the proposed method helps determine conflict-free route points at each time step with minimal deviations from the original trajectories.

Despite significant research successes, there are serious obstacles to the practical application of reinforcement learning methods in the field of air transport due to the tight certification standards

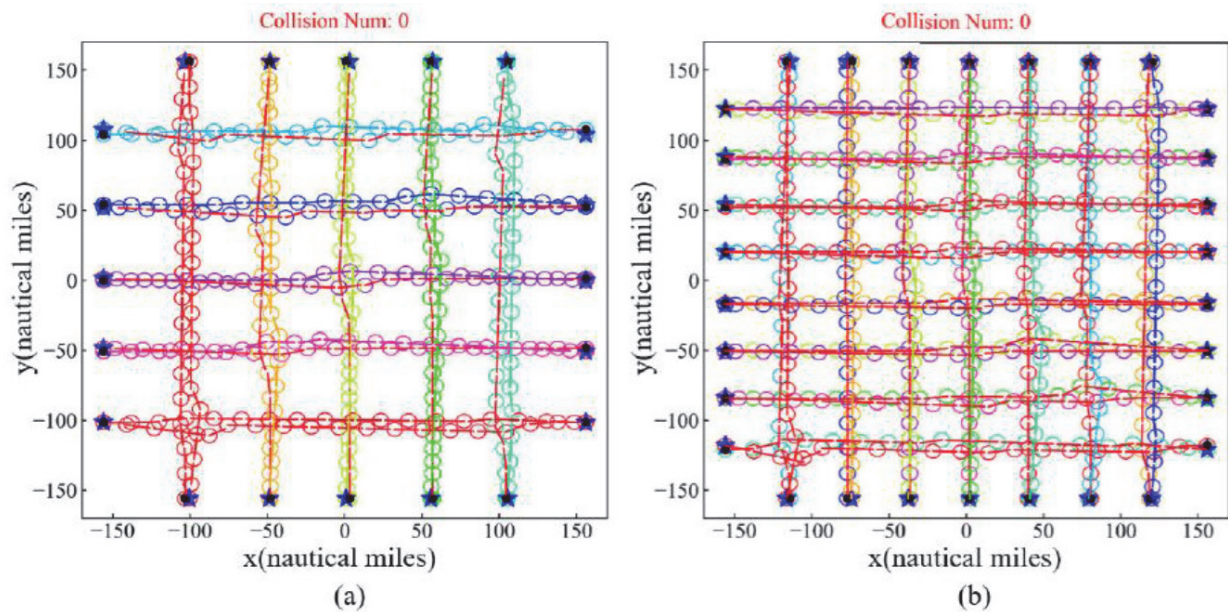


Fig. 7. Intersection conflict scenario.

in the aviation industry. The current regulatory and legal framework fails to provide adequate and acceptable means of meeting the requirements for reinforcement learning applications, and thus there is no legal framework yet in place for their safe use. It is necessary to develop certification recommendations for reinforcement learning models designed for air transport, so that these promising methods can be used in real conditions [94].

6. CONCLUSIONS

It follows from the review of publications that air traffic conflict prevention methods based on deep reinforcement learning are aimed at solving two principal tasks, viz. automatic generation of decision options to support air traffic controllers given the centralized air traffic control and support of autonomous conflict resolution systems in free flight. Models with discrete actions are mainly offered to support decision-making by air traffic controllers. Continuous-action models are designed for autonomous conflict resolution in free flight and allow all agents to perform trajectory correction actions at each time step.

The graph reinforcement learning approach seems to be most promising since the information represented in the form of a conflict graph that develops over time helps reduce the amount of redundant information processed and ensures the scalability of models for different numbers of aircraft. The attention mechanism allows singling out the most urgent information contained in the conflict graph, providing improved conflict prevention strategies in terms of security and efficiency.

With the development of research, the potential for practical application of conflict prevention methods between aircraft based on deep reinforcement learning is becoming more and more obvious. A review of publications shows that the studied reinforcement learning methods demonstrate promising results due to adaptive decision-making in real time to prevent air traffic conflicts. However, there are still serious unresolved issues preventing these methods from being applied in air traffic control practice, where safety is critical. It is impossible to train models in real-world conditions because of the potential damage, and it is impossible to perfectly simulate those conditions. The shift in distribution between the simulated environment and the reality may limit the efficiency of reinforcement learning models. Certification of such models should be one of the directions of research in this area.

REFERENCES

1. *International Civil Aviation Association. Doc 4444: Air Traffic Management, Procedures for Air Navigation Services*, 16th ed. ICAO: Montreal, QC, Canada. 2016.
2. Kulida, E.L. and Lebedev, V.G., Methods for Solving Some Problems of Air Traffic Planning and Regulation. PART I: Strategic Planning of 4D Trajectories, *Control Sciences*, 2023, no. 1, pp. 2–11. <https://doi.org/10.25728/cs.2023.1.1>
3. Brittain, M. and Wei, P., Autonomous Separation Assurance in a High-Density En Route Sector: A Deep Multi-Agent Reinforcement Learning Approach, *22nd IEEE Intelligent Transportation Systems Conference (ITSC)*. Auckland, New Zealand, 2019. <https://doi.org/10.109/ITSC.2019.8917217>
4. Ponomarev, K.Yu., A Method for Assessing the Dynamic Air Situation for Conflict by Polychromatic Display of Objects in the Information Support of the Air Traffic Control Dispatcher, *Cand. Sci. (Tech.) Dissertation Synopsis*, St. Petersburg: Saint Petersburg State University of Civil Aviation, 2023. 24 p.
5. Erzberger, H., Automated Conflict Resolution for Air Traffic Control, *25th International Congress of the Aeronautical Sciences (ICAS)*, Germany, Hamburg. 2006.
6. Farley, T., Field, M., and Erzberger, H., Fast-time Simulation Evaluation of a Conflict Resolution Algorithm Under High Air Traffic Demand, <https://www.researchgate.net/publication/255062615>, 2007.
7. Erzberger, H. and Heere, K., Algorithm and Operational Concept for Resolving Short-Range Conflicts, *Proc. Inst. Mechan. Engin., Part G: J. Aerospac. Engin*, 2010, vol. 224, no. 2, pp. 225–243. <https://doi.org/10.1243/09544100JAERO546>
8. Hoekstra, J.M., van Gent, R.N.H.W., and Ruigrok, R.C.J., Designing for Safety: The ‘Free Flight’ Air Traffic Management Concept, *Reliab. Engin. Syst. Safet.*, 2002, vol. 75, no. 2, pp. 215–232. [https://doi.org/10.1016/S0951-8320\(01\)00096-5](https://doi.org/10.1016/S0951-8320(01)00096-5)
9. Clari, M.V., Ruigrok, R.C.J., Hoekstra, J.M., and Visser, H.G., Cost-Benefit Study of Free Flight with Airborne Separation Assurance, *Air Traffic Control Quarterly*, 2001, vol. 9, no. 4, pp. 287–309. <https://doi.org/10.2514/atcq.9.4.287>
10. Marjin, N.P., Prospect of Introducing the Concept of “Free Flight,” *Problemy bezopasnosti poletov*, 2009, no. 5, pp. 42–55. (In Russ.)
11. Yang, Y., Zhang, J., Cai, K., and Prandini, M., Multi-aircraft Conflict Detection and Resolution Based on Probabilistic Reach Sets, *IEEE Transactions on Control Systems Technology*, 2016, vol. 25, no. 1, pp. 309–316. <https://doi.org/10.1109/TCST.2016.2542046>
12. Orlov, V.S., Development and Research of Algorithms for Detecting and Preventing Dangerous Approaches in the Air within the Framework of a Promising ATM System, *Cand. Sci. (Tech.) Dissertation*, Moscow: Moscow Aviation Institute. 2015. 116 p.
13. Burkin, V.S., Synthesis of Algorithms for the Detection and Resolution of Collision Conflicts Based on Data from the Automatic Dependent Surveillance System under Uncertainty, *J. Comput. Syst. Sci. Int.*, 2017, vol. 56, no. 3, pp. 492–504. <https://doi.org/10.1134/S106423071703008X>
14. Kumkov, S.I. and Pyatko, S.G., Fast Algorithms for Detecting Conflict Situations between Aircraft, *Theory of Optimal Control and Applications (OSTA 2022), Materials of the International Conference*. N.N. Krasovskii Institute of Mathematics and Mechanics of the Ural Branch of the Russian Academy of Sciences (IMM UB RAS), Yekaterinburg, 2022, pp. 126–131.
15. Pelegrin, M. and D’Ambrosio, C., Aircraft Deconfliction via Mathematical Programming: Review and Insights, *Transportation Science*, 2022, vol. 56, no. 1, pp. 118–140. <https://doi.org/10.1287/trsc.2021.1056>
16. Cafieri, S., Conn, A.R., and Mongeau, M., Mixed-Integer Nonlinear and Continuous Optimization Formulations for Aircraft Conflict Avoidance via Heading and Speed Deviations, *Eur. J. Oper. Res.*, 2023, vol. 310, no. 2, pp. 670–679. <https://doi.org/10.1016/j.ejor.2023.03.002>
17. Dias, F. and Rey, D., Aircraft Conflict Resolution with Trajectory Recovery Using Mixed-integer Programming, *J. Global Optim.*, 2024, vol. 90, pp. 1031–1067. <https://doi.org/10.1007/s10898-024-01393-1>

18. Cecen, R.K. and Cetek, C., Conflict-Free En-Route Operations with Horizontal Resolution Manoeuvres Using a Heuristic Algorithm, *Aeronaut. J.*, 2020, vol. 124, pp. 767–785.
<https://doi.org/10.1017/aer.2020.5>
19. Eby, M., A Self-Organizational Approach for Resolving Air Traffic Conflicts, *Lincoln Lab. J.*, 1994, vol. 7, no. 2, pp. 239–254.
20. Balasooriyan, S., Multi-Aircraft Conflict Resolution Using Velocity Obstacles, *MSc thesis*, Delft University of Technology, 2017, 126 p.
21. Durand, N., Constant Speed Optimal Reciprocal Collision Avoidance, *Transportation Research. Part C, Emerging Technologies*, 2018, pp. 366–379. <https://doi.org/10.1016/j.trc.2018.10.004>
22. Pan, W., Qin, L., He, Q., and Huang, Y., Three-Dimensional Flight Conflict Detection and Resolution Based on Particle Swarm Optimization, *Aerospace*, 2023, vol. 10, no. 9.
<https://doi.org/10.3390/aerospace10090740>
23. Sui, D. and Zhang, K., A Tactical Conflict Detection and Resolution Method for En Route Conflicts in Trajectory-Based Operations, *J. Advanc. Transport*, 2022, no. 2, pp. 1–16.
<https://doi.org/10.1155/2022/9283143>
24. Brittain, M. and Wei, P., Scalable Autonomous Separation Assurance with Heterogeneous Multi-agent Reinforcement Learning, *IEEE Transactions on Automation Science and Engineering*, 2022, vol. 19, no. 4, pp. 2837–2848. <https://doi.org/10.1109/TASE.2022.3151607>
25. Samoilov, V.A. and Dotsenko, D.V., The Possibility of Using Neural Networks to Find and Resolve Potential Conflict Situations between Aircraft When Flying in Upper Airspace, *Transport of Russia: Problems and Prospects – 2022. Materials of the International Scientific and Practical Conference*, N.S. Solomenko Institute of Transport Problems of the Russian Academy of Sciences, Saint Petersburg, 2022, pp. 180–184.
26. Wang, Z., Liang, M., and Delahaye, D., Data-Driven Conflict Detection Enhancement in 3D Airspace with Machine Learning, *2020 International Conference on Artificial Intelligence and Data Analytics for Air Transportation (AIDA-AT)*, Singapore, 2020.
<https://doi.org/10.1109/AIDA-AT48540.2020.904.9180>
27. Pinto Neto, E.C., Baum, D., Almeida, J.R., et. al., Deep Learning in Air Traffic Management (ATM): Applications, Opportunities, and Open Challenges, *Aerospace 2023*, vol. 10, no. 4.
<https://doi.org/10.3390/aerospace10040358>
28. Razzaghi, P., Tabrizian, A., Guo, W., et. al., A Survey on Reinforcement Learning in Aviation Applications, *Engineering Applications of Artificial Intelligence*, 2024, vol. 136, no. 3.
<https://doi.org/10.1016/j.engappai.2024.108911>
29. Kulida, E.L. and Lebedev, V.G., Methods for Solving Some Problems of Air Traffic Planning and Regulation. PART II: Application of Deep Reinforcement Learning, *Control Sciences*, 2023, no. 2, pp. 2–14.
<https://doi.org/10.25728/cs.2023.2.2>
30. Brittain, M.W. and Wei, P., One to Any: Distributed Conflict Resolution with Deep Multi-agent Reinforcement Learning and Long Short-Term Memory, *AIAA Scitech 2021 Forum*, Nashville, Tennessee, USA. <https://doi.org/10.2514/6.2021-1952>
31. Wang, Z., Pan, W., Li, H., et. al., Review of Deep Reinforcement Learning Approaches for Conflict Resolution in Air Traffic Control, *Aerospace*, 2022, vol. 9, no. 6.
<https://doi.org/10.3390/aerospace9060294>
32. Groot, J., Ribeiro, M., Ellerbroek, J., et. al., Improving Safety of Vertical Manoeuvres in a Layered Airspace with Deep Reinforcement Learning, *International Conference on Research in Air Transportation (ICRAT)*, Tampa, Florida, USA, 2022, pp. 19–23.
33. Ribeiro, M., Ellerbroek, J., and Hoekstra, J., Review of Conflict Resolution Methods for Manned and Unmanned Aviation, *Aerospace*, 2020, vol. 7, no. 6. <https://doi.org/10.3390/aerospace7060079>

34. Ribeiro, M., Ellerbroek, J., and Hoekstra, J., Distributed Conflict Resolution at High Traffic Densities with Reinforcement Learning, *Aerospace*, 2022, vol. 9, no. 9. <https://doi.org/10.3390/aerospace9090472>
35. Ribeiro, M., Conflict Resolution at High Traffic Densities with Reinforcement Learning, *PhD Thesis*, 2023. <https://doi.org/10.4233/uuid:a2979919-cb01-41d1-bbba-fefa9079463b>
36. Ribeiro, M., Ellerbroek, J., and Hoekstra, J., Improving Algorithm Conflict Resolution Manoeuvres with Reinforcement Learning, *Aerospace*, 2022, vol. 9, no. 12. <https://doi.org/10.3390/aerospace9120847>
37. Vizilter, Yu.V., Vishnyakov, B.V., and Zheltov, S.Yu., Modern Artificial Intelligence Technologies and Their Application in Aviation Complexes, *16th All-Russian Multi-conference on Management Problems (MCPU-2023)*, Volgograd, Materials of the multi-conference in 4 volumes, vol. 3, pp. 13–16.
38. Sui, D., Ma, C., and Wei, C., Tactical Conflict Solver Assisting Air Traffic Controllers Using Deep Reinforcement Learning, *Aerospace*, 2023, vol. 10, no. 2. <https://doi.org/10.3390/aerospace10020182>
39. Bastas, A. and Vouros, G., Data-Driven Modeling of Air Traffic Controllers' Policy to Resolve Conflicts, *Aerospace*, 2023, vol. 10, no. 6. <https://doi.org/10.3390/aerospace10060557>
40. Kulida, E.L. and Lebedev, V.G., Problems in the Application of Machine Learning Methods in Aviation, *Proc. of the 16th International Conference "Management of Large-Scale Systems Development" (MLSD'2023, Moscow)*, Moscow, Institute of Control Sciences of RAS, pp. 1315–1320. <https://doi.org/10.25728/mlsd.2023.1315>
41. Degas, A., Islam, M.R., Hurter, C., et. al., A Survey on Artificial Intelligence (AI) and eXplainable AI in Air Traffic Management: Current Trends and Development with Future Research Trajectory, *Appl. Sci.*, 2022, vol. 12, iss. 3. <https://doi.org/10.3390/app12031295>
42. Wang, L., Yang, H., Lin, Y., et. al., Enhancing Air Traffic Control: A Transparent Deep Reinforcement Learning Framework for Autonomous Conflict Resolution, *Expert Systems with Applications*, 2024, vol. 260(2). <https://doi.org/10.06/j.eswa.2024.125389>
43. Sutton, R.S. and Barto, A.G., *Reinforcement Learning: An Introduction*, MIT Press, 1998.
44. Graesser, L. and Keng, W.L., *Foundations of Deep Reinforcement Learning: Theory and Practice in Python*, Addison-Wesley Professional, 2019.
45. Morales, M., *Grokking Deep Reinforcement Learning*, Manning, 2010.
46. Sui, D., Ma, C., and Dong, J., Conflict Resolution Strategy Based on Deep Reinforcement Learning for Air Traffic Management, *Aviation*, 2023, vol. 27, iss. 3, pp. 177–186. <https://doi.org/10.3846/aviation.2023.19720>
47. Hasselt, H.V., Double Q-Learning, *24th Annual Conference on Neural Information Processing Systems*, Vancouver, Canada, 2010, pp. 2613–2621.
48. Brittain, M. and Wei, P., Autonomous Aircraft Sequencing and Separation with Hierarchical Deep Reinforcement Learning, *International Conference for Research in Air Transportation*, Castelfidefil, Spain, 2018.
49. Mollinga, J. and Hoof, H., An Autonomous Free Airspace En-route Controller Using Deep Reinforcement Learning Techniques, *9th International Conference on Research in Air Transportation (ICRAT)*, Tampa, Florida, USA, 2020.
50. Sui, D., Xu, W., and Zhang, K., Study on the Resolution of Multi-aircraft Flight Conflicts Based on an IDQN, *Chin. J. Aeronaut.*, 2021, vol. 35, no. 11, pp. 195–213. <https://doi.org/10.06/j.cja.2021.03.015>
51. Li, S., Egorov, M., and Kochenderfer, M.J., Optimizing Collision Avoidance in Dense Airspace Using Deep Reinforcement Learning, *Proc. of the 13th USA/Europe Air Traffic Management Research and Development Seminar*, Vienna, Austria. June 17–21, 2019. <https://doi.org/10.48550/arXiv.1912.10146>
52. Hermans, M.C., Towards Explainable Automation for Air Traffic Control Using Deep Q-Learning from Demonstrations and Reward Decomposition. *Master's Thesis*, Delft University of Technology, Delft, The Netherlands, May 2020.

53. Pham, D.-T., Tran, N.P., Goh, S.K., et. al., Reinforcement Learning for Two-aircraft Conflict Resolution in the Presence of Uncertainty, *IEEE-RIVF International Conference on Computing and Communication Technologies (RIVF)*, Singapore, 2019. <https://doi.org/10.1109/RIVF.2019.8713624>
54. Wang, Z., Li, H., Wang, J., et. al., Deep Reinforcement Learning Based Conflict Detection and Resolution in Air Traffic Control, *IET Intelligent Transport Systems*, 2019, vol. 13, iss. 6, pp. 1041–1047. <https://doi.org/10.1049/iet-its.2018.5357>
55. Badea, C.A., Groot, J., Morfin Veytia, A., et. al., Lateral and Vertical Air Traffic Control Under Uncertainty Using Reinforcement Learning, *Proc. of the 12th SESAR Innovation Days*, Budapest, Hungary, 2022.
56. Pham, D., Tran, N., Alam, S., et. al., A Machine Learning Approach for Conflict Resolution in Dense Traffic Scenarios with Uncertainties, *13th USA/Europe Air Traffic Management Research and Development Seminar*, Vienne, Austria, June 2019.
57. Wen, H., Li, H., and Wang, Z., Application of DDPG-based Collision Avoidance Algorithm in Air Traffic Control, *12nd International Symposium on Computational Intelligence and Design*, Hangzhou, China, 2019, pp. 130–133. <https://doi.org/10.1109/ISCID.2019.00036>
58. Pham, D.-T., Tran, P.N., Alam, S., et. al., Deep Reinforcement Learning Based Path Stretch Vector Resolution in Dense Traffic with Uncertainties, *Transportation Research, Part C*, 2022, vol. 135, no. 3. <https://doi.org/10.1016/j.trc.2021.103463>
59. Mukherjee, A., Guleria, Y., and Alam, S., Deep Reinforcement Learning for Air Traffic Conflict Resolution Under Traffic Uncertainties, *41st Digital Avionics Systems Conference (DASC)*, Portsmouth, USA, 2022. <https://doi.org/10.1109/DASC55683.2022.9925772>
60. Sunil, E., Ellerbroek, J., and Hoekstra, J. M., Camda: Capacity Assessment Method for Decentralized Air Traffic Control, *International Conference on Research in Air Transportation (ICRAT)*, Barcelona, Spain, 2018, pp. 26–29.
61. Dudoit, A., Rimsa, V., and Bogdevicius, M., Investigation of Aircraft Conflict Resolution Trajectories under Uncertainties, *Sensors*, 2024, vol. 24, no. 18. <https://doi.org/10.3390/s24185877>
62. Busoni, L., Babuska, R., and De Schutter, B., A Comprehensive Survey of Multi-agent Reinforcement Learning, *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 2008, vol. 38, no. 2, pp. 156–172. <https://doi.org/10.1109/TSMCC.2007.913919>
63. Tan, M., Multi-agent Reinforcement Learning: Independent vs. Cooperative Agents, *10th International Conference on Machine Learning (ICML)*, 1993, pp. 330–337. <https://doi.org/10.1016/B978-1-55860-307-3.50049-6>
64. Matignon, L., Laurent, G.J., and Le Fort-Piat, N., Independent Reinforcement Learners in Cooperative Markov Games: a Survey Regarding Coordination Problems, *Knowledge Engineering Review*, 2012, vol. 27, no. 1. <https://doi.org/10.1017/S0269888912000057>
65. Everett, M., Chen, Y.F., and How, J.P., Motion Planning Among Dynamic, Decision-making Agents with Deep Reinforcement Learning, *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2018, pp. 3052–3059. <https://doi.org/10.48550/arXiv.1805.01956>
66. Chen, Y., Hu, M., Yang, L., et. al., General Multi-agent Reinforcement Learning Integrating Adaptive Manoeuvre Strategy for Real-time Multi-Aircraft Conflict Resolution, *Transportation Research. Part C. Emerging Technologies*, 2023, vol. 151. <https://doi.org/10.1016/j.trc.2023.104125>
67. Xu, Q., Chen, Z., Li, F., et. al., An Efficient Aircraft Conflict Detection and Resolution Method Based on an Improved Reinforcement Learning Framework, *Int. J. Aerospac. Engin.*, 2023, vol. 1, pp. 1–16. <https://doi.org/10.1155/2023/6643903>
68. Ghosh, S., Laguna, S., Lim, S.H., et. al., A Deep Ensemble Method for Multi-Agent Reinforcement Learning: A Case Study on Air Traffic Control, *31st International Conference on Automated Planning and Scheduling*, SuiGuangzhou, China, 2021, pp. 468–476. <https://doi.org/10.1609/icaps.v31i1.15993>

69. Chen, Y., Xu, Y., Yang, L., et. al., General Real-Time Three-Dimensional Multi-Aircraft Conflict Resolution Method Using Multi-Agent Reinforcement Learning, *Transportation Research. Part C. Emerging Technologies*, vol. 157. <https://doi.org/10.1016/j.trc.2023.104367>
70. Hessel, M., Modayil, J., Van Hasselt, H., et. al., Rainbow: Combining Improvements in Deep Reinforcement Learning, *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018, vol. 32, no. 3. <https://doi.org/10.1609/aaai.v32i1.11796>
71. Nilsson, J., Unger, J., and Eilertsen, G., Self-Prioritizing Multi-Agent Reinforcement Learning for Conflict Resolution in Air Traffic Control with Limited Instructions, *Aerospace*, 2025, vol. 12, no. 2. <https://doi.org/10.3390/aerospace12020088>
72. Sui, D., Zhou, Z., and Cui, X., Priority-based Intelligent Resolution Method of Multi-Aircraft Flight Conflicts, *Aeronaut. J.*, 2024, vol. 129, pp. 326–350. <https://doi.org/10.1017/aer.2024.75>
73. Zhao, P. and Liu, Y., Physics Informed Deep Reinforcement Learning for Aircraft Conflict Resolution, *IEEE Transactions on Intelligent Transportation Systems*, 2021, vol. 23, no. 7, pp. 8288–8301.
74. Hochreiter, S. and Schmidhuber, J., Long Short-Term Memory, *Neural Computation*, 1997, vol. 9, no. 8, pp. 1735–1780. <https://doi.org/10.1162/neco.1997.9.8.1735>
75. Cho, K., Merriënboer, B., Gulcehre, C., et. al., Learning Phrase Representations Using RNN Encoder-Decoder for Statistical Machine Translation, *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Doha, Qatar, 2014, pp. 1724–1734. <https://doi.org/10.3115/v1/D14-1179>
76. Brittain, M.W., Yang, X., and Wei, P., Autonomous Separation Assurance with Deep Multi-agent Reinforcement Learning, *J. Aerospac. Inform. Syst.*, 2021, vol. 18, no. 12, pp. 890–905. <https://doi.org/10.254/1.1010973>
77. Hoekstra, J.M. and Ellerbroek, J., BlueSky ATC Simulator Project: an Open Data and Open Source Approach, *7th International Conference on Research in Air Transportation*, Philadelphia, USA, 2016, vol. 131, p. 132.
78. Schulman, J., Wolski, F., Dhariwal, P., et. al., Proximal Policy Optimization Algorithms, *ArXiv*, 2017. <https://doi.org/10.48550/arXiv.1707.06347>
79. Vaswani, A., Shazeer, N., Parmar, N., et. al., Attention is All You Need, *31st Conference on Neural Information Processing Systems (NIPS)*, Long Beach, USA, 2017. <https://doi.org/10.48550/arXiv.1706.03762>
80. Dalmau, R. and Allard, E., Air Traffic Control Using Message Passing Neural Networks and Multi-agent Reinforcement Learning, *Conference: SESAR Innovation Days (SID), Virtual Event*, 2020, pp. 7–10.
81. Groot, J., Ellerbroek, J., and Hoekstra, J., Using Relative State Transformer Models for Multi-Agent Reinforcement Learning in Air Traffic Control, *Conference: SESAR Innovation days (SID)*, Seville, Spain, 2023.
82. Wollkind, S., Valasek, J., and Ioerger, T., Automated Conflict Resolution for Air Traffic Management Using Cooperative Multi-Agent Negotiation, *AIAA Guidance, Navigation, and Control Conference and Exhibit*, 2004. <https://doi.org/10.2514/6.2004-4992>
83. Pritchett, R. and Genton, A., Negotiated Decentralized Aircraft Conflict Resolution, *IEEE Transactions on Intelligent Transportation Systems*, 2017, vol. 19, no. 1, pp. 81–91. <https://doi.org/10.1109/TITS.2017.2693820>
84. Lai, J., Cai, K., Liu, Z., et. al., A Multi-Agent Reinforcement Learning Approach for Conflict Resolution in Dense Traffic Scenarios, *IEEE/AIAA 40th Digital Avionics Systems Conference (DASC)*, San Antonio, USA, 2021. <https://doi.org/10.1109/DASC52595.2021.9594437>
85. Wu, Z., Pan, S., Chen, F., et. al., A Comprehensive Survey on Graph Neural Networks, 2019. <https://doi.org/10.48550/arXiv.1901.00596>
86. Mendonça, M., Ziviani, A., and Barreto, A., Graph-Based Skill Acquisition for Reinforcement Learning, *ACM Computing Surveys (CSUR)*, 2019, vol. 52, no. 1. <https://doi.org/10.1145/3291045>

87. Papadopoulos, G., Bastas, A., Vouros, G.A., et. al., Deep Reinforcement Learning in Service of Air Traffic Controllers to Resolve Tactical Conflicts, *Expert Systems with Applications*, 2024, vol. 236, no. 1. <https://doi.org/10.1016/j.eswa.2023.121234>
88. Isufaj, R., Aranega Sebastia, D., and Piera, M.A., Toward Conflict Resolution with Deep Multi-Agent Reinforcement Learning, *J. Air Transport*, 2022, vol. 30, no. 3, pp. 71–80. <https://doi.org/10.2514/1.DO296>
89. Vouros, G., Papadopoulos, G., Bastas, A., et. al., Automating the Resolution of Flight Conflicts: Deep Reinforcement Learning in Service of Air Traffic Controllers, *PAIS 2022*, pp. 72–85. <https://doi.org/10.48550/arXiv.2206.07403>
90. Kipf, T.N. and Welling, M., Semi-supervised Classification with Graph Convolutional Networks, 2017. <https://doi.org/10.48550/arXiv.1609.02907>
91. Velickovic, P., Cucurull, G., Casanova, A., et. al., Graph Attention Networks, 2018. <https://doi.org/10.48550/arXiv.170.0903>
92. Zhang, Y., Xu, S., Zhang, L., et. al., Short-Term Multi-Step-Ahead Sector-Based Traffic Flow Prediction Based on the Attention-Enhanced Graph Convolutional LSTM Network (AGC-LSTM), *Neural Computing and Applications*, 2024. <https://doi.org/10.1007/s00521-024-09827-3>
93. Li, Y., Zhang, Y., Guo, T., et. al., Graph Reinforcement Learning for Multi-Aircraft Conflict Resolution, *IEEE Transactions on Intelligent Vehicles*, 2024. <https://doi.org/10.1109/TIV.2024.3364652>
94. Ribeiro, M., Tseremoglou, I., and Santos, B., Certification of Reinforcement Learning Applications for Air Transport Operations Based on Criticality and Autonomy, *AIAA Science and Technology Forum and Exposition*, Orlando, Florida, USA, 2024. <https://doi.org/10.2514/6.2024-1463>

This paper was recommended for publication by N.I. Selvesyuk, a member of the Editorial Board

Development of the Design and Trajectory Planning of a Snake-Like Manipulator Robot Based Empirical Approach

Xinbin Zhang^{*,a}, Zhuo Chen^{**,b}, Haoxiang Su^{**,c}, Ligu Tan^{*,d}, Hongwei Liu^{**,e},
Manlu Liu^{**,f}, Jianwen Huo^{**,g}, and S. V. Novikova^{***,h}

^{*}Harbin Institute of Technology, Harbin, China

^{**}Southwest University of Science and Technology, Mianyang, China

^{***}Kazan National Research Technical University named after A.N. Tupolev, Kazan, Russia

e-mail: ^axinbinzhang@hit.edu.cn, ^b986052851@qq.com, ^c463099610@qq.com, ^dtanliguo@hit.edu.cn,
^eliuhongwei@swust.edu.cn, ^fliumanlu@swust.edu.cn, ^ghuojianwen2008@hotmail.com, ^hSVNovikova@kai.ru

Received May 14, 2024

Revised April 22, 2025

Accepted April 29, 2025

Abstract—A new design for a pneumatic flexible manipulator based on a series-parallel composite structure is proposed. The device's flexible skeleton is created following the principle of snake spinal articulations. Adjacent vertebrae are equipped with air reservoir-joints located on two sides, connected in parallel with the spine. The pneumatic joint can be bent left or right in the robot's plane of motion by changing the pressure in the pneumatic chamber. A kinematic model has been developed to calculate the position of the end effector based on the bending angles of the manipulator's joints, which, in turn, are calculated from the pressure in the pneumatic chamber-joints. For the developed design, a method for calculating the planned manipulator trajectory is proposed, based on solving inverse kinematics problems. Several examples of calculating optimal motion trajectories meeting various task requirements are provided. The results of full-scale experiments show that the developed flexible snake-like manipulator accurately follows a pre-planned trajectory and demonstrates extensive movement capabilities on a plane.

Keywords: snake-like manipulator, flexible manipulator, soft manipulator, pneumatic actuator, trajectory planning system synthesis, inverse kinematics problem, empirical approach, movement in confined spaces

DOI: 10.31857/S0005117925090059

1. INTRODUCTION

With the advancement of high-tech technologies, the designs of products and devices are becoming, on the one hand, increasingly complex, and on the other, more compact. As a result, their assembly, maintenance, and other operations often need to be performed in confined spaces of complex configurations. Automating routine maintenance processes for such units requires equipment capable of adapting to the space's topology, avoiding obstacles, and performing service functions without damaging the main structure. Traditional rigid manipulators, such as industrial robotic arms, generally possess the ability to move and satisfy diverse end-effector pose requirements only in open spaces. However, their relatively small number of joints, and consequently, degrees of freedom, significantly limit their capabilities when moving in narrow, confined spaces.

One way to address this issue is to use redundant continuum manipulators, which have the ability to bend while moving and bypass existing obstacles [1–7]. For example, [1] presents a multi-joint cable-driven manipulator for automotive spray painting, enhancing the manipulator's movement

capabilities and its ability to avoid obstacles. In [2, 3], linear or hybrid air hose actuators are used to create a series of flexible manipulators mimicking an elephant's trunk. In [4, 5], a series of flexible manipulators capable of bypassing obstacles and reaching a specific point in a confined space were developed. Continuum manipulators enhance movement capability in three-dimensional space by increasing the degrees of freedom of their joints. However, increasing the degrees of freedom leads to an increase in the size and weight of the drive mechanism due to the use of additional hoses and cables, and an increase in the complexity of kinematic calculations, which in turn increases control complexity and computational costs. All of this collectively limits their practical application. Furthermore, the components of such manipulators are entirely made of rigid materials, which can raise safety concerns when they interact with the environment.

In recent years, thanks to the emergence of new pliable materials, so-called soft manipulators have appeared, capable of easily penetrating complex and narrow spaces due to their ability to actively and passively change their geometric shape. At the same time, their high compliance allows them to safely interact with elements of the surrounding environment, causing minimal damage to both environmental elements and the robots themselves in case of collisions. Soft manipulators have prospects for application in industrial assembly, agricultural product harvesting, minimally invasive surgery, domestic service, and other fields. Soft manipulators often mimic bionic strategies observed in nature. For instance, [8] uses an elephant's trunk imitation, [9] imitates worm movements, in [10–12] the authors imitate the fixation methods of climbing plants, in [13] – the movement of octopus tentacles, and so on. Such manipulators use air cushions, pneumatic muscles, springs, etc., as structural support and are actuated by pneumatic actuators, shape memory alloys, or electroactive polymers [14–17]. The unique properties of soft manipulators are attracting increasing attention from researchers striving to overcome the limitations of existing rigid and flexible manipulator designs [18–27]. Examples of soft manipulators include the OctArm continuum manipulator [18], which is driven by pneumatic muscles and mounted on a tracked mobile robot. In [19], a soft manipulator using a honeycomb pneumatic network was developed, which provides stable motion control and high payload capacity. In [20], the authors developed a soft manipulator using artificial pneumatic muscles that mimics the movement of an elephant's trunk and possesses flexible spatial movement capabilities. In [21], a fluidically actuated soft manipulator is described, and the authors of [22] assembled a multi-joint soft manipulator that uses omnidirectional bending and twisting modules for efficient spatial movement. In [23], a soft manipulator for minimally invasive surgery is proposed, where omnidirectional bending motion is achieved by three pneumatic modules. In [24], a self-organizing soft manipulator based on biological mechanisms such as neurons, pollen tubes, and thick-walled tissue cells was developed. According to the authors, the self-organizing manipulator is capable of changing shape depending on environmental conditions. In [25], a three-joint soft manipulator suitable for underwater operation and allowing precise object gripping using an anti-bending and stretching mechanism was developed. In [26], a modular, hydraulically actuated soft robotic wrist consisting of rotating and bending modules is proposed, capable of performing delicate and precise manipulations in complex deep-sea environments. In [27], a tendon-driven soft manipulator using three modules based on Kresling patterns was developed, which can switch between two stable states: flexible joints with low bending stiffness and rigid links with high stiffness.

Despite numerous studies on soft manipulators, these works are still in their infancy, and their practical application is associated with many difficulties. One problem is that the low stiffness of soft robotic systems reduces their controllability and motion stability under the action of gravitational and inertial forces, although their easily changeable geometry and high plasticity allow them to adapt to complex and narrow spaces. One approach to solving this problem is the development of variable stiffness mechanisms, which can be achieved through the use of materials with controllable stiffness, such as low-melting point alloys [28], or jamming mechanisms [29, 30]. Another approach

involves the incorporation of flexible structures made from rigid materials to enhance the overall stiffness of the robot [31]. The first approach allows for changing the robot's plasticity and geometric shape by adjusting stiffness, but the use of variable stiffness mechanisms affects not only the inherent flexibility of the manipulator but also significantly complicates the design and control. With the second approach, some of the soft manipulator's capabilities for changing geometry will be lost. This article implements the second approach to improve the controllability and stability of soft manipulators. A planar flexible snake-like soft manipulator, inspired by the structure of a snake's body, has been developed for movement in narrow spaces with obstacles. By using a rigid-flexible composite structure for the supporting frame (spine), the manipulator demonstrates a good ability to change its geometric shape with simultaneous motion stability and ease of control. For the designed snake-like manipulator, it is necessary to develop a trajectory calculation algorithm that corresponds to the specific movements of the proposed design. Currently, the most widespread approach to manipulator trajectory planning is based on solving the inverse kinematics problem. This approach is applied to both rigid [32–34] and soft multi-joint manipulators [21, 35]. The inverse kinematics problem is solved using various strategies: in [33], trajectory calculation relies on solving kinematic differential equations, while in [32], an analytical solution is proposed without using derivatives. Heuristic approaches are gaining widespread use today. For instance, in the FABRIK (Forward and Backward Reaching Inverse Kinematics) algorithm [36], the coordinates of a multi-joint manipulator's nodes are calculated by repeated linear shifts towards the target position of the end effector. Trajectory calculation algorithms without using inverse kinematics are also described in the literature. For example, [37] uses a specific modification of a genetic algorithm for trajectory calculation, in [38] the authors propose using fuzzy logic for trajectory planning, and in [39] – hybrid neuro-fuzzy networks. It should be noted that such approaches generally require more computational resources and are more complex to implement than algorithms based on solving inverse kinematic problems.

The trajectory calculation algorithm proposed in this article is based on the inverse kinematic approach. The kinematic model of robot motion is built on a combination of theory and experiments. The forward kinematics problem of the flexible manipulator is formulated based on the results of previously conducted full-scale experiments, which allows the algorithm to be adapted to a specific manipulator with high accuracy. The solution to the inverse problem is obtained analytically. Full-scale experiments were conducted to verify the mobility of the flexible manipulator with the proposed trajectory planning algorithm, which confirmed its high effectiveness in bypassing obstacles.

2. DESIGN AND ASSEMBLY OF THE FLEXIBLE ROBOT

The wide range of motion and maneuverability of snakes in complex, confined environments can be attributed to their unique body structure. In general, a snake's body is a composite structure consisting of both rigid and flexible elements. The central spine provides robust support for the body and is covered by soft muscular tissue. A snake's spine comprises hundreds of vertebrae arranged sequentially. By manipulating muscles in various directions, adjacent vertebrae can achieve horizontal deflections of up to 10 degrees and vertical deflections of up to 3 degrees [40]. Although the bending angle of a single joint is limited, a snake can achieve significant overall body bends by accumulating numerous small bends across individual joints. Based on this, the manipulator should have an analogous rigid-flexible composite structure, consisting of a planar flexible spine and actuated air reservoirs. The entire snake-like manipulator is divided into six flexible sector-joints for independent control, with each flexible joint sequentially connected to the others, as shown in Fig. 1. Each sector-joint, in turn, consists of six serially connected small air reservoir-vertebrae. The vertebrae are joined into a joint using flat flexible hinges, allowing the sector to bend within

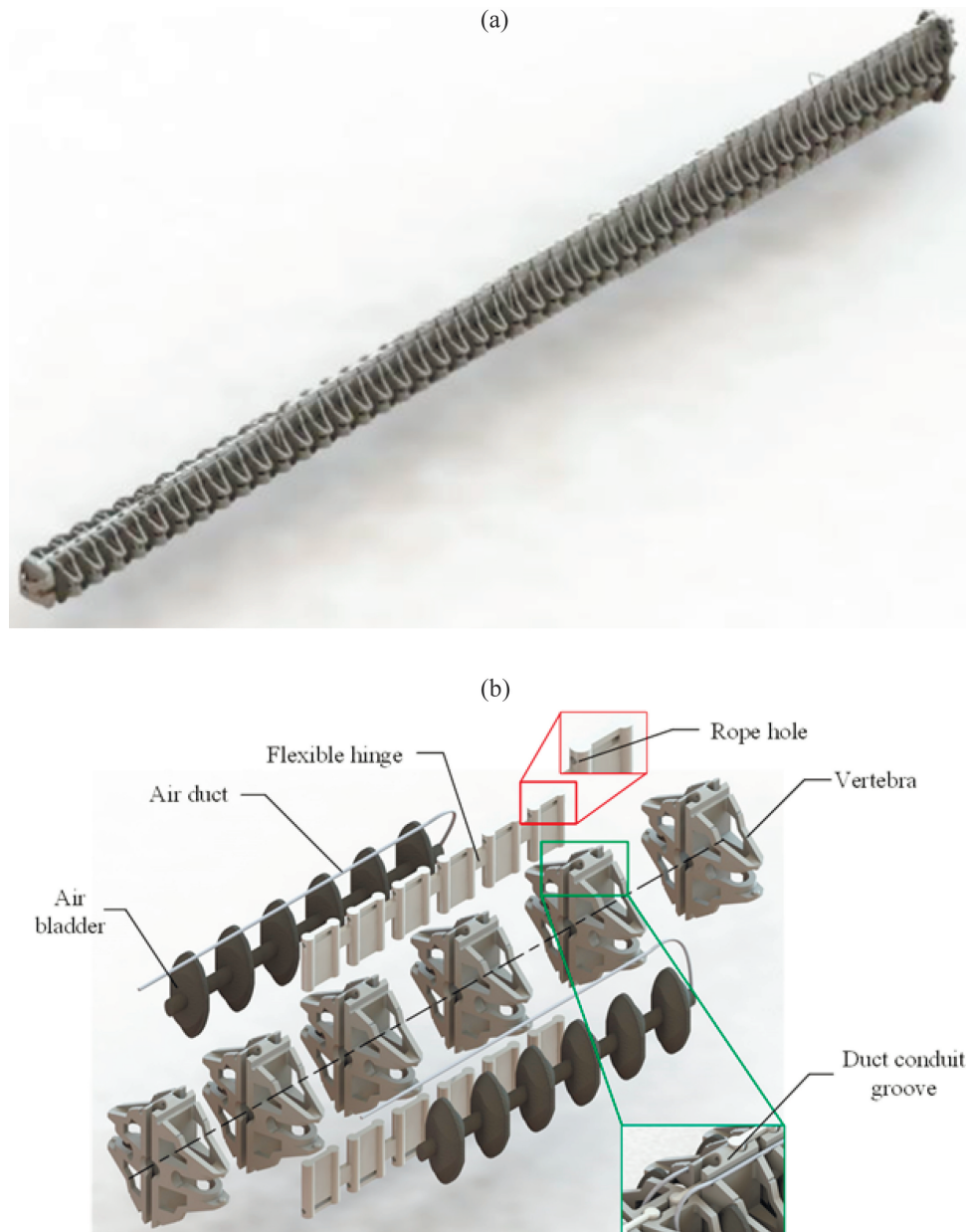


Fig. 1. Flexible manipulator in the shape of a snake: (a) overall view, (b) structure of the flexible joint.

the plane of the joint while restricting longitudinal bending. Additionally, high-strength cables are attached to the outer side of the hinge. Their number increases from tail to head, thereby enhancing the overall longitudinal strength of the flexible joints. Between the vertebrae, on both sides, two pneumatic chambers act as muscles for movement. Each pneumatic chamber contains six internal air reservoirs, corresponding to the operation of the six vertebrae. By changing the air pressure inside the pneumatic chambers on both sides of the structure, they expand and lengthen, thereby regulating the bending angle of the entire flexible joint. To simplify the design, except for the final flexible joint, the lower (“ventral”) and upper (“dorsal”) surfaces of the vertebrae are equipped with grooves. The depth of the grooves increases as the robot moves from head to tail, resulting in the manipulator acquiring a conical bio-inspired shape resembling an octopus tentacle. To reduce the robot’s weight and cost, its spine and flexible hinges were manufactured using 3D printing with

polylactic acid (PLA) and thermoplastic polyurethane (TPU), respectively. The air reservoirs were fabricated using silicone rubber (E640), known for its high elasticity, tear strength, and elongation up to 470%. The air reservoirs were produced by silicone molding. The mold for shaping the reservoirs consists of two parts. Casting is performed by inserting an iron rod into the upper cavity to create the two halves of the reservoir. A vacuum drying oven creates reduced pressure to prevent bubble formation inside the molded reservoirs during their manufacturing process. The finished snake-like manipulator consists of 6 sector-joints. The total length of the robot is 900 mm, and its weight is approximately 2 kg.

3. TRAJECTORY CONSTRUCTION OF TRANSLATING THE SNAKE-LIKE MANIPULATOR BASED ON EMPIRICAL APPROACH

In this section a model is being developed that allows control over the position of the manipulator's end-effector (working tool) in two-dimensional space. We will assume that the end-effector is located at the "head" of the robot. The forward kinematic model of the flexible snake-like manipulator includes two main components: the dependence of the end-effector's position on the bending angle of each flexible joint, and the dependence of the bending angles of the mechanical joints on the pressure in the controlling pneumatic chambers.

3.1. Dependence of the Manipulator's End-Effector Position on Joint Bending Angles

Figure 2 illustrates the motion model of a single sector-joint of the flexible snake-like manipulator. This represents a simplified model with constant curvature, resulting in a smooth continuous curve. The shape of the curve is determined by two variables related to the flexible joint: the length of the joint (L) and the bending angle (θ_i). Based on the kinematic model presented in Fig. 2, the transformation matrix from the initial coordinate system $P_{i-1}(x_{i-1}, y_{i-1})$ to the final coordinate system $P_i(x_i, y_i)$ can be determined. The matrix will thus represent a kinematic mapping from the joint space of the robot to the robot's position space in the external environment. The initial coordinate system $P_{i-1}(x_{i-1}, y_{i-1})$ is transformed into $P_i(x_i, y_i)$ by a single translation and a single rotation:

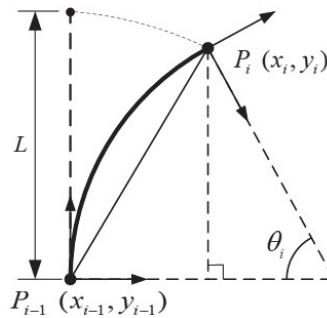


Fig. 2. Bending of a single flexible joint.

- (1) The coordinate system $P_{i-1}(x_{i-1}, y_{i-1})$ is translated along two coordinate axes to move point P_{i-1} to point P_i . Based on geometric relationships, we obtain the translation matrix:

$$\text{Trans} \left(L \frac{1 - \cos \theta_i}{\theta_i}, L \frac{\sin \theta_i}{\theta_i} \right) = \begin{pmatrix} 1 & 0 & L \frac{1 - \cos \theta_i}{\theta_i} \\ 0 & 1 & L \frac{\sin \theta_i}{\theta_i} \\ 0 & 0 & 1 \end{pmatrix}. \quad (1)$$

- (2) The coordinate system rotates around the z-axis by an angle θ_i . The rotation matrix is (2).

$$\text{Rot}(z_i, \theta_i) = \begin{pmatrix} \cos \theta_i & \sin \theta_i & 0 \\ -\sin \theta_i & \cos \theta_i & 0 \\ 0 & 0 & 1 \end{pmatrix}. \quad (2)$$

Multiplying these two matrices, we get the transformation matrix from the initial to the final coordinate system for a single joint:

$${}^{i-1}T_i = \text{Trans}\left(\frac{L}{\theta_i}(1 - \cos \theta_i), \frac{L}{\theta_i} \sin \theta_i\right) \text{Rot}(z_i, \theta_i) = \begin{bmatrix} \cos \theta_i & -\sin \theta_i & L \frac{1 - \cos \theta_i}{\theta_i} \\ \sin \theta_i & \cos \theta_i & L \frac{\sin \theta_i}{\theta_i} \\ 0 & 0 & 1 \end{bmatrix}. \quad (3)$$

Since the manipulator consists of several serially connected joints with a single degree of freedom of bending, the end-effector's position depends on the bending angle of each joint. If all joints use a constant circular arc curvature model, then according to the coordinate transformation formula (3), the homogeneous transformation matrix from the joint coordinate system $P_n(x_n, y_n)$ to the base coordinate system $P_0(x_0, y_0)$ for n segments can be obtained:

$${}^0T_n = {}^0T_1 \times {}^1T_2 \times \dots \times {}^{n-2}T_{n-1} \times {}^{n-1}T_n. \quad (4)$$

Furthermore, if the manipulator is mounted on a slider that can move along the y-axis, the homogeneous transformation matrix from the fixed coordinate system to the base coordinate system is as follows:

$${}^bT_0 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & y_b \\ 0 & 0 & 1 \end{bmatrix}. \quad (5)$$

In this equation, the parameter y_b represents the distance the slider moves. Multiplying matrix y_b by the transformation matrix 0T_n gives the homogeneous transformation matrix from the fixed end of the manipulator to the manipulator's end-effector:

$${}^bT_n = {}^bT_0 \times {}^0T_n \quad (n = 6). \quad (6)$$

The position and orientation vectors of the end-effector are given by the relations:

$$\begin{cases} P_n = {}^bT_n [0 \quad 0 \quad 1]^T, \\ D_n = {}^bT_n [0 \quad 0 \quad 1]^T - P_n. \end{cases} \quad (7)$$

3.2. Dependence of Joint Bending Angle on Pressure in Controlling Pneumatic Chambers

Under a constant load, there is a unique relationship between the deformation of the controlling pneumatic chambers and the pressure inside them. During operation, the joint's bending angle depends on the following factors: the pressure applied to the air reservoirs on both sides of the spine; the physical characteristics of the reservoirs' expansion; and the structure of the connecting hinges. Different pressures and varying degrees of reservoir expansion in different sector-joints lead to different bending angles. The unique bending characteristics of each joint of the flexible snake-like manipulator were determined through physical experiments. Hysteresis is a property of systems whose instantaneous response to applied stimuli depends, in part, on their current state. Hysteresis is observed when the current state of a system is determined by external conditions not only at that

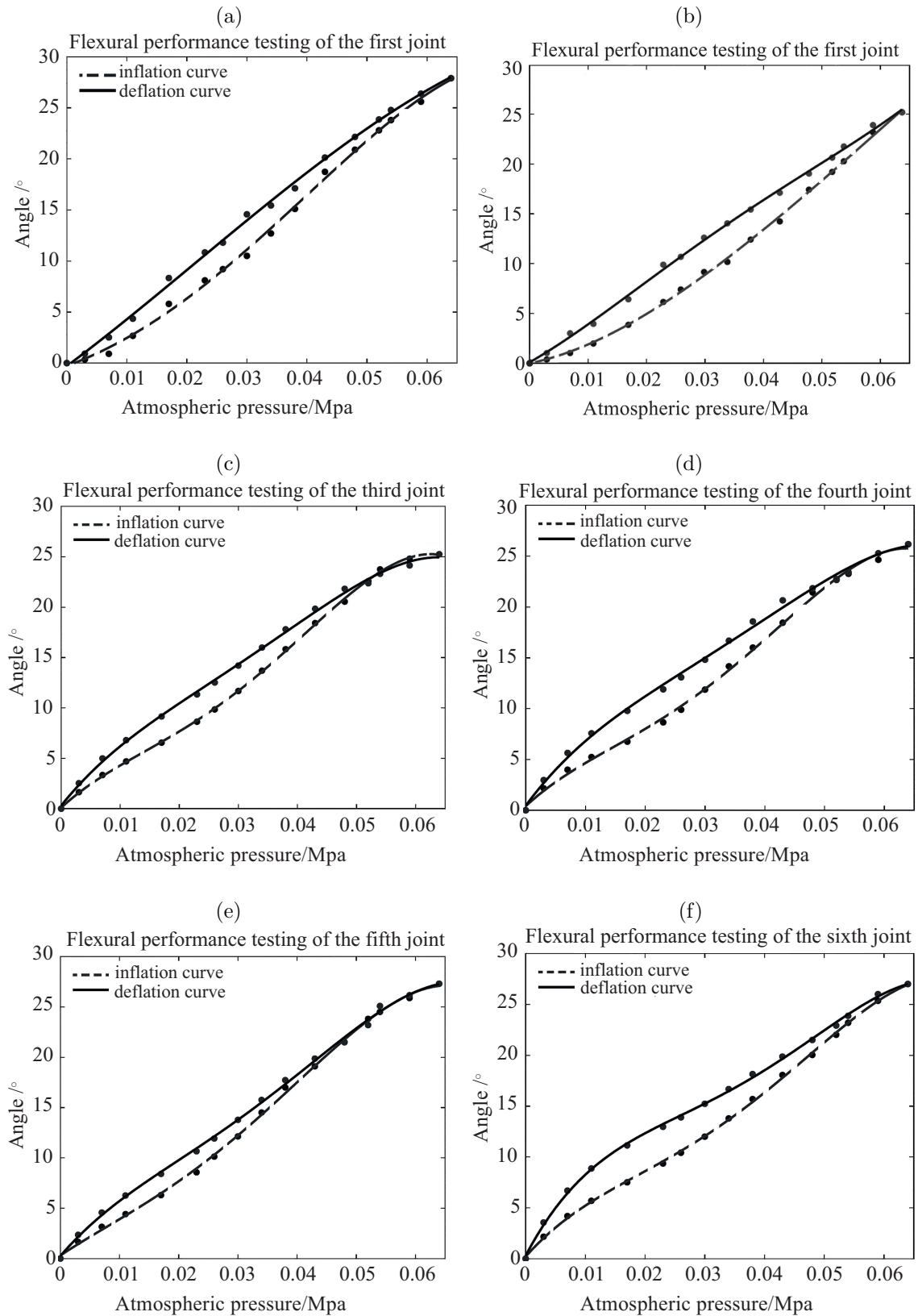


Fig. 3. Mapping of the relationship between joint bending angles and pressure in the controlling pneumatic chamber.

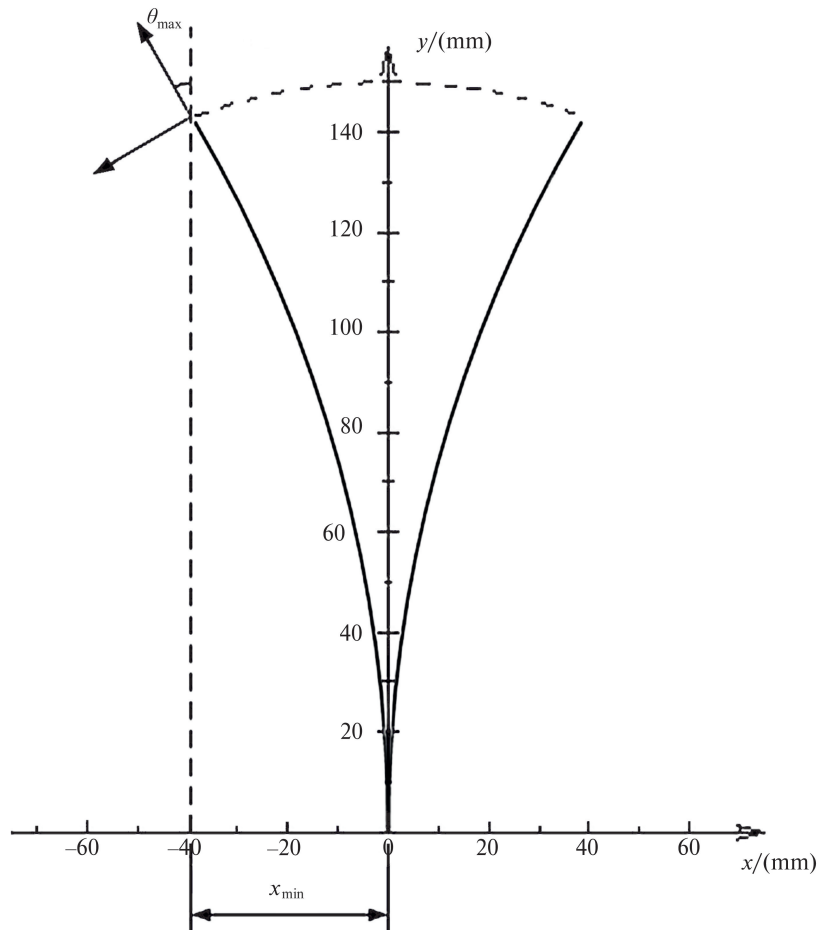


Fig. 4. Reachable range of the end joint.

moment but also at preceding moments in time. For the problem under consideration, this means that the pneumatic chamber's deformation is influenced by its historical states. If two actuated pneumatic chambers are involved in the joint motion, the dependency becomes more complex, and solving for the joint's bending angle becomes difficult. Therefore, the developed flexible manipulator uses a single pneumatic chamber control method, described as follows: When the joint bending angle increases on one side, the pressure in the pneumatic chamber on the opposite side increases. When the joint bending angle decreases on one side, the pneumatic chamber on the opposite side is depressurized and contracts. This means that one pneumatic chamber leads to a unilateral bend of the joint. This control method establishes a direct relationship between the joint's bending angle and the pressure in the controlling pneumatic chamber, which simplifies calculations and facilitates subsequent trajectory planning for the manipulator.

The setup consists of an STM32 microcontroller for regulating the valve, which controls the pressure in the air chambers to induce bending in the snake-like manipulator. A high-precision WT61C bending angle sensor is installed on the first vertebra of each segment to measure the joint's bending angle. Experiments were conducted by inflating the pneumatic chamber on one side while the opposite side remained unloaded. The air pressure range was set from 0 to 0.064 MPa. Several experiments were performed, and the resulting curves were approximated by a fourth-degree polynomial. As a result, "pressure-bending angle" relationships were obtained for all six joints, as shown in Fig. 3. The experimental results with two mirror-imaged pneumatic chambers of the same joint are practically identical; therefore, this article presents results only for one side of the robot. The discrepancy between the approximating "pressure-bending angle" relationship

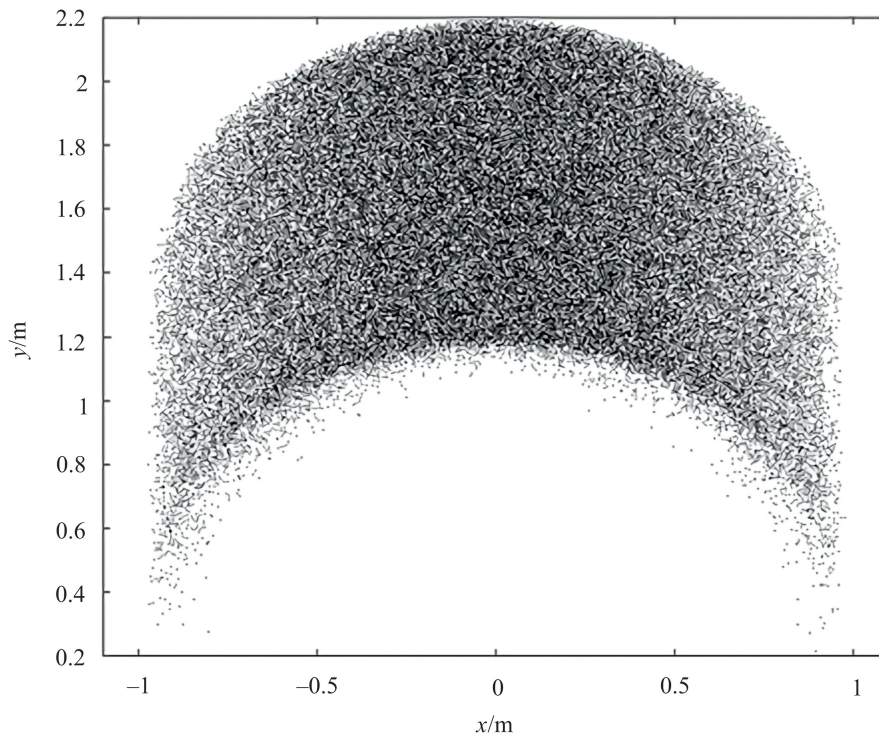


Fig. 5. Workspace of the flexible snake-like manipulator.

and the actual experimental values is less than 5%, reflecting the kinematic characteristics of the joint with high accuracy. The obtained curves serve as the basis for solving the forward and inverse kinematics problems of the robot when calculating the manipulator's end-effector position based on the joint bending angles. As seen in Fig. 3, at the same air pressure levels, the joint's bending angle during the inflation process is smaller compared to the deflation process. The maximum angular lag is about 5 degrees, indicating the pneumatic chamber's hysteresis characteristics. Moreover, due to structural differences between the flexible joints, small variations in corresponding bending angles are observed at the same air pressure, with a maximum deviation of about 3 degrees. Based on this, when solving the forward (inverse) kinematics problem between air pressure and joint bending angle, it is first necessary to determine the air pressure value (or joint bending angle) at the previous articulation, select one of the two experimental fit curves (for inflation or deflation), and then obtain the solution.

The robot's workspace includes the space that can be reached by the base point of the snake-like manipulator's wrist (last link) with its end-effector attached. The workspace is a key criterion for evaluating the manipulator's dynamic performance. Based on the analysis in the previous section, the rotation range of the flexible joints is $[-30^\circ, 30^\circ]$. The reachable range for the final vertebra of the last joint (head) is shown in Fig. 4. To model the workspace of the robot moving on a mobile slider, the Monte Carlo method was applied with a slider movement range of 1000 mm. In each range, 20 000 sets of random numbers were generated to calculate the end-effector's position using the forward kinematics equations. The resulting workspace is demonstrated in Fig. 5.

3.3. Calculating Bending Angles of a Joint Sequence

To solve the inverse kinematics problem for calculating the required bend of each joint, the approach of equal joint bending and linear direction compensation was used. To move the end-effector by an amount x_l in the x-direction (perpendicular to the slider's direction) and an amount y_l in the y-direction (along the slider's direction), the following steps must be performed:

- (1) Calculate the required displacement of a single joint in the x-direction by dividing the total displacement by the total number of joints.
- (2) Calculate the joint bending angle based on the obtained displacement value along the x-axis.
- (3) Calculate the displacement of a single joint along the y-axis when bent by the amount calculated in the previous step.
- (4) Calculate the total displacement of the end-effector along the y-axis due to bending by multiplying the displacement of a single joint by their total number.
- (5) Calculate the difference between the end-effector's displacement due to bending and the required displacement.
- (6) Move the guiding slider by the obtained value, compensating for the lack of robot displacement along the y-axis.

4. TRAJECTORY PLANNING BASED ON EMPIRICAL MODEL

4.1. Planning Based on End-Effector Position Tracking

Given the large number of joints in a pneumatic flexible manipulator, individual trajectory planning with obstacle avoidance for each position of every joint would significantly reduce the efficiency of inverse kinematics calculations. This work adopts a head-tracking strategy, where only the trajectory of the final vertebra of the last joint is tracked. This significantly increases the efficiency of trajectory planning while maintaining the manipulator's flexibility. For the developed flexible snake-like manipulator, the main challenge when using the head-tracking strategy is determining the actual movement trajectory of the end-effector. Essentially, this is the position of the flexible snake-like manipulator upon reaching the operational goal. The following conditions must be met:

- (1) The maximum curvature of the trajectory must be less than the curvature corresponding to the maximum joint bending angle.
- (2) The guiding slider must be tangent to the trajectory of the manipulator's first joint, and the end of the trajectory must reach the target point.
- (3) The total length of the trajectory is an integer multiple of the joint length.
- (4) The trajectory must not intersect obstacles.

Based on these requirements, this article considers the construction of end-effector movement trajectories for three different task scenarios.

4.1.1. Reaching a Specified End-Effector Position and Orientation. Based on the modeling and analysis from previous sections, it becomes clear that there is a connection between the end-points of the flexible sector-joints and the joint bending angles. When solving the inverse kinematics problem through path interpolation, there is a possibility of encountering points where no solution exists. To overcome this problem, the article uses a geometric method for finding suitable paths, as shown in Fig. 6. The method consists of the following:

- (1) Since the target point and direction of movement are predefined, the target point T can be taken as the starting point, and the direction opposite to the target direction ($-y_e$) as the initial direction. The path search is performed in reverse, starting from the end-effector and moving towards the robot's base. The direction of reverse movement y (see Fig. 4) is chosen as the direction opposite to y_T (see Fig. 6). This allows for defining the workspace for the end vertebra of the first sector-joint from the head. Within this area, the corresponding position is determined to sketch the first segment of the flexible joint.
- (2) The end-point position for the next joint is calculated as the solution to the following empirical optimization problem, consisting of stages A–D:

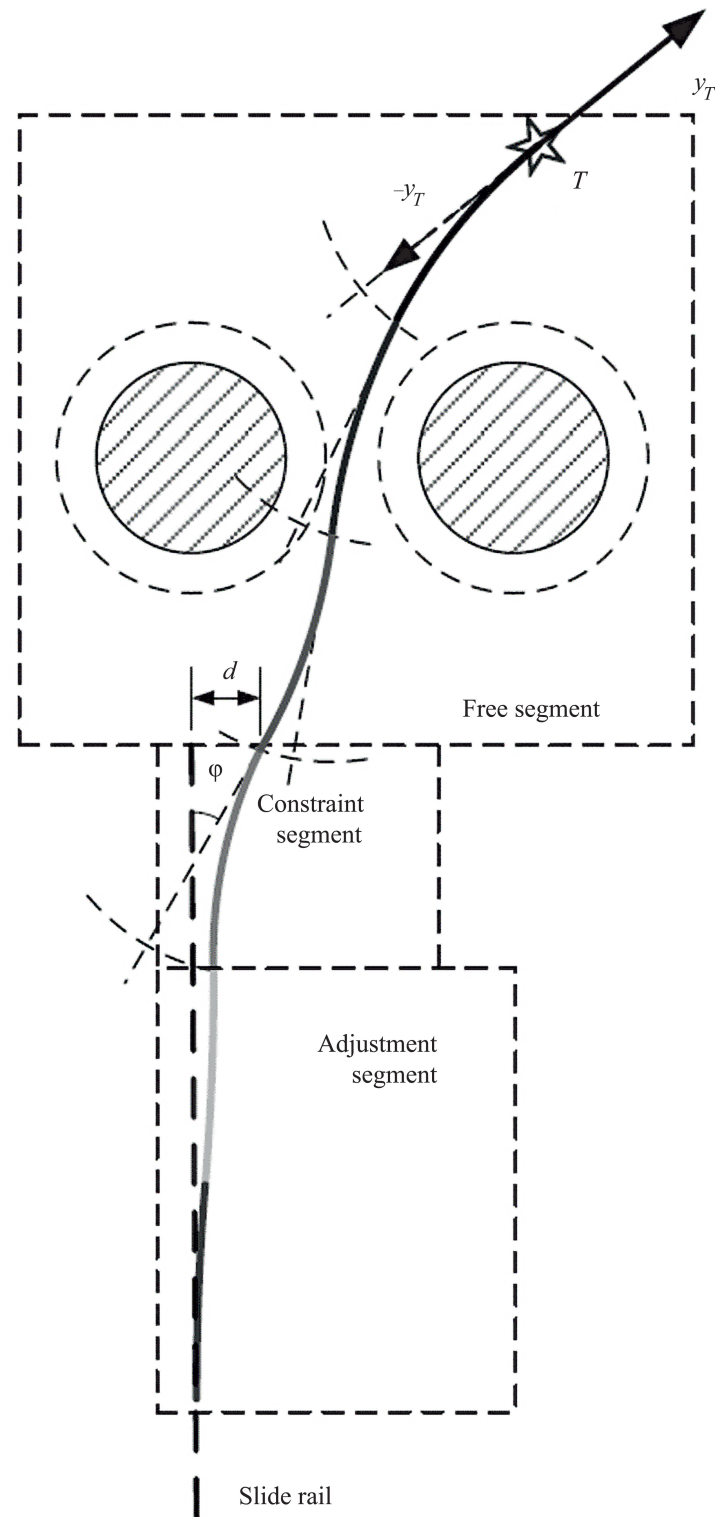


Fig. 6. Robot trajectory search method for a given end-effector position and direction.

(A) The joint's end-point does not intersect obstacles, i.e.:

$$P_1 \notin R_0 \quad (8)$$

Here: R_0 is the forbidden region (set of points of obstacle models). In Fig. 6, examples of forbidden regions are the two circles R_1 and R_2 .

- (B) The distance d from the end-point to the guiding slider's line must be minimal.
 - (C) If, as a result of applying the two above-mentioned principles, the distance d between the end-point and the guiding line is greater than $2x_{\max}$, optimality criterion B changes to D.
 - (D) The angle φ between the direction of the vertebra's end and the guiding line must be minimal.
- (3) The positions of subsequent segment-joints are determined similarly when moving from the end to the base of each joint. The choice of curves for mapping joint positions is made considering approximation to a straight line – the guiding line of the slider's movement without collisions with obstacles. These segments are here called “free segments” (see Fig. 6).
- (4) For the subsequent joints to reach the guiding mobile slider, on which the entire robot's base is attached, and for the guiding line to be tangent to the first vertebra of the base joint, the last joint of the free segment must satisfy the following conditions:
- (a) The continuation of the last joint's base direction must intersect the guiding line.
 - (b) The distance d from the base point to the guiding slider's line must not be greater than $x_{\max}$.
 - (c) The angle φ between the joint's base direction and the slider's guiding line must not be greater than $\theta_{\max}$.

If these conditions cannot be met, it becomes necessary to recalculate the joint angles for the free segment joints. If the requirement for distance d cannot be met, no acceptable solution exists. If the requirement for the deviation angle φ cannot be met, the angle deviation constraint should be introduced into the optimization problem during replanning to ensure that the segment end is sufficiently close to the guiding line while satisfying the maximum angle deviation requirement.

- (5) The next section will be called the “limiting segment.” It consists of a single flexible joint. Its bending angle is limited by the angle φ , with its base direction parallel to the guiding line.
- (6) The last section will be called the “adjustment segment.” It includes two joints. By setting the angles of these two joints equal in magnitude but opposite in direction, it's possible to ensure that the first joint's orientation remains unchanged when the distance from the guiding line changes. The bending angle must be chosen to ensure that the first vertebra of the robot's first joint passes tangentially to the guiding line.

At this point, the search for the end-effector's movement trajectory is complete. In this trajectory search method, the critical aspect is using the first three joints for the robot's initial orientation, as this is the minimum number capable of ensuring the correct manipulator position relative to the slider's guiding line. Consequently, only three joints of the free segment are capable of bypassing obstacles. Therefore, under these conditions, the developed flexible snake-like manipulator cannot perform complex obstacle avoidance maneuvers.

4.1.2. Fitting to a Given Trajectory. The method for fitting joint positions to a given trajectory is similar to the previous scenario. For the free segment, the joint ends that intersect the given trajectory are directly selected. The fitting method essentially aligns the free segment's joints along the trajectory. Therefore, the actual trajectory may have a rather large error compared to the planned one. For complex curves, if the end of the free segment does not satisfy the requirements listed above, trajectory fitting may be unattainable.

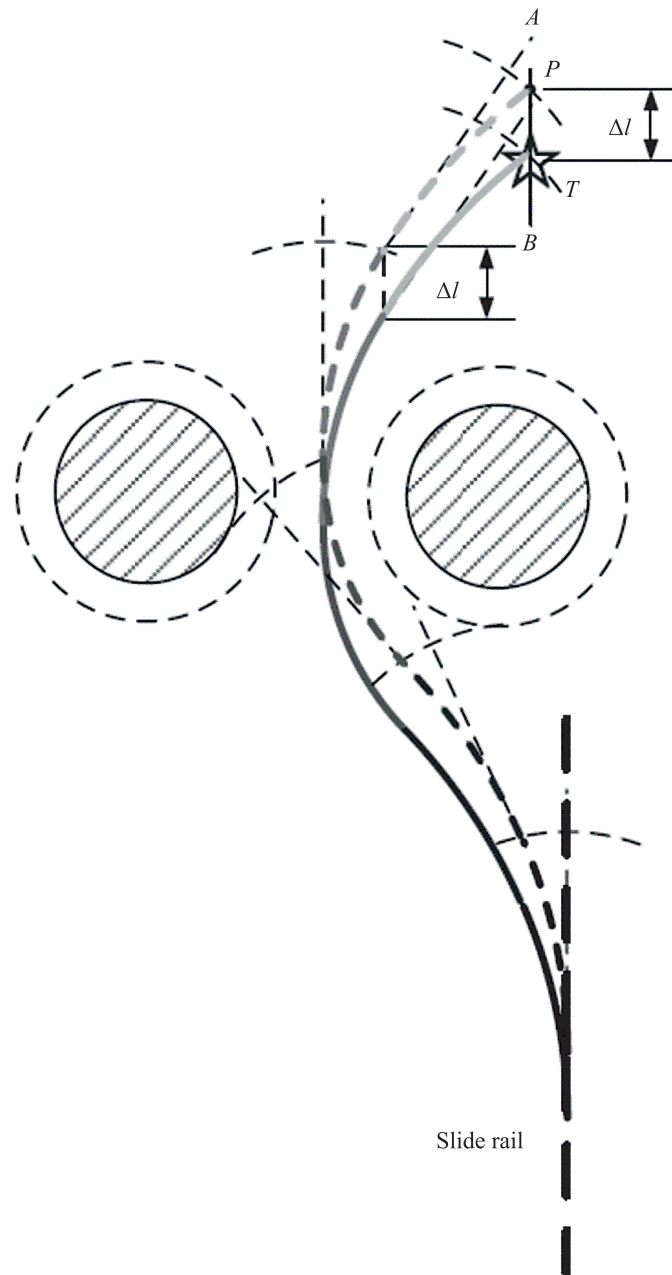


Fig. 7. Robot trajectory search method using permissible regions.

4.1.3. Reaching a Specified End-Effector Position. The trajectory planning methods described above for reaching a target point have limitations in maintaining the manipulator's flexibility and adaptability to the environment. This necessitates the development of a new method for searching the end-effector's trajectory. In accordance with the previously described trajectory planning stages, it is clear that the main problem is achieving a position for the robot's first joint such that the slider's guiding line is tangent to the joint's curve at the robot's attachment point. In many cases, this requires at least three articulations. Given the unfixed orientation of each segment's end, planning to reach a given position, starting from the guiding line, can be done from the base to the segment's end, which increases the number of free joints and simplifies the trajectory planning process.

Let the permissible regions for the base point's location be defined for the base of each segment. Then the planning steps are as follows (see Fig. 7).

- (1) Let the target point of the trajectory be denoted by T . A straight line AB is drawn through point T parallel to the slider's guiding line.
- (2) At a distance no greater than the joint length L from line AB , a point from the permissible region is selected as the temporary starting point for the last joint's bending trajectory.
- (3) The reachable area for the end of the first joint from the temporary starting point is determined.
- (4) The position of the joint's end-point in the reachable area is calculated based on constructing the optimization problem $A-D$, where distance d is replaced by the distance between the end and target points.
- (5) Calculations for subsequent joints are repeated similarly, moving towards the end of each joint.
- (6) When the joint's end is close to the target point ($d \leq 2L$), optimality criterion B changes to D: the angle φ between the final direction and the line connecting the end point and the target point must be minimal.

After the algorithm completes, a bending trajectory for the robot's joints will be obtained, ensuring the achievement of an alternative target point P . The distance between the alternative P and the actual T target points along the line AB is given by Δl . To reach the true target point, the entire trajectory curve needs to be shifted by the amount Δl along the direction AB . If, after the shift, the trajectory intersects obstacle models, it is necessary to plan the trajectory again by selecting different points from the permissible regions at each step as temporary base points for the joints. If there are no intersections, planning is considered complete. This method, compared to previous methods, allows the robotic manipulator to use more joints to directly perform the given tasks, as all joints are utilized to reach the target point and avoid obstacles. Consequently, the flexible robot can effectively perform tasks in more complex conditions, leading to an increased percentage of successfully completed tasks.

4.2. Trajectory Planning Based on Passive Geometric Shape Change

Given the limited force applied for pneumatic chamber deformation, it should be noted that the range of motion of the manipulator's end-effector during active joint bending is often limited. At the same time, the snake-like manipulator possesses flexibility and extensive capabilities for passive geometric shape change. When the sector-joints come into contact with obstacles in the external environment, the manipulator can autonomously bypass these obstacles through its inherent flexibility. This allows the manipulator to adapt to various complex tasks and operating conditions. In the example shown in Fig. 8, during the flexible manipulator's movement, active bending was

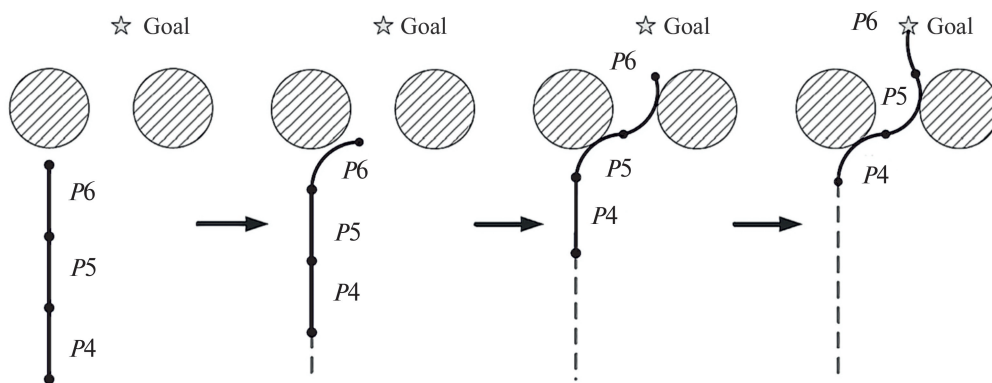


Fig. 8. Trajectory planning based on passive geometric shape change.

only required for the final joint (head). The remaining joints bend upon contact with obstacles and complete their movement solely due to passive bending. The formula for calculating the active bending angle of the last i th joint is:

$$\theta_i = CP_{i-1}L \cdot \arccos\left(\frac{FD_{i-1}}{F \cdot D_{i-1}}\right). \quad (9)$$

where C is the contact point with the obstacle, and F is the tangent vector to the contact surface at the base mark. This strategy effectively utilizes the snake-like manipulator's flexibility, simplifies the trajectory planning process, and consequently, the solution of the given task, but it can lead to frequent collisions with environmental elements. Therefore, this approach is effective in scenarios where the robotic manipulator does not cause damage upon contact with an obstacle or where safety requirements are relatively low.

5. EXPERIMENTS

5.1. Experiment Preparation

The snake-like manipulator is equipped with 12 pneumatic actuating chambers for 6 joints and 12 proportional pneumatic control valves connected to a compressor. The robot is mounted on a linear slider, allowing it to extend and retract by controlling the movement of the slider's linear support. Figure 9 shows the composition of the robotic manipulator's control system. A personal computer (PC) serves as the host for the main control program, written in C language. It communicates with the microcontroller unit (MCU) via a serial communication channel to transmit data. Based on a predefined manipulator motion strategy, the computer calculates the bending angles of each flexible joint at different times and determines the linear guide's displacement. The necessary air pressure values in the 12 control air chambers are determined based on the relationship between air pressure and joint bending angles. The MCU then converts the calculated values into corresponding electrical signals and transmits them to the proportional valves and stepper motors. As a result, the snake-like manipulator performs various movements and obstacle avoidance actions. Two series of experiments were conducted: moving the end-effector along a predefined trajectory without obstacles to investigate its performance, and moving the end-effector to a target point with obstacles.

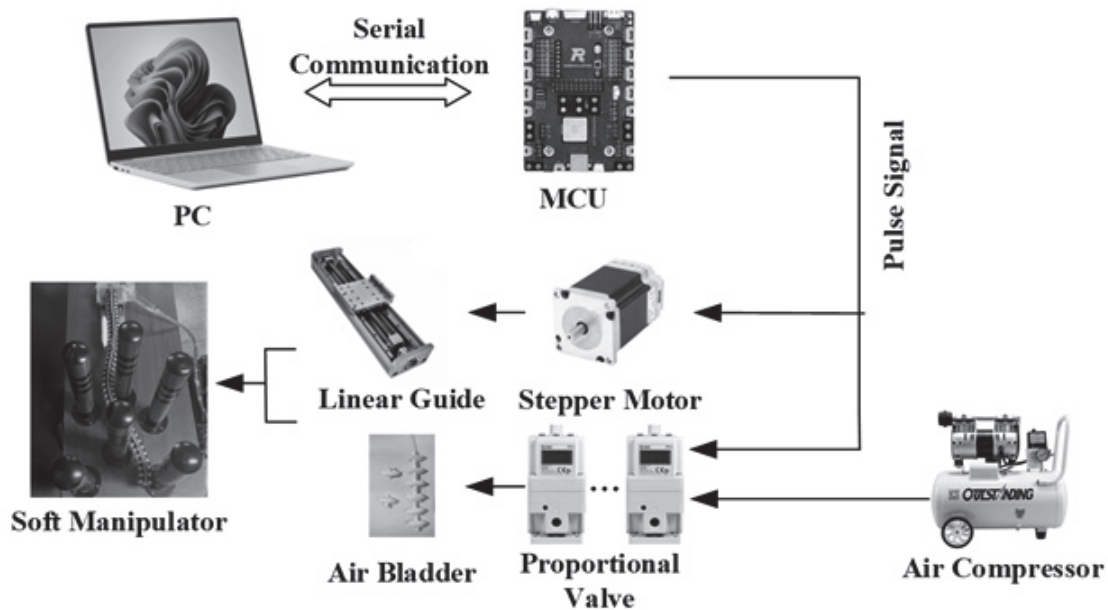


Fig. 9. Schematic diagram of the flexible manipulator's control system layout.

5.2. Experiments on Movement Along a Predefined Trajectory Without Obstacles

This experiment involves creating specific end-effector movement trajectories to test the overall performance of the flexible snake-like manipulator, analyze its movement capabilities, and evaluate its potential application in two-dimensional operations. In these experiments, movement along three trajectories was tested: a straight line, a rectangle, and a circle. A laser is mounted on

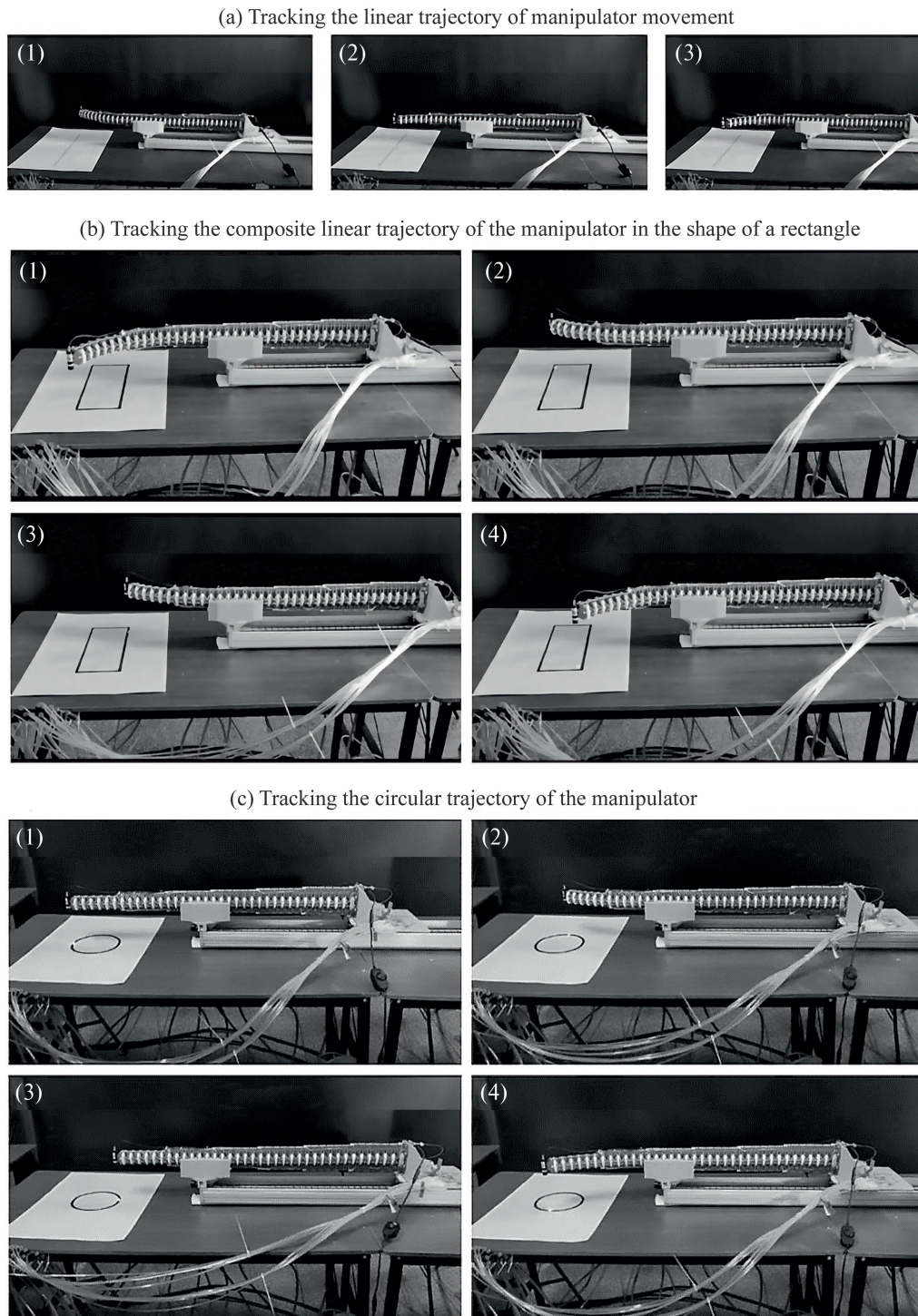


Fig. 10. Experimental results for controlling end-effector movement without obstacles along a given trajectory.

the manipulator's end-joint to track the end-effector's movement trajectory and calculate the error between the actual and predefined trajectories. Figure 10 demonstrates the progress of the three experiments without obstacles: a) tracking a linear trajectory, b) tracking a composite linear trajectory in the shape of a rectangle, and c) tracking a circular trajectory. The error was defined as the deviation of the actual trajectory from the predefined one at the final time point (the end point of a straight line, the corner of a rectangle, and the end point of a circle). As a result, the maximum deviation of the soft manipulator's final position did not exceed 1%, and the deviation of the joint angle in the real experiment from the calculated value did not exceed 2%. This confirms the high effectiveness and accuracy of the flexible snake-like manipulator's movement in two-dimensional space.

5.3. Experiments on Active Obstacle Avoidance Based on End-Effector Position Tracking

The obstacle avoidance experiment, shown in Fig. 11, involves two parallel support pillars as obstacles, and the end-effector is a steel needle attached to the manipulator's last vertebra. The target is a balloon located behind the obstacles. Based on the previously proposed path planning algorithm, the manipulator's end-effector must bypass the obstacles to reach the balloon's position and puncture it. The predefined experimental scenario corresponds to the task scenarios described earlier. Based on the end-effector trajectory planning methods described in previous sections, the trajectories shown in Fig. 11 were constructed. Analyzing the constructed trajectories, it can be concluded that using the method with achieving a specified position and orientation requires a longer guiding slider and involves a greater number of joints. Since there are no strict requirements for the end-effector's position in this experiment, a trajectory corresponding to the method of achieving a specified position was chosen. The speed of the moving slider was set at 25 mm/s, and the time required to bypass the obstacles was set at 20 seconds. To calculate the time-

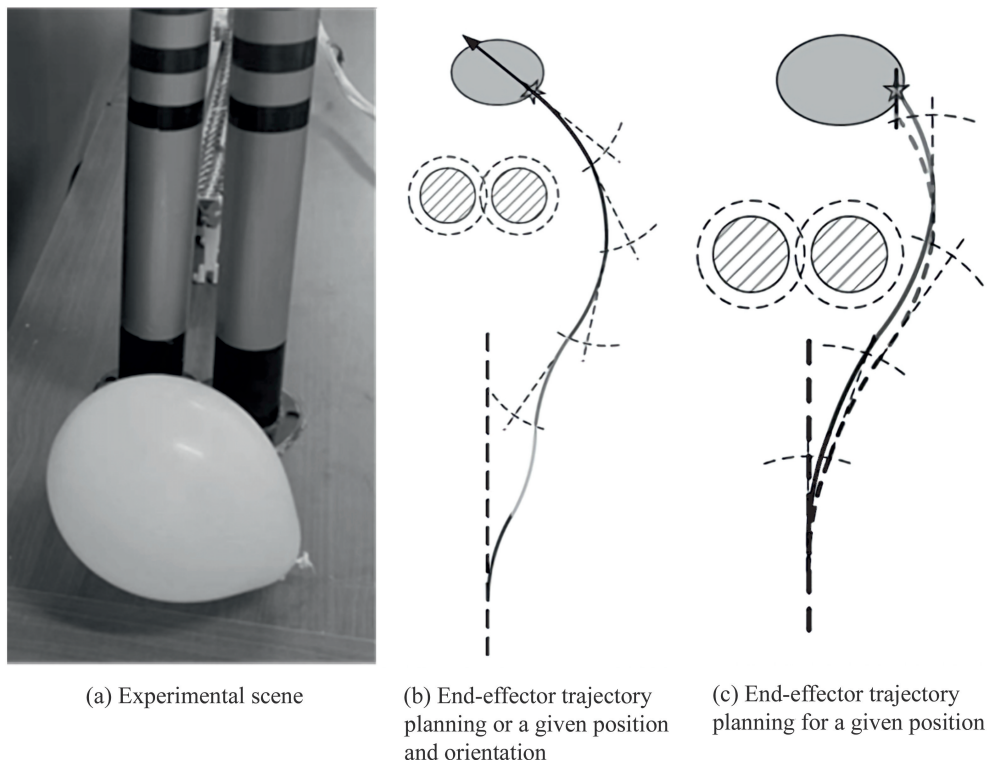


Fig. 11. Experiment on active obstacle avoidance.

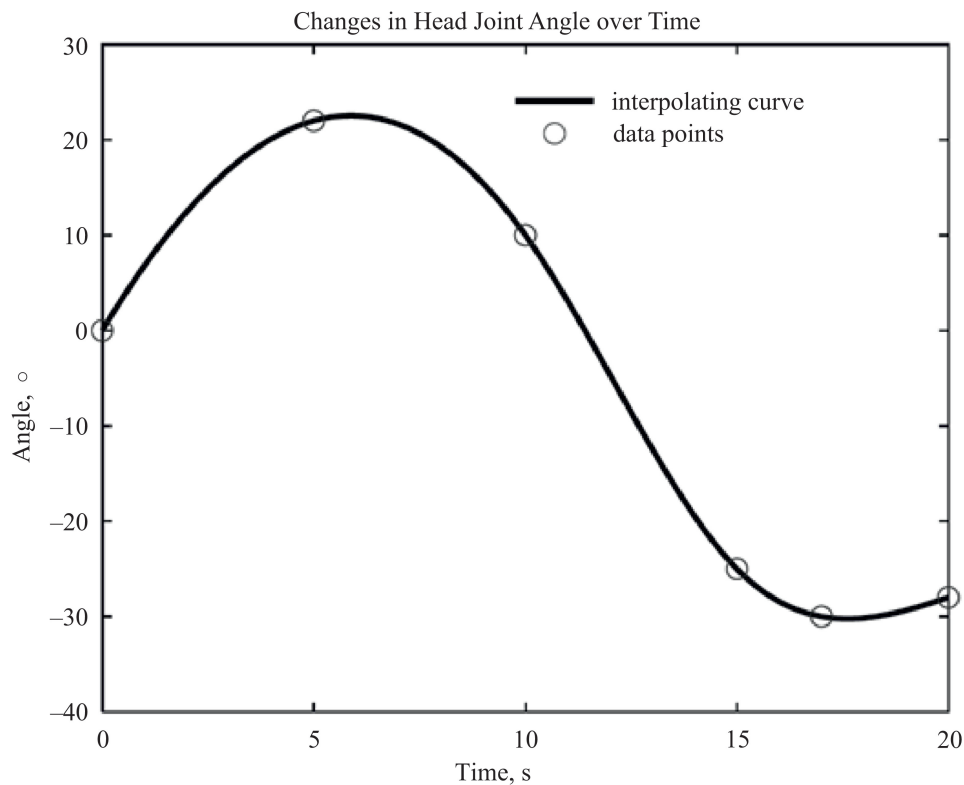


Fig. 12. Calculation of the end-joint angle change trajectory during obstacle circumvention.

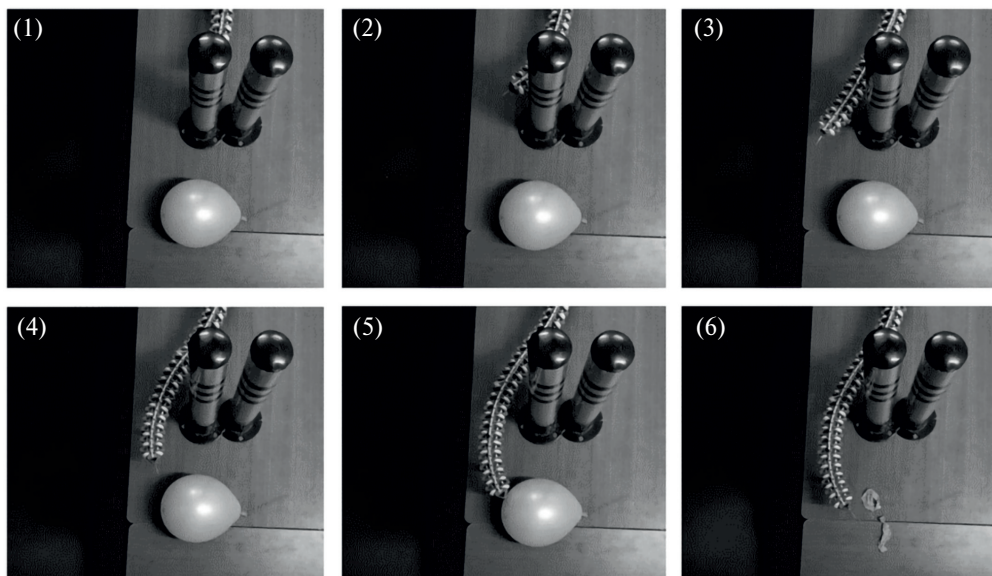


Fig. 13. Experiment on the movement of a flexible snake-like manipulator through active geometric shape change, bypassing a single obstacle.

varying curvature of the end-joint carrying the effector, cubic spline interpolation was applied to four calculated angular positions of the end-effector, as shown in Fig. 12. At the 17-second mark, an additional point of 30° was added to prevent the joint from exceeding its allowable bend. Obstacle avoidance by the flexible manipulator is achieved based on the obtained curve of the end-joint's angle change using the kinematic model and the end-effector tracking strategy. Figure 13

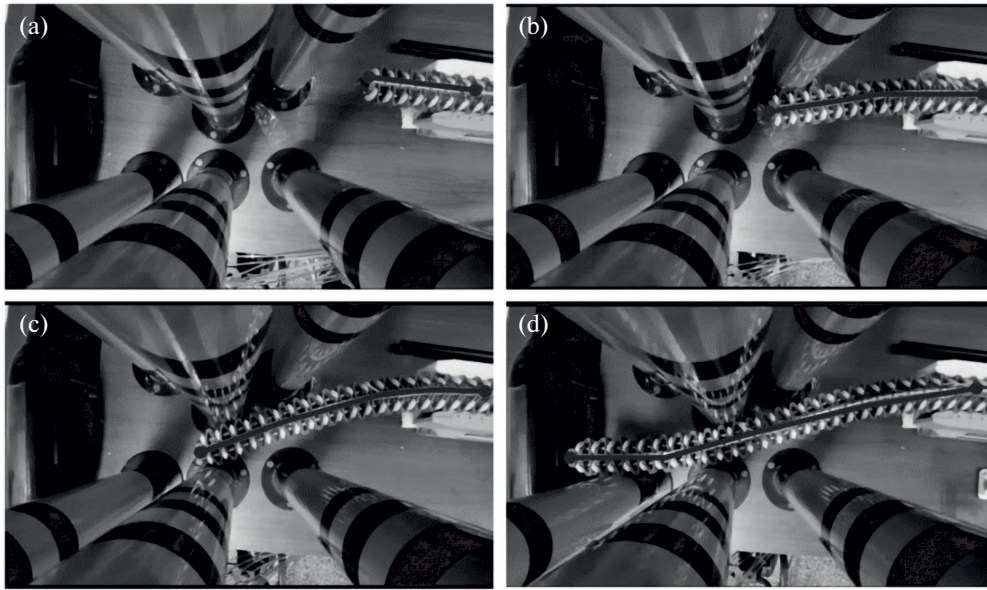


Fig. 14. Experiment on the movement of a flexible snake-like manipulator based on passive geometric shape change, bypassing a group of obstacles.

demonstrates the experimental setup for movement through active geometric shape change of the flexible manipulator during obstacle avoidance. The manipulator is initially positioned in front of the obstacle, moves around the obstacle, bypassing it from the right as it moves, and punctures the balloon located behind the obstacle. It is evident that the snake-like manipulator successfully completed its movement, bypassing the obstacles along the predefined trajectory, with joint angle deviations from the calculated values of no more than 5%.

5.4. Experiment on Obstacle Avoidance Based on Passive Geometric Shape Change

For obstacle avoidance scenarios based on passive geometric shape change, as proposed above, control is only required to calculate the trajectory of the head joint with the effector to ensure the manipulator moves in the desired direction between obstacles. Figure 14 shows the progress of the experiment on soft manipulator movement based on passive geometric shape change to bypass a group of obstacles. Changes in the joint bending angles during movement are also illustrated in Fig. 14. As a result of the experiment, the snake-like manipulator successfully completed its movement, bypassing multiple obstacles and reaching the designated target position, using its active and passive capabilities for geometric change.

6. COMPARATIVE ANALYSIS OF THE PROPOSED APPROACH WITH KNOWN METHODS

This study proposes an original design for a flexible snake-like manipulator and a methodology for constructing its trajectory planning system based on real experiments. To analyze the effectiveness of the developed design and the corresponding trajectory planning system, a comparison of the proposed approach with known designs and methods for calculating flexible manipulator trajectories was conducted.

6.1. Comparison of Designs

A comparison of the features of the developed flexible snake-like manipulator design with known counterparts is presented in Table 1.

Table 1. Characteristics of Flexible Manipulators of Various Designs

Manipulator Type	Construction Type	Actuation Type	Degrees of Freedom	Geometric Shape Change Method	Attachment /Movement Method	Trajectory Calculation Method
Industrial Robotic Arm (KUKA LBR IIWA, Rozum PULSE, etc.)	Constant stiffness (high stiffness)	Mechanical	3–7	Active	On a stationary stand	Trajectory generation based on industrial software
Multi-joint Robotic Arm [1]	Constant stiffness (medium stiffness)	Cable-driven	5–20	Active	On a stationary stand	No information
Super-soft Liquid Elastomer Manipulator [21]	Constant stiffness (low stiffness)	Hydraulic	15	Passive /Active	On a stationary suspension	Trajectory calculation based on position tracking with computer vision
Soft Manipulator based on Jamming [29]	Variable stiffness	Pressure-hardening granulate	–	Active	On a mobile carrier/slider	Trajectory generation based on impulse commands
Proposed Flexible Snake-like Manipulator Model	Variable stiffness	Pneumatic	16	Passive /Active	On a slider	Trajectory calculation based on experimental studies

As Table 1 illustrates, the flexible snake-like manipulator design presented in this work has several advantages over its counterparts, particularly the ability for passive geometric change, variable stiffness, and a large number of degrees of freedom. A key feature is the specific trajectory calculation algorithm, which is developed based on the physical data of the specific snake-like manipulator obtained through a series of experiments.

6.2. Comparison of Trajectory Planning Approaches

Let's consider several approaches for trajectory planning described in modern literature.

- **Trajectory planning for a multi-joint redundant manipulator based on a genetic algorithm [37]:** The manipulator's trajectory is represented by the coordinates of its nodes, with the end node associated with the end-effector. The trajectory must ensure minimal discrepancy between the end-effector's position and the target point. The trajectory is formed as a solution to an optimization problem where the objective function is the deviation of the end-effector's coordinates from the target coordinates, and the constraints are the requirements that the manipulator's links do not intersect obstacles or each other, and that joint rotation angles are permissible. Optimization is performed using a genetic algorithm, where trajectories act as chromosomes and node coordinates as genes. The algorithm is repeated many times to obtain

a “genealogical tree” – a sequence of suitable trajectories that move the manipulator from a given initial position to a final one.

- **Manipulator trajectory calculation based on an analytical solution to the inverse kinematics problem [32]:** The rotation angles of a three-link manipulator (shoulder-forearm-gripper) are determined based on the required position of the last link (gripper). The method involves sequentially formulating and solving systems of nonlinear equations reflecting the dependencies between the rotation angles of subsequent and preceding links. Due to the relatively simple configuration and limitation on the number of links, the problem has a unique solution.
- **Multi-joint manipulator trajectory calculation based on the FABRIK (Forward and Backward Reaching Inverse Kinematics) algorithm [36]:** The trajectory of a multi-joint manipulator is calculated according to repeated forward and backward iterations – passes. The initial node is considered rigidly fixed, and the end-effector is conditionally located at the last node. In the forward pass, the nodes move from the end-effector to the initial node. The last node, with the end-effector, moves to the position specified as the target. Then each subsequent node moves to a position lying on the straight line connecting it and the previous, already shifted, node, at a distance equal to the link length. As a result, after all nodes have shifted sequentially, the initial node will be offset from its original position. Since the initial node is considered fixed and its position cannot change, the initial node’s position is corrected – it returns to its initial position. After this, a backward pass is performed – all manipulator nodes are shifted again, but now from the initial to the end node, following the same principle as in the forward pass. Forward and backward passes are repeated until the end-effector is sufficiently close to the given target position.

A comparison of the characteristics of the reviewed methods, as well as the proposed method based on an empirical approach, is presented in Table 2.

Table 2. Features of Trajectory Generation Methods for Multi-joint Manipulators

	Method Group	Algorithm Type	Number of Manipulator Links	Workspace Dimension	Consideration of Specific Manipulator Features	Obstacle Avoidance	Structural Movement
Trajectory calculation based on genetic algorithms	Random Search	Single-pass, iterative	Unlimited	3D	No	Yes	Requires algorithm modification
Analytical trajectory calculation	Analytical Model	Single-pass, non-iterative	2–3	3D	Partially	No	No
Forward and Backward Reaching Inverse Kinematics (FABRIK)	Heuristic Algorithm	Two-pass, iterative	Unlimited	3D	No	Requires algorithm modification	No
Trajectory calculation based on empirical approach	Empirical Algorithm	Two-pass, iterative	Unlimited	2D	Yes	Yes	Yes

The proposed method, unlike other existing trajectory generation methods, is tailored to a specific manipulator, as the calculation is performed based on empirical data obtained through experiments. The analytical method, which also involves calculations for a specific manipulator, is

based on theoretical, but not practically measured, device characteristics. The proposed algorithm is not designed for changing the manipulator's position in three-dimensional space, but it can account for the movement of the initial node due to the mobile slider.

6.3. Comparison of Computational Costs

To compare the proposed algorithm with other manipulator trajectory generation algorithms in terms of computational costs, we will evaluate the computational time complexity for the respective algorithms. Computational time complexity is estimated by a function that expresses the number of required operations as a function of the input data size n . This section presents values of time complexity as $n \rightarrow \infty$, i.e., asymptotic computational time complexity [41].

- **Assessment of Genetic Algorithm computational complexity:** With typical choices of transformation methods (point mutation, single-point crossover, roulette wheel selection), the complexity of genetic algorithms is $O(g(nm + nm + n))$, where g is the number of generations, n is the population size, and m is the size of individuals. Thus, the computational complexity of the algorithm can be estimated as $O(n^2)$.
- **Assessment of Analytical Trajectory Calculation Method computational complexity:** The computational complexity of the analytical method is the lowest. Since the algorithm does not involve iterative calculations, the computational volume remains constant, and the complexity is $O(1)$.
- **Assessment of Forward and Backward Reaching Inverse Kinematics (FABRIK) algorithm computational complexity:** The FABRIK algorithm consists of inner and outer loops. In the inner loop, forward and backward passes of the algorithm are sequentially performed. Each such pass computationally generates a decomposition tree of depth $\log_2 n$. Consequently, the

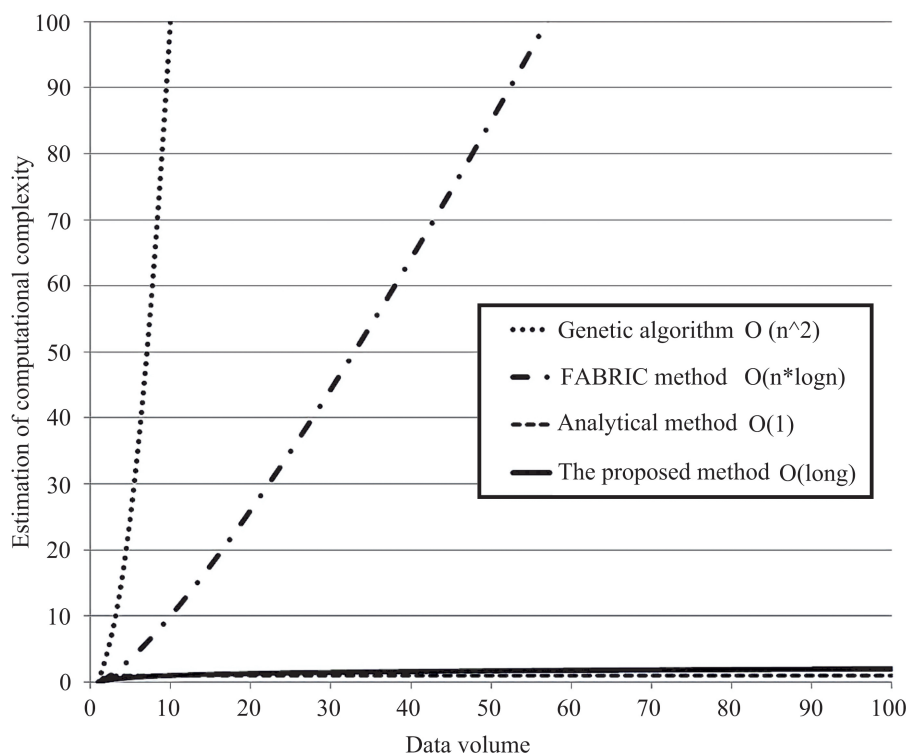


Fig. 15. Growth of computational complexity with increasing data volume for the reviewed trajectory calculation methods of flexible manipulators.

complexity of the inner loop with a double pass is $O(n \log_2 n)$, which for $n \rightarrow \infty$ is equivalent to a complexity of $O(n)$. The outer loop of the algorithm repeats the inner loop a number of times proportional to the input data volume n . Therefore, the computational complexity is estimated as $O(n^2)$.

- **Assessment of Trajectory Calculation Algorithm Based on Empirical Approach computational complexity:** The proposed algorithm, unlike the FABRIK algorithm discussed above, consists of a single loop. There is no outer loop, as the end link is located on a slider and can change its position. Consequently, the complexity of the proposed algorithm is $O(n)$.

A graphical comparison of the computational complexity estimates is presented in Fig. 15.

7. CONCLUSION

This article, building upon international research in soft robotic manipulators, presents the design and development of a 6-joint, 13-degree-of-freedom snake-like flexible manipulator. Its design is based on the anatomical structure of a snake's spine. The overall configuration features a series-parallel composite structure, controlled by 12 sets of pneumatic chambers on both sides of the manipulator and a linear guide-slider. We performed a kinematic mapping from the manipulator's joint space to its workspace using a constant curvature model. Through experiments, the relationship between air pressure in the pneumatic chambers and the joint bending angle was established. A unilateral movement method along a linear slider was employed to model the robot's forward kinematics. To meet the requirements of practical engineering applications for a flexible manipulator, we developed trajectory planning methods for obstacle avoidance. Two methods for calculating the robot's movement trajectory were proposed based on the end-effector position tracking approach. We also analyzed a method for obstacle avoidance based on the passive geometric shape change of the soft robot. The conducted physical experiments confirmed the feasibility and correctness of the developed methods. A comparison of the proposed flexible manipulator design approach with existing design solutions demonstrated the originality of the developed structure. A key advantage of the designed snake-like flexible manipulator is its ability to passively change its geometric shape during movement for obstacle avoidance, as well as its variable stiffness with a large number of degrees of freedom. The manipulator is designed for movement and operation on a planar surface, making its application preferable in a range of practical tasks. The algorithm developed for calculating the manipulator's trajectory is based on experimental measurements of the characteristics of the pneumatic chamber-joints, which enable the bending and straightening of the flexible snake-like manipulator. This algorithm has low computational complexity, surpassed only by analytical modeling, which, in turn, cannot offer broad trajectory modification capabilities. Future research plans include further expansion of studies on the design, trajectory planning, and control of the flexible snake-like manipulator. Specifically, we will investigate control methods that allow for variable stiffness of the manipulator. Additionally, the manipulator may be equipped with multiple integrated sensors for real-time position tracking, which will enhance the robot's response speed and control accuracy.

REFERENCES

1. Larson, O. and Davidson, C., Flexible arm, particularly a robot arm, U.S. Patent No. 4,393,728, 19 Jul. 1983.
2. Hannan, M.W. and Walker, I.D., Kinematics and the implementation of an elephant's trunk manipulator and other continuum style robots, *J. Field Robot.*, 2003, vol. 20, no. 2, pp. 45–63.
3. McMahan W., Jones B., and Walker I. D. Design and implementation of a multi-section continuum robot: Air-Octor. *2005 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2005. P. 2578–2585.

4. Robert, B., Robots in the nuclear industry: a review of technologies and applications, *Ind. Robot.*, 2011, vol. 38, no. 2, pp. 113–118.
5. Rob, B. and Andrew, G., Nuclear snake-arm robots, *Ind. Robot.*, 2012, vol. 39, no. 1, pp. 6–11.
6. Camarillo, D.B., Milne, C.F., Carlson, C.R., et al., Mechanics modeling of tendon-driven continuum manipulators, *IEEE Trans. Rob.*, 2008, vol. 24, no. 6, pp. 1262–1273.
7. Xu, W., Liu, T., and Li, Y., Kinematics, dynamics, and control of a cable-driven hyper-redundant manipulator, *IEEE/ASME Trans. Mechatronics*, 2018, vol. 23, no. 4, pp. 1693–1704.
8. Rolf, M. and Steil, J., Constant curvature continuum kinematics as fast approximate model for the bionic handling assistant, *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2012, pp. 3440–3446.
9. Yang, G.Z., Bellingham, J., Dupont, P.E., et al., The grand challenges of Science Robotics, *Sci. Robot.*, 2018, vol. 3, no. 14, pp. eaar7650.
10. Dumais, J. and Forterre, Y., Vegetable dynamics: the role of water in plant movements, *Annu. Rev. Fluid Mech.*, 2012, vol. 44, pp. 453–478.
11. Argiolas, A., Puleo, G.L., Sinibaldi, E., et al., Osmolyte cooperation affects turgor dynamics in plants, *Sci. Rep.*, 2016, vol. 6, pp. 30139.
12. Chen, Z.C., Yamaji, N., Fujii-Kashino, M., et al., A cation-chloride cotransporter gene is required for cell elongation and osmoregulation in rice, *Plant Physiol.*, 2016, vol. 171, pp. 494–507.
13. Margheri, L., Laschi, C., and Mazzolai, B., Soft manipulator inspired by the octopus: I. From biological functions to artificial requirements, *Bioinspir. Biomim.*, 2012, vol. 7, no. 2, pp. 1–12.
14. Wirekoh, J. and Park, Y.L., Design of flat pneumatic artificial muscles, *Smart Mater. Struct.*, 2017, vol. 26, no. 3, pp. 035009.
15. Rodrigue, H., Wang, W., Han, M.W., et al., An overview of shape memory alloy-coupled actuators and robots, *Soft Robot*, 2017, vol. 4, pp. 3–15.
16. Miriyev, A., Stack, K., and Lipson, H., Soft material for soft actuators, *Nat. Commun.*, 2017, vol. 8, no. 1, pp. 596.
17. Mirvakili, S.M., Sim, D., Hunter, I.W., et al., Actuation of untethered pneumatic artificial muscles and soft robots using magnetically induced liquid-to-gas phase transitions, *Sci. Robot.*, 2020, vol. 5, no. 41, pp. eaaz4239.
18. Neppalli, S., Jones, B., McMahan, W., et al., OctArm - A soft robotic manipulator, *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2007, pp. 2569–2569.
19. Hao, J., Wang, Z., Jin, Y., Chen, X., Peijin, L., Gan, Y., and Chen, X., Hierarchical control of soft manipulators towards unstructured interactions, *Int. J. Robot. Res.*, 2021, vol. 40, pp. 027836492097936.
20. Escande, C., Chettibi, T., Merzouki, R., Coelen, V., and Pathak, P.M., Kinematic calibration of a multisection bionic manipulator, *IEEE/ASME Trans. Mechatronics*, 2015, vol. 20, no. 2, pp. 663–674.
21. Marchese, A.D. and Rus, D., Design, kinematics, and control of a soft spatial fluidic elastomer manipulator, *Int. J. Robot. Res.*, 2015, vol. 35, no. 7, pp. 840–869.
22. Yan, J., Zhang, X., Xu, B., and Zhao, J., A new spiral-type inflatable pure torsional soft actuator, *Soft Robot.*, 2018, vol. 5, no. 5, pp. 527–540.
23. Thuruthel, T.G., Shih, B., Laschi, C., and Tolley, M.T., Soft robot perception using embedded soft sensors and recurrent neural networks, *Sci. Robot.*, 2019, vol. 4, no. 26, pp. eaav1488.
24. Greer, J.D., Blumenschein, L.H., Alterovitz, R., Hawkes, E.W., and Okamura, A.M., Robust navigation of a soft growing robot by exploiting contact with the environment, *Int. J. Robot. Res.*, 2020, vol. 39, no. 14, pp. 1724–1738.
25. Gong, Z., Fang, X., Chen, X., Cheng, J., Xie, Z., Liu, J., and Wen, L., A soft manipulator for efficient delicate grasping in shallow water: Modeling, control, and real-world experiments, *Int. J. Robot. Res.*, 2021, vol. 40, no. 1, pp. 449–469.

26. Kurumaya, S., Phillips, B.T., Becker, K.P., Rosen, M.H., Gruber, D.F., Galloway, K.C., Suzumori, K., and Wood, R.J., A modular soft robotic wrist for underwater manipulation, *Soft Robot.*, 2018, vol. 5, pp. 399–409.
27. Kaufmann, J., Bhovad, P., and Li, S., Harnessing the multistability of kresling origami for reconfigurable articulation in soft manipulators, *Soft Robot.*, 2022, vol. 9, no. 2, pp. 212–223.
28. Schubert, B.E. and Floreano, D., Variable stiffness material based on rigid low-melting-point-alloy microstructures embedded in soft poly(dimethylsiloxane) (PDMS), *RSC Adv.*, 2013, vol. 3, no. 46, pp. 24671–24679.
29. Brown, E., Rodenberg, N., Amend, J., et al., Universal robotic gripper based on the jamming of granular material, *Proc. Natl. Acad. Sci.*, 2010, vol. 107, no. 44, pp. 18809–18814.
30. Kim, Y.J., Cheng, S., Kim, S., et al., A novel layer jamming mechanism with tunable stiffness capability for minimally invasive surgery, *IEEE Trans. Robot.*, 2013, vol. 29, no. 4, pp. 1031–1042.
31. Hao, J., Wang, Z., Jin, Y., Chen, X., Peijin, L., Gan, Y., and Chen, X., Hierarchical control of soft manipulators towards unstructured interactions, *Int. J. Robot. Res.*, 2021, vol. 40, pp. 027836492097936.
32. Bratchikov, S.A., Abramova, E.A., and Fedosov, Yu.V., Solution of the inverse kinematics problem of the manipulator, *Bull. Tomsk State Univ., Manag., Comput. Inform.*, 2021, vol. 56, pp. 4–11. <https://doi.org/10.17223/19988605/56/1>
33. Molotkov, A.V. and Chelnokov, Yu.N., Solution of the inverse problem of kinematics of the robot-manipulator “Puma” using the biquaternion theory of kinematic control, *Math. Mech.*, 2002, no. 4, pp. 204–206.
34. Antsiporovich, P.P., Akulich, V.K., and Dubovskaya, E.M., Solution of the inverse problem of manipulator kinematics, *Theor. Appl. Mech., Int. Sci. Tech. Collect.*, 2017, no. 32, pp. 306–309.
35. Svinin, M.M., and Uchiyama, M., A new compensation scheme for the inverse kinematics tasks of flexible robot arms, *Proc. IEEE Int. Conf. Robot. Autom.*, 1994, vol. 1, pp. 315–320. <https://doi.org/10.1109/ROBOT.1994.351276>
36. Lamb, M., Lee, S., Billing, E., Högberg, D., and Yang, J., Forward and Backwards Reaching Inverse Kinematics (FABRIK) solver for DHM: A pilot study, *Proc. 7th Int. Digit. Hum. Model. Symp. (DHM 2022) Iowa Virtual Hum. Summit*, 2022, pp. dhm.31772. <https://doi.org/10.17077/dhm.31772>
37. Kamilianov, A.R., Planning of motion trajectories of a multi-link manipulator in a complex three-dimensional workspace based on evolutionary methods, Dissertation for the degree of candidate of technical sciences, Ufa, 2007.
38. Plotnikova, N.V., Control of manipulation robots based on fuzzy logic, *Sci. SUSU: Mater. 62nd Sci. Conf.*, 2010, vol. 2, pp. 170–174.
39. Rahim, F.A., Neuro-fuzzy structure for planning the movement of a robot manipulator online in an unknown dynamic environment, *Izv. Vyssh. Uchebn. Zaved., Sev.-Kavk. Reg., Tekh. Nauki*, 2008, no. 6, pp. 41–49.
40. Song, T.L., Lu, Y.P., and Li, Z.Y., Structural design and research of the bionic snake-like robot, *Adv. Mater. Res.*, 2012, vol. 538, pp. 3034–3037.
41. Oluwade, B., Uwadia, C., and Ayeni, J., Asymptotic time complexity of an algorithm for finding the error pattern of a uniform digital code, *J. Sci. Res. Dev.*, 2001, vol. 6, pp. 127–134.

This paper was recommended for publication by R.A. Munasypov, a member of the Editorial Board