

**Volume 86, Number 5,
May 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. 5, May 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. 5, 2025

Stochastic Systems

Asymptotic-Diffusion Analysis of the Retrial Queueing System $M^{(2)}|M^{(2)}|1$ with Priority Customers for a Non-Priority Component

A. A. Nazarov and Y. E. Izmailova

375

Robust, Adaptive, and Network Control

Adaptive Output Control with the Given Control Quality Guaranteed

A. A. Gukov

392

Control in Technical Systems

Design of Self-Checking Discrete Devices Based on Polynomial Codes with Computation Control via Several Diagnostic Attributes

D. V. Efanov and D. V. Pivovarov

402

Intellectual Control Systems, Data Analysis

Regression Models for the AI Gaming Chatbot for Learning Programming Based on Wordle-Type Puzzles

A. N. Varnavsky

417

Optimization, System Analysis, and Operations Research

Pareto Set Design When Combining Feasible Solutions of the Multicriteria Axial Assignment Problem

L. G. Afraimovich and M. D. Emelin

432

Statistical Study of the Quality of the Cycle Merging Algorithm for Solving the Traveling Salesman Problem at Minimum

Yu. F. Leonova and A. V. Panuykov

445

Relations between Average Shortest Path Length and Another Centralities in Graphs

M. A. Tuzhilin

457

Selected Articles from *Avtomatizatsiya v Promyshlennosti*

A Method for Augmentation of Multidimensional Time Series in Equipment Health Monitoring Tasks

A. V. Erpalov, V. V. Sinitsin, O. L. Ibryaeva, and A. L. Shestakov

464

Estimating Induction Motor State Based on Current Signals in the Conditions of Limited Data Set Using Recurring Images

A. L. Shestakov, D. V. Galyshev, O. L. Ibryaeva, and V. A. Ereemeeva

474

Determination of Optimal Parameters of a Model Predictive Controller with Priorities of Control Objectives under Conditions of Plant Uncertainty

D. V. Shtakin, O. Yu. Snegirev, and A. Yu. Torgashov

484

Asymptotic-Diffusion Analysis of the Retrial Queueing System $M^{(2)}|M^{(2)}|1$ with Priority Customers for a Non-Priority Component

A. A. Nazarov^{*,a} and Y. E. Izmailova^{*,b}

^{*}National Research Tomsk State University, Tomsk, Russia
e-mail: ^anazarov.tsu@gmail.com, ^bevgenevna.92@mail.ru

Received April 22, 2024

Revised March 2, 2025

Accepted March 6, 2025

Abstract—In this paper, we consider retrial queueing system. Two classes of customers come to the system according Poisson arrival processes. The first flow is a flow of priority customers, the second flow is a flow of non-priority customers. Service times have exponential distributions. If a priority customer finds the server occupying by the customer of the same class, it goes to an orbit (orbit for priority customers) and makes a repeated attempt after a random delay. Inter-retrial times have exponential distributions. If an arrival priority customer finds the non-priority customer on the server, it can interrupt its service and starts servicing itself. The preempted customers moves into the orbit for non-priority customers. If a non-priority customer finds the server occupying, it goes to an orbit (orbit for non-priority customers). The customers from the orbit behave the same way. Customers are submitted to the system after successful completion of service. We propose an asymptotic-diffusion analysis of the system. Probability distribution of the number of customers in a non-priority orbit and in a priority orbit are obtained.

Keywords: queueing system, retrial queueing system, orbit, asymptotic-diffusion analysis, diffusion process, probability distribution

DOI: 10.31857/S0005117925050017

1. INTRODUCTION

The formation of queues in bank offices, hospitals, shopping centers and other facilities is a common problem that modern people face in every day life. Sometimes, unable to wait in queue, people leave and attempt to get service later, when the load is reduced. This phenomenon has led to special systems in the queue theory, called retrial queueing systems (RQ-systems). These systems are characterized by the customer (call) finding the server occupied, moves to a virtual orbit where it remains for some random time, then attempts to be served again. A significant effort has been done by a number of scientists involved in RQ-systems research. An overview of this topic can be found in the following sources [1–5].

But in practice, traditional RQ-system models are difficult to apply widely in real-life conditions due to their lack of versatility. Almost every field of life is affected by the prioritization of one customer over another. In service systems such as airlines, first class customers are prioritized over economy class. In telecommunication system, voice packs have higher priority than other data packs like email etc. In [6] various RQ-systems with priority calls and many other system parameters like customer loss, feedback etc. are analyzed.

Recently, mobile traffic has increased rapidly, resulting in a shortage of wireless spectrum. Cognitive radio networks are promising technologies to solve the spectrum scarcity problem. In cognitive radio networks, there are priority and non-priority users. Primary users (priority users) provide

some spectrum bands to secondary users (non-priority users). Secondary users can cognitively utilize these bands when they are not used by the primary users, but when the primary user has arrived for service, the current secondary user must leave the server and attempt to occupy the channel again later. From a mathematical point of view, the study of RQ-systems with two types of customers is significantly more difficult than with a single type of customer. A number of works are devoted to the study of systems with two flows. In [7], a priority system with urgent requests and heterogeneous service is studied. In [8], the authors considered a system in which a prioritized request, having found a server occupied, either preempts a request on the server or queues up with some probability. In [8], the authors take into account the possibility of server breakdown. In [9–11] the functioning metrics of RQ-systems of different configurations with two flows (priority and non-priority) and a queue for priority flows are also found.

Notably, in the works described above, the priority customers, having found the server occupied, enter the queue. In this paper, we propose a system with repeated calls, in which priority customers, as well as non-priority ones, go into orbit, i.e., we need to consider a system with two orbits. In [12], the authors considered a tandem RQ-system with two orbits. The peculiarity of the system studied in this paper is that prioritized customers can displace the serviced non-prioritized customers.

2. MATHEMATICAL MODEL AND PROBLEM STATEMENT

Consider a system with repeated calls (retrial queueing system, RQ-system), Fig. 1.

Two simple flows with parameters λ_1 and λ_2 are received. The first flow is the flow of priority customers and the second is the flow of non-priority customers. When at the moment of arrival a customer finds the server free, it starts to be serviced during the time distributed according to the exponential law with parameters μ_1 and μ_2 or the priority and non-priority flows calls, respectively. After successful completion of the service, the customer leaves the system. When a priority flow customer finds the server occupied at the time of arrival, then:

- if a priority customer was serviced at the server, the incoming one moves to the orbit where it performs a random delay having an exponential distribution with parameter σ_1 . After a random delay, it reaccesses the server with a second attempt to capture again;
- if a non-priority customer was serviced at the server, the incoming customer displaces the serviced one and starts to be serviced itself, and the preempted one moves to the orbit for non-priority customers, where it performs a random delay exponentially distributed with the parameter σ_2 , after which it contacts the device with a repeated attempt to capture the server.

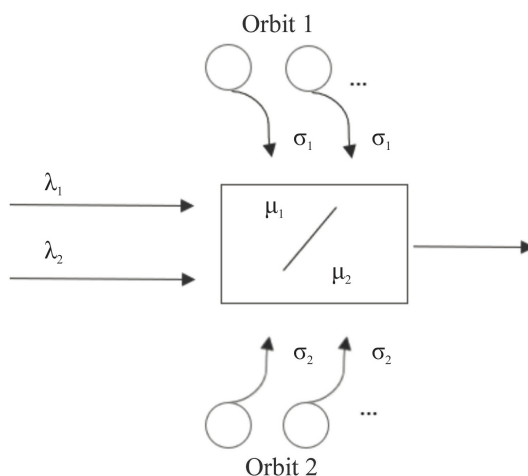


Fig. 1. Mathematical model of the RQ-system $M^{(2)}|M^{(2)}|1$.

If it's free it starts service, if it's occupied it goes back instantly. The discipline for addressing priority calls from orbit to the server is the same as for addressing priority calls newly arrived in the system. Denoting: $i_1(t)$ — number of customers in the first orbit, $i_2(t)$ — number of customers in the second orbit. The server state is determined by the value $k(t)$: $k(t) = 0$, if the server is free for service; $k(t) = 1$, if the server is occupied with servicing a priority customer; $k(t) = 2$, if the server is occupied with servicing a non-priority customer. Then the probability that at time t the device is in state k , in the first orbit the number of customers is i_1 , in the second orbit the number of customers is i_2 , denote as

$$P\{k(t) = k, i_1(t) = i_1, i_2(t) = i_2\} = P_k(i_1, i_2, t). \quad (1)$$

For the distribution (1), using the formula of full probability, we make a system of Kolmogorov differential equations:

$$\begin{aligned} \frac{\partial P_0(i_1, i_2, t)}{\partial t} &= -(\lambda_1 + \lambda_2 + i_1\sigma_1 + i_2\sigma_2)P_0(i_1, i_2, t) + \mu_1 P_1(i_1, i_2, t) + \mu_2 P_2(i_1, i_2, t), \\ \frac{\partial P_1(i_1, i_2, t)}{\partial t} &= -(\lambda_1 + \lambda_2 + \mu_1)P_1(i_1, i_2, t) + \lambda_1 P_1(i_1 - 1, i_2, t) \\ &\quad + \lambda_2 P_1(i_1, i_2 - 1, t) + \lambda_1 P_0(i_1, i_2, t) + (i_1 + 1)\sigma_1 P_0(i_1 + 1, i_2, t) \\ &\quad + \lambda_1 P_2(i_1, i_2 - 1, t) + (i_1 + 1)\sigma_1 P_2(i_1 + 1, i_2 - 1, t), \\ \frac{\partial P_2(i_1, i_2, t)}{\partial t} &= -(\lambda_1 + \lambda_2 + \mu_2 + i_1\sigma_1)P_2(i_1, i_2, t) + \lambda_2 P_2(i_1, i_2 - 1, t) \\ &\quad + \lambda_2 P_0(i_1, i_2, t) + (i_2 + 1)\sigma_2 P_0(i_1, i_2 + 1, t). \end{aligned} \quad (2)$$

To obtain the characteristics of a given queuing system, it is necessary to determine the probability distribution (1). In the limit case, i.e., by solving the system of equations (2), it is difficult to accomplish this because the system is non-trivial. Therefore, it is proposed to construct a diffusion process with the help of which one can approximate the probability distribution (1), thus solving the objective.

Definition 1. Let consider as a partial derivative-characteristic function a function of the order

$$H_k(z, u, t) = \sum_{i_1=0}^{\infty} z^{i_1} \sum_{i_2=0}^{\infty} e^{ju i_2} P_k(i_1, i_2, t). \quad (3)$$

For further investigations, we pass from the system of equations (2) the system of equations for functions (3). We obtain:

$$\begin{aligned} \frac{\partial H_0(z, u, t)}{\partial t} &= -(\lambda_1 + \lambda_2)H_0(z, u, t) - \sigma_1 z \frac{\partial H_0(z, u, t)}{\partial z} + j\sigma_2 \frac{\partial H_0(z, u, t)}{\partial u} \\ &\quad + \mu_1 H_1(z, u, t) + \mu_2 H_2(z, u, t), \\ \frac{\partial H_1(z, u, t)}{\partial t} &= -(\lambda_1 + \lambda_2 + \mu_1)H_1(z, u, t) + \lambda_1 z H_1(z, u, t) + \lambda_2 e^{ju} H_1(z, u, t) \\ &\quad + \lambda_1 H_0(z, u, t) + \lambda_1 e^{ju} H_2(z, u, t) + \sigma_1 \frac{\partial H_0(z, u, t)}{\partial z} + \sigma_1 e^{ju} \frac{\partial H_2(z, u, t)}{\partial z}, \\ \frac{\partial H_2(z, u, t)}{\partial t} &= -(\lambda_1 + \lambda_2 + \mu_2)H_2(z, u, t) + \lambda_2 e^{ju} H_2(z, u, t) + \lambda_2 H_0(z, u, t) \\ &\quad - \sigma_1 z \frac{\partial H_2(z, u, t)}{\partial z} - j\sigma_2 e^{-ju} \frac{\partial H_0(z, u, t)}{\partial u}. \end{aligned} \quad (4)$$

Summing the equations of system (4), setting $z = 1$ and denoting $H_0(1, u, t) + H_1(1, u, t) + H_2(1, u, t) = H(1, u, t)$, we obtain another additive equation:

$$\begin{aligned} \frac{\partial H(1, u, t)}{\partial t} &= (e^{ju} - 1) \\ &\times \left(\lambda_2 H_1(1, u, t) + (\lambda_1 + \lambda_2) H_2(1, u, t) + j\sigma_2 e^{-ju} \frac{\partial H_0(1, u, t)}{\partial u} + \sigma_1 \frac{\partial H_2(1, u, t)}{\partial z} \right), \end{aligned} \quad (5)$$

which we shall solve jointly with system (4) by the asymptotic-diffusion analysis method.

The diffusion analysis will be performed for the non-priority component in the limit condition of large delay of customers in the second orbit, i.e. at $\sigma_2 \rightarrow 0$, in several steps:

- (1) performing the first-order asymptotics, we obtain the transfer coefficient of some diffusion process, by applying it to approximate the probability distribution of the customers number on the orbit. Also at this stage, we find expressions for the stationary probability distribution of the server states and for the partial derivative function of the customers number on the first orbit;
- (2) performing the second order asymptotics, we obtain the diffusion coefficient of some diffusion process;
- (3) at the third stage we obtain an approximation of the customers number probability distribution in the non-priority orbit.

3. ASYMPTOTIC-DIFFUSION ANALYSIS BY NON-PRIORITY COMPONENT

3.1. First Order Asymptotics

In system (4) and equation (5), denoting $\sigma_2 = \epsilon$, we make the following substitutions:

$$\sigma_2 t = \tau, \quad u = \sigma_2 \epsilon = \epsilon w, \quad H_k(z, u, t) = F_k(z, w, \tau, \epsilon).$$

Then (4) and (5), taking into account the substitutions, will be described in the form:

$$\begin{aligned} \epsilon \frac{\partial F_0(z, w, \tau, \epsilon)}{\partial \tau} &= -(\lambda_1 + \lambda_2) F_0(z, w, \tau, \epsilon) - \sigma_1 z \frac{\partial F_0(z, w, \tau, \epsilon)}{\partial z} + j \frac{\partial F_0(z, w, \tau, \epsilon)}{\partial w} \\ &\quad + \mu_1 F_1(z, w, \tau, \epsilon) + \mu_2 F_2(z, w, \tau, \epsilon), \\ \epsilon \frac{\partial F_1(z, w, \tau, \epsilon)}{\partial \tau} &= -(\lambda_1 + \lambda_2 + \mu_1) F_1(z, w, \tau, \epsilon) + \lambda_1 z F_1(z, w, \tau, \epsilon) + \lambda_2 e^{j\epsilon w} F_1(z, w, \tau, \epsilon) \\ &\quad + \lambda_1 F_0(z, w, \tau, \epsilon) + \lambda_1 e^{j\epsilon w} F_2(z, w, \tau, \epsilon) + \sigma_1 \frac{\partial F_0(z, w, \tau, \epsilon)}{\partial z} + \sigma_1 e^{j\epsilon w} \frac{\partial F_2(z, w, \tau, \epsilon)}{\partial z}, \\ \epsilon \frac{\partial F_2(z, w, \tau, \epsilon)}{\partial \tau} &= -(\lambda_1 + \lambda_2 + \mu_2) F_2(z, w, \tau, \epsilon) + \lambda_2 e^{j\epsilon w} F_2(z, w, \tau, \epsilon) + \lambda_2 F_0(z, w, \tau, \epsilon) \\ &\quad - \sigma_1 z \frac{\partial F_2(z, w, \tau, \epsilon)}{\partial z} - j e^{-j\epsilon w} \frac{\partial F_0(z, w, \tau, \epsilon)}{\partial w}. \end{aligned} \quad (6)$$

$$\begin{aligned} \epsilon \frac{\partial F(1, w, \tau, \epsilon)}{\partial \tau} &= (e^{j\epsilon w} - 1) \\ &\times \left(\lambda_2 F_1(1, w, \tau, \epsilon) + (\lambda_1 + \lambda_2) F_2(1, w, \tau, \epsilon) + j e^{-j\epsilon w} \frac{\partial F_0(1, w, \tau, \epsilon)}{\partial w} + \sigma_1 \frac{\partial F_2(1, w, \tau, \epsilon)}{\partial z} \right), \end{aligned} \quad (7)$$

where $F(1, w, \tau, \epsilon) = F_0(1, w, \tau, \epsilon) + F_1(1, w, \tau, \epsilon) + F_2(1, w, \tau, \epsilon)$.

Formulate the following theorem.

Theorem 1. Denote the equation system solution (6):

$$\lim_{\epsilon \rightarrow 0} F_k(z, w, \tau, \epsilon) = F_k(z, w, \tau), \quad k = \overline{0, 2}.$$

Then the following statement is true

$$F_k(z, w, \tau) = G_k(z, x(\tau))e^{jwx(\tau)}, \quad k = \overline{0, 2}. \quad (8)$$

For convenience of notation, we will omit the argume τ : $x(\tau) = x$. Functions $G_k(z, x)$, $k = \overline{0, 2}$ – are partial derivative functions of the customers number on the first orbit, which have the form

$$\begin{aligned} G_0(z, x) &= \frac{(\mu_1 - \lambda_1)^{\frac{\lambda_1}{\sigma_1} + 1}}{\mu_1(\mu_1 - \lambda_1 z)^{\frac{\lambda_1}{\sigma_1}}} - \frac{\lambda_2 + x}{\sigma_1} z^{-\frac{\lambda_1 + \lambda_2 + \mu_2 + x}{\sigma_1}} \int_0^z y^{\frac{\lambda_1 + \lambda_2 + \mu_2 + x - \sigma_1}{\sigma_1}} \frac{(\mu_1 - \lambda_1)^{\frac{\lambda_1}{\sigma_1} + 1}}{\mu_1(\mu_1 - \lambda_1 y)^{\frac{\lambda_1}{\sigma_1}}} dy, \\ G_1(z) &= \left(\frac{\mu_1 - \lambda_1}{\mu_1 - \lambda_1 z} \right)^{\frac{\lambda_1 + \sigma_1}{\sigma_1}} \frac{\lambda_1}{\mu_1}, \\ G_2(z, x) &= \frac{\lambda_2 + x}{\sigma_1} z^{-\frac{\lambda_1 + \lambda_2 + \mu_2 + x}{\sigma_1}} \int_0^z y^{\frac{\lambda_1 + \lambda_2 + \mu_2 + x - \sigma_1}{\sigma_1}} \frac{(\mu_1 - \lambda_1)^{\frac{\lambda_1}{\sigma_1} + 1}}{\mu_1(\mu_1 - \lambda_1 y)^{\frac{\lambda_1}{\sigma_1}}} dy, \end{aligned} \quad (9)$$

value $x(\tau)$ is the differential equation solution

$$x'(\tau) = -x(\tau)G_0(1, x) + \lambda_2 G_1(1) + (\lambda_1 + \lambda_2)G_2(1, x) + \sigma_1 \frac{\partial G_2(z, x)}{\partial z} \Big|_{z=1}. \quad (10)$$

Proof. In the system (6) we perform the limit transition at $\epsilon \rightarrow 0$, obtain

$$\begin{aligned} &-(\lambda_1 + \lambda_2)F_0(z, w, \tau) - \sigma_1 z \frac{\partial F_0(z, w, \tau)}{\partial z} + j \frac{\partial F_0(z, w, \tau)}{\partial w} \\ &+ \mu_1 F_1(z, w, \tau) + \mu_2 F_2(z, w, \tau) = 0, \\ &-(\lambda_1 + \mu_1)F_1(z, w, \tau) + \lambda_1 z F_1(z, w, \tau) \\ &+ \lambda_1 F_0(z, w, \tau) + \lambda_1 F_2(z, w, \tau) + \sigma_1 \frac{\partial F_0(z, w, \tau)}{\partial z} + \sigma_1 \frac{\partial F_2(z, w, \tau)}{\partial z} = 0, \\ &-(\lambda_1 + \mu_2)F_2(z, w, \tau) + \lambda_2 F_0(z, w, \tau) - \sigma_1 z \frac{\partial F_2(z, w, \tau)}{\partial z} - j \frac{\partial F_0(z, w, \tau)}{\partial w} = 0. \end{aligned} \quad (11)$$

The solution of the equations system (11) will be found in the form (8). Then (11) will be rewritten as

$$\begin{aligned} &-(\lambda_1 + \lambda_2 + x(\tau))G_0(z, x) - \sigma_1 z \frac{\partial G_0(z, x)}{\partial z} + \mu_1 G_1(z) + \mu_2 G_2(z, x) = 0, \\ &-(\lambda_1(1 - z) + \mu_1)G_1(z) + \lambda_1 G_0(z, x) + \lambda_1 G_2(z, x) + \sigma_1 \frac{\partial G_0(z, x)}{\partial z} + \sigma_1 \frac{\partial G_2(z, x)}{\partial z} = 0, \\ &-(\lambda_1 + \mu_2)G_2(z, x) + (\lambda_2 + x(\tau))G_0(z, x) - \sigma_1 z \frac{\partial G_2(z, x)}{\partial z} = 0. \end{aligned} \quad (12)$$

Let's introduce the notation $G_{02}(z, x) = G_0(z, x) + G_2(z, x)$. Then by differentiating $G_{02}(z, x)$ by z , we obtain $\frac{\partial G_{02}(z, x)}{\partial z} = \frac{\partial G_0(z, x)}{\partial z} + \frac{\partial G_2(z, x)}{\partial z}$.

Sum the first and the third equation (12) and make a system of two equations (the sum of the first and the third equations and the second equation (12)). Taking into account the introduced notations, we obtain

$$\begin{aligned} -\lambda_1 G_{02}(z, x) + \mu_1 G_1(z) - \sigma_1 z \frac{\partial G_{02}(z, x)}{\partial z} &= 0, \\ -(\lambda_1(1 - z) + \mu_1) G_1(z) + \lambda_1 G_{02}(z, x) + \sigma_1 \frac{\partial G_{02}(z, x)}{\partial z} &= 0. \end{aligned} \quad (13)$$

Multiplying the second equation by z and summing the equations of the system, we obtain

$$\lambda_1 G_{02}(z, x) - \mu_1 G_1(z) + \lambda_1 z G_1(z) = 0. \quad (14)$$

In (14), express $G_1(z)$ through $G_{02}(z, x)$ and substitute it into the first equation of system (13) to obtain a homogeneous differential equation with reference to the function $G_{02}(z, x) = G_{02}(z)$:

$$\frac{\lambda_1^2 z}{\mu_1 - \lambda_1 z} G_{02}(z) - \sigma_1 z \frac{\partial G_{02}(z)}{\partial z} = 0,$$

whose solution is

$$G_{02}(z) = (\mu_1 - \lambda_1 z)^{-\frac{\lambda_1}{\sigma_1}} C. \quad (15)$$

For the function $G_1(z)$ we similarly obtain the differential equation

$$(\lambda_1 + \sigma_1) G_1(z) + \left(\sigma_1 z - \frac{\sigma_1 \mu_1}{\lambda_1} \right) \frac{\partial G_1(z)}{\partial z} = 0,$$

whose solution is

$$G_1(z) = (\mu_1 - \lambda_1 z)^{-\frac{\lambda_1 + \sigma_1}{\sigma_1}} C. \quad (16)$$

We shall seek the constant C . In functions $G_k(z, x)$, $k = \overline{0, 2}$, we will assume $z = 1$ and denote $G_0(1, x) = R_0(x)$, $G_1(1) = R_1$, $G_2(1, x) = R_2(x)$. The values of $R_0(x)$, R_1 , $R_2(x)$ satisfy the normalization condition $R_0(x) + R_1 + R_2(x) = 1$. If we switch to stationary mode in equation (10) and denote $x = \kappa$ by the solution of the stationary equation, we obtain the stationary probability distribution of the server states $R_0(\kappa)$, R_1 , $R_2(\kappa)$.

In equation (14) we assume $z = 1$ and, adding the normalization condition, we obtain the system of equations

$$\begin{aligned} \lambda_1(R_0(x) + R_2(x)) - \mu_1 R_1 + \lambda_1 R_1 &= 0, \\ R_0(x) + R_1 + R_2(x) &= 1. \end{aligned}$$

From where we obtain

$$R_1 = \frac{\lambda_1}{\mu_1}, \quad R_0(x) + R_2(x) = \frac{\mu_1 - \lambda_1}{\mu_1}.$$

Then assuming $z = 1$ in (15) and (16) and using the found $R_k(x)$, we obtain

$$G_1(z) = \left(\frac{\mu_1 - \lambda_1}{\mu_1 - \lambda_1 z} \right)^{\frac{\lambda_1 + \sigma_1}{\sigma_1}} \frac{\lambda_1}{\mu_1}, \quad G_{02}(z) = \left(\frac{\mu_1 - \lambda_1}{\mu_1 - \lambda_1 z} \right)^{\frac{\lambda_1}{\sigma_1}} \frac{\mu_1 - \lambda_1}{\mu_1}.$$

Consider the third equation of the system (12) and found $G_{02}(z) = G_{02}(z, x) = G_0(z, x) + G_2(z, x)$:

$$\begin{aligned} -(\lambda_1 + \mu_2) G_2(z, x) + (\lambda_2 + x(\tau)) G_0(z, x) - \sigma_1 z \frac{\partial G_2(z, x)}{\partial z} &= 0, \\ G_0(z, x) + G_2(z, x) &= \left(\frac{\mu_1 - \lambda_1}{\mu_1 - \lambda_1 z} \right)^{\frac{\lambda_1}{\sigma_1}} \frac{\mu_1 - \lambda_1}{\mu_1}. \end{aligned} \quad (17)$$

Expressing $G_0(z, x)$ from the second equation and substituting it into the first system equation (17), we obtain an inhomogeneous differential equation with reference to the function $G_2(z, x)$:

$$(G_2(z, x))'_z + \frac{\lambda_1 + \lambda_2 + \mu_2 + x}{\sigma_1 z} G_2(z, x) = \frac{\lambda_2 + x}{\sigma_1 z} \left(\frac{\mu_1 - \lambda_1}{\mu_1 - \lambda_1 z} \right)^{\frac{\lambda_1}{\sigma_1}} \frac{\mu_1 - \lambda_1}{\mu_1},$$

whose solution is

$$G_2(z, x) = \frac{\lambda_2 + x}{\sigma_1} z^{-\frac{\lambda_1 + \lambda_2 + \mu_2 + x}{\sigma_1}} \int_0^z y^{\frac{\lambda_1 + \lambda_2 + \mu_2 + x - \sigma_1}{\sigma_1}} \frac{(\mu_1 - \lambda_1)^{\frac{\lambda_1}{\sigma_1} + 1}}{\mu_1 (\mu_1 - \lambda_1 y)^{\frac{\lambda_1}{\sigma_1}}} dy.$$

Substituting the found expression for the function $G_2(z, x)$ into the system (17), we obtain the solution for the function $G_0(z, x)$:

$$G_0(z, x) = \frac{(\mu_1 - \lambda_1)^{\frac{\lambda_1}{\sigma_1} + 1}}{\mu_1 (\mu_1 - \lambda_1 z)^{\frac{\lambda_1}{\sigma_1}}} - \frac{\lambda_2 + x}{\sigma_1} z^{-\frac{\lambda_1 + \lambda_2 + \mu_2 + x}{\sigma_1}} \int_0^z y^{\frac{\lambda_1 + \lambda_2 + \mu_2 + x - \sigma_1}{\sigma_1}} \frac{(\mu_1 - \lambda_1)^{\frac{\lambda_1}{\sigma_1} + 1}}{\mu_1 (\mu_1 - \lambda_1 y)^{\frac{\lambda_1}{\sigma_1}}} dy.$$

The expressions found for the functions $G_k(z, x)$, $k = 0, 1, 2$ coincide with (9).

Consider equation (7). Decompose the exponent into a Taylor series

$$e^{j\epsilon w} = 1 + j\epsilon w + O(\epsilon^2)$$

and divide the left and right parts of (7) by $j\epsilon w$. Then we execute the limit transition at $\epsilon \rightarrow 0$, we obtain

$$\frac{\partial F(1, w, \tau)}{jw\partial\tau} = \lambda_2 F_1(1, w, \tau) + (\lambda_1 + \lambda_2) F_2(1, w, \tau) + j \frac{\partial F_0(1, w, \tau)}{\partial w} + \sigma_1 \frac{\partial F_2(1, w, \tau)}{\partial z}.$$

Execute substitutions

$$F_k(1, w, \tau) = G_k(1, x) e^{jwx} = R_k(x) e^{jwx}, \quad k = 0, 1, 2, \quad F(1, w, \tau) = G(1, x) e^{jwx} = e^{jwx}.$$

Then we obtain the following differential equation for $x(\tau)$:

$$x'(\tau) = -x(\tau) G_0(1, x) + \lambda_2 G_1(1) + (\lambda_1 + \lambda_2) G_2(1, x) + \sigma_1 \frac{\partial G_2(z, x)}{\partial z} \Big|_{z=1},$$

which coincides with (10). Thus, Theorem 1 is proved.

Denoting:

$$\frac{\partial G_k(z, x)}{\partial z} \Big|_{z=1} = \frac{\partial G_k(1, x)}{\partial z}, \quad k = 0, 1, 2, \quad \frac{\partial G_{02}(z)}{\partial z} \Big|_{z=1} = \frac{\partial G_{02}(1)}{\partial z}.$$

Denote the right-hand side of equation (10) by $a(x)$ and simplify it using (9); we obtain

$$a(x) = x'(\tau) = -x(\tau) G_0(1, x) + \lambda_2 G_1(1) + (\lambda_1 + \lambda_2) G_2(1, x) + \sigma_1 \frac{\partial G_2(1, x)}{\partial z} = \lambda_2 - \mu_2 R_2(x).$$

The value $a(x)$ has the meaning of the transfer coefficient of some diffusion process, by which we obtain an approximation of the calls number probability distribution on the orbit.

Corollary 1. *The ergodicity condition of the considered RQ-system has the following form*

$$\frac{\lambda_1}{\mu_1} + \frac{\lambda_2}{\mu_2} < 1.$$

Proof. A sufficient condition of the system ergodicity is the inequality

$$\lim_{x \rightarrow \infty} a(x) < 0.$$

Thus, we obtain

$$\lim_{x \rightarrow \infty} a(x) = \lim_{x \rightarrow \infty} (\lambda_2 - \mu_2 R_2(x)) = \lambda_2 - \mu_2 \left(\frac{\mu_1 - \lambda_1}{\mu_1} \right) < 0.$$

From where we obtain the condition

$$\frac{\lambda_1}{\mu_1} + \frac{\lambda_2}{\mu_2} < 1.$$

Corollary 2. *The derivative function of the number of calls in a priority orbit has the following form*

$$G(z) = \frac{\lambda_1}{\mu_1} \left(\frac{\mu_1 - \lambda_1}{\mu_1 - \lambda_1 z} \right)^{\frac{\lambda_1 + \sigma_1}{\sigma_1}} + \frac{\mu_1 - \lambda_1}{\mu_1} \left(\frac{\mu_1 - \lambda_1}{\mu_1 - \lambda_1 z} \right)^{\frac{\lambda_1}{\sigma_1}}.$$

Thus, the number of calls probability distribution on the first orbit has the form of a weighted sum of two negative-binomial distributions with weights $\frac{\lambda_1}{\mu_1}$ and $\frac{\mu_1 - \lambda_1}{\mu_1}$.

3.2. Second Order Asymptotics

Consider the characteristic function of a random process $i(t) - \frac{x(\sigma_2 t)}{\sigma_2}$. For this purpose, in system (4) and equation (5) we make a substitution:

$$H_k(z, u, t) = e^{\frac{jux(\sigma_2 t)}{\sigma_2}} H_k^{(2)}(z, u, t), \quad k = 0, 1, 2.$$

Denoting $\sigma_2 = \epsilon^2$, in the equations system for $H_k^{(2)}(z, u, t)$, $k = 0, 1, 2$ we substitute

$$\tau = \epsilon^2 t, \quad u = \epsilon w, \quad H_k^{(2)}(z, u, t) = F_k^{(2)}(z, w, \tau, \epsilon),$$

we obtain a system for functions $F_k^{(2)}(z, w, \tau, \epsilon)$, $k = 0, 1, 2$ and an additive equation

$$\begin{aligned} \epsilon^2 \frac{\partial F_0^{(2)}(z, w, \tau, \epsilon)}{\partial \tau} + j\epsilon w a(x) F_0^{(2)}(z, w, \tau, \epsilon) &= -(\lambda_1 + \lambda_2 + x) F_0^{(2)}(z, w, \tau, \epsilon) \\ -\sigma_1 z \frac{\partial F_0^{(2)}(z, w, \tau, \epsilon)}{\partial z} + j\epsilon \frac{\partial F_0^{(2)}(z, w, \tau, \epsilon)}{\partial w} &+ \mu_1 F_1^{(2)}(z, w, \tau, \epsilon) + \mu_2 F_2^{(2)}(z, w, \tau, \epsilon), \\ \epsilon^2 \frac{\partial F_1^{(2)}(z, w, \tau, \epsilon)}{\partial \tau} + j\epsilon w a(x) F_1^{(2)}(z, w, \tau, \epsilon) &= -(\lambda_1 + \lambda_2 + \mu_1) F_1^{(2)}(z, w, \tau, \epsilon) \\ + \lambda_1 z F_1^{(2)}(z, w, \tau, \epsilon) + \lambda_2 e^{j\epsilon w} F_1^{(2)}(z, w, \tau, \epsilon) &+ \lambda_1 F_0^{(2)}(z, w, \tau, \epsilon) \\ + \lambda_1 e^{j\epsilon w} F_2^{(2)}(z, w, \tau, \epsilon) + \sigma_1 \frac{\partial F_0^{(2)}(z, w, \tau, \epsilon)}{\partial z} &+ \sigma_1 e^{j\epsilon w} \frac{\partial F_2^{(2)}(z, w, \tau, \epsilon)}{\partial z}, \end{aligned} \quad (18)$$

$$\begin{aligned} \epsilon^2 \frac{\partial F_2^{(2)}(z, w, \tau, \epsilon)}{\partial \tau} + j\epsilon w a(x) F_2^{(2)}(z, w, \tau, \epsilon) &= -(\lambda_1 + \lambda_2 + \mu_2) F_2^{(2)}(z, w, \tau, \epsilon) \\ + \lambda_2 e^{j\epsilon w} F_2^{(2)}(z, w, \tau, \epsilon) + \lambda_2 F_0^{(2)}(z, w, \tau, \epsilon) & \\ -\sigma_1 z \frac{\partial F_2^{(2)}(z, w, \tau, \epsilon)}{\partial z} - j\epsilon e^{-j\epsilon w} \frac{\partial F_0^{(2)}(z, w, \tau, \epsilon)}{\partial w} &+ x e^{-j\epsilon w} F_0^{(2)}(z, w, \tau, \epsilon). \end{aligned}$$

$$\begin{aligned} \epsilon^2 \frac{\partial F^{(2)}(1, w, \tau, \epsilon)}{\partial \tau} + j\epsilon w a(x) F^{(2)}(1, w, \tau, \epsilon) &= (e^{j\epsilon w} - 1) \\ \times \left(-x e^{-j\epsilon w} F_0^{(2)}(1, w, \tau, \epsilon) + \lambda_2 F_1^{(2)}(1, w, \tau, \epsilon) + (\lambda_1 + \lambda_2) F_2^{(2)}(1, w, \tau, \epsilon) \right. & \\ \left. + j\epsilon e^{-j\epsilon w} \frac{\partial F_0^{(2)}(1, w, \tau, \epsilon)}{\partial w} + \sigma_1 \frac{\partial F_2^{(2)}(1, w, \tau, \epsilon)}{\partial z} \right). & \end{aligned} \quad (19)$$

Denote: $F_k^{(2)}(z, w, \tau) = \lim_{\epsilon \rightarrow 0} F_k^{(2)}(z, w, \tau, \epsilon)$, $k = 0, 1, 2$. We formulate and prove the following theorem.

Theorem 2. *The function $F_k^{(2)}(z, w, \tau)$, $k = 0, 1, 2$ have the form $F_k^{(2)}(z, w, \tau) = \Phi(w, \tau)G_k(z, x)$, where the function $\Phi(w, \tau)$ is a characteristic function of process $y(\tau) = \lim_{\sigma_2 \rightarrow 0} \sqrt{\sigma_2}(i(\frac{\tau}{\sigma_2}) - \frac{x(\tau)}{\sigma_2})$, and satisfies the differential equation*

$$\frac{\partial \Phi(w, \tau)}{\partial \tau} = a'(x)w \frac{\partial \Phi(w, \tau)}{\partial w} - \frac{w^2}{2}b(x)\Phi(w, \tau), \quad (20)$$

where

$$a(x) = \lambda_2 - \mu_2 R_2(x), \quad (21)$$

$$b(x) = a(x) + 2((\lambda_2 - a(x))R_2(x) - \mu_2 h_2(x)). \quad (22)$$

Here $h_2(x) = h_2(1, x)$,

$$h_2(1, x) = \frac{1}{\sigma_1} \int_0^1 y^{\frac{\lambda_1 + \lambda_2 + \mu_2 + x - \sigma_1}{\sigma_1}} \cdot D(y, x) dy, \quad (23)$$

$$D(z, x) = (\lambda_2 + x)h_{02}(z, x) + (\lambda_2 - a(x))G_2(z, x) - xG_0(z, x), \quad (24)$$

$$h_{02}(z, x) = \left(\frac{\mu_1 - \lambda_1}{\mu_1 - z\lambda_1} \right)^{\frac{\lambda_1}{\sigma_1}} \cdot \left(\frac{1}{\mu_1} A(1, x) - \frac{1}{\sigma_1} \int_z^1 \left(\frac{\mu_1 - \lambda_1}{\mu_1 - y\lambda_1} \right)^{\frac{-\lambda_1}{\sigma_1}} B(y, x) dy \right), \quad (25)$$

$$A(1, x) = a(x) \frac{\partial G_{02}(1)}{\partial z} - (\lambda_1 + \lambda_2 + \sigma_1) \frac{\partial G_2(1, x)}{\partial z} + x \frac{\partial G_0(1, x)}{\partial z} - \lambda_1 R_2(x) + (a(x) - \lambda_2)R_1 + (a(x) - \lambda_2) \frac{\partial G_1(1)}{\partial z} - \sigma_1 \frac{\partial^2 G_2(1, x)}{\partial z^2}, \quad (26)$$

$$B(z, x) = (a(x) - \lambda_2)G_1(z) - \lambda_1 G_2(z, x) - \sigma_1 \frac{\partial G_2(z, x)}{\partial z} - \frac{(z-1)\lambda_1 - \mu_1}{(z\lambda_1 - \mu_1)(z-1)} \times \left(a(x)G_{02}(z) - (\lambda_2 + z\lambda_1)G_2(z, x) + xG_0(z, x) + z(a(x) - \lambda_2)G_1(z) - z\sigma_1 \frac{\partial G_2(z, x)}{\partial z} \right). \quad (27)$$

Proof. The solution of the equation system (18) is written in the following form:

$$F_k(z, w, \tau, \epsilon) = \Phi(w, \tau) (G_k(z, x) + j\epsilon w g_k(z, x)) + O(\epsilon^2), \quad k = 0, 1, 2.$$

Substituting these functions into the system of equation (18), making simple transformations and performing the limit transition at $\epsilon \rightarrow 0$, we obtain a system of equations:

$$\begin{aligned} a(x)G_0(z, x) &= -(\lambda_1 + \lambda_2 + x)g_0(z, x) + \mu_1 g_1(z, x) + \mu_2 g_2(z, x) \\ &\quad - \sigma_1 z \frac{\partial g_0(z, x)}{\partial z} + G_0(z, x) \frac{\partial \Phi(w, \tau)}{w \Phi(w, \tau) \partial w}, \\ a(x)G_1(z) &= -(\lambda_1(1-z) + \mu_1)g_1(z, x) + \lambda_1 g_0(z, x) + \lambda_1 g_2(z, x) + \lambda_2 G_1(z, x) \\ &\quad + \lambda_1 G_2(z, x) + \sigma_1 \frac{\partial g_0(z, x)}{\partial z} + \sigma_1 \frac{\partial g_2(z, x)}{\partial z} + \sigma_1 \frac{\partial G_2(z, x)}{\partial z}, \\ a(x)G_2(z, x) &= -(\lambda_1 + \mu_2)g_2(z, x) + (\lambda_2 + x)g_0(z, x) + \lambda_2 G_2(z, x) \\ &\quad - xG_0(z, x) - \sigma_1 z \frac{\partial g_2(z, x)}{\partial z} - G_0(z, x) \frac{\partial \Phi(w, \tau)}{w \Phi(w, \tau) \partial w}. \end{aligned} \quad (28)$$

The functions $g_k(z, x)$, $k = 0, 1, 2$ will be solved as the sum of a general homogeneous solution and a partial solution of an inhomogeneous differential equation:

$$g_k(z, x) = CG_k(z, x) + h_k(z, x) - \phi_k(z, x) \frac{\partial \Phi(w, \tau)}{w \Phi(w, \tau) \partial w}, \quad k = 0, 1, 2.$$

Substituting $g_k(z, x)$ into (28), we can confirm that the coefficient in front of the constant C is zero. Then equating the coefficients at $\frac{\partial \Phi(w, \tau)}{w \Phi(w, \tau) \partial w}$, we obtain the equations system to find functions $\phi_k(z, x)$. If the equation system for the functions $G_k(z, x)$ is differentiated by the variable x , we can confirm that it coincides with the equation system for finding the functions $\phi_k(z, x)$. Thus, we conclude that

$$\phi_k(z, x) = \frac{\partial G_k(z, x)}{\partial x}, \quad k = 0, 1, 2.$$

Equating the remaining summands, we can write the equation system for determining the functions $h_k(z, x)$, $k = 0, 1, 2$:

$$\begin{aligned} & -(\lambda_1 + \lambda_2 + x)h_0(z, x) + \mu_1 h_1(z, x) + \mu_2 h_2(z, x) - \sigma_1 z \frac{\partial h_0(z, x)}{\partial z} = a(x)G_0(z, x), \\ & \lambda_1 h_0(z, x) - (\lambda_1(1 - z) + \mu_1)h_1(z, x) + \lambda_1 h_2(z, x) + \sigma_1 \frac{\partial h_0(z, x)}{\partial z} + \sigma_1 \frac{\partial h_2(z, x)}{\partial z} \\ & \quad = (a(x) - \lambda_2)G_1(z) - \sigma_1 \frac{\partial G_2(z, x)}{\partial z} - \lambda_1 G_2(z, x), \\ & (\lambda_2 + x)h_0(z, x) - (\lambda_1 + \mu_2)h_2(z, x) - \sigma_1 z \frac{\partial h_2(z, x)}{\partial z} = (a(x) - \lambda_2)G_2(z, x) + xG_0(z, x). \end{aligned} \quad (29)$$

Sum the first and the third system equation (29) and add the second equation to the resulting equation to obtain the system:

$$\begin{aligned} & -\lambda_1(h_0(z, x) + h_2(z, x)) + \mu_1 h_1(z, x) - \sigma_1 z \left(\frac{\partial h_0(z, x)}{\partial z} + \frac{\partial h_2(z, x)}{\partial z} \right) \\ & \quad = a(x)(G_0(z, x) + G_2(z, x)) - \lambda_2 G_2(z, x) + xG_0(z, x), \\ & \lambda_1(h_0(z, x) + h_2(z, x)) - (\lambda_1(1 - z) + \mu_1)h_1(z, x) + \sigma_1 \left(\frac{\partial h_0(z, x)}{\partial z} + \frac{\partial h_2(z, x)}{\partial z} \right) = \\ & \quad = (a(x) - \lambda_2)G_1(z) - \sigma_1 \frac{\partial G_2(z, x)}{\partial z} - \lambda_1 G_2(z, x). \end{aligned}$$

Denote:

$$h_{02}(z, x) = h_0(z, x) + h_2(z, x).$$

Then the equation system will be rewritten in the form:

$$\begin{aligned} & -\lambda_1 h_{02}(z, x) + \mu_1 h_1(z, x) - \sigma_1 z \frac{\partial h_{02}(z, x)}{\partial z} \\ & \quad = a(x)(G_0(z, x) + G_2(z, x)) - \lambda_2 G_2(z, x) + xG_0(z, x), \\ & \lambda_1 h_{02}(z, x) - (\lambda_1(1 - z) + \mu_1)h_1(z, x) + \sigma_1 \frac{\partial h_{02}(z, x)}{\partial z} \\ & \quad = (a(x) - \lambda_2)G_1(z) - \sigma_1 \frac{\partial G_2(z, x)}{\partial z} - \lambda_1 G_2(z, x). \end{aligned} \quad (30)$$

Multiply the second equation by z and sum the system equations, we obtain:

$$\begin{aligned} \lambda_1 h_{02}(z, x) - (\mu_1 - \lambda_1 z) h_1(z, x) &= \frac{1}{z-1} \left(a(x)(G_0(z, x) + G_2(z, x) + zG_1(z)) \right. \\ &\quad \left. + xG_0(z, x) - \lambda_2 zG_1(z) - (\lambda_1 z + \lambda_2)G_2(z, x) - \sigma_1 z \frac{\partial G_2(z, x)}{\partial z} \right). \end{aligned} \quad (31)$$

Denote the right side of the equation by $A(z, x)$ and seek the limit at $z \rightarrow 1$:

$$\begin{aligned} \lim_{z \rightarrow 1} \frac{(a(x) + x)G_0(z, x) + z(a(x) - \lambda_2)G_1(z) - (\lambda_1 z + \lambda_2 - a(x))G_2(z, x) - \sigma_1 z \frac{\partial G_2(z, x)}{\partial z}}{z-1} \\ = \lim_{z \rightarrow 1} \left(a(x) \frac{\partial G_{02}(z)}{\partial z} - (\lambda_1 + \lambda_2 + \sigma_1) \frac{\partial G_2(z, x)}{\partial z} + x \frac{\partial G_0(z, x)}{\partial z} - \lambda_1 G_2(z, x) \right. \\ \left. + (a(x) - \lambda_2)G_1(z) + (a(x) - \lambda_2) \frac{\partial G_1(z)}{\partial z} - \sigma_1 z \frac{\partial^2 G_2(z, x)}{\partial z^2} \right) \\ = \lim_{z \rightarrow 1} A(z, x) = a(x) \frac{\partial G_{02}(1)}{\partial z} - (\lambda_1 + \lambda_2 + \sigma_1) \frac{\partial G_2(1, x)}{\partial z} + x \frac{\partial G_0(1, x)}{\partial z} - \lambda_1 R_2(x) \\ + (a(x) - \lambda_2)R_1 + (a(x) - \lambda_2) \frac{\partial G_1(1)}{\partial z} - \sigma_1 \frac{\partial^2 G_2(1, x)}{\partial z^2} = A(1, x), \end{aligned}$$

which coincides with (26).

Then equation (31) can be written in the form

$$\lambda_1 h_{02}(z, x) - (\mu_1 - \lambda_1 z) h_1(z, x) = A(1, x).$$

Then we express the function $h_1(z, x)$ and substitute into the second equation of the system (30), we obtain an inhomogeneous differential equation with reference to the function $h_{02}(z, x)$. Denote the right part of the equation by $B(z, x)$:

$$\begin{aligned} B(z, x) &= (a(x) - \lambda_2)G_1(z) - \lambda_1 G_2(z, x) - \sigma_1 \frac{\partial G_2(z, x)}{\partial z} - \frac{(z-1)\lambda_1 - \mu_1}{(z\lambda_1 - \mu_1)(z-1)} \\ &\times \left(a(x)G_{02}(z) - (\lambda_2 + z\lambda_1)G_2(z, x) + xG_0(z, x) + z(a(x) - \lambda_2)G_1(z) - z\sigma_1 \frac{\partial G_2(z, x)}{\partial z} \right). \end{aligned}$$

Then the differential equation is written in the following form:

$$\lambda_1 \left(1 - \frac{\lambda_1(1-z) + \mu_1}{\mu_1 - \lambda_1 z} \right) h_{02}(z, x) + \sigma_1 \frac{\partial h_{02}(z, x)}{\partial z} = B(z, x).$$

Its solution has the form

$$h_{02}(z, x) = \left(\frac{\mu_1 - \lambda_1}{\mu_1 - z\lambda_1} \right)^{\frac{\lambda_1}{\sigma_1}} \left(\frac{1}{\mu_1} A(1, x) - \frac{1}{\sigma_1} \int_z^1 \left(\frac{\mu_1 - \lambda_1}{\mu_1 - y\lambda_1} \right)^{\frac{-\lambda_1}{\sigma_1}} B(y, x) dy \right),$$

which coincides with (25).

Since $h_0(z, x) = h_{02}(z, x) - h_2(z, x)$, we substitute this into the third equation of the system (29) and denote the right side as

$$D(z, x) = (\lambda_2 + x)h_{02}(z, x) + (\lambda_2 - a(x))G_2(z, x) - xG_0(z, x),$$

obtain a differential equation with reference to $h_2(z, x)$:

$$(\lambda_1 + \lambda_2 + \mu_2 + x)h_2(z, x) + \sigma_1 z \frac{\partial h_2(z, x)}{\partial z} = D(z, x),$$

the solution of which has the form

$$h_2(z, x) = z^{-\frac{\lambda_1 + \lambda_2 + \mu_2 + x}{\sigma_1}} \frac{1}{\sigma_1} \int_0^z y^{\frac{\lambda_1 + \lambda_2 + \mu_2 + x - \sigma_1}{\sigma_1}} D(y, x) dy.$$

Assuming $z = 1$, we obtain (23).

Consider equation (19). Substitute into it

$$F_k(z, w, \tau, \epsilon) = \Phi(w, \tau) (G_k(z, x) + j\epsilon w g_k(z, x)) + O(\epsilon^2), \quad k = 0, 1, 2,$$

making simple transformations and performing the limit transition at $\epsilon \rightarrow 0$, we obtain

$$\begin{aligned} -\frac{\partial \Phi(w, \tau)}{w^2 \Phi(w, \tau) \partial \tau} + a(x) g(1, x) &= \lambda_2 g_1(1, x) + (\lambda_1 + \lambda_2) g_2(1, x) \\ + G_0(1, x) \frac{\partial \Phi(w, \tau)}{w \Phi(w, \tau) \partial w} - x g_0(1, x) &+ x G_0(1, x) + \sigma_1 \frac{\partial g_2(1, x)}{\partial z} + \frac{1}{2} a(x). \end{aligned}$$

The functions $g_k(1, x)$ are written in the form

$$g_k(1, x) = C \cdot G_k(1, x) + h_k(1, x) - \phi_k(1, x) \frac{\partial \Phi(w, \tau)}{w \Phi(w, \tau) \partial w}, \quad k = 0, 1, 2.$$

Considering the conditions $\sum_{k=0}^2 h_k(z, x) \Big|_{z=1} = 0$, $\sum_{k=0}^2 \phi_k(z, x) \Big|_{z=1} = 0$ we obtain the equation

$$\begin{aligned} \frac{\partial \Phi(w, \tau)}{\partial \tau} &= \left(\lambda_2 \phi_1(1, x) + (\lambda_1 + \lambda_2) \phi_2(1, x) - G_0(1, x) - x \phi_0(1, x) + \sigma_1 \frac{\partial \phi_2(1, x)}{\partial z} \right) \\ &\quad \times w \frac{\partial \Phi(w, \tau)}{\partial w} - \frac{w^2}{2} \Phi(w, \tau) \\ &\times \left(a(x) + 2 \left(\lambda_2 h_1(1, x) + (\lambda_1 + \lambda_2) h_2(1, x) - x h_0(1, x) + x G_0(1, x) + \sigma_1 \frac{\partial h_2(1, x)}{\partial z} \right) \right). \end{aligned}$$

Convert the coefficients in front of $w \frac{\partial \Phi(w, \tau)}{\partial w}$ and $\frac{w^2}{2} \Phi(w, \tau)$, we obtain

$$\begin{aligned} \lambda_2 \phi_1(1, x) + (\lambda_1 + \lambda_2) \phi_2(1, x) - G_0(1, x) - x \phi_0(1, x) + \sigma_1 \frac{\partial \phi_2(1, x)}{\partial z} &= a'(x), \\ a(x) + 2 \left(\lambda_2 h_1(1, x) + (\lambda_1 + \lambda_2) h_2(1, x) - x h_0(1, x) + x G_0(1, x) + \sigma_1 \frac{\partial h_2(1, x)}{\partial z} \right) &= \\ = a(x) + 2((\lambda_2 - a(x)) R_2(x) - \mu_2 h_2(1, x)) &= b(x). \end{aligned}$$

Then we obtain the equation

$$\frac{\partial \Phi(w, \tau)}{\partial \tau} = a'(x) w \frac{\partial \Phi(w, \tau)}{\partial w} - \frac{w^2}{2} b(x) \Phi(w, \tau).$$

The theorem is proved.

The function $b(x)$ has the meaning of the diffusion coefficient of some diffusion process, using which we could approximate the probability distribution of the number of calls on the orbit.

3.3. Diffusion Approximation of Probability Distribution of the Number of Calls on a Non-Priority Orbit

Apply the inverse Fourier transform to equation (20). We obtain the Fokker-Planck equation for the probability distribution density function $P(y, \tau) = \frac{\partial P\{y(\tau) < y\}}{\partial y}$:

$$\frac{\partial P(y, \tau)}{\partial \tau} = -\frac{\partial\{ya'(x)P(y, \tau)\}}{\partial y} + \frac{1}{2} \frac{\partial^2\{b(x)P(y, \tau)\}}{\partial y^2}.$$

It can be concluded that $y(\tau) = \lim_{\sigma_2 \rightarrow 0} \sqrt{\sigma_2} \left(i \left(\frac{\tau}{\sigma_2} \right) - \frac{x(\tau)}{\sigma_2} \right)$ is the solution of the stochastic differential equation

$$dy(\tau) = a'(x)y d\tau + \sqrt{b(x)} dW(\tau),$$

where $W(\tau)$ is a Wiener random process, $a'(x)y$ – transfer coefficient, $\sqrt{b(x)}$ – diffusion coefficient.

Consider the diffusion process $z(\tau) = x(\tau) + \sqrt{\sigma_2}y(\tau)$. Note that $z(\tau) = \lim_{\sigma_2 \rightarrow 0} \sigma_2 i \left(\frac{\tau}{\sigma_2} \right)$. Denote by $V(z)$ the stationary probability distribution density function of the process $z(\tau)$. It may be shown that the density $V(z)$ has the form

$$V(z) = \frac{C}{b(z)} \cdot \exp \left(\frac{2}{\sigma_2} \int_0^z \frac{a(x)}{b(x)} dx \right).$$

A more detailed description of the procedure for constructing the diffusion approximation and finding the kind of density $V(z)$ can be found in [13, 14].

To construct a diffusion approximation of the probability distribution of the number of requests on a non-priority orbit, we will use the formula:

$$P(i) = \frac{V(i\sigma_2)}{\sum_{n=0}^{\infty} V(n\sigma_2)}. \quad (32)$$

Thus, there is no need to determine the value of the constant C .

4. THE DIFFUSION APPROXIMATION ALGORITHM IN MATHCAD SOFTWARE

The obtained theoretical results were realized in MathCAD software package. The numerical realization algorithm for finding the diffusion approximation of the number of requests on the non-priority orbit is given below.

Algorithm 1.

- (1) set system parameter: $\lambda_1, \lambda_2, \mu_1, \mu_2, \sigma_1, \sigma_2$. Set a sufficiently large number of N ;
- (2) by formulas (9) write down the functions $G_k(z, x), k = 0, 1, 2$ and $R_k(x) = G_k(1, x), k = 0, 1, 2$;
- (3) by formula (21) write down the function $a(x)$ and find the solution κ of the stationary equation $a(x) = 0$, using the built-in function of MathCAD software

$$\kappa := \text{root}(a(x), x, 0, N);$$

- (4) determine the values $R_0(\kappa)$, $R_1, R_2(\kappa)$, which are the stationary probabilities of the server states;
 (5) write a function

$$G_{02}(z) = \left(\frac{\mu_1 - \lambda_1}{\mu_1 - \lambda_1 z} \right)^{\frac{\lambda_1}{\sigma_1}} \frac{\mu_1 - \lambda_1}{\mu_1};$$

- (6) count the derivatives z from the functions $G_k(z, x)$, $k = 0, 1, 2$ and $G_{02}(z, x)$:

$$\begin{aligned} \frac{\partial G_{02}(z)}{\partial z} &= G_{02}(z) \frac{\lambda_1}{\sigma_1} \frac{\lambda_1}{\mu_1 - z\lambda_1}; & \frac{\partial G_1(z)}{\partial z} &= \frac{\lambda_1 G_1(z) + \lambda_1 \frac{\partial G_{02}(z)}{\partial z}}{\mu_1 - z\lambda_1}, \\ \frac{\partial G_0(z, x)}{\partial z} &= \frac{-(\lambda_1 + \lambda_2)G_0(z, x) + \mu_1 G_1(z) + \mu_2 G_2(z, x) - xG_0(z, x)}{\sigma_1 z}, \\ \frac{\partial G_2(z, x)}{\partial z} &= \frac{(\lambda_2 + x)G_{02}(z) - (\lambda_1 + \lambda_2 + \mu_2 + x)G_2(z, x)}{\sigma_1 z}, \\ \frac{\partial^2 G_2(z, x)}{\partial z^2} &= \frac{(\lambda_2 + x)G_{02}(z) - (\lambda_1 + \lambda_2 + \mu_2 + x + \sigma_1) \frac{\partial G_2(z, x)}{\partial z}}{\sigma_1 z}; \end{aligned}$$

- (7) write $A(z, x)$ by the formula

$$\begin{aligned} A(z, x) &= \frac{1}{z-1} \left(a(x) + x \right) G_0(z, x) + z(a(x) - \lambda_2) G_1(z) \\ &\quad - (\lambda_1 z + \lambda_2 - a(x)) G_2(z, x) - \sigma_1 z \frac{\partial G_2(z, x)}{\partial z}, \end{aligned}$$

$B(z, x)$ by the formula (27), for the $h_{02}(z, x)$ – (25):

$$h_{02}(z, x) := if \left(z = 1, \frac{1}{\mu_1} A(1, x), h_{02}(z, x) \right);$$

- (8) determine $D(z, x)$ (24) and $h_2(1, x)$ (23);
 (9) write the diffusion coefficient $b(x)$ (22);
 (10) construct $P1(i)$ (32):

$$P1(i) = \frac{1}{b(\sigma_2 i)} \exp \left(\frac{2}{\sigma_2} \int_0^{\sigma_2 i} \frac{a(x)}{b(x)} dx \right);$$

- (11) perform normalization and obtain an approximation of the discrete probability distribution of the number of calls on the non-priority orbit

$$P(i) := P1(i) \left(\sum_{i=0}^N P1(i) \right)^{-1}.$$

Example 1. Set the parameters of the system:

$$\lambda_1 = 0.3, \quad \lambda_2 = 0.9, \quad \mu_1 = 1, \quad \mu_2 = 2, \quad \sigma_1 = 1.$$

We obtain the probability distribution of the server states:

$$R_0 = 0.25, \quad R_1 = 0.3, \quad R_2 = 0.45.$$

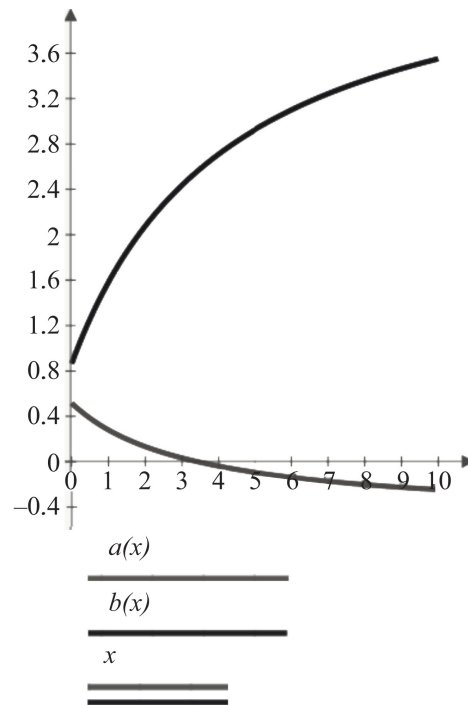


Fig. 2. Transfer coefficient and diffusion coefficient plots.

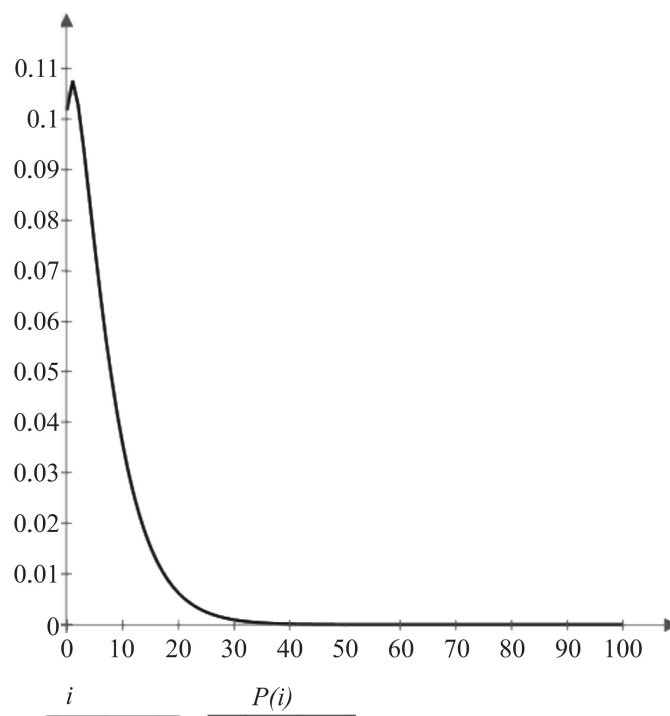


Fig. 3. Diffusion approximation $P(i)$ at $\sigma_2 = 1$, $N = 100$.

Figure 2 shows the transfer coefficient $a(x)$ and diffusion coefficient $b(x)$ dependence on the number of calls in the non-priority orbit. We can conclude that as the number of applications increases, the spread relative to the average increases. Figures 3–5 show diffusion approximation plots of probability distribution of the number of calls on the non-priority orbit.

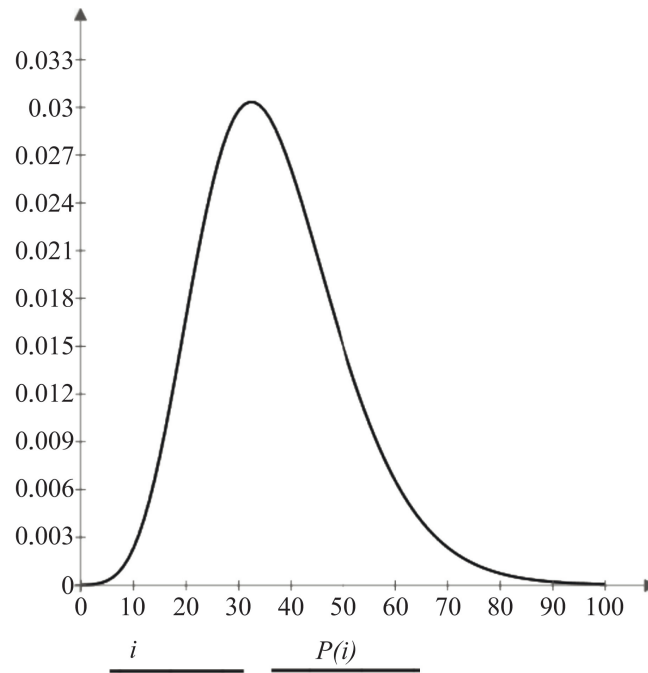


Fig. 4. Diffusion approximation $P(i)$ at $\sigma_2 = 0, 1$, $N = 100$.

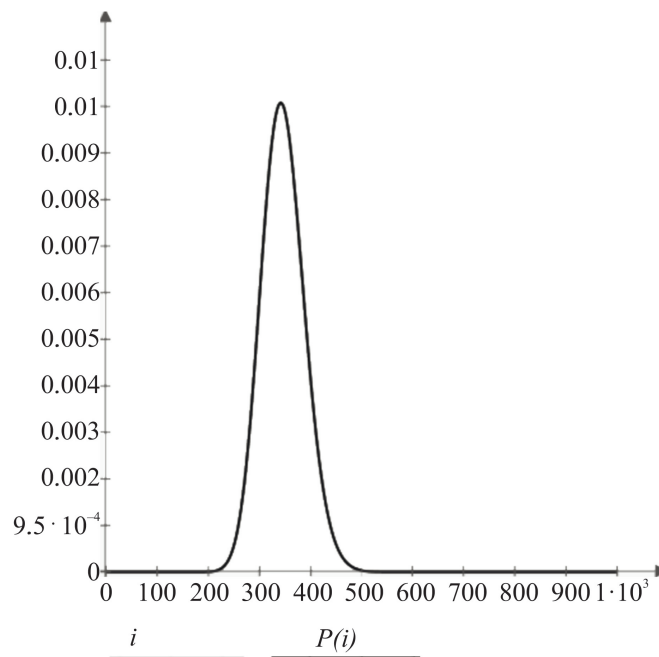


Fig. 5. Diffusion approximation $P(i)$ at $\sigma_2 = 0.01$, $N = 1000$.

Numerical results show the proposed research approach implementing possibility to problems of this type. Also, we can conclude from the distribution graph that for small values of σ_2 the distribution is close to the Gaussian probability distribution. This conclusion can be confirmed by referring to [15], where the system $M^{(2)}|M^{(2)}|1$ is studied by the Gaussian approximation method.

5. CONCLUSION

In this paper, a queuing system with repeated calls and two incoming flows of calls (priority and non-priority stream) is investigated.

For the number of priority calls on the orbit, the derivative function is found in the form of a weighted sum of derivative functions of negative-binomial distributions.

For the number of non-priority requests in an orbit, we obtain a probability distribution, which we call the diffusion approximation.

The found probability distributions allow us to determine all the necessary probabilistic temporal characteristics of the system for the received number of calls. The obtained theoretical results can be used to solve a number of practical problems in which it is necessary to separate requests by priority: cognitive radio, data transmission, where it is necessary to separate by the volume of transmitted data packets. The investigation of problems of a more general type (with arbitrary service time distribution function) is of concern.

REFERENCES

1. Yang, T. and Templeton, J.G.C., A survey on retrial queues, *Queueing Syst.*, 1987, vol. 2, no. 3, pp. 201–233.
2. Artalejo, J.R., A classified bibliography of research on retrial queues, *Progress in 1990–1999*, 1999, vol. 7, no. 2, pp. 187–211.
3. Artalejo, J.R., Accessible bibliography on retrial queues, *Math. Comput. Model.*, 1999, vol. 30, no. 3–4, pp. 1–6.
4. Artalejo, J.R. and Falin, G.I., Standard and retrial queueing systems: a comparative analysis, *Revista Matematica Complutense*, 2002, vol. 15, pp. 101–129.
5. Artalejo, J.R., Accessible bibliography on retrial queues, *Math. Comput. Model.*, 2010, vol. 51, pp. 1071–1081.
6. Choi, B.D. and Chang, Y., Single server retrial queues with priority calls, *Math. Comput. Model.*, 1999, vol. 30, pp. 7–32.
7. Yin, M., Yan, M., Guo, Y., and Liu, M., Analysis of a preemptive two-priority queueing system with impatient customers and heterogeneous servers, *Mathematics*, 2023, vol. 11, pp. 3878.
8. Razumchik, R., Two-priority queueing system with lfs service, probabilistic priority and batch arrivals, *AIP Conference Proceedings*, 2019, vol. 2116, pp. 090011.
9. Dimitriou, I., A mixed priority retrial queue with negative arrivals, unreliable server and multiple vacations, *Appl. Math. Model.*, 2013, vol. 37, pp. 1295–1309.
10. Jain, M., Bhagat, A., and Shekhar, C., Double orbit finite retrial queues with priority customers and service interruptions, *Appl. Math. Comput.*, 2015, vol. 253, pp. 324–344.
11. Atencia, I., Moreno, P., and Bouza, G., An $m2/g2/1$ retrial queue with priority customers, 2nd optional service and linear retrial policy, *Invest. Oper.*, 2006, vol. 27, no. 3, pp. 229–248.
12. Nazarov, A., Phung-Duc, T., Paul, S., and Morozova, M., Scaling limits of a tandem queue with two infinite orbits, *Mathematics*, 2023, vol. 11, no. 11, pp. 2454.
13. Nazarov, A., Phung-Duc, T., and Izmailova, Ya., Asymptotic-diffusion analysis of multiserver retrial queueing system with priority customers, *Commun. in Comput. and Inform. Sci.*, 2021, vol. 1391, pp. 236–250.
14. Nazarov, A., Phung-Duc, T., Paul, S., and Lizyura, O., Diffusion limit for single-server retrial queues with renewal input and outgoing calls, *Mathematics*, 2022, vol. 10, no. 6, pp. 948.
15. Nazarov, A. and Izmaylova, Ya., Study of an rq -system $m(2)|m(2)|1$ with r -persistent preemption of alternative calls, *Siberian Aerospace Journal*, 2016, vol. 17, no. 2, pp. 328–334.

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

Adaptive Output Control with the Given Control Quality Guaranteed

A. A. Gukov

*Institute for Problems in Mechanical Engineering, Russian Academy of Sciences,
St. Petersburg, Russia*

e-mail: artemiy.gukoff4u@mail.ru

Received August 21, 2024

Revised March 10, 2025

Accepted March 16, 2025

Abstract—An algorithm for output control of linear plants with an arbitrary relative degree under the conditions of parametric uncertainty and bounded perturbations is proposed. Unlike classical adaptive control algorithms, the proposed algorithm allows ensuring the plants output tracking of the reference signal, with the tracking error being in the set specified by the developer. An example illustrating the efficiency of the proposed method is given.

Keywords: adaptive control, dynamic system, change of coordinates, stability, control

DOI: 10.31857/S0005117925050022

1. INTRODUCTION

In this work, we consider the adaptive output tracking problem with respect to the reference signal with the given control quality guaranteed at any instant. We consider linear systems under parametric uncertainty and external perturbations. First stated in the 1950s, the model reference adaptive control problem is among the most studied ones. It is still relevant despite its respectable age. There are new problems arising, such as increasing the computational performance of algorithms, improving the quality of regulator tuning, generalizing the methods to include wider classes of systems, etc. The first solutions of adaptive control problems were associated with a number of assumptions, such as measurability of the state vector, the assumption for the systems transfer function to be strictly positive and real, or the master signal or external perturbations to be reproducible using an autonomous generator [1–4]. Approaches based on the apparatus of the second Lyapunov method [5] and hyperstability theory [6] required measuring higher derivatives of the tracking error. Later, this problem was solved using the augmented error method [7], high-order adaptation algorithms [8], perturbation compensation methods [9, 10], etc. One of the key issues faced by the above methods is that there is no way to influence the quality of the transient processes [11]. Various options have been proposed to partially resolve this, such as the accelerated convergence scheme [12, 13], the scheme with big coefficients in feedback [14], the nonvanishing excitation condition met [15], non-smooth control laws [16], etc. Thus, the methods [7–10, 12–16] impose significant constraints and solve the problem incompletely, with the control objective attained only in asymptotics. At the same time, the obtained estimates of the characteristics of the limit set are sufficiently rough. In [17–21], a method is proposed that ensures the output signal is in the given set. In [22], based on the method from [17–21], an adaptive control law with the output signal belonging to the given set is proposed. However, in [22], the relative degree of the plant is assumed to equal the unity. In this work, using the approaches from [17–22] and a modified adaptation algorithm [23], we propose a new solution to the adaptive control problem with the given control quality guaranteed [7, 8, 12, 14–16] for minimum-phase plants with an arbitrary

relative degree. The structure of the work is as follows. In Section 2, we state the adaptive tracking problem with constraints on the output variable. In Section 3, we first synthesize a control law that assumes measurability of the derivatives of the plants output signal. The obtained solution is then generalized to the case when these derivatives are not measurable. In Section 4, we give numerical simulations to illustrate the efficiency of the obtained solution.

2. STATEMENT OF THE PROBLEM

We consider the dynamic system

$$Q(p)y(t) = kR(p)u(t) + f(t), \quad (1)$$

where $t \geq 0$, $u(t) \in \mathbb{R}$ is the control signal, $y(t) \in \mathbb{R}$ is the measurable controlled signal, $f(t) \in \mathbb{R}$ is the bounded external perturbation, $Q(s)$ and $R(s)$ are normalized polynomials (i.e., polynomials with higher coefficients equaling the unity) with unknown real coefficients and with the degrees equaling n and m , respectively, $\rho = n - m \geq 1$, the polynomial $R(s)$ is Hurwitz, $p = d/dt$ is the differentiation operator, the coefficient $k > 0$ is unknown, the boundary conditions $y^{(i)}(0)$, $i = \overline{2, n}$ are unknown, but the set of initial conditions $y(0)$ is known. Throughout this work, s is a complex variable.

We give the reference model

$$T(p)y_m(t) = k_m g(t), \quad (2)$$

where $g(t) \in \mathbb{R}$ is a bounded and $(\rho - 1)$ times differentiable master control, $y_m(t) \in \mathbb{R}$ is the output of the reference model, $T(s)$ is the known normalized Hurwitz polynomial with real coefficients and the degree ρ , $k_m > 0$.

The purpose of this work is to synthesize a control law that ensures the tracking error $e(t) = y(t) - y_m(t)$ belongs to the set

$$\mathcal{E} = \left\{ (t, e) \in \mathbb{R}^2 \mid t \geq 0, \underline{g}(t) < e(t) < \overline{g}(t) \right\}, \quad (3)$$

where the functions $\underline{g}(t) < 0$, $\overline{g}(t) > 0$, $\overline{g}(t) - \underline{g}(t) > \delta$, $\delta > 0$ are bounded and have bounded first derivatives for any $t \geq 0$, and $e(0) \in \mathcal{E}$. The derivatives $\underline{g}(t)$ and $\overline{g}(t)$ should be bounded for the method to be applied [19].

3. PRINCIPAL RESULT

3.1. Synthesizing an Ideal Control Law

We introduce an auxiliary control v and first assume its derivatives to be measurable. We consider the control law

$$u(t) = \frac{T(p)}{p} v(t). \quad (4)$$

We represent the polynomials $Q(s)$ and $R(s)$ as $Q(s) = Q_m(s) + \Delta Q(s)$ and $R(s) = R_m(s) + \Delta R(s)$, where the normalized Hurwitz polynomials $Q_m(s)$ and $R_m(s)$ have the degrees n and m , respectively, and $Q_m(s)/R_m(s) = T(s)$. The unknown polynomials $\Delta Q(s)$ and $\Delta R(s)$ have degrees $n - 1$ and $m - 1$, respectively. Taking into account (4), we transform (1) to the form

$$Q_m(p)y(t) = \frac{kR_m(p)T(p)}{p} v(t) + \frac{k\Delta R(p)T(p)}{p} v(t) - \Delta Q(p)y(t) + f(t). \quad (5)$$

Dividing (5) by $Q_m(p)$, $kR_m(p)T(p)$ and p , we write the result as

$$y(t) = \frac{k}{p} \left[v(t) + \frac{\Delta R(p)}{R_m(p)} v(t) - \frac{p\Delta Q(p)}{kQ_m(p)} y(t) + \frac{p}{kQ_m(p)} f(t) + \epsilon_1(t) \right], \quad (6)$$

where $\epsilon_1(t)$ is an exponentially damped function depending on initial conditions (1). We express $y_m(t)$ from (2) as

$$y_m(t) = \frac{k}{p} \left[\frac{k_m}{k} g_r(t) + \epsilon_2(t) \right], \quad (7)$$

where $g_r(t) = \frac{p}{T(p)} g(t)$, $\epsilon_2(t)$ is an exponentially damped function depending on initial conditions (2). Taking into account the structure of (6) and (7), we rewrite $e(t)$ as

$$e(t) = \frac{k}{p} \left[v(t) - c_{01} y(t) - c_{02}^\top \zeta_y(t) - c_{03}^\top \zeta_v(t) - \frac{k_m}{k} g_r(t) + \frac{p}{k Q_m(p)} f(t) + \epsilon(t) \right], \quad (8)$$

where $\epsilon(t) = \epsilon_1(t) - \epsilon_2(t)$, the signals $\zeta_y(t)$, $\zeta_v(t)$ and $g_r(t)$ are formed using the following filters

$$\begin{aligned} \dot{\zeta}_y(t) &= F_y \zeta_y(t) + b_y y(t), \quad \zeta_y(0) = 0, \\ \dot{\zeta}_v(t) &= F_v \zeta_v(t) + b_v v(t), \quad \zeta_v(0) = 0, \\ \dot{\zeta}_g(t) &= F_g \zeta_g(t) + b_g g(t), \quad \zeta_g(0) = 0, \quad g_r(t) = L_2 \zeta_g(t). \end{aligned} \quad (9)$$

Here, $b_i^\top = [0 \dots 0 \ 1]$ is the vector-column with the unity at the last position and zeros at the others, $i \in \{y, v, g, \eta\}$, $L_j = [0 \dots 0 \ 1 \ 0 \dots 0]$ is the row vector with the unity at the j th position and zeros at the others. Here and in what follows, the matrices b_i and L_j have the same structure, and their dimensionality is clear from the context. The matrices F_y , F_v and F_g of filters (9) are given in the Frobenius form with the characteristic polynomials $Q_m(s)$, $R_m(s)$ and $T(s)$, respectively. The coefficient c_{01} is obtained from the expression

$$\frac{p \Delta Q(p)}{k Q_m(p)} = c_{01} + \frac{\Delta \tilde{Q}(p)}{k Q_m(p)},$$

where $\Delta \tilde{Q}(s)$ is an unknown polynomial of the degree $n - 1$. The vectors c_{02} and c_{03} consist of the coefficients of the polynomials $\Delta \tilde{Q}(s)/k$ and $\Delta R(s)/k$, respectively. We introduce the vector of constant unknown parameters $c_0^\top = [c_{01} \ c_{02}^\top \ c_{03}^\top \ k_m/k]$ and the regression vector $\omega^\top(t) = [y(t) \ \zeta_y^\top(t) \ \zeta_v^\top(t) \ g_r(t)]$. We rewrite (8) as

$$\dot{e}(t) = k \left(v(t) - c_0^\top \omega(t) + \bar{f}(t) + \epsilon(t) \right), \quad (10)$$

where $\bar{f}(t) = \frac{p}{k Q_m(p)} f(t)$ is the signal bounded since $Q_m(s)$ is Hurwitz.

We apply the method from [19] to set the auxiliary control $v(t)$ that ensures the error $e(t)$ is in set (3). To do this, we introduce an auxiliary signal $\varepsilon(t)$ calculated by the formula

$$e(t) = \Phi(\varepsilon(t), t), \quad (11)$$

where the function $\Phi(\varepsilon(t), t)$ satisfies the conditions

- (a) $\underline{g}(t) < \Phi(\varepsilon(t), t) < \overline{g}(t)$ for any $t \geq 0$ and $\varepsilon \in \mathbb{R}$;
- (b) the function $\Phi(\varepsilon(t), t)$ is continuously differentiable on $\varepsilon(t)$ and t , and for any $e \in \mathcal{E}$ and $t \geq 0$, the following holds

$$\frac{\partial \Phi}{\partial \varepsilon}(\varepsilon(t), t) \neq 0;$$

- (c) the function $\frac{\partial \Phi}{\partial t}(\varepsilon(t), t)$ is bounded for any $\varepsilon \in \mathbb{R}$ and $t \geq 0$.

We give an example of transformation (11) in the form

$$\Phi(\varepsilon(t), t) = \frac{\overline{g}(t) \exp(\varepsilon(t)) + \underline{g}(t)}{\exp(\varepsilon(t)) + 1}. \quad (12)$$

For other examples of such functions (11) of coordinates, see [19].

Taking into account (11), we consider the total time derivative of $e(t)$ in the form

$$\dot{e}(t) = \frac{\partial \Phi}{\partial \varepsilon}(\varepsilon(t), t) \dot{\varepsilon}(t) + \frac{\partial \Phi}{\partial t}(\varepsilon(t), t).$$

Taking into account (10) and property (b), we express $e(t)$ as

$$\dot{e}(t) = \left(\frac{\partial \Phi}{\partial \varepsilon}(\varepsilon(t), t) \right)^{-1} \left(k(v(t) - c_0^\top \omega(t) + \bar{f}(t) + \epsilon(t)) - \frac{\partial \Phi}{\partial t}(\varepsilon(t), t) \right). \quad (13)$$

Now, we choose an auxiliary control action and an adaptation algorithm in the form

$$v(t) = -\operatorname{sgn} \left\{ \frac{\partial \Phi}{\partial \varepsilon}(\varepsilon(t), t) \right\} \alpha \varepsilon(t) + c^\top(t) \omega(t), \quad (14)$$

$$\dot{c}(t) = - \left(\frac{\partial \Phi}{\partial \varepsilon}(\varepsilon(t), t) \right)^{-1} \varepsilon(t) \omega(t) - \gamma c(t), \quad (15)$$

where $\alpha > 0$ and $\gamma > 0$, $c(t)$ is the vector of adjustable parameters. The transformation $\Phi(\varepsilon(t), t)$ is chosen beforehand, and the degree of $\frac{\partial \Phi}{\partial \varepsilon}(\varepsilon(t), t)$ is constant due to property (b), so the value of the function $\operatorname{sgn}\{\cdot\}$ in (14) is known. Substituting (14) into (13), we obtain the following expression

$$\dot{e}(t) = \left(\frac{\partial \Phi}{\partial \varepsilon}(\varepsilon(t), t) \right)^{-1} \left(-\operatorname{sgn} \left\{ \frac{\partial \Phi}{\partial \varepsilon}(\varepsilon(t), t) \right\} k \alpha \varepsilon(t) + k(c(t) - c_0)^\top \omega(t) + \Xi(t) \right), \quad (16)$$

where $\Xi(t) = k(\bar{f}(t) + \epsilon(t)) - \frac{\partial \Phi}{\partial t}(\varepsilon(t), t)$ is a bounded function. We state a theorem and prove it in the Appendix.

Theorem 1. Suppose the functions $\underline{g}(t)$ and $\bar{g}(t)$ satisfy the imposed requirements (see the paragraph below (3)), and properties (a)–(c) of transformation (11) hold. Then, for any $\alpha > 0$ and $\gamma > 0$ in closed system (1), (2), (9), (15), (16), control objective (3) is attained.

Remark 1. In [22], the problem with the plant's relative degree equaling 1 and the known coefficient k is considered. Unlike the algorithm presented in [22], the procedure proposed in this work allows excluding the term containing $e(t)$ from the dynamics $\varepsilon(t)$, choose a simplified control law, and overcome the issue with the unknown k .

3.2. Synthesizing a Feasible Control Law

Control law (4) contains the derivatives of the auxiliary control v up to the $(\rho - 1)$ th order. We introduce the estimate $\tilde{v}$ of auxiliary control (14). Then, the new control law looks like

$$u(t) = \frac{T(p)}{p} \tilde{v}(t), \quad (17)$$

$$\dot{\xi}(t) = G_0 \xi(t) + D_0(\tilde{v}(t) - v(t)), \quad \tilde{v}(t) = \xi_1(t) = L_1 \xi(t). \quad (18)$$

Here, $\xi(t) \in \mathbb{R}^\rho$ is the vector of signal estimates v and its derivatives,

$$G_0 = \begin{bmatrix} 0 & I_{\rho-1} \\ 0 & 0 \end{bmatrix} \quad D_0^\top = \begin{bmatrix} -\frac{d_1}{\mu} & -\frac{d_2}{\mu^2} & \cdots & -\frac{d_\rho}{\mu^\rho} \end{bmatrix},$$

the numbers d_i , $i = \overline{1, \rho}$ are chosen so that the matrix $G = G_0 + DL_1$ is Hurwitz, $D^\top = \begin{bmatrix} d_1 & d_2 & \cdots & d_\rho \end{bmatrix}$, and $\mu > 0$ is a sufficiently small number. We introduce the vector

$$\eta(t) = \Gamma^{-1}(\xi(t) - \theta(t)), \quad \Gamma = \operatorname{diag} \left\{ \mu^{\rho-1}, \mu^{\rho-2}, \dots, \mu, 1 \right\}, \quad (19)$$

where $\theta^\top(t) = [v(t) \quad \dot{v}(t) \quad \dots \quad v^{(\rho-1)}(t)]$. Taking into account (19), we write

$$\Delta v(t) = \tilde{v}(t) - v(t) = \mu^{\rho-1} \eta_1(t) = \mu^{\rho-1} L_1 \eta(t), \quad \tilde{v}(t) = v(t) + \mu^{\rho-1} L_1 \eta(t). \quad (20)$$

Taking into account (17) and (20), we express $\dot{\varepsilon}$ as

$$\begin{aligned} \dot{\varepsilon}(t) = & \left(\frac{\partial \Phi}{\partial \varepsilon}(\varepsilon(t), t) \right)^{-1} \left(-\operatorname{sgn} \left\{ \frac{\partial \Phi}{\partial \varepsilon}(\varepsilon(t), t) \right\} k \alpha \varepsilon(t) \right. \\ & \left. + k(c(t) - c_0)^\top \omega(t) + \mu^{\rho-1} k L_1 \eta(t) + \Xi(t) \right). \end{aligned} \quad (21)$$

Taking into account (18) and (19), we obtain

$$\mu \dot{\eta}(t) = G \eta(t) - \mu b_\eta v^{(\rho)}(t). \quad (22)$$

We state the principal result of the work.

Theorem 2. *Suppose the hypotheses of Theorem 1 hold. Then, there exists a number μ_0 such that control objective (3) is attained for $\mu \leq \mu_0$ in closed system (1), (2), (9), (15), (21), (22).*

Remark 2. Equation (22) is a singularly perturbed dynamical system. The analysis of such systems shows [24] that under certain conditions on the right-hand side of the system and for sufficiently small μ , the system has the same dissipativity domain and the same attraction domain as the system for $\mu = 0$, which is equivalent to using an ideal control law. As we showed in Theorem 1, such a system attains objective (3), so it will be sufficient to use one of the results of the theory of singularly perturbed systems to prove the theorem.

Theorem 2 shows the existence of a sufficiently small parameter μ_0 while the search for its actual value is an unsolved problem. The search for some quantitative characteristics, whose structure includes high-order observers, is not always possible and often remains in the qualitative form [28, 29]. Also, works [30, 31] point out that the iterative search for the value μ_0 , for which the stability of the closed-loop system is achieved, can be carried out at the simulation stage.

Proof of Theorem 2. We rewrite (21) and (22) as

$$\begin{aligned} \dot{\varepsilon}(t) = & \left(\frac{\partial \Phi}{\partial \varepsilon}(\varepsilon(t), t) \right)^{-1} \left(-\operatorname{sgn} \left\{ \frac{\partial \Phi}{\partial \varepsilon}(\varepsilon(t), t) \right\} k \alpha \varepsilon(t) \right. \\ & \left. + k(c(t) - c_0)^\top \omega(t) + \mu_2^{\rho-1} k L_1 \eta(t) + \Xi(t) \right), \end{aligned} \quad (23)$$

$$\mu_1 \dot{\eta}(t) = G \eta(t) - \mu_2 b_\eta v^{(\rho)}(t),$$

where $\mu_1 = \mu_2 = \mu$. We use the lemma [29].

Lemma 1. *If the system is described by the equation $\dot{x} = f(x, \mu_1, \mu_2)$, $x \in \mathbb{R}^{m_0}$, where $f(x, \mu_1, \mu_2)$ is a continuous function Lipschitzian with respect to x and has a bounded closed dissipativity domain $\mu_2 = 0$ for $D = \{x \mid F(x) < C\}$, where $F(x)$ is a continuous piecewise smooth function positive definite in $\mathbb{R}^{m_0}$ (in the sense of [26]), then there exists $\mu_0 > 0$ such that for $\mu_2 \leq \mu_0$ the original system has the same dissipativity domain D if the relation*

$$\sup_{|\mu_1| \leq \bar{\mu}_1} \left\{ \left\langle \frac{\partial F}{\partial x}(x), f(x, \mu_1, 0) \right\rangle \right|_{F(x)=C} \right\} \leq -C_1$$

holds when $\mu_2 = 0$ for some $C_1 > 0$ and $\bar{\mu}_1 > 0$.

Substituting $\mu_2 = 0$ into (23), we have the closed-loop system with the ideal control law obtained in the previous subsection. Theorem 1 shows that the solutions of the closed-loop system tend to a bounded set, and the additional equation $\mu_1 \dot{\eta}(t) = G\eta(t)$ does not violate this condition since G is Hurwitz. Hence, for a sufficiently small μ , the system preserves the dissipativity domain. The theorem is proved.

4. SIMULATION

Example 1. We consider control plant (1) with the following linear differential operators

$$Q(p) = p^4 + 6p^3 - 3p^2 - p + 2 \quad \text{and} \quad kR(p) = p + 1.$$

The initial conditions are $p^3 y(0) = p^2 y(0) = p y(0) = y(0) = 1$. The external perturbation is $f(t) = 2 \sin(1.5t) + d(t)$, where $d(t) = \text{sat}\{\hat{d}(t)\}$, $\text{sat}\{\cdot\}$ is the saturation function, $\hat{d}(t)$ is white noise simulated in Matlab Simulink using the Band-Limited White Noise block with the Noise power parameter equaling 1 and the Sample time parameter equaling 0.1. Reference model (2)

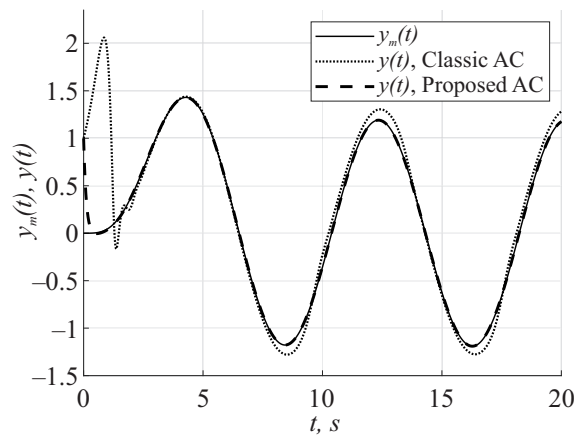


Fig. 1. The output of the reference model (the black curve), the output of plant (1) when the high-order adaptive robust algorithm is used (the blue curve), and the output of plant (1) when the proposed algorithm is used (the red curve).

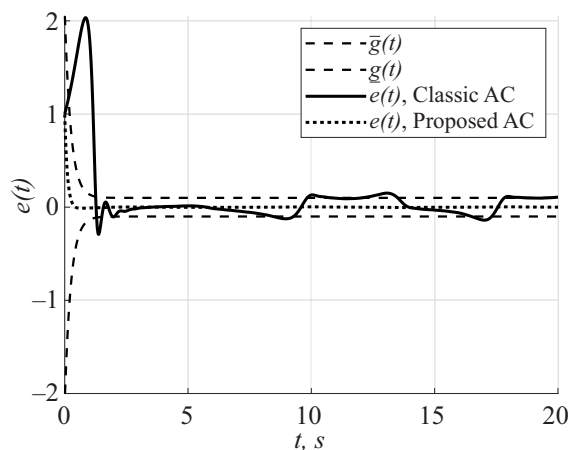


Fig. 2. The constraints $\bar{g}(t)$ and $g(t)$ (the blue and red curves), the control error when the high-order adaptive robust algorithm is used (the green curve), and the control error when the proposed algorithm is used (the black curve).

that receives the master control $g(t) = 2.5 \sin(0.8t)$ has the following parameters

$$T(p) = p^3 + 3p^2 + 3p + 1 \quad \text{and} \quad k_m = 1.$$

According to (4), we set the control law in the form $u(t) = \frac{T(p)}{p}v(t)$. As the function $\Phi(\varepsilon(t), t)$ we take (12). We give the constraints for the tracking error by the functions

$$\begin{aligned} \bar{g}(t) &= 2 \exp(-4t) + 0.1, \\ \underline{g}(t) &= -\bar{g}(t). \end{aligned}$$

We choose $Q_m(p) = p^4 + 4p^3 + 6p^2 + 4p + 1$, $R_m(p) = p + 1$ and set in filters (9):

$$F_v = -1, \quad F_g = \begin{bmatrix} 0 & I_2 & \\ -1 & -3 & -3 \end{bmatrix}, \quad F_y = \begin{bmatrix} 0 & I_3 & \\ -1 & -4 & -6 & -4 \end{bmatrix}.$$

We choose the parameters $\alpha = 5$, $\gamma = 1$ and $\mu = 10^{-3}$ in (14), (15), and (18). We compare the proposed control algorithm with the classical high-order adaptive algorithm [31] with the parameters $\sigma_c = 5$, $\gamma_c = 100$ and $\mu_c = 200$.

Figures 1 and 2 show that the classical algorithm fails. First, there is no way to set the control quality during transient processes. Secondly, the limit set is not defined in advance and turns out to be bigger than that specified using the proposed algorithm.

5. CONCLUSIONS

In this work, we propose a new model reference adaptive tracking algorithm with the given control quality guaranteed over the entire interval of system operation. The method combines the idea of adaptive control [23] and the approach [19] used to synthesize a nonlinear control law under constraints. We compared the obtained result with the classical high-order adaptive control law [27] by simulation. The proposed algorithm allowed us to ensure that the tracking error is in a predetermined set at any instant while the methods from [23, 27] do not allow controlling the values of the tracking error in the transient mode. Moreover, the estimates of the value of the limit objective set obtained in [23, 27] depend on the values of unknown parameters and have overestimated values, therefore, they cannot be used to set the control accuracy in the steady-state mode. Thus, the proposed approach solves both these problems, and the control quality is completely determined by the choice of bounding functions.

FUNDING

The work was performed in the Institute for Problems in Mechanical Engineering of the Russian Academy of Sciences within State Assignment no. FFNF-2024-0008 (no. 124041100006-1 in the Unified State Information System for Accounting of Research, Experimental Design and Technological Works).

APPENDIX

Proof of Theorem 1. By property (b) of transformation (11), the function $\frac{\partial \Phi}{\partial \varepsilon}(\varepsilon(t), t)$ is sign-definite. First, suppose that $\frac{\partial \Phi}{\partial \varepsilon}(\varepsilon(t), t) > 0$. We choose a Lyapunov function of the form

$$V_1 = 0.5\varepsilon^2(t) + 0.5k(c(t) - c_0)^\top(c(t) - c_0) + \chi \int_t^\infty \left(\frac{\partial \Phi}{\partial \varepsilon}(\varepsilon(s), s) \right)^{-1} \varepsilon^2(s) ds,$$

where $\chi > 0$. Taking into account (15) and (16), we find $\dot{V}_1$ in the form

$$\begin{aligned} \dot{V}_1 = & -\alpha k \left(\frac{\partial \Phi}{\partial \varepsilon}(\varepsilon(t), t) \right)^{-1} \varepsilon^2(t) + \left(\frac{\partial \Phi}{\partial \varepsilon}(\varepsilon(t), t) \right)^{-1} \varepsilon(t)(\psi(t) + k\varepsilon(t)) \\ & - \gamma k(c(t) - c_0)^\top c(t) - \chi \left(\frac{\partial \Phi}{\partial \varepsilon}(\varepsilon(t), t) \right)^{-1} \varepsilon^2(t), \end{aligned} \quad (\text{A.1})$$

where $\psi(t) = k\bar{f}(t) - \frac{\partial \Phi}{\partial t}(\varepsilon(t), t)$ is bounded. We use the following relations

$$\begin{aligned} \varepsilon(t)\psi(t) & \leq 0.5(\nu^{-1}\varepsilon^2(t) + \nu\psi^2(t)), \\ \varepsilon(t)\varepsilon(t) & \leq 0.5(\nu^{-1}\varepsilon^2(t) + \nu\varepsilon^2(t)), \\ -(c(t) - c_0)^\top c(t) & = -0.5((c(t) - c_0)^\top (c(t) - c_0) + c^\top(t)c(t) - c_0^\top c_0). \end{aligned} \quad (\text{A.2})$$

Taking into account (A.2), we rewrite (A.1) as

$$\begin{aligned} \dot{V}_1 \leq & -\left(\frac{\partial \Phi}{\partial \varepsilon}(\varepsilon(t), t) \right)^{-1} (\alpha k - 0.5\nu^{-1}(1+k))\varepsilon^2(t) \\ & + 0.5 \left(\frac{\partial \Phi}{\partial \varepsilon}(\varepsilon(t), t) \right)^{-1} \nu\psi^2(t) + 0.5\gamma k c_0^\top c_0 \\ & - 0.5\gamma k((c(t) - c_0)^\top (c(t) - c_0) + c^\top(t)c(t)) \\ & - \left(\frac{\partial \Phi}{\partial \varepsilon}(\varepsilon(t), t) \right)^{-1} (\chi - 0.5k\nu)\varepsilon^2(t). \end{aligned} \quad (\text{A.3})$$

Note that it follows from properties (a) and (b) of function (11) that it is differentiable everywhere, strictly monotone and bounded, and hence $\sup_{t \geq 0} \left\{ \frac{\partial \Phi}{\partial \varepsilon}(\varepsilon(t), t) \right\} < \infty$ is bounded. Moreover, $\sup_{t \geq 0} \{\psi(t)\} < \infty$ is bounded by property (c). Then, it follows from (A.3) that $\dot{V}_1 < 0$ is attained when the condition

$$\begin{aligned} \nu & > 0.5\alpha^{-1}(k^{-1} + 1), \quad \chi > 0.5k\nu, \\ |\varepsilon(t)| & > \sqrt{\frac{0.5}{\alpha k - 0.5\nu^{-1}(1+k)} \left(\frac{\nu}{k} \sup_{t \geq 0} \{\psi(t)\}^2 + \sup_{t \geq 0} \left\{ \frac{\partial \Phi}{\partial \varepsilon}(\varepsilon(t), t) \right\} \gamma k c_0^\top c_0 \right)} \end{aligned} \quad (\text{A.4})$$

holds.

Obviously, there will always be ν and χ such that (A.4) holds. Hence, $\dot{V}_1 < 0$.

Now, we consider the case $\frac{\partial \Phi}{\partial \varepsilon}(\varepsilon(t), t) < 0$. We choose a Lyapunov function of the form

$$V_2 = 0.5\varepsilon^2(t) + 0.5k(c(t) - c_0)^\top (c(t) - c_0).$$

We find $\dot{V}_2$ as

$$\dot{V}_2 = \left(\frac{\partial \Phi}{\partial \varepsilon}(\varepsilon(t), t) \right)^{-1} k\alpha\varepsilon^2(t) + \left(\frac{\partial \Phi}{\partial \varepsilon}(\varepsilon(t), t) \right)^{-1} \varepsilon(t)(\psi(t) + k\varepsilon(t)) - \gamma k(c(t) - c_0)^\top c(t). \quad (\text{A.5})$$

Using relations (A.2) for $\nu = 1$, we rewrite (A.5) as

$$\begin{aligned} \dot{V}_2 \leq & \left(\frac{\partial \Phi}{\partial \varepsilon}(\varepsilon(t), t) \right)^{-1} (\alpha k + 0.5k + 0.5)\varepsilon^2(t) + 0.5 \left(\frac{\partial \Phi}{\partial \varepsilon}(\varepsilon(t), t) \right)^{-1} \psi^2(t) + 0.5\gamma k c_0^\top c_0 \\ & - 0.5\gamma k((c(t) - c_0)^\top (c(t) - c_0) + c^\top(t)c(t)) + 0.5 \left(\frac{\partial \Phi}{\partial \varepsilon}(\varepsilon(t), t) \right)^{-1} k\varepsilon^2(t). \end{aligned} \quad (\text{A.6})$$

Since $\frac{\partial \Phi}{\partial \varepsilon}(\varepsilon(t), t) < 0$, all terms but $0.5\gamma k c_0^\top c_0$ are negative. Previously, we have shown that $\sup_{t \geq 0} \left\{ \frac{\partial \Phi}{\partial \varepsilon}(\varepsilon(t), t) \right\} < \infty$. Then, it follows from (A.6) that $\dot{V}_2 < 0$ is attained when the condition

$$|\varepsilon(t)| > \sqrt{-\inf_{t \geq 0} \left\{ \frac{\partial \Phi}{\partial \varepsilon}(\varepsilon(t), t) \right\} \frac{0.5\gamma k c_0^\top c_0}{\alpha k + 0.5k + 0.5}} \quad (\text{A.7})$$

holds.

Inequalities (A.4) and (A.7) define the sets trajectories (16) tend to in each of the cases involved. Then, by Theorem 3.1 from [19], trajectories (10) will belong to some subset $\mathcal{E}$, which means that objective (3) is fulfilled. The theorem is proved.

REFERENCES

1. Landau, I.D., A survey of model reference adaptive techniques-Theory and applications, *Automatica*, 1974, vol. 10, no. 4, pp. 353–379, ISSN 0005-1098. [https://doi.org/10.1016/0005-1098\(74\)90064-8](https://doi.org/10.1016/0005-1098(74)90064-8)
2. Chang-chieh Hang and Parks, P.C., Comparative studies of model reference adaptive control systems, *IEEE Trans. Autom. Control.*, 1973, vol. 18, no. 5, pp. 419–428.
3. Johnson, C.D., Accommodation of external disturbances in linear regulator and servomechanism problems, *IEEE Trans. Autom. Control.*, 1971, vol. 16, no. 6, pp. 635–644.
4. Davison, E.J., The robust control of a servomechanism problem for linear timeinvariant multivariable systems, *IEEE Trans. Autom. Control.*, 1976, vol. 21, pp. 25–34.
5. Parks, P., Liapunov redesign of model reference adaptive control systems, *IEEE Trans. Autom. Control.*, 1976, vol. 11, no. 3, pp. 362–367.
6. Landau, I.D. and Ioan, D., A hyperstability criterion for model reference adaptive control systems, *IEEE Trans. Autom. Control.*, 1969, vol. 14, pp. 552–555.
<https://api.semanticscholar.org/CorpusID:121498902>
7. Morse, A.S., Global stability of parameter adaptive control systems, *IEEE Trans. Autom. Control.*, 1980, vol. 25, no. 3, pp. 433–439.
8. Morse, A.S., High-order parameter tuners for the adaptive control of nonlinear systems, *US-Italy Seminar Syst. Model. Feedback Theor. Appl.*, 1992, vol. 12, pp. 339–364.
https://doi.org/10.1007/978-1-4757-2204-8_23
9. Andrievsky, B.R. and Furtat, I.B., Disturbance observers: methods and applications. I. Methods, *Autom. Remote Control*, 2020, vol. 81, no. 9, pp. 1563–1610.
10. Andrievsky, B.R. and Furtat, I.B., Disturbance observers: methods and applications. II. Applications, *Autom. Remote Control*, 2020, vol. 81, no. 10, pp. 1775–1818.
11. Zang, Z. and Bitmead, R., Transient bounds for adaptive control systems, *IEEE Trans. Autom. Control.*, 1991, vol. 5, no. 39, pp. 2724–2729. <http://dx.doi.org/10.1109/CDC.1990.203273>
12. Kreisselmeier, G., Adaptive observers with exponential rate of convergence, *IEEE Trans. Autom. Control.* 1977, vol. 22, no. 1, pp. 2–8. <https://doi.org/10.1109/TAC.1977.1101401>
13. Lion, P.M., Rapid identification of linear and nonlinear systems, *AIAA J.*, 1967, vol. 5, pp. 1835–1842.
14. Sun, J., A modified model reference adaptive control scheme for improved transient performance, *IEEE Trans. Autom. Control.*, 1993, vol. 38, no. 8, pp. 1255–1259. <https://doi.org/10.1109/9.233162>
15. Arteaga, M. and Tang, Y., Adaptive control of robots with an improved transient performance, *IEEE Trans. Autom. Control.* 2002, vol. 47, pp. 1198–1202. <http://dx.doi.org/10.1109/TAC.2002.800672>
16. Morse, A.S., Supervisory Control of Families of Linear Set-Point Controllers-Part I. Exact Matching, *IEEE Trans. Autom. Control.*, 1996, vol. 41, no. 10, pp. 1413–1431. <https://doi.org/10.1109/9.539424>

17. Furtat, I.B., Divergent stability conditions of dynamic systems, *Avtomat. i Telemekh.*, 2020, no. 2, pp. 62–75; *Autom. Remote Control*, 2020, vol. 81, no. 2, pp. 247–257.
18. Furtat, I.B. and Gushchin P.A., Control of dynamical plants with a guarantee for the controlled signal to stay in a given set, *Autom. Remote Control*, 2021, vol. 82, no. 4, pp. 654–669.
19. Furtat, I.B. and Gushchin, P.A., Nonlinear feedback control providing plant output in given set, *Int. J. Control*, 2022, vol. 95, no. 6, pp. 1533–1542. <https://doi.org/10.1080/00207179.2020.1861336>
20. Furtat, I.B., Density systems: analysis and control, *Avtomat. i Telemekh.*, 2023, no. 11, pp. 55–76.
21. Furtat, I.B., Gushchin, P.A., and Nguyen, B., Control of dynamic systems under input and output constraints, *Autom. Remote Control*, 2023, vol. 84, no. 4, pp. 397–411.
22. Furtat, I.B., Gushchin, P.A., Nguyen, B., and Kolesnik, N.S., Adaptive control with a guarantee of a given performance, *UBS*, 2023, vol. 102, pp. 44–57. <https://doi.org/10.25728/ubs.2023.102.3>
23. Tsykunov, A.M., A modified high-order adaptive output feedback control algorithm for linear plants, 2006, *Autom. Remote Control*, vol. 67, no. 8, pp. 1311–1321.
24. Khalil, H.K., *Nonlinear systems*, Moscow-Izhevsk: Institute of Computer Research, 2009.
25. Atassi, A.N. and Khalil, H.K., A separation principle for the stabilization of a class of nonlinear systems, *IEEE Transact. Autom. Control*, 1999, vol. 44, no. 9, pp. 1672–1687. <https://doi.org/10.1109/9.788534>
26. Borisov, O.I., Pyrkin, A.A., and Isidori, A., Robust Output Regulation of Permanent Magnet Synchronous Motors by Enhanced Extended Observer, *IFAC-PapersOnLine*, 2020m vol. 53, no. 2, pp. 4881–4886, ISSN 2405-8963. <https://doi.org/10.1016/j.ifacol.2020.12.1056>
27. Tsykunov, A.M., An Algorithm of Robust Control of a Non-Stationary Linear Plant with Perturbation Compensation, *Journal of Computer and Systems Sciences International*, 2008, vol. 47, no. 4, pp. 527–534.
28. Tsykunov, A.M., Robust control algorithms with compensation of bounded perturbations, *Autom. Remote Control*, 2007, vol. 68, no. 7, pp. 1213–1224.
29. Brusin, V.A., On a class of singularly disturbed adaptive systems. I, *Autom. Remote Control*, 1995, vol. 56, no. 4, pp. 552–559. <https://doi.org/10.25728/ubs.2023.102.3>
30. Rouche, N., Abets, P., and Lalua M., *Direct Lyapunov Method in Stability Theory*, Moscow, Mir Publishers, 1980.
31. Miroyschnik, I.V., Nikiforov, V.O., and Fradkov, A.L., *Nelineinoe i adaptivnoe upravlenie slozhnymi dinamicheskimi sistemami [Nonlinear and Adaptive Control of Complex Dynamical Systems]*, Saint Petersburg: Nauka, 2000.

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

Design of Self-Checking Discrete Devices Based on Polynomial Codes with Computation Control via Several Diagnostic Attributes

D. V. Efanov^{*,**,a} and D. V. Pivovarov^{***,b}

**Peter the Great St. Petersburg Polytechnic University, St. Petersburg, Russia*

***Russian University of Transport (MIIT), Moscow, Russia*

****Emperor Alexander I St. Petersburg State Transport University, St. Petersburg, Russia*

e-mail: ^aTrES-4b@yandex.ru, ^bpivovarov.d.v.spb@gmail.com

Received November 28, 2024

Revised March 17, 2025

Accepted March 27, 2025

Abstract—It is proposed to use polynomial codes in the design of self-checking discrete devices with computation control via several diagnostic attributes. A fast algorithm is developed to obtain functions describing check symbols of polynomial codes in the form of logical expressions. As is shown, encoders of polynomial codes can be referred to devices of three types: 1) those with only self-dual Boolean functions realized at the outputs, 2) those with only nearly self-dual (self-quasidual) Boolean functions realized at the outputs, and 3) those with both self-dual and self-quasidual Boolean functions realized at the outputs. A classification of polynomial codes considering this feature is presented. The structural diagram of computation control at the outputs of self-dual discrete devices via several diagnostic attributes is described. A novel algorithm is proposed to design a fully self-checking discrete device with computation control via several diagnostic attributes. In contrast to the well-known ones, this algorithm considers the nature of errors occurring at the outputs of discrete devices and their preliminary detection by checkers of self-dual and/or self-quasidual signals. The results can be used in the development of automated design tools for discrete devices for a wide range of applications.

Keywords: self-checking discrete devices, computation control via several diagnostic attributes, control of self-duality of signals, control of self-quasiduality of signals, polynomial codes

DOI: 10.31857/S0005117925050038

1. INTRODUCTION

The growing complexity of the developed and implemented technical systems that support critical industrial processes, as well as the increase in their performance and the appearance of advanced functionality, including artificial intelligence, require special attention to the reliability and safety of operation. In this regard, the most important aspect is to provide devices with the ability to identify faults during operation.

When designing discrete systems and control devices, their structures are endowed with various properties to easily ensure the detection of occurring faults. There exist methods for designing controllable devices [1] and developing self-checking structures for them [2]. Such methods apply to any critical-purpose systems, including those with rarely changing input data (e.g., control systems in the nuclear industry, electrical interlocking in railway transport, air defense systems, etc.). In such systems, hidden faults and the accumulation of errors are not allowed: one way or another, they will ultimately violate the readiness property of the control algorithm [3, 4].

The self-checking implementation of discrete devices is required to detect their faults in due time. This is done by various methods with the introduction of structural redundancy according to definite algorithms, e.g., state coding and the use of self-checking concurrent error-detection (CED) circuits [5].

An effective approach to implementing self-checking CED circuits involves the so-called data inversion: all input and output signals are represented as sequences 0101...01 and 1010...10 instead of constant values 0 and 1 [6]. This approach to implementing self-checking discrete devices is associated with using hardware redundancy and, moreover, temporal redundancy. It includes computation control via a definite diagnostic attribute, considering the constant change of input signals and the supply of input argument sets in pairs, i.e., the operating and check sets of argument values. Self-dual functions and Boolean functions [7], “close” to the former, possess the peculiarities that can be considered when organizing diagnostic support in this operation mode. The issues of implementing self-dual discrete devices were studied in several research works, e.g., [6, 8–11]. Also, note three monographs covering the basic results from the theory of self-dual discrete device design [12–14].

Computation control via the self-duality attribute requires the self-dual implementation of discrete devices, implying an appropriate structure to describe all their outputs by self-dual functions. Not all discrete devices are self-dual. Here are some examples: a full adder, a majority element, encoders of some separable block codes, etc. [6]. However, since the number of self-dual functions of t variables equals 2^{2^t-1} , it is possible to design $C_{2^{2^t-1}}^m$ combinational devices with $m \leq 2^{2^t-1}$ outputs. For example, for $t = 4$ and $m = 4$, the number of different self-dual combinational discrete devices reaches 174 792 640. In addition, any structure of a combinational discrete device can be transformed into a self-dual one using only a single variable based on the well-known Shannon decomposition [15]. For a self-dual implementation of memory devices, it suffices to make the combinational parts self-dual [6]. The design peculiarities of self-dual discrete devices are well-studied; for example, see [9].

In this paper, the reader’s attention is focused on the issues of organizing computation control at the outputs of self-dual discrete devices, using not only the diagnostic attribute of signal self-duality but also additional attributes. As shown by numerous experiments, this self-checking device design approach allows increasing checkability (in terms of error observability) under a smaller level of hardware redundancy compared to the well-known duplication method [16]. This effect is achieved by using properties of some noise-immune codes with self-dual and “close” functions describing their check symbols [17]. As such, we consider a wide class of polynomial, or algebraic, codes [18].

2. POLYNOMIAL CODES AND THEIR PECULIARITIES

2.1. Construction Principles of Polynomial Codes

Polynomial codes are based on the division, with remainder (residue), of binary polynomials corresponding to data messages. When dividing different polynomials of degree q by a generating polynomial $G(y)$, 2^q residues are obtained. Each check symbol of the polynomial code is thus described by a logical expression containing only mod $M = 2$ addition (the XOR operation) with the values of definite data symbols. For the sake of simplicity, let the functions describing the check symbols of the codewords of polynomial codes be called *check functions*.

When forming the codewords of polynomial codes, each bit of the code vector is assigned a variable y with degree j , which corresponds to the bit location: the least significant bit is assigned the value $j = 0$, the next the value $j = 1$, and so on. The algebraic representation of the code vector is obtained by multiplying the value of y^j by the value of the corresponding bit. For example, the codeword $< 1011 >$ can be written as a polynomial as follows: $1 \times y^3 + 0 \times y^2 + 1 \times y^1 + 1 \times y^0$. By removing the zero terms, we have the polynomial $y^3 + y + 1$.

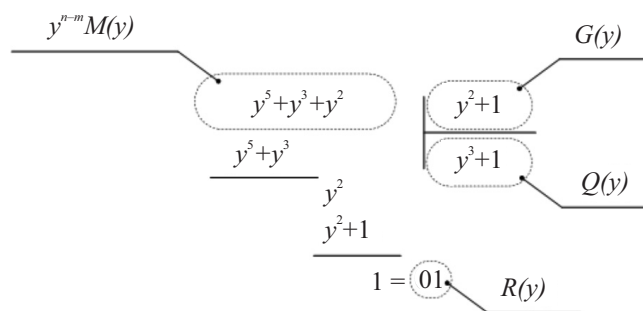


Fig. 1. The division of polynomials with residue: an illustrative example.

To form the codeword $V(y)$ of a polynomial code, the residual polynomial $R(y)$ is determined by dividing the polynomial $M(y)$, with the factor y^{n-m} , by the generating polynomial [18]: $y^{n-m}M(y) = G(y)Q(y) + R(y)$, where $Q(y)$ and $R(y)$ denote the quotient and residue from dividing $y^{n-m}M(y)$ by $G(y)$. In this case, the residue $R(y)$ will be the check vector occupying the k least significant bits of the codeword of length n . The m most significant bits will correspond to the data vector.

We encode the data vector $\langle 1011 \rangle$ by a polynomial code with the generating polynomial $y^2 + 1$. This polynomial has degree 2. (It must not exceed the degree of the data polynomial.) When multiplication is performed, the polynomial $y^{n-m}M(y)$ takes the form $y^2(y^3 + y + 1) = y^5 + y^3 + y^2$.

Let us divide it by the generating polynomial $y^2 + 1$ (Fig. 1).

The polynomial corresponding to the codeword is obtained by summing the polynomials $y^{n-m}M(y)$ and $R(y)$: $y^5 + y^3 + y^2 + 1$. Thus, the encoded message becomes $\langle 101101 \rangle$ (instead of the original combination $\langle 1011 \rangle$). In fact, the data vector has been supplemented by the check vector $\langle 01 \rangle$.

Since a generating polynomial can be mapped into a binary number, we will use the decimal equivalent N of this binary number to indicate the generating polynomial. Some error detection properties in CED circuits based on polynomial codes with different values of N were investigated in [19, 20]. Note that the generating polynomial $G(y)$ is chosen based on the condition of maximum error detection. For example, there is no sense in choosing y^0 as a generating polynomial: when dividing any polynomial of arbitrary degree by this polynomial, the residue will always be 0. (Also, zero residue is obtained when dividing by several polynomials even of the highest degree, e.g., y^2 or y^3 .) Such polynomials allow detecting no errors in the code vector. We exclude from further analysis the generating polynomials without the free term y^0 (the numbers N corresponding to such polynomials are even). Therefore, we will consider polynomial codes whose generating polynomials correspond to odd numbers N . Only for such polynomials is it possible to obtain a complete set of check vectors with k bits and, accordingly, easier to provide the full self-checking of encoders and checkers of these codes in CED circuits [20].

2.2. Logical Expressions for the Check Functions of Polynomial Codes

As a rule, the values of check symbols for polynomial codes are obtained using shift registers [18], which is due to the specifics of their application. However, in some applications of polynomial codes, polynomial code encoders representing combinational circuits are required. One example is the design of CED circuits for discrete devices. When constructing an encoder, one needs a formula describing each check symbol of a polynomial code, since this device should be universal and generate a check vector on the incoming data vector in one cycle.

The following trivial method can be used to get formulas for the check symbols of a polynomial code: for the complete set of data vectors with m bits, compute the values of check symbols and form check vectors; then, for each check symbol, obtain logical expressions. Based on such functions, the encoder of the polynomial code under consideration (a combinational circuit) is easily designed.

To get logical expressions for the check functions, it is necessary to determine the values of all check symbols when substituting as arguments the Boolean vectors corresponding to the code data vectors from the complete set 2^m . Here, m denotes the number of arguments. However, this operation becomes rather computationally intensive even for $m \geq 6$. The number of data vectors can be reduced to m by utilizing the following result.

Theorem 1. *To obtain functions describing the check symbols of polynomial codes, it is necessary and sufficient to determine the values of check symbols when substituting the arguments from data vectors with weight $r = 1$.*

The proof of this theorem is given in the Appendix.

Theorem 1 leads to an algorithm for determining logical expressions for the check functions of a polynomial code.

Algorithm 1. Rules to form check functions for polynomial codes:

1. Specify a generating polynomial N and the number m of bits in the data vector.
2. Determine the number of bits in the check vectors, $k = \lfloor \log_2 N \rfloor$.
3. Form the set Q of data vectors with weight $r = 1$ and cardinality $|Q| = m$. Rank the data vectors by the significant bit precedence and compile their list.
4. Let $i = 1$.
5. Consider the i th data vector and determine the values of bits of the check vector by division.
6. Compile formulas for check functions: if the value of the check symbol equals 1, then include the i th data bit in the formula; otherwise, do not.
7. Write the bits in formulas through the XOR sign.
8. Let $i := i + 1$.
9. $i + 1 > m$? If “yes,” proceed to Step 10; otherwise, get back to Step 5.
10. The formulas have been obtained.

Using Algorithm 1, we obtain the formulas for the check functions of the above polynomial code with $N = 13$ and $m = 6$. All the data vectors with weight $r = 1$ are provided in Table 1.

Table 1. The codewords of a polynomial code with the generating polynomial corresponding to $N = 13$ given $m = 3$, for data vectors with weight $r = 1$

m	y_6	y_5	y_4	y_3	y_2	y_1	g_3	g_2	g_1
6	1	0	0	0	0	0	1	0	1
5	0	1	0	0	0	0	1	1	1
4	0	0	1	0	0	0	1	1	0
3	0	0	0	1	0	0	0	1	1
2	0	0	0	0	1	0	1	0	0
1	0	0	0	0	0	1	0	1	0

Based on Table 1, it is easy to get the formula for each function g_j . One should proceed as follows.

Algorithm 2. Rules to form the check functions of polynomial codes by the codeword table for data vectors with weight $r = 1$:

1. Determine m codewords for the data vectors with weight $r = 1$.
2. Let $j = 1$.

3. Consider the j th column corresponding to the j th check function.
4. Compile the formula by including those data symbols f_i for which 1 is written in the j th column.
5. Let $j := j + 1$.
6. $j + 1 > k$? If “yes,” proceed to Step 7; otherwise, get back to Step 3.
7. The formulas have been obtained.

We arrive at the following system of Boolean functions:

$$\begin{cases} g_1 = y_3 \oplus y_5 \oplus y_6 \\ g_2 = y_1 \oplus y_3 \oplus y_4 \oplus y_5 \\ g_3 = y_2 \oplus y_4 \oplus y_5 \oplus y_6. \end{cases}$$

Algorithms 1 and 2 yield formulas for any polynomial code with the linear rate of computational intensity. They can be used to design encoders of polynomial codes as part of checkers in CED circuits.

3. A CLASSIFICATION OF POLYNOMIAL CODES BY THE TYPE OF CHECK FUNCTIONS

3.1. Special Properties of Polynomial Codes

The check functions of polynomial codes are linear. The following peculiarities of linear Boolean functions are well known [17].

Theorem 2. *A linear Boolean function is self-dual only if it has an odd number of essential arguments.*

Theorem 3. *A linear Boolean function is self-quasidual only if it has an even number of essential arguments.*

Let us recall several well-known definitions [7].

Definition 1. A function belongs to the class of self-dual Boolean functions if it takes opposite values when inverting all its arguments:

$$SD = \{f(x_1, x_2, \dots, x_t) | f(\overline{x_1}, \overline{x_2}, \dots, \overline{x_t}) = \overline{f(x_1, x_2, \dots, x_t)}\}.$$

Definition 2. A function belongs to the class of self-quasidual¹ Boolean functions if it takes the same values when inverting all its arguments:

$$SQD = \{f(x_1, x_2, \dots, x_t) | f(\overline{x_1}, \overline{x_2}, \dots, \overline{x_t}) = f(x_1, x_2, \dots, x_t)\}.$$

According to Theorems 2 and 3, polynomial codes with different generating polynomials N and with different number m of data symbols belong to one of the three classes: codes whose encoders are described only by self-dual Boolean functions (class I), codes whose encoders are described only by self-quasidual Boolean functions (class II), and codes whose encoders are described by both self-dual and self-quasidual Boolean functions (class III).

Definition 3. Discrete devices whose outputs are described only by self-dual functions are called self-dual devices (SD devices).

Definition 4. Discrete devices whose outputs are described only by self-quasidual functions are called self-quasidual devices (SQD devices).

Definition 5. Discrete devices whose outputs are described, in one part, by self-dual Boolean functions and, in the other part, by self-quasidual Boolean functions are called self-dual/self-quasidual devices (SD/SQD devices).

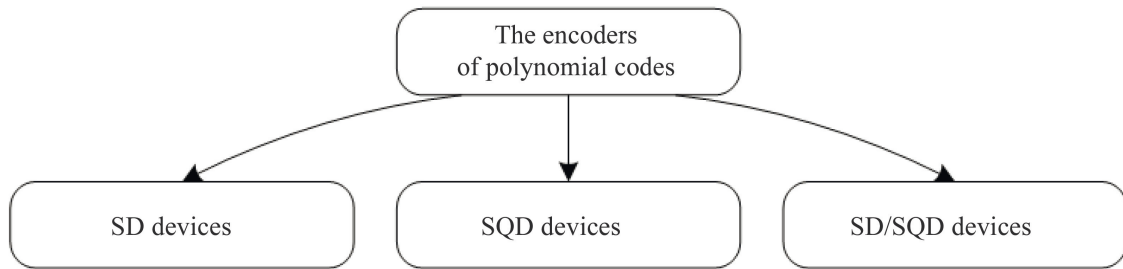


Fig. 2. The encoders of polynomial codes: classification.

Thus, the encoders of polynomial codes may be devices of only three types (Fig. 2).

It is necessary to emphasize the following important peculiarities inherent in the encoders of polynomial codes [17].

Theorem 4. *In the circuit implementation of a linear Boolean function with an odd number of essential arguments, computation control via the self-duality attribute does not check faults of logic gates connected by paths with an even number of inputs.*

Theorem 5. *In the circuit implementation of a linear Boolean function with an even number of essential arguments, computation control via the self-quasiduality attribute does not check faults of logic gates connected by paths with an even number of inputs.*

According to Theorems 4 and 5, it is impossible to design a fully self-checking CED circuit via the attribute of self-duality and/or self-quasiduality of check functions since it is impossible to detect all faults of polynomial encoders when controlling only such attributes. Indeed, the errors caused by faults of XORs in linear circuits are always translated to their outputs, and computation control via the attribute of self-duality or self-quasiduality checks the two-rail property of the spatial signal but not of that coming in parallel to the checker's inputs. Therefore, under the conditions of Theorems 4 and 5, the error is manifested when supplying both sets of argument values to the inputs of the device and is not detected by the checker of a self-dual or self-quasidual signal. Additional control via one more diagnostic attribute is required. The codeword's belonging to the chosen polynomial code may be such an attribute. Indeed, on each set of argument values in the presence of an error, either the correct values of all bits of the check vector will be computed or a distortion will occur at least in one bit. The checker of a polynomial code will detect this event. Thus, polynomial codes can be effectively used in the structure of computation control via several diagnostic attributes, which will be discussed below.

3.2. The Classes of Polynomial Codes by the Encoder Type

Based on the properties of linear functions and the logical expressions for the functions describing the check symbols of polynomial codes (see the simple method above), we have classified polynomial codes by the type of their encoders used in checkers within CED circuits [21]. Table 2 gives a fragment of this classification.

The first column in the table contains the numbers N corresponding to the generating polynomials. Table 2 provides all generating polynomials for constructing polynomial codes with $k = 1, \dots, 6$ check symbols.

In practice, a convenient representation for different classes of polynomial codes is a matrix form that indicates the belonging of an encoder of a particular code with a given generating polynomial to a certain class under a given number of data symbols. Table 3 presents such a matrix for codes

¹ The author [7] used the somewhat inappropriate term of self-antidual functions to denote self-quasidual ones.

with $k \leq 4$ check symbols. Here, the cells at the intersection of particular rows and columns are set off in three different colors: black, gray, and white (meaning that the encoder of the polynomial code belongs to the class of SD devices, SQD devices, and SD/SQD devices, respectively).

Table 2. The detailed classification of polynomial codes by encoder types

N	SD	SQD	SD/SQD
$k = 1$			
3	$m \equiv 1(\text{mod}2)$	$m \equiv 0(\text{mod}2)$	–
$k = 2$			
5	$m \equiv 2(\text{mod}4)$	$m \equiv 0(\text{mod}4)$	$m \equiv 1(\text{mod}2)$
7	$m \equiv 2(\text{mod}3)$	$m \equiv 0(\text{mod}3)$	$m \equiv 1(\text{mod}3)$
$k = 3$			
9	$m \equiv 3(\text{mod}6)$	$m \equiv 0(\text{mod}6)$	$m \equiv \alpha(\text{mod}8), \alpha \in \{0, 1, \dots, 5\} \setminus \{0, 3\}$
11	$m \equiv 2(\text{mod}7)$	$m \equiv 0(\text{mod}7)$	$m \equiv \alpha(\text{mod}7), \alpha \in \{0, 1, \dots, 6\} \setminus \{0, 2\}$
13	$m \equiv 6(\text{mod}7)$	$m \equiv 0(\text{mod}7)$	$m \equiv \alpha(\text{mod}7), \alpha \in \{0, 1, \dots, 6\} \setminus \{0, 6\}$
15	–	$m \equiv 0(\text{mod}4)$	$m \equiv \alpha(\text{mod}7), \alpha \in \{1, 2, 3\}$
$k = 4$			
17	$m \equiv 4(\text{mod}8)$	$m \equiv 0(\text{mod}8)$	$m \equiv \alpha(\text{mod}8), \alpha \in \{0, 1, \dots, 7\} \setminus \{0, 4\}$
19	$m \equiv 3(\text{mod}15)$	$m \equiv 0(\text{mod}15)$	$m \equiv \alpha(\text{mod}15), \alpha \in \{0, 1, \dots, 14\} \setminus \{0, 3\}$
21	–	$m \equiv 0(\text{mod}6)$	$m \equiv \alpha(\text{mod}6), \alpha \in \{1, 2, 3, 4, 5\}$
23	$m \equiv 2(\text{mod}14)$	$m \equiv 0(\text{mod}14)$	$m \equiv \alpha(\text{mod}14), \alpha \in \{0, 1, \dots, 13\} \setminus \{0, 2\}$
25	$m \equiv 8(\text{mod}15)$	$m \equiv 0(\text{mod}15)$	$m \equiv \alpha(\text{mod}15), \alpha \in \{0, 1, \dots, 14\} \setminus \{0, 8\}$
27	$m \equiv 4(\text{mod}12)$	$m \equiv 0(\text{mod}12)$	$m \equiv \alpha(\text{mod}12), \alpha \in \{0, 1, \dots, 11\} \setminus \{0, 4\}$
29	$m \equiv 6(\text{mod}14)$	$m \equiv 0(\text{mod}14)$	$m \equiv \alpha(\text{mod}14), \alpha \in \{0, 1, \dots, 13\} \setminus \{0, 6\}$
31	$m \equiv 4(\text{mod}5)$	$m \equiv 0(\text{mod}5)$	$m \equiv \alpha(\text{mod}5), \alpha \in \{1, 2, 3\}$
$k = 5$			
33	$m \equiv 5(\text{mod}10)$	$m \equiv 0(\text{mod}10)$	$m \equiv \alpha(\text{mod}10), \alpha \in \{0, 1, \dots, 9\} \setminus \{0, 5\}$
35	$m \equiv 2(\text{mod}42)$	$m \equiv \alpha(\text{mod}42),$ $\alpha \in \{0, 20, 31\}$	$m \equiv \alpha(\text{mod}42),$ $\alpha \in \{0, 1, \dots, 41\} \setminus \{0, 2, 20, 31\}$
37	$m \equiv \alpha(\text{mod}62),$ $\alpha \in \{40, 56\}$	$m \equiv \alpha(\text{mod}62),$ $\alpha \in \{0, 38\}$	$m \equiv \alpha(\text{mod}62),$ $\alpha \in \{0, 1, \dots, 61\} \setminus \{0, 38, 40, 56\}$
39	$m \equiv 1(\text{mod}28)$	$m \equiv 0(\text{mod}28)$	$m \equiv \alpha(\text{mod}28), \alpha \in \{0, 1, \dots, 27\} \setminus \{0, 1\}$
41	$m \equiv \alpha(\text{mod}62),$ $\alpha \in \{18, 60\}$	$m \equiv \alpha(\text{mod}62),$ $\alpha \in \{0, 39, 62\}$	$m \equiv \alpha(\text{mod}62),$ $\alpha \in \{0, 1, \dots, 61\} \setminus \{0, 18, 39, 60, 62\}$
43	$m \equiv 11(\text{mod}30)$	$m \equiv 0(\text{mod}30)$	$m \equiv \alpha(\text{mod}30), \alpha \in \{0, 1, \dots, 29\} \setminus \{0, 11\}$
45	–	$m \equiv 0(\text{mod}12)$	$m \equiv \alpha(\text{mod}12), \alpha \in \{1, 2, \dots, 11\}$
47	$m \equiv 2(\text{mod}31)$	$m \equiv 0(\text{mod}31)$	$m \equiv \alpha(\text{mod}31), \alpha \in \{0, 1, \dots, 30\} \setminus \{0, 2\}$
49	–	$m \equiv 0(\text{mod}21)$	$m \equiv \alpha(\text{mod}21), \alpha \in \{1, 2, \dots, 20\}$
51	$m \equiv 5(\text{mod}8)$	$m \equiv 0(\text{mod}8)$	$m \equiv \alpha(\text{mod}8), \alpha \in \{0, 1, \dots, 7\} \setminus \{0, 5\}$
53	$m \equiv 1(\text{mod}30)$	$m \equiv 0(\text{mod}30)$	$m \equiv \alpha(\text{mod}30), \alpha \in \{0, 1, \dots, 29\} \setminus \{0, 1\}$
55	$m \equiv \alpha(\text{mod}62),$ $\alpha \in \{34, 42\}$	$m \equiv \alpha(\text{mod}62),$ $\alpha \in \{0, 10\}$	$m \equiv \alpha(\text{mod}62),$ $\alpha \in \{0, 1, \dots, 62\} \setminus \{0, 10, 34, 42\}$
57	$m \equiv 17(\text{mod}28)$	$m \equiv 0(\text{mod}28)$	$m \equiv \alpha(\text{mod}28), \alpha \in \{0, 1, \dots, 27\} \setminus \{0, 17\}$
59	$m \equiv 18(\text{mod}31)$	$m \equiv 0(\text{mod}31)$	$m \equiv \alpha(\text{mod}31), \alpha \in \{0, 1, \dots, 30\} \setminus \{0, 18\}$
61	$m \equiv 22(\text{mod}31)$	$m \equiv 0(\text{mod}31)$	$m \equiv \alpha(\text{mod}31), \alpha \in \{0, 1, \dots, 30\} \setminus \{0, 22\}$
63	–	$m \equiv 0(\text{mod}6)$	$m \equiv \alpha(\text{mod}6), \alpha \in \{1, 2, \dots, 5\}$

Table 3. The matrix of polynomial codes with $k = 1, \dots, 4$ check symbols

m	N														
	$k = 1$	$k = 2$		$k = 3$				$k = 4$							
	3	5	7	9	11	13	15	17	19	21	23	25	27	29	31
4															
5															
6															
7															
8															
9															
10															
11															
12															
13															
14															
15															
16															
17															
18															
19															
20															
21															
22															
23															
24															
25															
26															
27															
28															
29															
30															

4. USING POLYNOMIAL CODES IN THE DESIGN OF SELF-CHECKING DISCRETE DEVICES WITH COMPUTATION CONTROL VIA SEVERAL DIAGNOSTIC ATTRIBUTES

4.1. The Structural Diagram of a CED Circuit

Consider the peculiarities of using polynomial codes in the design of self-checking discrete devices with computation control via several diagnostic attributes. The approach to implementing CED circuits via several diagnostic attributes simultaneously was used earlier in [22] when organizing computation control of devices based on the principle of Boolean signal correction and the properties of balanced codes of the form “ r of $2r$,” where r specifies the weight of a codeword. It was also investigated in [16] in the context of computation control via several diagnostic attributes using the well-known Hamming codes. Figure 3 shows the structural diagram of computation control via several diagnostic attributes.

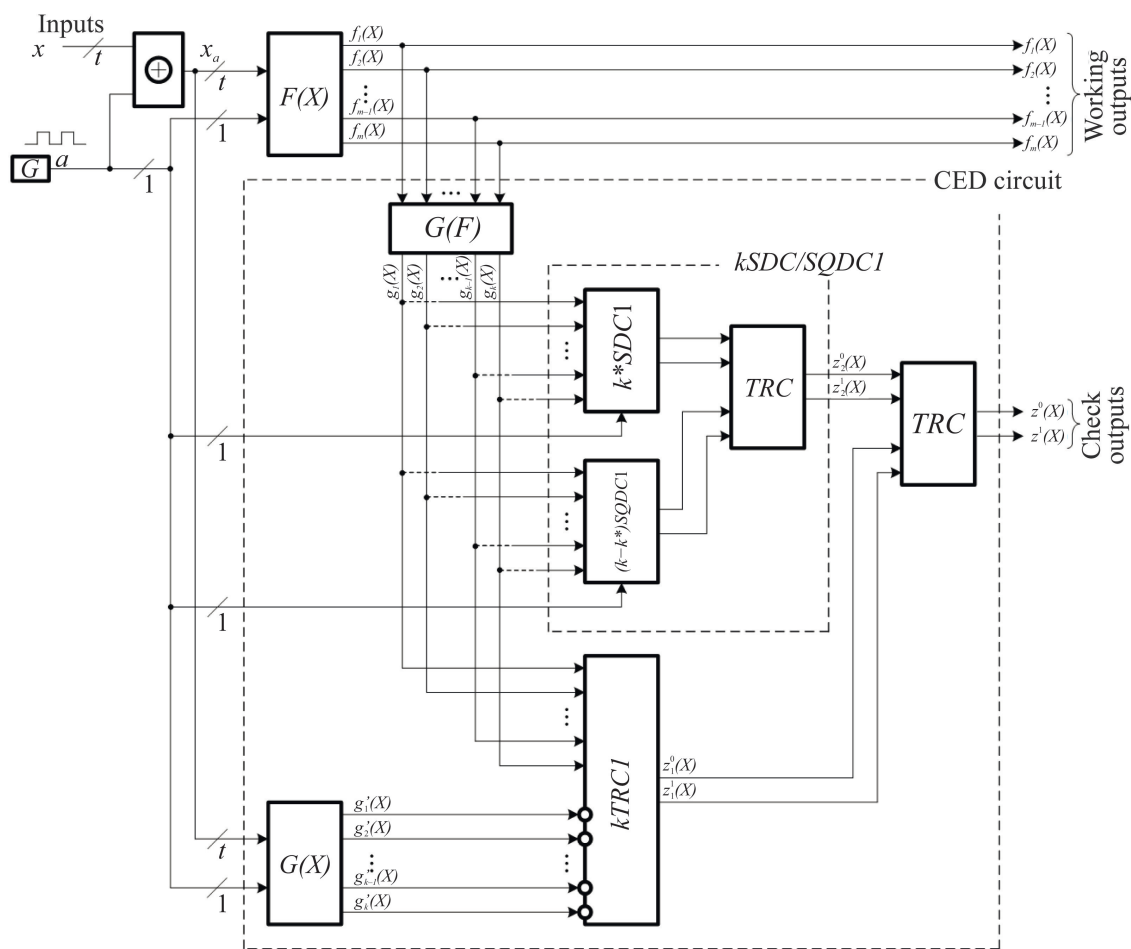


Fig. 3. The structural diagram of a CED circuit for self-dual discrete devices via several diagnostic attributes.

In this structure, the initial discrete device (the object under diagnosis) is a block $F(X)$ intended to compute the values of Boolean functions $f_1(X), f_2(X), \dots, f_{m-1}(X), f_m(X)$ when supplying the sets of argument values $\langle X \rangle = \langle x_t x_{t-1} \dots x_2 x_1 \rangle$ to the inputs. This object is an SD device. A special CED circuit with several diagnostic attributes is used to control computations at the outputs of the block $F(X)$. The CED circuit is implemented on the basis of polynomial codes. The object's outputs are identified with a data vector with m bits. Given the values of functions $f_1(X), f_2(X), \dots, f_{m-1}(X), f_m(X)$, the block $G(F)$ forms the values of check symbols $g_1(X), g_2(X), \dots, g_{k-1}(X), g_k(X)$ for the codewords of a selected polynomial code, thus generating a particular check vector. On the other hand, the block $G(X)$ generates the values of check symbols $g'_1(X), g'_2(X), \dots, g'_{k-1}(X), g'_k(X)$ for the codewords of the same polynomial code when supplying the sets of argument values $\langle X \rangle = \langle x_t x_{t-1} \dots x_2 x_1 \rangle$ to the inputs. This solution provides a computation control subcircuit based on the belonging of the codewords formed in the CED circuit to the selected polynomial code. More precisely, the same-name outputs of the blocks $G(F)$ and $G(X)$ are connected to the inputs of the two-rail comparator $kTRC1$, which compares k pairs of signals at the inputs and forms a single pair of check signals $z_1^0(X)$ and $z_1^1(X)$. This block is implemented in the two-rail logic and represents a two-rail compression circuit based on $k - 1$ elementary *two-rail checkers* (TRC) [23]. Therefore, the signals from one of the blocks $G(F)$ or $G(X)$ are pre-inverted. (In Fig. 3, these are the signals from $G(X)$ are inverted.) In general, when designing the block $G(X)$, one can immediately receive the inverted signals $g_1(X), g_2(X), \dots, g_{k-1}(X), g_k(X)$; then inverters at the comparator's inputs will not be needed. Computation errors at the outputs of

the blocks $G(F)$ and $G(X)$, as well as in the comparator's structure, violate the two-rail property of the signal at the outputs $z_1^0(X)$ and $z_1^1(X)$, which indirectly indicates the presence of faults in the structure of the entire system. This structure of computation control (with check symbols supplementing data symbols) has been studied quite well [24].

Since the outputs of the block $G(F)$ are described by self-dual and/or self-quasidual functions, the self-duality/self-quasiduality control subcircuit is used for the signals $g_1(X), g_{k-1}(X), g_k(X)$ to control computations. For self-duality, the block k^*SDC1 is installed to control k^* self-dual signals and generate one pair of two-rail outputs. Similarly, for self-quasiduality, the block $(k - k^*)SQDC1$ is installed to control $k - k^*$ self-quasidual signals and generate one pair of two-rail outputs. These blocks are built based on compression circuits for self-dual and self-quasidual signals [11] and checkers of self-dual and self-quasidual signals [25]. The outputs of both blocks k^*SDC1 and $(k - k^*)SQDC1$ are connected to the inputs of the elementary compression block TRC of two-rail signals. Its outputs are the check outputs $z_2^0(X)$ and $z_2^1(X)$ of the computation control subcircuit via self-duality/self-quasiduality. The outputs of the computation control subcircuits via different diagnostic attributes are connected to the inputs of one block TRC . Its outputs $z^0(X)$ and $z^1(X)$ are the check outputs of the CED circuit.

The operation principles and tuning of control devices for self-dual and self-quasidual signals were described in [25]. The entire structure involves temporal redundancy and the data inversion mode [6]. In this case, signals are represented as pulse sequences: zero is encoded by the sequence 0101...01 and one by 1010...10. As a result, the operation of devices is implemented by supplying sets of input argument values in pairs: the first is an operating set, and the second is a check set. The sets of input arguments in a pair are orthogonal in all variables (inverse in all variables). This peculiarity of the computation control approach requires a rectangular pulse generator G , forming a signal a with a pulse ratio of 2. A sequence of signals corresponding to the input variables is obtained using a cascade of two-input XORs: their first inputs receive signals from particular inputs and the second ones the signal a . The same signal can be supplied to the inputs of the blocks $F(X)$ and $G(X)$ when they are designed from non-self-dual structures using the Shannon decomposition with a single variable. The signal a is also required for the operation of self-duality/self-quasiduality checkers.

According to [16], the computation control structure with several diagnostic attributes has better checkability in terms of error observability at the outputs of objects under diagnosis, owing to the operation mode and the possibility of versatile computation control.

Note the following important peculiarity of the computation control structure with several diagnostic attributes.

Theorem 6. *In the CED circuit with computation control via several diagnostic attributes, any single errors are detected at the outputs of the computation control subcircuit by the attributes of the self-duality and/or self-quasiduality of the signals formed.*

The proof of this result is given in the Appendix.

In practice, Theorem 6 simplifies the design of CED circuits with detection of any errors at the outputs of the object under diagnosis by excluding from consideration the object's faults that cause single errors at their outputs.

4.2. An Algorithm for Selecting the Generating Polynomial

The general algorithm for designing a self-checking CED circuit with several diagnostic attributes using polynomial codes contains the following steps.

Algorithm 3. Rules to design a CED circuit with several diagnostic attributes using polynomial codes:

1. Consider the structure of a combinational object under diagnosis with m outputs and t inputs.

2. From the complete set Ω of faults within a given model (e.g., single stuck-at faults at the outputs of logic gates), select a subset $\Omega' \subset \Omega$ of only those occurring at the outputs of logic gates connected by paths with two or more outputs of the object under diagnosis.

3. If $\Omega' = \emptyset$, organize computation control using a polynomial code with a generating polynomial $N = 3$ and design a CED circuit with several diagnostic attributes. If $\Omega' \neq \emptyset$, consider the possibilities of error detection using polynomial codes with the generating polynomials $N = 3 + 2i$, $i \in \mathbb{N}$. In this case, take only polynomials for which $k = \lfloor \log_2 N \rfloor < m$. In other words, the maximum value of this number is $N = 2^{m-1} - 1$, and the list contains the numbers $N = 3 + 2i$, $i = \{1, \dots, 2^{m-2} - 2\}$.

4. Simulate the operation of the self-dual device by supplying all pairs $(A_i^j, B_{2^n-1-i}^j)$ to its inputs, where A_i^j and $B_{2^n-1-i}^j$ are the values of the j th function formed at the j th output of the object under diagnosis under an input combination corresponding to the decimal equivalent of i and $2^n - 1 - i$, sets of argument values, and the impact of single stuck-at faults from the set Ω' . Fix the errors caused by them at the outputs.

5. Compile the list of generating polynomials $N = 3 + 2i$, $i = \{1, 2, \dots, 2^{m-2} - 2\}$, and arrange them in ascending order of N . With such an ordering, one first considers the generating polynomials yielding codes with the smallest number of check symbols.

6. Take the first member of this list and remove it from there.

7. On the complete family of the sets of argument values, determine the values of the functions defining the check symbols for the codewords of the polynomial code with the chosen generating polynomial N .

8. Check error detection by computation control via the self-duality and/or self-quasiduality attribute of the check functions obtained. From the complete set Ψ of errors occurring at the circuit outputs, eliminate single errors and all errors with multiplicities $d \geq 2$, detected by computation control via the self-duality and/or self-quasiduality attribute. Form the set $\Psi' \subset \Psi$ of errors undetectable via this attribute.

9. If $\Psi' = \emptyset$, organize computation control using a polynomial code with the generating polynomial corresponding to the number N . If $\Psi' \neq \emptyset$, check the detection of errors from the set Ψ' using the polynomial code selected.

10. If all errors are detected, design the CED circuit based on the polynomial code selected; otherwise, take the next generating polynomial from the list (if non-empty) and return to Step 4 of the algorithm. In the case of the empty list, it is impossible to design a fully self-checking structure with computation control via several diagnostic attributes.

Consider an example of computation control using this algorithm for a self-dual discrete device described by the following system of Boolean functions:

$$\begin{aligned}
 f_1 &= \underbrace{x_1 x_2}_1 \vee \underbrace{x_1 \bar{x}_4}_2 \vee \underbrace{x_2 \bar{x}_4}_3, \\
 f_2 &= \underbrace{x_3 x_4}_4 \vee \underbrace{x_1 x_4}_5 \vee \underbrace{x_1 x_3}_6, \\
 f_3 &= \underbrace{x_1 \bar{x}_2 \bar{x}_3}_7 \vee \underbrace{\bar{x}_2 x_3 x_4}_8 \vee \underbrace{x_2 \bar{x}_3 x_4}_9 \vee \underbrace{x_1 x_2 x_3}_{10}, \\
 f_4 &= \underbrace{\bar{x}_2 x_3 x_4}_8 \vee \underbrace{x_2 \bar{x}_3 x_4}_9 \vee \underbrace{x_2 x_3 \bar{x}_4}_{11} \vee \underbrace{x_1 x_3 x_4}_{12} \vee \underbrace{x_1 \bar{x}_2 \bar{x}_3 \bar{x}_4}_{13}, \\
 f_5 &= \underbrace{x_1 \bar{x}_4}_2 \vee \underbrace{x_1 \bar{x}_3}_{14} \vee \underbrace{x_1 \bar{x}_2}_{15} \vee \underbrace{\bar{x}_2 \bar{x}_3 \bar{x}_4}_{16}, \\
 f_6 &= \underbrace{x_1 \bar{x}_3 x_4}_{17} \vee \underbrace{x_1 x_3 \bar{x}_4}_{18} \vee \underbrace{x_1 \bar{x}_2 x_3}_{19} \vee \underbrace{\bar{x}_1 x_3 x_4}_{20} \vee \underbrace{x_1 \bar{x}_2 \bar{x}_3 \bar{x}_4}_{21}.
 \end{aligned}$$

At Step 1 of the algorithm, we take a particular two-level implementation of the device by the above formulas, with the inversions of arguments transferred to the inputs of the first-cascade gates. Let all non-repeating conjunctions in the expressions be numbered. The structural diagram of this implementation is omitted here due to its bulkiness. It contains 21 multi-input AND gates in the first cascade and 6 multi-input OR gates in the second one.

At Steps 2 and 3, we determine the subset of single stuck-at faults of logic gates causing multiple errors at the device outputs and error detection possibilities using a polynomial code with the generating polynomial $N = 3$. Multiple errors at the device outputs can be caused only by the faults of the gates realizing conjunctions with numbers 2, 8, and 9. (They correspond to possible simultaneous distortions in the values of the functions f_1 and f_5 , the functions f_3 and f_4 , and the functions f_3 and f_4 , respectively.) The polynomial code with the generating polynomial $N = 3$ is not suitable for computation control in the case considered.

At Step 4, we simulate the operation of the device under single stuck-at faults and memorize the results in tabular form.

We compile the list of generating polynomials $N = 3 + 2i$, $i = \{1, \dots, 14\}$.

Next, we choose the polynomial corresponding to the number $N = 5$, removing it from the list. We check the error detection possibilities at the outputs of the device.

According to the simulation results, the polynomial code with the chosen generating polynomial allows detecting all multiple errors caused by the faults of the gates realizing conjunctions with numbers 2, 8, and 9, except for five double errors caused by a stuck-at-1 fault of the gate realizing conjunction with number 2. (These five faults are detected neither by self-duality control of the functions describing the check symbols of the codewords (see the matrix in Table 3) nor by the control by the belonging of the check vector to this polynomial code.) The cardinality of the set is $|\Psi'| \neq 5$. Hence, it is necessary to choose the next polynomial from the updated list. This polynomial corresponds to the number $N = 7$.

We perform the same actions as before. Here, judging by the matrix in Table 3, computations are controlled via the following attributes: the belonging of the functions describing the check symbols of the codewords to the class of self-quasidual ones and the belonging of the check vector to the codewords of the polynomial code selected. According to the simulation results, all multiple errors are detected by computation control via only one of the diagnostic attributes. Next, we design the CED circuit with two diagnostic attributes using the polynomial code selected.

Note that computation control via the attributes of self-duality and self-quasiduality insignificantly complicates CED circuits compared to CED circuits with computation control via polynomial codes: the compression schemes of self-dual/self-quasidual signals, as well as the checkers of these signals, have a rather simple structure [11, 25]. Therefore, it is possible to design less redundant structures than, e.g., in the case of duplication, while obtaining improved checkability in terms of error observability at the outputs of CED circuits.

An advantage of Algorithm 3 is that, with an appropriate choice of the generating polynomial from small $N = 3$ to $N = 2^{m-2} - 2$, it is possible to achieve the full detection of errors using codes with small k , which also affects the performance of CED circuit implementation. A drawback is the necessity of fault modeling and error detection at the outputs of objects under diagnosis, which limits the applicability of this method for a sufficiently large number of inputs. The applicability of the method for CED circuit design in some discrete devices also depends on the complexity of Boolean functions computed at their outputs [26]. The complexity of the algorithm for constructing a self-checking device can be asymptotically estimated as $2^{O(t)}$, i.e., the problem is solved in exponential time with a linear index. Nevertheless, it is always possible to construct CED circuits using decomposition methods. Besides, considering the structures of objects under diagnosis is

an effective CED circuit design method when standard modular redundancy-based ones are not applied.

Note also that polynomial codes detect a large number of errors under high degrees of the polynomials $M(y)$ and $G(y)$, which is widely used in computation control and data transmission. However, in the above application of polynomial codes, the degree of the polynomial $M(y)$ is determined by the number of outputs in the object under diagnosis, whereas the degrees of the generating polynomials are selected considering the detection of errors caused by faults in the structures of such objects. As a matter of fact, the earlier the number N is found by Algorithm 3, the smaller the degree of the residue $R(y)$ will be, and the less structural redundancy the final self-checking discrete device will have.

5. CONCLUSIONS

Compared to the widely known approaches, the design method proposed above endows the resulting fully self-checking discrete devices with improved checkability indicators. Moreover, the method allows detecting a subset of errors in computations under fixed input actions. This is due to the binary-sequence representation of signals in discrete devices.

The CED circuit design algorithm developed above based on the properties of polynomial codes involves a sequential search of generating polynomials from the smallest, associated with the number $N = 3$, to the corresponding number $N = 2^{m-2} - 2$. Under $m < 6$, the number of generating polynomials satisfying this condition is small, reducing the possibilities for constructing a fully self-checking device. Hence, the approach under consideration should be most effective for devices with sufficiently many outputs: $m \geq 6$. As N grows to the limit, the number of check symbols in the codewords changes from $k = 1$ to $k = m - 1$, also increasing the cardinality of the set of errors detected by polynomial codes. Therefore, the algorithm will be effective when constructing self-checking devices for almost any structures of objects under diagnosis with $m \geq 6$. Nevertheless, for each particular discrete device, it is necessary to compare the indicators of their self-checking implementation with those of self-checking implementations obtained by other methods, e.g., duplication.

Possible directions of further research include setting various experiments with test discrete devices and estimating the efficiency indicators of their self-checking implementations, developing other algorithms for choosing generating polynomials considering the structural peculiarities of objects under diagnosis and the peculiarities of other linear codes in CED circuit design via several diagnostic attributes, and studying the peculiarities of constructing such devices on different (particularly programmable) element bases.

As we believe, the use of computation control via several diagnostic attributes in the design of self-checking discrete control devices is a promising and underinvestigated approach to the realization of components of highly reliable control systems.

APPENDIX

Proof of Theorem 1.

Necessity. Each check function is described by the general expression $g_j = f_{i_1} \oplus f_{i_2} \oplus \dots \oplus f_{i_q}$, $j \in \{1, k\}$, $i_1, i_2, \dots, i_q \in \{1, \dots, m\}$. To obtain detailed expressions for each check function, it is necessary to establish significant data symbols. Consider all data vectors with weight $r = 1$ to determine sequentially the values of the check symbols by substituting a single significant data symbol. If the function value is $g_j = 0$, the corresponding data symbol will not affect the particular check function; otherwise ($g_j = 1$), it will be included in the logical expression describing the check function.

Sufficiency. When substituting the data vector with weight $r = 1$, the value of the function g_j will be determined only by the value of the significant data symbol; hence, it will be included in the descriptive formula for the corresponding check function. Consider all data vectors with weight $r = 1$ to determine all arguments summed in each function g_j . This yields the formulas describing the check symbols of polynomial codes.

Proof of Theorem 6.

All functions of the encoder of a polynomial code are linear: $g_j = f_{i_1} \oplus f_{i_2} \oplus \dots \oplus f_{i_q}$, $j \in \{1, k\}$, $i_1, i_2, \dots, i_q \in \{1, \dots, m\}$. Consider one of them and suppose the occurrence of a single error at the output $f_q \in \{1, \dots, m\}$. Then the value of the function f_p is distorted, becoming equal to $f_p = 0$ or $f_p = 1$. The original expression for the check function is $g_j = f_{i_1} \oplus f_{i_2} \oplus \dots \oplus f_p \oplus \dots \oplus f_{i_q}$. Due to the distortion, it takes the form $g_j = f_{i_1} \oplus f_{i_2} \oplus \dots \oplus (f_p = 0) \oplus \dots \oplus f_{i_q} = f_{i_1} \oplus f_{i_2} \oplus \dots \oplus f_{i_q}$ or $g_j = f_{i_1} \oplus f_{i_2} \oplus \dots \oplus (f_p = 1) \oplus \dots \oplus f_{i_q} = \overline{f_{i_1} \oplus f_{i_2} \oplus \dots \oplus f_{i_q}}$. In the first case, we have a parity change in the number of arguments of the original function and its dependence on all variables except f_p . In the second case, the parity of the number of arguments changes as well but, besides, the function of mod 2 addition is inverted, which preserves all the attributes of belonging to the classes of self-dual and/or self-quasidual functions. In other words, the function has turned from self-quasidual/self-dual into self-dual/self-quasidual. Due to the new type, the function will be detected in the computation control subcircuit with the attribute selected: it will change under a single error.

REFERENCES

1. Bennetts, R.G., *Design of Testable Logic Circuits*, Addison-Wesley, 1984.
2. Sogomonyan, E.S. and Slabakov, E.V., *Samoproveryaemye ustroystva i otkazoustoichivyye sistemy* (Self-Checking Devices and Fault-Tolerant Systems), Moscow: Radio i Svyaz', 1989.
3. Drozd, A., Kharchenko, V., Antoshchuk, S., et. al., Checkability of the Digital Components in Safety-Critical Systems: Problems and Solutions, *Proceedings of the 9th IEEE East-West Design and Test Symposium (EWDTS'2011)*, Sevastopol, Ukraine, 2011, pp. 411–416. <https://doi.org/10.1109/EWDTS.2011.6116606>
4. Drozd, A.V., Kharchenko, V.S., Antoshchuk, S.G., et al., *Rabochee diagnostirovanie bezopasnykh informatsionno-upravlyayushchikh sistem* (Operational Diagnosis of Safe Information and Control Systems), Khar'kov: Zhukovskii National Aerospace University, 2012.
5. Sapozhnikov, V.V. and Sapozhnikov, V.I., *Samoproveryaemye diskretnyye ustroystva* (Self-Checking Discrete Devices), St. Petersburg: Energoatomizdat, 1992.
6. Aksenova, G.P., Restoration in Duplicated Units by the Method of Data Inversion, *Avtomat. Telemekh.*, 1987, no. 10, pp. 144–153.
7. Shalyto, A.A., Modules Universal in the Class of Self-dual Functions and in Close Classes, *Izv. Akad. Nauk. Teor. Sist. Upravlen.*, 2001, no. 5, pp. 110–120.
8. Saposhnikov, V.I., Dmitriev, A., Goessel, M., and Saposhnikov, V.V., Self-Dual Parity Checking—a New Method for On-line Testing, *Proceedings of the 14th IEEE VLSI Test Symposium*, USA, Princeton, 1996, pp. 162–168. <https://doi.org/10.1109/VTEST.1996.510852>
9. Gessel', M., Moshanin, V.I., Sapozhnikov, V.V., and Sapozhnikov, V.I., Fault Detection in Self-Test Combination Circuits Using the Properties of Self-Dual Functions, *Avtomat. Telemekh.*, 1997, no. 12, pp. 193–200.
10. Gessel', M., Dmitriev, A.V., Sapozhnikov, V.V., and Sapozhnikov, V.I., A Functional Fault-Detection Self-test for Combinational Circuits, *Autom. Remote Control*, 1999, vol. 60, no. 11, pp. 1653–1663.
11. Dmitriev, A., Saposhnikov, V., Saposhnikov, V., and Goessel, M., New Self-Dual Circuits for Error Detection and Testing, *VLSI Design*, 2000, vol. 11, no. 1, pp. 1–21. <https://doi.org/10.1155/2000/84720>

12. Sapozhnikov, V.V., Sapozhnikov, V.I., and Gessel', M., *Samodvoistvennyye diskretnye ustroystva* (Self-Dual Discrete Devices), St. Petersburg: Energoatomizdat, 2001.
13. Sapozhnikov, V.V., Sapozhnikov, V.I., and Valiev, R.Sh., *Sintez samodvoistvennykh diskretnykh sistem* (Design of Self-Dual Discrete Systems), St. Petersburg: Elmor, 2006.
14. Goessel, M., Ocheretny, V., Sogomonyan, E., and Marienfeld, D., *New Methods of Concurrent Checking*, 1st ed., Dordrecht: Springer, 2008.
15. Yablonskii, S.V., *Vvedenie v diskretnuyu matematiku* (Introduction to Discrete Mathematics), Sadovnichii, V.A., Ed., Moscow: Vysshaya Shkola, 2003.
16. Efanov, D.V. and Pogodina, T.S., Properties Investigation of Self-Dual Combinational Devices with Calculation Control Based on Hamming Codes, *Informatics and Automation*, 2023, vol. 22, no. 2, pp. 349–392. <https://doi.org/10.15622/ia.22.2.5>
17. Efanov, D.V., The Self-Checking Structures Implementation Features Based on the Inverting Data and Linear Block Code Method, *Tomsk State University Journal of Control and Computer Science*, 2023, no. 65, pp. 126–138. <https://doi.org/10.17223/19988605/65/13>
18. Sellers, F.F., Hsiao, M.-Y., and Bearnson, L.W., *Error Detecting Logic for Digital Computers*, New York: McGraw-Hill, 1968.
19. Efanov, D., Plotnikov, D., Sapozhnikov, V., Sapozhnikov, V.I., and Abdullaev, R., Experimental Studies of Polynomial Codes in Concurrent Error Detection Systems of Combinational Logical Circuits, *Proceedings of the 16th IEEE East-West Design and Test Symposium (EWDTS'2018)*, Kazan, Russia, September 14–17, 2018, pp. 184–190. <https://doi.org/10.1109/EWDTS.2018.8524684>
20. Abdullaev, R. and Efanov, D., Polynomial Codes Properties Application in Concurrent Error-Detection Systems of Combinational Logic Devices, *Proceedings of the 19th IEEE East-West Design and Test Symposium (EWDTS'2021)*, Batumi, Georgia, September 10–13, 2021, pp. 40–46. <https://doi.org/10.1109/EWDTS52692.2021.9580992>
21. Efanov, D.V. and Pivovarov, D.V., Organization of Control of Calculations at the Outputs of Self-Dual Digital Devices Using Cyclic Redundant Codes, *Trudy 14-go Vserossiyskogo soveshchaniya po problemam upravleniya* (Proceedings of the 14th All-Russian Meeting on Control Problems), Novikov, D.A., Ed., Moscow, June 17–20, 2024, pp. 2395–2399.
22. Efanov, D.V. and Pivovarov, D.V., The Hybrid Structure of a Self-Dual Built-In Control Circuit for Combinational Devices with Pre-Compression of Signals and Checking of Calculations by Two Diagnostic Parameters, *Proceedings of the 19th IEEE East-West Design and Test Symposium (EWDTS'2021)*, Batumi, Georgia, September 10–13, 2021, pp. 200–206. <https://doi.org/10.1109/EWDTS52692.2021.9581019>
23. Lala, P.K., *Self-Checking and Fault-Tolerant Digital Design*, San Francisco: Morgan Kaufmann, 2001.
24. Sahana, A.R., Chiraag, V., Suresh, G., Thejaswini, P., and Nandi, S., Application of Error Detection and Correction Techniques to Self-Checking VLSI Systems: An Overview, *Proceedings of 2023 IEEE Guwahati Subsection Conference (GCON)*, Guwahati, 2023. <https://doi.org/10.1109/GCON58516.2023.10183449>
25. Efanov, D.V. and Pivovarov, D.V., Checkers of Self-Dual and “Close in Meaning” Signals, *Journal of Instrument Engineering*, 2024, vol. 67, no. 1, pp. 5–19. <https://doi.org/10.17586/0021-3454-2024-67-1-5-19>
26. Korshunov, A.D., Computational Complexity of Boolean Functions, *Russian Math. Surveys*, 2012, vol. 67, no. 1, pp. 93–165. <https://doi.org/10.1070/RM2012v067n01ABEH004777>

This paper was recommended for publication by L.Yu. Filimonyuk, a member of the Editorial Board

Regression Models for the AI Gaming Chatbot for Learning Programming Based on Wordle-Type Puzzles

A. N. Varnavsky

National Research University “Higher School of Economics”, Moscow, Russia

e-mail: avarnavsky@hse.ru

Received November 29, 2024

Revised January 9, 2025

Accepted January 14, 2025

Abstract—Programming is one of the most important skills of the 21st century. However, for many students, learning programming is a rather complex process. In such cases, it is important to maintain the interest and involvement of students in the learning process. It is believed that digital games can solve this problem. One of the types of games that are well suited for the field of computer science are puzzles, which are aimed, among other things, at developing cognitive abilities. The purpose of the work is to develop models, an algorithm for the operation and structure of a game chatbot with artificial intelligence for learning programming using Wordle-like puzzles. Game Wordle was chosen due to its worldwide popularity and adapted as a gaming chatbot for use in learning programming. Artificial intelligence in the chatbot is necessary to control the appropriateness and appropriate time of its use, as well as adaptively forming the difficulty level of puzzle. Based on the data collected as a result of using a non-intelligent gaming chatbot, regression models of the influence of student indicators on the level of interest and difficulty of puzzles offered by the gaming chatbot were built. The developed models formed the basis of the algorithm of operation and structure of the gaming chatbot with artificial intelligence. When using an intelligent gaming chatbot, it is possible to further train the models and adjust the previously obtained coefficient values.

Keywords: gaming chatbot, learning to program, artificial intelligence

DOI: 10.31857/S0005117925050046

1. INTRODUCTION

For many students, learning to program is quite a difficult process. This is due to the fact that solving programming problems requires not only knowledge of programming languages, but also appropriate levels of cognitive abilities (logical, critical, computational thinking). Low levels of such abilities often create difficulties in learning and lead to a decrease in student engagement and motivation.

The use of educational computer games is one of the methods for maintaining student interest in learning in various disciplines and areas [1], including programming [2]. Cognitive abilities and functions (working memory, processing speed, executive functions, etc.) can be significantly improved by using games [3], and the specific effect depends on the types of games used [4]. Thus, logical games and puzzle games are well suited for teaching computer science [5].

Traditionally, Parsons programming puzzles [6], which are randomly jumbled code fragments that the learner must arrange in the correct order to obtain a working program, are well suited for teaching programming at the initial stage. They are effective for mastering basic syntactic

structures and programming logic, but their capabilities are limited and they are not sufficiently focused on developing more complex cognitive skills.

At the same time, word puzzles, which are widely used in teaching foreign languages and medicine, are practically not used in teaching programming, despite their enormous potential for developing cognitive abilities [7]: vocabulary, working memory, cognitive processing, strategies for retrieving information from working memory, reasoning and thinking abilities [8].

Effective programming directly depends on the knowledge of the syntax of the programming language, vocabulary related to specific terminology, syntactic rules [9]. Even a minor deviation from the syntax can lead to compilation or execution errors, requiring significant time spent on debugging. Moreover, unclear and imprecise use of terminology in the code significantly complicates its understanding both by the author and other developers, increasing the cost and time spent on supporting and upgrading the software [10]. This is especially critical in large projects where the code is written and supported by many people, and unclear vocabulary can lead to code incompatibility, loss of time, and even to the occurrence of errors leading to a failure of the entire system. Given the potential of word puzzles, it is necessary to develop the direction of their use in teaching programming and computer science. Moreover, it is necessary to search for options for implementing word puzzles that would be interesting and exciting for students of different levels, while developing their skills.

One approach to creating educational games is to adapt well-known computer games (for example, the adaptation of Super Mario in [11]). In recent years, a fairly popular word game is Wordle (<https://www.nytimes.com/games/wordle/index.html>). Wordle is a word puzzle game in which players must guess a word in several attempts. Scientific works appear that study strategies, the process of solving puzzles, the complexity of words, and the behavior of players in this game [12–14]. Therefore, the principle of the Wordle game can be adapted to create a puzzle game for recognizing program code for use in the process of teaching programming. Given the growing popularity of chatbots in various fields, including education [15], as well as the text format of the Wordle game, the adaptation can be implemented in the form of a chatbot.

The gaming chatbot with puzzle tasks can be embedded with models that will make it intelligent, for example, in terms of generating tasks with an appropriate level of difficulty and interest. This will increase the efficiency of its use, in particular due to the fact that interest affects engagement and motivation, and the difficulty of tasks can affect interest. Similar gaming chatbots were not found in the literature.

The aim of the work is to develop models, an algorithm for the operation and structure of the gaming chatbot with artificial intelligence for learning programming using like Wordle-like puzzle. To achieve this aim, it is necessary to collect data on the use of the first version of the non-intelligent gaming chatbot, on the basis of which regression models can be built. These models will form the basis for the algorithm for the operation and structure of the new version of the gaming chatbot.

2. RELATED WORK

2.1. Use of Educational Games

Although some studies have not found any positive effects from using educational games, most scientists consider games to be a very effective and inexpensive teaching method that increases interest in the training course, student engagement, academic performance, and knowledge assessment after using games [17, 18]. Games can influence various aspects, including affective and motivational elements, behavioral changes, knowledge acquisition, and the development of cognitive and social skills [19].

There are computer games for teaching computer science and programming, and their number has been steadily increasing in recent years [5, 20–22]. There are games for different levels of education, although games for teaching programming to beginners and schoolchildren predominate. Among the existing games, puzzles and logic games are a separate area, since they are well suited for teaching computer science and allow developing thinking skills and logical thinking [23]. This is manifested in the improvement of such indicators as the speed of information processing, the volume of working memory, cognitive control, argumentation and problem solving.

Word puzzles are a fairly popular type of game used in educational practice, especially in teaching foreign languages and medicine. Their integration with courses, in combination with other learning tools, helps to reduce the cognitive load on students, stimulate motivation, improve academic performance [16], effectively develop general knowledge [25] and cognitive abilities [7, 24]. However, word puzzles are not very common in programming education.

In recent years, one of the most popular word puzzles is the Wordle game. This game is intended for entertainment, but it is possible to note the appearance of a fairly large number of scientific works studying this game, the solution process, options for enumeration and search for a solution, etc. [12–14]. Therefore, it is possible to consider the direction of adapting this game, as applied to the educational process, in particular for use in teaching programming.

2.2. AI Chatbots

Puzzle games can be implemented in a variety of forms, including a chatbot. This format may work well for some text/word puzzles. Basic chatbots use a rule-based approach [26]. In contrast, AI chatbots use machine learning models.

AI chatbots are one type of conversational AI application. According to forecasts, the global conversational AI market size is expected to grow at a CAGR of 21.8%, reaching \$18.4 billion by 2026 (<https://www.marketsandmarkets.com>). Conversational AI has the potential to become a mainstream technology in almost all market verticals (<https://www.emarketer.com/content/tech-catches-up-conversational-ai-ambitions>).

Overall, the benefits of AI chatbots include providing a natural language, user-friendly interface that can improve the user experience [27]. Using AI chatbots can reduce stress and help users avoid information overload [28], providing users with a unique and more personalized experience. Such chatbots can influence users' emotions in a way that conventional enterprise systems cannot [29].

AI conversational systems can exhibit personality and have anthropomorphic characteristics [30], better match user and organizational contexts, and are able to continually evolve by learning from the content of requests and how users interact with them. Therefore, the continued use of AI systems is important for their continued improvement, since they are usually based on machine learning technologies. Since the continuous improvement of an AI chatbot depends on its use, it is important to consider user behavior and engage usage patterns [31].

Currently, the development of AI chatbots is focused on the development of algorithm training methods and approaches related to customization, anthropomorphism [32], personalization [33], improving algorithms to obtain more accurate results [34] and expanding functionality, including, for example, sentiment analysis [35]. There are proposals to create a new human-machine intelligence [36].

Examples include AI chatbots that are used in the workplace [37], in business processes [38], in organizing effective interactions with users and clients [39], in training [40]. Such chatbots are more effective due to the automatic detection of user needs [41, 42] and the formation of user-oriented behavior [43]. For example, in education, AI-powered chatbots can increase users' interest in learning mathematics [44] and reading [45].

2.3. Formulation of the Task

In general, there are very few educational gaming chatbots described in the literature, and even fewer educational gaming chatbots with artificial intelligence. [46] can be used as an example.

Each described chatbot has its own logical/scenario or AI/ML behavior model. Given the originality of the task, such models are not suitable for building an intelligent gaming chatbot with puzzles like Wordle and need to be developed. To do this, we will use the non-intelligent gaming chatbot and conduct an experiment with it. The data collected as a result of the experiment will be used to create machine learning models to control the behavior of the intelligent gaming chatbot. The models will control the level of difficulty of tasks and assess the feasibility of using a chatbot at a given stage of learning a programming language by predicting interest in the task. The models can be further trained based on new data obtained when using an intelligent gaming chatbot. Based on the obtained models, we will create an algorithm for the operation and structure of the gaming chatbot with artificial intelligence for teaching programming using Wordle-like puzzles.

3. GAMING CHATBOT WITH PUZZLE FOR LEARNING PROGRAMMING

The non-intelligent gaming chatbot “Identify the Python command” for the Telegram messenger with text puzzle similar to the Wordle game [47] has been created, which is designed to teach a text programming language, in particular Python (Fig. 1), and has the following game mechanics.

1. The aim of each task is to recognize a line of a certain program in several steps-attempts. Initially, the line of the program is presented as a sequence of symbols *, the number of which is equal to the length of the line (with spaces).

2. Simplifications in the formation of the task are carried out in relation to the names of variables, values of numbers, strings, lists, dictionaries and other structures. Thus, variables are changed and used in the order a, b, c, ..., numbers — in the order 1, 2, 3, ..., symbols and strings — in the order ‘a’, ‘b’, ‘c’, ..., mathematical operations — in the order ‘+’, ‘-’, ‘*’, lists, tuples, dictionaries, etc. are simplified to 1 element.

3. The text entered at each step is checked for compliance with the length of the program line, the syntax rules of the language and the rules of PEP8 (in the case of Python), and a check is also made for the presence of already open and known symbols. In case of violations, the student is informed about the discrepancy and incorrectness of the input, and in case of absence, a line with open symbols is given, followed by a hint after the # symbol. The hint is given as a set of symbols sorted in ascending order of ASCII code.

4. Open symbols and a hint after each attempt are formed based on the result of a symbol-by-symbol comparison of the entered text with the required line of the program according to the same rules as in the Wordle game: if the symbols match, i.e. are in the correct places, these symbols appear instead of *, and if the symbols are in other positions, i.e. are not in their places, these symbols appear in the hint.

5. The maximum number of correct attempts given to complete the task is 20.

6. There are 3 levels of tasks of varying difficulty and different scoring, the ability to change and move between levels. When completing at least 3 tasks at the current level, the level can be changed to the next one. The level can always be changed to a lower one. When completing task in 5 or fewer attempts, on the first level 10 scores are awarded, on the second level 20 scores are awarded, on the third level 30 scores are awarded. If the task is completed in more attempts, the number of scores received decreases.

7. The number of characters in the task depends on the level: the higher the level, the more characters in the used line of the program. Program lines with widespread functions, commands, methods are used at levels 1 and 2, less common ones are used at level 3.



The gameplay of “Identify the Python command” is as follows.

- AUTOMATION AND REMOTE CONTROL Vol. 86 No. 5 2025

6. Interaction with the gaming chatbot is aimed at supporting the development of hard skills (memorization and recall of commands, functions, methods of the programming language), as well as cognitive skills (working memory, attention and logical thinking) [48].

4. MODELS DEVELOPMENT

4.1. Conducted Experiment

The experiment involved 60 first-year IT students studying the Python programming language as part of a two-semester course. Most of the students had no programming experience when they began their studies at the university. The experiment itself took place in the second semester.

The experiment used a non-intelligent game chatbot “Identify the Python command” and 2 questionnaires. The indicators in the questionnaires were assessed using a 5-point Likert scale.

Stages of the experiment.

1. Filling out the first questionnaire and indicating self-assessment of the level of writing programs in Python and the complexity of the course.

2. Using the chatbot for 10 days outside the classroom and solving one puzzle per day. Automatic collection of data on the progress and results of solving tasks for each student.

3. Filling out the second questionnaire and indicating the interest and difficulty of the tasks, sensation of involvement of working memory, attention and logical thinking when solving tasks.

4.2. Developed Models

To build the models, we used three data sources: the first questionnaire, the second questionnaire, and statistics on students’ work with the gaming chatbot.

The following indicators were obtained from the first questionnaire:

- Self-assessment of Python coding skill (SAWP);
- Assessment of Python course difficulty (ACD).

The following indicators were obtained from the second questionnaire:

- Self-assessment of sensation of working memory involvement in solving tasks (SAWM);
- Self-assessment of sensation of logical thinking involvement in solving tasks (SALT);
- Self-assessment of sensation of remembering Python function in solving tasks (SARF);
- Perceived difficulty of tasks (PDT);
- Interest in tasks (PIT).

The following indicators were obtained from the student work statistics:

- Final level of tasks in the gaming chatbot (FLT);
- Number of steps/attempts spent on solving a separate task (NS);
- Average number of steps spent on solving tasks (ANS);
- Final number of points for completing tasks (SRT).

We assume that the PDT perceived difficulty of tasks is influenced by SAWM, SALT, SARF, FLT (Fig. 2). This influence can be explained as follows. If a student was at 2 or 3 levels, he faced more difficult tasks, compared to those who were at 1 or 2 levels and were not at 3. Therefore, such students will obviously rate the tasks as more difficult, i.e. FLT influences PDT. However, it should be taken into account that if a student cannot solve a task while at a level, he will most likely move to a lower level. Working memory and logical thinking play a key role in solving various problems. To successfully complete tasks, it is necessary to remember Python functions and commands and apply logical reasoning at each step (attempt) when open symbols appear. Working memory will

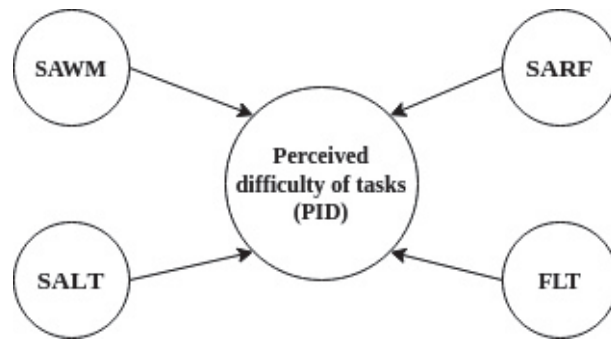


Fig. 2. Model of the influence of indicators on the perceived difficulty of tasks.

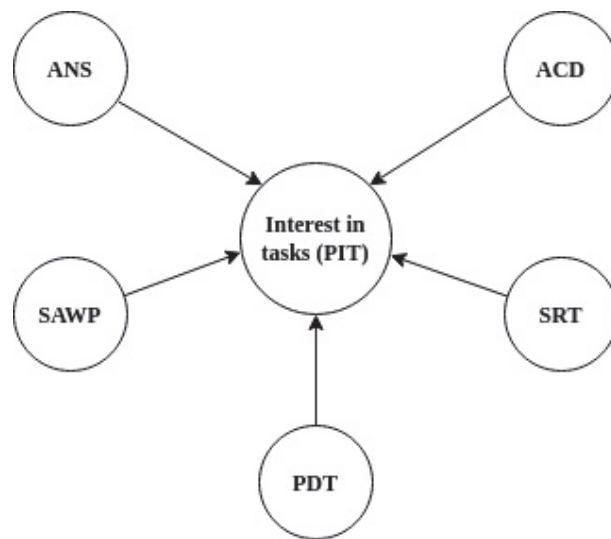


Fig. 3. Model of the influence of indicators on the interest in tasks.

ensure the process of solving tasks, so SAWM will influence PDT. It should be noted that simple tasks, unlike complex ones, do not exert a significant load on working memory. In addition to working memory, the student's level of logical thinking and the number of previously memorized Python functions and commands will affect the perceived difficulty of the tasks. With a low level of logical thinking and a low number of memorized functions, the student may experience difficulties in solving even problems of average complexity, and the tasks will be perceived as difficult. Hence the influence of SARF and SALT on PDT [49].

We assume that PIT interest in tasks is influenced by ANS, SAWP, PDT, ACD, SRT (Fig. 3). This influence can be explained as follows. A large number of attempts spent on solving a task may indicate both high task complexity and the use of brute force. Both of these options negatively affect the interest in the tasks being performed. Therefore, ANS affects PIT. To successfully write code, students must know and remember the functions of the Python language. Students who rate their programming skills highly usually know a sufficient number of functions and have developed logical and algorithmic thinking. Therefore, solving problems will be more interesting for them than for students who are unsure of their programming abilities, are not familiar with a sufficient number of functions, and do not have developed logical and algorithmic thinking. Therefore, SAWP affects PIT. If students spend a lot of time solving complex problems, especially if they have a low level of working memory and logical thinking, this can cause some discomfort and reduce the level of interest. Therefore, PDT affects PIT. The use of gamification tools and digital games can reduce the

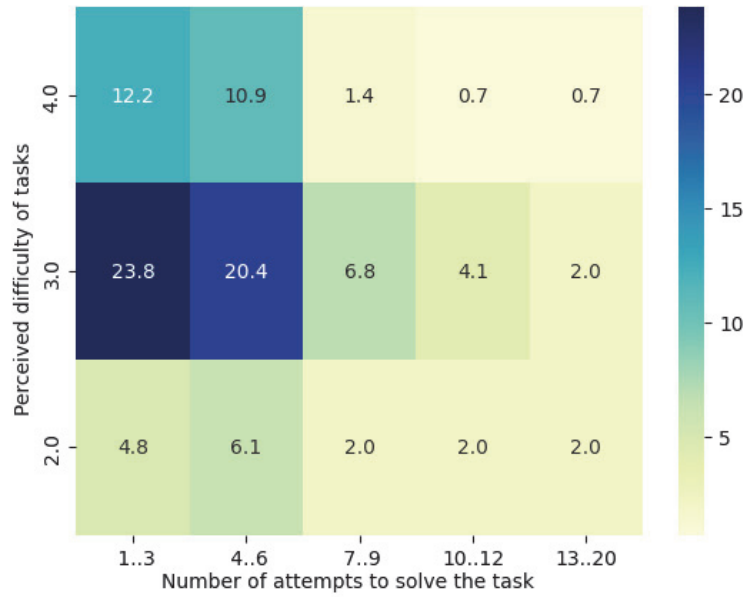


Fig. 4. Heat map of the frequency of occurrence of combinations of perceived task difficulty and the number of attempts to solve tasks.

cognitive load in complex courses. Therefore, ACD affects PIT. The more points a student scored as a result of working with a game chatbot, the more engaged he was and the more interesting it was for him to work with the chatbot. Thus, SRT affects PIT [49].

We conducted a regression analysis and built models of the impact of the parameters on the interest and perceived difficulty of the gaming chatbot. To assess the importance and contribution of each parameter to the value of the dependent variable, standardization was used with a mathematical expectation of 0 and a standard deviation of 1. The dependent variables were not standardized.

To assess the perceived difficulty of the gaming chatbot tasks, a linear regression with a significance level of $p < 0.01$ was obtained:

$$PDT = a_{SAWM} \cdot SAWM + a_{SARF} \cdot SARF + a_{SALT} \cdot SALT + a_{FLT} \cdot FLT + a_0^{PDT}, \quad (1)$$

where the weighting coefficients take the following values:

$$a_{SAWM} = 0.63^{***}, \quad a_{SARF} = -0.52^{**}, \quad a_{SALT} = -0.41^{**}, \quad a_{FLT} = 0.23^*, \\ a_0^{PDT} = 3.07^{***} \quad (***-p < 0.001, **-p < 0.01, *-p < 0.1).$$

The coefficient of determination of the model was 0.82 (adjusted — 0.74).

To assess the interest in the gaming chatbot tasks, a linear regression with a significance level of $p < 0.001$ was obtained:

$$PIT = a_{ANS} \cdot ANS + a_{SAWP} \cdot SAWP + a_{PDT} \cdot PDT + a_{ACD} \cdot ACD + a_{SRT} \cdot SRT + a_0^{PIT}, \quad (2)$$

where the weighting coefficients take the following values:

$$a_{ANS} = -0.68^{**}, \quad a_{SAWP} = 0.64^{**}, \quad a_{PDT} = -0.47^*, \\ a_{ACD} = 0.44^*, \quad a_{SRT} = 0.27^*, \quad a_0^{PIT} = 3.73^{***}.$$

The coefficient of determination of the model was 0.86 (adjusted — 0.78).

In the model (1), the SAWM, SALT, SARF indicators change little over a short period of time. The FLT indicator characterizes the final level of tasks in the chatbot. With different FLT values, different values of the number of steps to solve tasks are observed. So, with $FLT = 2$, the average value of $ANS = 5.4$, and with $FLT = 3$, the average value of $ANS = 5.0$.

In the model (2), the SAWP, ACD indicators change little over a short period of time. The SRT indicator characterizes the final number of points scored for completing tasks in the chatbot. This indicator is correlated with ANS, and the Pearson correlation coefficient is -0.43 . The PDT indicator used in the model (2) is the output of the model (1).

We constructed a heat map that shows the frequency of occurrence of various combinations of perceived task difficulty and the number of attempts used to solve the tasks. We plotted the frequencies on the heat map as a percentage of the total number of tasks solved by students (Fig. 4).

The heat map showed that the highest frequencies of combinations of PDT values and the number of attempts to solve tasks are observed among students who solved tasks in 1–6 attempts and for whom PDT is medium or high. An increase in the number of attempts is observed among students for whom the perceived difficulty of the gaming chatbot tasks was medium.

Therefore, it is possible to propose to use models (1) and (2) in the task issuance control algorithm based on the ANS value.

4.3. Limitations of the Developed Models

Despite the statistical significance of the obtained results, it is necessary to take into account a number of limitations associated with the developed models.

- Limited sample of students. The experiment was conducted on a limited sample of students (60 people), which may not reflect the diversity of educational contexts and different groups of students. Students with different levels of training, motivation and previous experience of interacting with chatbots, programming in Python, using traditional puzzles may demonstrate different reactions to tasks, which limits the generalizability of the results.
- Limited set of factors. The models do not take into account demographic factors of students, the influence of external factors. These factors can affect the perception of difficulty and interest in tasks, but they were not included in the analysis. In addition, time parameters for completing tasks in the chatbot, such as duration or frequency of attempt input, were not used.
- Subjectivity of assessments. The perception of difficulty and interest in tasks are subjective and may vary among different students. The models are based on self-assessments by participants, which may introduce biases related to individual preferences.
- Linear dependencies. The assumption of a linear relationship between the parameters and perceived difficulty and interest in tasks may be an oversimplification. Real relationships may be more complex and nonlinear, which requires further research and more complex models.
- Long-term effects. The experiment does not take into account the long-term effects of using the gaming chatbot, such as the impact on academic performance, motivation, and the sustainability of interest in the learning process over time. This limits the ability to draw conclusions about the long-term effects of introducing such technologies into the educational process.

5. STRUCTURE OF THE GAMING CHATBOT WITH ARTIFICIAL INTELLIGENCE

The effectiveness of using the gaming educational chatbot in learning programming depends on a number of factors related to the characteristics of the students. These factors include:

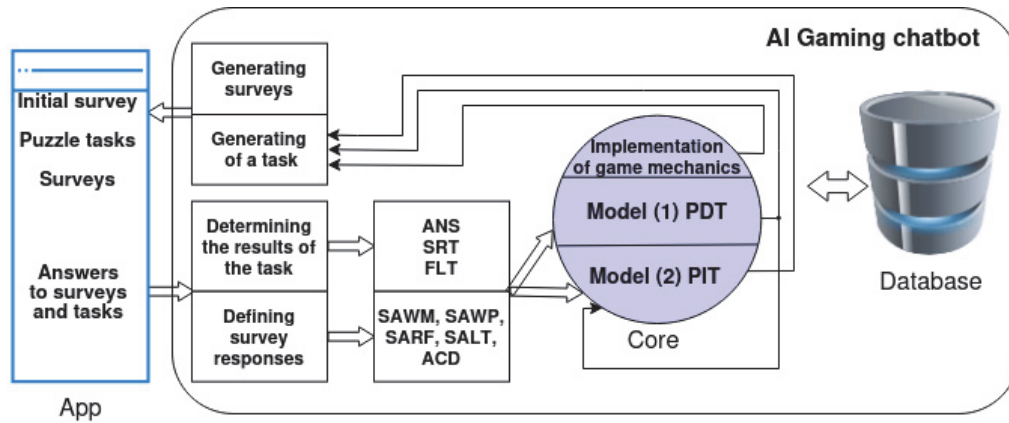


Fig. 5. Structure of the gaming chatbot with artificial intelligence.

- Programming proficiency: students with basic knowledge and experience with a programming language (in this case, Python) will benefit more from using the chatbot.
- Task-solving strategies and trajectories: students ability to effectively analyze tasks, plan solution steps, and use adequate algorithms is directly related to their success in solving tasks.

Perception of difficulty task is a key factor influencing interest in the chatbot. Perception of difficulty is influenced by:

- Cognitive abilities: the volume of working memory and the level of logical thinking play an important role in students ability to cope with tasks.
- Objective difficulty of task: the difficulty of the tasks should be adequate to the level of students preparation.
- Solution trajectory: the number of steps required to solve a problem can affect the perception of its difficulty.
- Knowledge of programming language functions and commands.

Based on this, we can conclude that the use of the gaming chatbot by beginners in programming is associated with a certain risk. And a certain “threshold” level of student preparation is required, including:

- Knowledge of basic Python functions and commands.
- Skills in reading and writing program code.

Students with a low level of working memory, logical thinking, or insufficient knowledge of Python commands may face increased difficulty of tasks, which will negatively affect their interest in the chatbot. Therefore, when introducing the gaming chatbot into the educational process, it is important to take into account the individual characteristics of students and provide a “threshold” level of preparation for its effective use.

It is possible to propose a structure of the gaming chatbot with models for managing Wordle-like puzzle for learning programming. Here, based on the results of the survey and data transfer to the chatbot, an assessment of the level of difficulty and interest of the tasks can be made, showing the feasibility of using the chatbot at the current moment in time. Accordingly, the chatbot, depending on the received values, will monitor the feasibility of use, and in case of feasibility, depending on the strategy used and the completion of tasks, it will select puzzles of the appropriate level (Fig. 5).

We presented the algorithm of the intelligent gaming chatbot (Algorithm 1) in the form of pseudocode.

Algorithm 1. Intelligent chatbot behavior**Input** : Models (1) and (2), survey and task completion results**Output:** Chatbot usage recommendation, puzzle

```

1 Setting  $PDT_i$  and  $PIT_i$  thresholds.
2 Determining  $SAWM$ ,  $SARF$ ,  $SALT$ ,  $SAWP$ ,  $ACD$  values based on testing and survey
  results.
3 Using models (1) and (2) and determining initial values  $PDT_0$  and  $PIT_0$ .
4 if  $PDT_0 > PDT_i$  or  $PIT_0 < PIT_i$  then
5   | Issuing a recommendation about undesirable and premature use of the chatbot at this
6   | point in time.
7 else
8   | Issuing the first puzzle.
9   | Monitoring the solution of the first puzzle.
10  | Determining  $NS$ .
11 end
12 if Puzzle solved then
13   | Based on  $NS$  and models (1), (2), the values of  $PDT$ ,  $PIT$  are determined and a new
14   | puzzle is generated taking these values into account.
15   | As puzzles are solved,  $ANS$  is determined.
16   | The values of  $PDT$ ,  $PIT$  are recalculated and each new puzzle is formed in accordance
17   | with them.
18 else
19   | Formation of a new puzzle.
20 end

```

6. CONCLUSION

Wordle-like puzzles can be used at the stage of learning programming. In general, they are quite an interesting educational tool for students. However, they should be used when the level of students' competencies is not low. In this case, solving puzzles will be exciting and useful. Also, this type of puzzles may not be very interesting for students with a low level of RAM and logical thinking.

Therefore, the introduction of machine learning models into the gaming chatbot, taking into account the student's indicators and the progress of the task, will allow the chatbot to predict the perceived difficulty of tasks and interest in them. Due to this, it is possible to adapt the behavior of the chatbot to the student's level in order to make the difficulty of the tasks optimal, and the interest — maximum.

The developed machine learning models formed the basis for the algorithm of operation and the structure of the game chatbot with artificial intelligence. Based on the developed structure, a new version of the game chatbot was implemented. Testing of the resulting chatbot showed great interest in its use among students studying programming.

Some parameters of the models (FLT, ANS, SRT, PDT) can be determined by the chatbot based on the results of completing the task and using the model 1. However, for some parameters (SAWM, SARF, SALT, SAWP, ACD), it is necessary to conduct surveys at least at the initial stage of connecting and launching the chatbot and, preferably, adjust them as the chatbot is used by more frequent surveys. In this regard, it is of interest to replace subjective indicators with objective ones and search for models that do not contain subjective indicators, or in which these indicators

are expressed through the parameters of task completion. This will be the main direction of further work.

In our future work, we also plan to expand the sample and take into account other indicators in the models, in particular, demographic data of students and their experience with chatbots, Python programming and solving traditional puzzles. We plan to consider nonlinear models of the influence of indicators on the perceived difficulty and interest in tasks.

FUNDING

The publication was prepared within the framework of the Academic Fund Program at the HSE University (grant no. 22-00-052).

REFERENCES

1. Mayer, R.E., Computer Games in Education, *Ann. Rev. Psychol.*, 2019, vol. 70, pp. 531–549. <https://doi.org/10.1146/ANNUREV-PSYCH-010418-102744>
2. Zhan, Z., He, L., Tong, Y., et al., The effectiveness of gamification in programming education: Evidence from a meta-analysis, *Computers and Education: Artificial Intelligence*, 2022, vol. 3, p. 100096. <https://doi.org/10.1016/J.CAEAI.2022.100096>
3. Nouchi, R., Taki, Y., Takeuchi, H., et al., Brain Training Game Boosts Executive Functions, Working Memory and Processing Speed in the Young Adults: A Randomized Controlled Trial, *PLoS One*, 2013, vol. 8, p. e55518. <https://doi.org/10.1371/JOURNAL.PONE.0055518>
4. Huang, V., Young, M., and Fiocco, A.J., The Association Between Video Game Play and Cognitive Function: Does Gaming Platform Matter?, *Cyberpsychol Behav Soc Netw*, 2017, vol. 20, pp. 689–694. <https://doi.org/10.1089/CYBER.2017.0241>
5. Harteveld, C., Smith, G., Carmichael, G., et al., A design-focused analysis of games teaching computer science, *Proceedings of Games+Learning+Society*, 2014, pp. 1–8.
6. Parsons, D. and Haden, P., Parsons Programming Puzzles: A Fun and Effective Learning Tool for First Programming Courses, in *Tolhurst D. and Mann S. (eds.) Proceedings of the 8th Australasian Conference on Computing Education (Vol. 52)*, 2006, pp. 157–163. Australian Computer Society, Inc.
7. Murphy, M., Spillane, K., Cully, J., et al., Can Word Puzzles be Tailored to Improve Different Dimensions of Verbal Fluency? A Report of an Intervention Study, *J. Psychol.*, Elsevier Publ., 2016, vol. 150, pp. 743–754. <https://doi.org/10.1080/00223980.2016.1182887>
8. Friedlander, K.J. and Fine, P.A., Fluid Intelligence is Key to Successful Cryptic Crossword Solving, *Journal of Expertise*, 2020, vol. 3, pp. 101–132.
9. Juricic, V. and Radošević, M., Puzzle-like programming languages in teaching programming, in *2019 42nd International Convention on Information and Communication Technology, Electronics and Microelectronics, MIPRO 2019 — Proceedings*, Institute of Electrical and Electronics Engineers Inc., 2019, pp. 542–546. <https://doi.org/10.239/r/mipro.2019.8757192>
10. Fakhoury, S., Ma, Y., Arnaoudova, V., et al., The Effect of Poor Source Code Lexicon and Readability on Developers Cognitive Load, in *2018 IEEE/ACM 26th International Conference on Program Comprehension (ICPC)*, Gothenburg, Sweden, 2018, pp. 286–28610.
11. Wassila, D. and Tahar, B., Using serious game to simplify algorithm learning, in *2012 International Conference on Education and e-Learning Innovations, ICEELI 2012*, 2012, pp. 1–5. <https://doi.org/10.1109/ICEELI.2012.6360569>
12. Liu, Y., Tan, J., Yang, Z., et al., Application Study of Collaborative Multi-Model in Wordle Game, in *2024 5th International Conference on Machine Learning and Computer Application (ICMLCA)*, 2024, pp. 591–596.

13. Zhang, Z., Wang, S., and Sun, D., A Study of Wordle Games Based on Mathematical Models, in *2024 IEEE 2nd International Conference on Control, Electronics and Computer Technology (ICCECT)*, 2024, pp. 813–817.
14. Wang, S., Zhou, G., Zhang, Z., and Liu, J., Exploring Wordle: A Data-Driven Analysis of Gameplay Dynamics, Mathematical Modeling, and Educational Implications, in *Fourth International Conference on Applied Mathematics, Modelling, and Intelligent Computing (CAMMIC 2024)*, 2024.
15. Cunningham-Nelson, S., Boles, W., Trouton, L., and Margerison, E., A review of chatbots in education: Practical steps forward, in *30th Annual Conference for the Australasian Association for Engineering Education (AAEE 2019): Educators Becoming Agents of Change: Innovate, Integrate, Motivate*, Engineers Australia, Brisbane, Queensland, 2019, pp. 299–306.
16. Khamparia, A. and Pandey, B., Effects of visual map embedded approach on students learning performance using Briggs-Myers learning style in word puzzle gaming course, *Computers & Electrical Engineering*, 2018, vol. 66, pp. 531–540. <https://doi.org/10.1016/J.COMPELECENG.2017.12.041>
17. Setiawan, H. and Phillipson, S., The effectiveness of Game-Based Science Learning (GBSL) to improve students academic achievement: A meta-analysis of current research from 2010 to 2017, *REID (Research and Evaluation in Education)*, 2019, vol. 5, pp. 152–168. <https://doi.org/10.21831/REID.V5I2.28073>
18. Karakoç, B., Eryılmaz, K., Turan Özpolat, E., et al., The Effect of Game-Based Learning on Student Achievement: A Meta-Analysis Study, *Technology, Knowledge and Learning*, 2020, vol. 27, pp. 207–222. <https://doi.org/10.1007/S10758-020-09471-5>
19. Connolly, T.M., Boyle, E.A., MacArthur, E., Hainey, T., and Boyle, J.M., A systematic literature review of empirical evidence on computer games and serious games, *Comput. Educ.*, 2012, vol. 59, pp. 661–686. <https://doi.org/10.1016/J.COMPEDU.2012.03.004>
20. Vahldick, A., Mendes, A.J., and Marcelino, M.J., A review of games designed to improve introductory computer programming competencies, in *Proceedings — Frontiers in Education Conference, FIE*, 2014, pp. 1–7. Institute of Electrical and Electronics Engineers Inc. <https://doi.org/10.1109/FIE.2014.7044114>
21. Miljanovic, M.A. and Bradbury, J.S., A review of serious games for programming, in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2018, vol. 11243 LNCS, pp. 204–216. https://doi.org/10.1007/978-3-030-02762-9_21/COVER
22. Giannakoulas, A. and Xinogalos, S., A Review of Educational Games for Teaching Programming to Primary School Students, in *Handbook of Research on Tools for Teaching Computational Thinking in P-12 Education*, 2020, pp. 1–30. <https://doi.org/10.4018/978-1-7998-4576-8.ch001>
23. Falkner, N., Sooriamurthi, R., and Michalewicz, Z., Puzzle-based learning for engineering and computer science, *Computer*, 2010, vol. 43, pp. 20–28. <https://doi.org/10.1109/MC.2010.113>
24. Toma, M., Halpern, D.F., and Berger, D.E., Cognitive Abilities of Elite Nationally Ranked SCRABBLE and Crossword Experts, *Applied Cognitive Psychology*, 2014, vol. 28, pp. 727–737. <https://doi.org/10.1002/ACP.3059>
25. Hambrick, D.Z., Salthouse, T.A., and Meinz, E.J., Predictors of crossword puzzle proficiency and moderators of age-cognition relations, *Journal of Experimental Psychology: General*, 1999, vol. 128, pp. 131–164. <https://doi.org/10.1037/0096-3445.128.2.131>
26. Makasi, T., Nili, A., Desouza, K.C., and Tate, M., A Typology of Chatbots in Public Service Delivery, *IEEE Software*, 2022, vol. 39, pp. 58–66. <https://doi.org/10.1109/MS.2021.3073674>
27. Marikyan, D., Papagiannidis, S., Rana, O.F., et al., “Alexa, let’s talk about my productivity”: The impact of digital assistants on work productivity, *Journal of Business Research*, 2022, vol. 142, pp. 572–584. <https://doi.org/10.1016/J.JBUSRES.2022.01.015>
28. Kimani, E., Rowan, K., McDuff, D., Czerwinski, M., and Mark, G., A Conversational Agent in Support of Productivity and Wellbeing at Work, in *2019 8th International Conference on Affective Computing and Intelligent Interaction, ACII 2019*, 2019, pp. 332–338. <https://doi.org/10.1109/ACII.2019.8925488>

29. Gkinko, L. and Elbanna, A., Hope, tolerance and empathy: employees emotions when using an AI-enabled chatbot in a digitalised workplace, *Information Technology and People*, 2022, vol. 35, pp. 1714–1743. <https://doi.org/10.1108/ITP-04-2021-0328/FULL/PDF>
30. Bavaresco, R., Silveira, D., Reis, E., et al., Conversational agents in business: A systematic literature review and future research directions, *Computer Science Review*, 2020, vol. 36, article 100239. <https://doi.org/10.1016/J.COSREV.2020.100239>
31. Brandtzaeg, P.B. and Følstad, A., Why people use chatbots, in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2017, vol. 10673, pp. 377–392. https://doi.org/10.1007/978-3-319-70284-1_30
32. Fotheringham, D. and Wiles, M.A., The effect of implementing chatbot customer service on stock returns: an event study analysis, *Journal of the Academy of Marketing Science*, 2023, vol. 51, pp. 802–822. <https://doi.org/10.1007/S11747-022-00841-2/TABLES/6>
33. Ciechanowski, L., Przegalinska, A., Magnuski, M., and Gloor, P., In the shades of the uncanny valley: An experimental study of human-chatbot interaction, *Future Generation Computer Systems*, 2019, vol. 92, pp. 539–548. <https://doi.org/10.1016/J.FUTURE.2018.01.055>
34. Thakkar, J., Raut, P., Doshi, Y., and Parekh, K., Erasmus AI Chatbot, *International Journal of Computer Sciences and Engineering*, 2018, vol. 6, pp. 498–502. <https://doi.org/10.26438/IJCSE/V6I10.498502>
35. Sidaoui, K., Jaakkola, M., and Burton, J., AI feel you: customer experience assessment via chatbot interviews, *Journal of Service Management*, 2020, vol. 31, pp. 745–766. <https://doi.org/10.1108/JOSM-11-2019-0341/FULL/PDF>
36. Zhou, L., Paul, S., Demirkan, H., et al., Intelligence Augmentation: Towards Building Human-Machine Symbiotic Relationship, *AIS Trans. Hum.-Comput. Interact.*, 2021, vol. 13, pp. 243–264. <https://doi.org/10.17705/1thci.00149>
37. Gkinko, L. and Elbanna, A., The appropriation of conversational AI in the workplace: A taxonomy of AI chatbot users, *Int. J. Inf. Manage.*, 2023, vol. 69, p. 102568. <https://doi.org/10.1016/J.IJINFOMGT.2022.102568>
38. Rizomyliotis, I., Kastanakis, M.N., Giovanis, A., et al., “How mAy I help you today?” The use of AI chatbots in small family businesses and the moderating role of customer affective commitment, *J. Bus. Res.*, 2022, vol. 153, pp. 329–340. <https://doi.org/10.1016/J.JBUSRES.2022.08.035>
39. Chen, Q., Gong, Y., Lu, Y., and Tang, J., Classifying and measuring the service quality of AI chatbot in frontline service, *J. Bus. Res.*, 2022, vol. 145, pp. 552–568. <https://doi.org/10.1016/J.JBUSRES.2022.02.088>
40. Tam, W., Huynh, T., Tang, A., et al., Nursing education in the age of artificial intelligence powered Chatbots (AI-Chatbots): Are we ready yet?, *Nurse Educ. Today*, 2023, vol. 129, p. 105917. <https://doi.org/10.1016/J.NEDT.2023.105917>
41. Zhu, Y., Zhang, J., Wu, J., and Liu, Y., AI is better when I’m sure: The influence of certainty of needs on consumers’ acceptance of AI chatbots, *J. Bus. Res.*, 2022, vol. 150, pp. 642–652. <https://doi.org/10.1016/J.JBUSRES.2022.06.044>
42. Ruan, Y. and Mezei, J., When do AI chatbots lead to higher customer satisfaction than human frontline employees in online shopping assistance? Considering product attribute type, *J. Retail. Consum. Serv.*, 2022, vol. 68, p. 103059. <https://doi.org/10.1016/J.JRETCONSER.2022.103059>
43. Janssen, A., Rodríguez Cardona, D., Passlick, J., and Breitner, M.H., How to Make chatbots productive — A user-oriented implementation framework, *Int. J. Hum. Comput. Stud.*, 2022, vol. 168, p. 102921. <https://doi.org/10.1016/J.IJHCS.2022.102921>
44. Lee, D. and Yeo, S., Developing an AI-based chatbot for practicing responsive teaching in mathematics, *Comput. Educ.*, 2022, vol. 191, p. 104646. <https://doi.org/10.1016/J.COMPEDU.2022.104646>

45. Liu, C.C., Liao, M.G., Chang, C.H., and Lin, H.M., An analysis of childrens interaction with an AI chatbot and its impact on their interest in reading, *Comput. Educ.*, 2022, vol. 189, p. 104576. <https://doi.org/10.1016/J.COMPEDU.2022.104576>
46. Johnson, C., Urazov, M., and Zanolli, E., Escapeling: A Gamified, AI-Supported Chatbot for Collaborative Language Practice, *Lect. Notes Networks Syst.*, 2022, vol. 349, pp. 141–148. https://doi.org/10.1007/978-3-030-90677-1_14/COVER
47. Varnavsky, A.N., Comparison of perception indicators of the gaming chatbot with puzzles tasks for teaching programming with perception indicators of other educational tools, *Informatics and education*, 2024, vol. 39(4), pp. 41–50. (In Russ.) <https://doi.org/10.32517/0234-0453-2024-39-4-41-50>
48. Varnavsky, A.N., Using the gaming chatbot to develop algorithmic and logical thinking skills, *Informatics and education*, 2024, vol. 39(2), pp. 48–58. (In Russ.) <https://doi.org/10.32517/0234-0453-2024-39-2-48-58>
49. Varnavsky, A.N., Models of Interest, Difficulty, and Perceived Usefulness of the Gaming Chatbot with Wordle Like Puzzles for Learning Programming, *Information technologies*, 2024, vol. 7, no. 30, pp. 372–381. (In Russ.) <https://doi.org/10.17587/it.30.372-381>

This paper was recommended for publication by A.A. Galyaev, a member of the Editorial Board

Pareto Set Design When Combining Feasible Solutions of the Multicriteria Axial Assignment Problem

L. G. Afraimovich^{*,a} and M. D. Emelin^{*,b}

^{*}Lobachevsky Nizhny Novgorod State University, Nizhny Novgorod, Russia
e-mail: ^alevafraimovich@gmail.com, ^bmakcum888e@mail.ru

Received November 11, 2024

Revised February 12, 2025

Accepted February 26, 2025

Abstract—This paper considers a two-criteria three-index axial assignment problem, representing a classical NP-hard problem even in the single-criterion case. Within this formulation, the problem of combining feasible solutions is posed; it is an assignment problem on the set of solutions containing only the components of the feasible solutions selected. A polynomial algorithm is proposed to find Pareto optimal solutions in the combination problem of two feasible solutions. Based on this algorithm, a heuristic approach is constructed to estimate the Pareto set of the multicriteria axial assignment problem.

Keywords: axial assignment problem, multi-index problems, multicriteria problems, Pareto optimality, combining solutions, polynomial solvability, NP-hardness

DOI: 10.31857/S0005117925050057

1. INTRODUCTION

There is a wide class of applied problems formalized by multi-index axial assignment problems. Examples include resource allocation, scheduling, object tracking, robotic logistics [1–5], and others. The class of multi-index axial assignment problems is NP-hard even in the three-index case [6]. Particular polynomially solvable subclasses [1, 7] and subclasses with polynomial approximation algorithms [1, 8, 9] are known. The three-index axial assignment problem was studied in [10–16], including the construction of approximate approaches and the implementation of exact algorithms. Multi-criteria formulations of multi-index assignment problems were discussed in [17–20].

This paper addresses the issue of constructing the Pareto set in a two-criteria three-index axial assignment problem. We formulate the problem of combining feasible solutions, which is a multicriteria assignment problem on the set of solutions obtained by combining the components of given feasible solutions. We propose a polynomial algorithm for constructing a subset of Pareto optimal solutions on the set of combinations of two feasible solutions. This algorithm is applied to develop a heuristic algorithm for approximating the Pareto set of the two-criteria three-index assignment problem. A computational experiment is carried out to illustrate the approach. The paper continues the series of research works [21–24] devoted to combining the solutions of the axial assignment problem.

The remainder of this paper is organized as follows. In Section 2, we formulate the two-criteria three-index axial assignment problem and the corresponding combination problem of feasible solutions. Section 3 describes an algorithm for finding a subset of Pareto optimal solutions of the combination problem. The results of computational experiments are provided in Section 4.

2. PROBLEM STATEMENT

Let I, J , and K be disjoint index sets, $I \cap J = \emptyset$, $I \cap K = \emptyset$, $J \cap K = \emptyset$ and $|I| = |J| = |K| = n$; c_{ijk} and d_{ijk} , where $i \in I$, $j \in J$, and $k \in K$, are three-index cost matrices; finally, x_{ijk} , where $i \in I$, $j \in J$, and $k \in K$, is the three-index matrix of the variables. Then the two-criteria three-index axial assignment problem is formulated as the following integer linear programming problem:

$$\sum_{i \in I} \sum_{j \in J} x_{ijk} = 1, \quad k \in K, \quad (1)$$

$$\sum_{i \in I} \sum_{k \in K} x_{ijk} = 1, \quad j \in J, \quad (2)$$

$$\sum_{j \in J} \sum_{k \in K} x_{ijk} = 1, \quad i \in I, \quad (3)$$

$$x_{ijk} \in \{0, 1\}, \quad i \in I, \quad j \in J, \quad k \in K, \quad (4)$$

$$\sum_{i \in I} \sum_{j \in J} \sum_{k \in K} c_{ijk} x_{ijk} \rightarrow \min, \quad (5)$$

$$\sum_{i \in I} \sum_{j \in J} \sum_{k \in K} d_{ijk} x_{ijk} \rightarrow \min. \quad (6)$$

For the sake of convenience, we denote by Z_2 the two-criteria problem (1)–(6) and introduce

$$C(x) = \sum_{i \in I} \sum_{j \in J} \sum_{k \in K} c_{ijk} x_{ijk},$$

$$D(x) = \sum_{i \in I} \sum_{j \in J} \sum_{k \in K} d_{ijk} x_{ijk}.$$

Then x^* is called a Pareto optimal solution of the problem Z_2 if x^* satisfies the constraints (1)–(4) and there is no x' satisfying (1)–(4) such that $C(x^*) > C(x')$ and $D(x^*) \geq D(x')$ or $C(x^*) \geq C(x')$ and $D(x^*) > D(x')$.

As is well known, the (single-criterion) axial assignment problem (1)–(5) is NP-hard [6]. Hence, the problem of constructing a Pareto optimal solution of the two-criteria axial assignment problem Z_2 is also NP-hard. To prove this fact, we need to define $c_{ijk} = d_{ijk}$, $i \in I$, $j \in J$, $k \in K$, in the corresponding problem Z_2 .

Proposition 1. *The problem of constructing a Pareto optimal solution of the problem Z_2 is NP-hard.*

Thus, it is topical to develop effective heuristic approaches to estimating the Pareto set of the problem Z_2 . The heuristic approach proposed below is based on combining feasible solutions of the problem Z_2 . Here, combining means solving the problem on the set of solutions containing only the components of given feasible solutions.

Now we state the combination problem. Let a given set $W \subset I \times J \times K$ define a subset of allowed assignments. We introduce the auxiliary constraint

$$x_{ijk} = 0, \quad (i, j, k) \notin W. \quad (7)$$

The two-criteria problem (1)–(4), (7), (5), (6) for a given set W will be denoted by $Z_2(W)$. Obviously, problem (1)–(6) corresponds to the problem $Z_2(I \times J \times K)$.

Then x^* is called a Pareto optimal solution of the problem $Z_2(W)$ if x^* satisfies the constraints (1)–(4), (7) and there is no x' satisfying (1)–(4), (7) such that $C(x^*) > C(x')$ and $D(x^*) \geq D(x')$ or $C(x^*) \geq C(x')$ and $D(x^*) > D(x')$.

Checking the consistency of all constraints in the problem $Z_2(W)$ with an arbitrary set W , i.e., the system (1)–(4), (7), is an NP-complete problem [1]. When solving the combination problem, we will consider the sets W corresponding to the assignments of a given subset of feasible solutions.

Let x_{ijk} , where $i \in I$, $j \in J$, and $k \in K$, be a feasible solution of the system of constraints (1)–(4). Then $W(x)$ will denote the following set of allowed assignments:

$$W(x) = \{(i, j, k) | x_{ijk} = 1, i \in I, j \in J, k \in K\}.$$

Consider $x_{ijk}^1, x_{ijk}^2, \dots, x_{ijk}^r$, where $i \in I$, $j \in J$, and $k \in K$, representing arbitrary r feasible solutions of the system of constraints (1)–(4). Then

$$W(x^1, x^2, \dots, x^r) = W(x^1) \cup W(x^2) \cup \dots \cup W(x^r).$$

Below, we will investigate the problem $Z_2(W(x^1, x^2, \dots, x^r))$.

3. CONSTRUCTING A SUBSET OF PARETO OPTIMAL SOLUTIONS

For $r = 2$, let x^1 and x^2 be given feasible solutions of the system of constraints (1)–(4). Consider the problem $Z_2(W(x^1, x^2))$ and the issue of constructing its Pareto optimal solutions.

Algorithm 1. Constructing a subset of Pareto optimal solutions of the problem $Z_2(W(x^1, x^2))$.

Step 1. Construct a graph $G = (V, A)$, where

$$V = \{I \cup J \cup K\}, \quad A = \{(i, j), (i, k), (j, k) | (i, j, k) \in W(x^1, x^2)\}.$$

Step 2. Find the connected components V_l , $l = \overline{1, q}$, of the graph G and construct the subgraphs $G_l = (V_l, A_l)$, $l = \overline{1, q}$, generated by the corresponding connected components.

Step 3. Construct the sets

$$D_l^1 = \{(i, j, k) | (i, j, k) \in W(x^1), (i, j), (i, k), (j, k) \in A_l\}, \quad l = \overline{1, q},$$

$$D_l^2 = \{(i, j, k) | (i, j, k) \in W(x^2), (i, j), (i, k), (j, k) \in A_l\}, \quad l = \overline{1, q}.$$

Step 4. Let

$$P_l = \left\{ p \left| \sum_{(i, j, k) \in D_l^p} c_{ijk} = \min_{p' \in \{1, 2\}} \sum_{(i, j, k) \in D_l^{p'}} c_{ijk}, p \in \{1, 2\} \right. \right\},$$

$$p_l = \operatorname{argmin}_{p \in P_l} \sum_{(i, j, k) \in D_l^p} d_{ijk}, \quad l = \overline{1, q}.$$

Step 5. Construct the Pareto optimal solution x^{*0} using the following algorithm. Let $x_{ijk}^{*0} := 0$, where $i \in I$, $j \in J$, and $k \in K$. For each $l = \overline{1, q}$, execute $x_{ijk}^{*0} := 1, (i, j, k) \in D_l^{p_l}$.

Step 6. Construct the following index set of the connected components:

$$L = \left\{ l \mid \sum_{(i,j,k) \in D_l^{\bar{p}}} c_{ijk} > \sum_{(i,j,k) \in D_l^{p_l}} c_{ijk} \quad \text{and} \quad \sum_{(i,j,k) \in D_l^{\bar{p}}} d_{ijk} < \sum_{(i,j,k) \in D_l^{p_l}} d_{ijk}, \quad l \in \{1, \dots, q\} \right\},$$

where $\bar{p} = 3 - p$ for $p \in \{1, 2\}$.

Step 7. Sort this set in ascending order of the value

$$tg(l) = \frac{\sum_{(i,j,k) \in D_l^{p_l}} d_{ijk} - \sum_{(i,j,k) \in D_l^{\bar{p}}} d_{ijk}}{\sum_{(i,j,k) \in D_l^{\bar{p}}} c_{ijk} - \sum_{(i,j,k) \in D_l^{p_l}} c_{ijk}},$$

i.e., let $L = \{l_1, \dots, l_{|L|}\}$ and $tg(l_s) \geq tg(l_{s+1})$, $s = \overline{1, |L| - 1}$.

Step 8. Construct $|L|$ Pareto optimal solutions x^{*s} , $s = \overline{1, |L|}$, in the following way. Let $x_{ijk}^{*s} := 0$, $i \in I$, $j \in J$, $k \in K$ and, for each $t = \overline{1, |q|}$, execute:

if $t \in \{l_1, \dots, l_s\}$, then $x_{ijk}^{*s} := 1$, $(i, j, k) \in D_t^{\bar{p}t}$;

otherwise, $x_{ijk}^{*s} := 1$, $(i, j, k) \in D_t^{p_t}$.

Here is a numerical example for the operation of Algorithm 1.

Example 1. An illustration of Algorithm 1.

Let $n = 6$ and let the matrices c_{ijk} and d_{ijk} be

$$c_{ijk} = \begin{cases} 0, (i, j, k) \in \{(1, 1, 1), (1, 1, 2), (2, 2, 1), (3, 3, 3), (4, 4, 4), (3, 3, 4), \\ (5, 5, 5), (5, 5, 6), (6, 6, 5)\}, \\ 1, (i, j, k) \in \{(4, 4, 3)\}, \\ 2, (i, j, k) \in \{(6, 6, 6)\}, \\ 3, (i, j, k) \in \{(2, 2, 2)\}, \\ 10 \text{ otherwise.} \end{cases}$$

$$d_{ijk} = \begin{cases} 0, (i, j, k) \in \{(1, 1, 1), (1, 1, 2), (2, 2, 2), (3, 3, 3), (4, 4, 3), (3, 3, 4), \\ (5, 5, 5), (5, 5, 6), (6, 6, 6)\}, \\ 1, (i, j, k) \in \{(2, 2, 1), (4, 4, 4)\}, \\ 4, (i, j, k) \in \{(6, 6, 5)\}, \\ 10 \text{ otherwise,} \end{cases}$$

Consider two feasible solutions x^1 and x^2 of the form

$$x_{ijk}^1 = \begin{cases} 1, (i, j, k) \in \{(1, 1, 1), (2, 2, 2), (3, 3, 3), (4, 4, 4), (5, 5, 5), (6, 6, 6)\}, \\ 0 \text{ otherwise,} \end{cases}$$

$$x_{ijk}^2 = \begin{cases} 1, (i, j, k) \in \{(1, 1, 2), (2, 2, 1), (3, 3, 4), (4, 4, 3), (5, 5, 6), (6, 6, 5)\}, \\ 0 \text{ otherwise.} \end{cases}$$

We describe the operation of Algorithm 1 on this example.

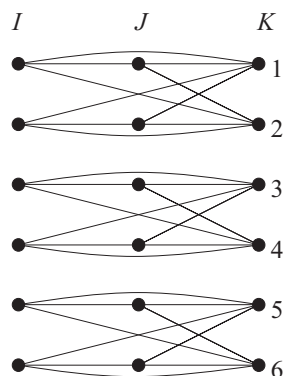


Fig. 1. The graph G at Step 1.

Figure 1 shows the graph G yielded by Step 1.

At Step 2 the connected components of the graph G are constructed. Obviously, the graph G has three connected components of the same structure. The sets obtained at Step 3 have the form

$$\begin{aligned} D_1^1 &= \{(1, 1, 1), (2, 2, 2)\}, & D_2^1 &= \{(3, 3, 3), (4, 4, 4)\}, & D_3^1 &= \{(5, 5, 5), (6, 6, 6)\}, \\ D_1^2 &= \{(1, 1, 2), (2, 2, 1)\}, & D_2^2 &= \{(3, 3, 4), (4, 4, 3)\}, & D_3^2 &= \{(5, 5, 6), (6, 6, 5)\}. \end{aligned}$$

At Step 4 we obtain

$$\begin{aligned} P_1 &= \{2\}, & p_1 &= 2, \\ P_2 &= \{1\}, & p_2 &= 1, \\ P_3 &= \{2\}, & p_3 &= 2. \end{aligned}$$

At Step 5 the first Pareto optimal solution x^{*0} is constructed as follows:

$$x_{ijk}^{*0} = \begin{cases} 1, & (i, j, k) \in \{(1, 1, 2), (2, 2, 1), (3, 3, 3), (4, 4, 4), (5, 5, 6), (6, 6, 5)\}, \\ 0 & \text{otherwise.} \end{cases}$$

The criteria values achieved at this solution are $C(x^{*0}) = 0$ and $D(x^{*0}) = 6$.

At Step 6 the set L contains all three connected components; at Step 7 it will be sorted to $l_1 = 3$, $l_2 = 2$, and $l_3 = 1$. The next three Pareto optimal solutions constructed at Step 8 are

$$\begin{aligned} x_{ijk}^{*1} &= \begin{cases} 1, & (i, j, k) \in \{(1, 1, 2), (2, 2, 1), (3, 3, 3), (4, 4, 4), (5, 5, 5), (6, 6, 6)\}, \\ 0 & \text{otherwise,} \end{cases} \\ x_{ijk}^{*2} &= \begin{cases} 1, & (i, j, k) \in \{(1, 1, 2), (2, 2, 1), (3, 3, 4), (4, 4, 3), (5, 5, 5), (6, 6, 6)\}, \\ 0 & \text{otherwise,} \end{cases} \\ x_{ijk}^{*3} &= \begin{cases} 1, & (i, j, k) \in \{(1, 1, 1), (2, 2, 2), (3, 3, 4), (4, 4, 3), (5, 5, 5), (6, 6, 6)\}, \\ 0 & \text{otherwise.} \end{cases} \end{aligned}$$

The corresponding criteria values are

$$\begin{aligned} C(x^{*1}) &= 2, & D(x^{*1}) &= 2, \\ C(x^{*2}) &= 3, & D(x^{*2}) &= 1, \\ C(x^{*3}) &= 6, & D(x^{*3}) &= 0. \end{aligned}$$

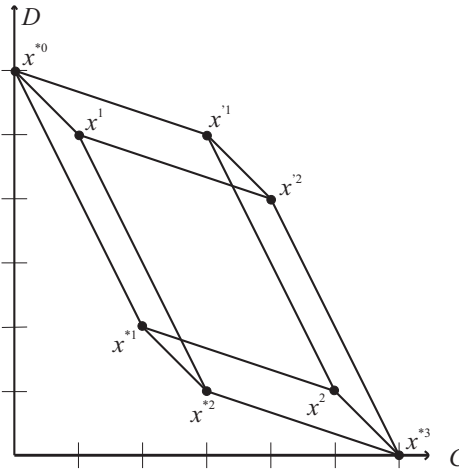


Fig. 2. The solutions found on the CD plane.

The resulting solutions are displayed on the CD plane, according to the value of their criterion (Fig. 2). Here, x^1 and x^2 are the initial solutions; x^{*0}, x^{*1}, x^{*2} , and x^{*3} are the Pareto optimal solutions of the combination problem obtained using Algorithm 1; finally, x'^1 and x'^2 are feasible solutions of the combination problem. In this example, there is one more Pareto optimal solution x^1 in addition to those found by the algorithm.

Next, we prove the correctness of Algorithm 1.

Consider the multicriteria problem. Let $X \subseteq R^n$ be the set of feasible solutions of the problem and $\overrightarrow{f(x)} \in R^m$ be the objective function of the problem, $f_i(x) \rightarrow \min, i = \overline{1, m}$.

Proposition 2. Let a set $X \subseteq R^n$ be finite. If $x \in X$ is not a Pareto optimal solution, then there exists a Pareto optimal solution $y \in X$ dominating x , i.e., $f_i(y) \leq f_i(x), i = \overline{1, m}$, and there exists an index $j \in \{1, \dots, m\}$ such that $f_j(y) < f_j(x)$.

Proof. Under the hypotheses of this proposition, suppose that $x \in X$ is not a Pareto optimal solution. In this case, we construct an algorithm for obtaining such a solution y .

Step 1. Find a solution x' dominating the solution x . If no such solution existed, the solution x would be Pareto optimal (a contradiction).

Step 2. Let $x := x'$ if x is Pareto-optimal, and return x ; otherwise, go to Step 1.

By the definition of Pareto optimality, the total number of Steps 2 of the algorithm will not exceed the cardinality of the solution set. Since the set is finite, the algorithm will output a Pareto optimal solution in a finite number of steps. The proof is complete.

Proposition 3. Any feasible solution x satisfying the system of constraints of the problem $Z_2(W(x^1, x^2))$ can be constructed by choosing triples only from the first or only from the second solution for each connected component independently.

Proof. This result was established when proving Theorem 1 in [21].

Therefore, we have

$$p_l(x) = \begin{cases} 1 & \text{if } x_{ijk} = 1 \text{ for all } (i, j, k) \in D_l^1 \\ 2 & \text{if } x_{ijk} = 1 \text{ for all } (i, j, k) \in D_l^2. \end{cases}$$

Proposition 4. For any Pareto optimal solution x' of the problem $Z_2(W(x^1, x^2))$,

$$\sum_{(i,j,k) \in D_l^{p_l(x')}} c_{ijk} = \sum_{(i,j,k) \in D_l^{p_l(x^{*0})}} c_{ijk} \quad \text{and} \quad \sum_{(i,j,k) \in D_l^{p_l(x')}} d_{ijk} = \sum_{(i,j,k) \in D_l^{p_l(x^{*0})}} d_{ijk}, \quad l \notin L.$$

Proof. Assume on the contrary the existence of a Pareto optimal solution x' with an index $l \notin L$ such that

$$\sum_{(i,j,k) \in D_l^{p_l(x')}} c_{ijk} \neq \sum_{(i,j,k) \in D_l^{p_l(x^{*0})}} c_{ijk} \quad \text{or} \quad \sum_{(i,j,k) \in D_l^{p_l(x')}} d_{ijk} \neq \sum_{(i,j,k) \in D_l^{p_l(x^{*0})}} d_{ijk}.$$

Consider two cases.

$$1. \quad \sum_{(i,j,k) \in D_l^{p_l(x')}} c_{ijk} \neq \sum_{(i,j,k) \in D_l^{p_l(x^{*0})}} c_{ijk}.$$

According to Step 5 of Algorithm 1,

$$\sum_{(i,j,k) \in D_l^{p_l(x')}} c_{ijk} > \sum_{(i,j,k) \in D_l^{p_l(x^{*0})}} c_{ijk}.$$

Then:

$$a) \text{ If } \sum_{(i,j,k) \in D_l^{p_l(x')}} d_{ijk} < \sum_{(i,j,k) \in D_l^{p_l(x^{*0})}} d_{ijk}, \text{ we have } l \in L \text{ (a contradiction).}$$

$$b) \text{ Otherwise, } \sum_{(i,j,k) \in D_l^{p_l(x')}} d_{ijk} \geq \sum_{(i,j,k) \in D_l^{p_l(x^{*0})}} d_{ijk}, \text{ and the triples corresponding to connected}$$

component l in the solution x' can be replaced as follows:

$$\begin{aligned} x'_{ijk} &:= 0, \quad (i, j, k) \in D_l^{p_l(x')}, \\ x'_{ijk} &:= 1, \quad (i, j, k) \in D_l^{p_l(x^{*0})}. \end{aligned}$$

The values of both criteria will be reduced accordingly; therefore, x' is not a Pareto optimal solution (a contradiction).

$$2. \quad \sum_{(i,j,k) \in D_l^{p_l(x')}} c_{ijk} = \sum_{(i,j,k) \in D_l^{p_l(x^{*0})}} c_{ijk}, \quad \sum_{(i,j,k) \in D_l^{p_l(x')}} d_{ijk} \neq \sum_{(i,j,k) \in D_l^{p_l(x^{*0})}} d_{ijk}.$$

According to Step 5 of Algorithm 1,

$$\sum_{(i,j,k) \in D_l^{p_l(x')}} d_{ijk} > \sum_{(i,j,k) \in D_l^{p_l(x^{*0})}} d_{ijk},$$

and the triples corresponding to connected component l in the solution x' can be replaced as follows:

$$\begin{aligned} x'_{ijk} &:= 0, \quad (i, j, k) \in D_l^{p_l(x')}, \\ x'_{ijk} &:= 1, \quad (i, j, k) \in D_l^{p_l(x^{*0})}. \end{aligned}$$

Again, the values of both criteria will be reduced, and hence x' is not a Pareto optimal solution (a contradiction). The proof of this proposition is complete.

We denote by P_t , $t = \overline{1, |L|}$, the straight line connecting the points x^{*t} and x^{*t+1} on the CD plane. The equation of P_t has the form

$$\frac{D - D(x^{*t})}{D(x^{*t-1}) - D(x^{*t})} = \frac{C - C(x^{*t})}{C(x^{*t-1}) - C(x^{*t})}.$$

For the sake of convenience, it can be reduced to

$$D = D(x^{*t}) + \frac{D(x^{*t}) - D(x^{*t-1})}{C(x^{*t-1}) - C(x^{*t})} (C(x^{*t}) - C) = D(x^{*t}) + tg(l_t)(C(x^{*t}) - C).$$

Also, this equation is equivalently written as

$$D = D(x^{*t-1}) + tg(l_t)(C(x^{*t-1}) - C).$$

We say that a feasible solution x of the problem $Z_2(W(x^1, x^2))$ is not below the straight line P_t if

$$D(x) \geq D(x^{*t}) + tg(l_t)(C(x^{*t}) - C(x)).$$

Obviously, if a solution x is not below the straight line P_t , it will not dominate any of the solutions x^{*t-1}, x^{*t} .

Proposition 5. *If for some $e \in \{1, \dots, |L|\}$ a solution x is not below the straight line P_e and $C(x) \leq C(x^{*e})$, then it is not below any straight line P_t , $t = \overline{e, |L|}$.*

Proof. Since x is not below the straight line P_e , we have

$$D(x) \geq D(x^{*e}) + tg(l_e)(C(x^{*e}) - C(x)).$$

By construction, $tg(l_t) \geq tg(l_{t+1}) \geq 0$, $t = \overline{1, |L| - 1}$. Due to the hypothesis of this proposition, $(C(x^{*e}) - C(x)) \geq 0$. Therefore,

$$tg(l_e)(C(x^{*e}) - C(x)) \geq tg(l_{e+1})(C(x^{*e}) - C(x))$$

and consequently,

$$D(x) \geq D(x^{*e}) + tg(l_{e+1})(C(x^{*e}) - C(x)).$$

In other words, the solution x is not below the straight line P_{e+1} . By induction, we establish that x is not below the straight line P_t , $t \geq e$. The proof is complete.

Theorem 1. *There exists no Pareto optimal solution x' of the problem $Z_2(W(x^1, x^2))$ that dominates any of the solutions constructed by Algorithm 1.*

Proof. Assume on the contrary the existence of a Pareto optimal solution x' dominating at least one of the solutions $x^{*0}, \dots, x^{*|L|}$ constructed by the algorithm. Due to Proposition 4, such a solution can differ from x^{*0} only in the triples of components $l \in L$.

We construct $|L| + 1$ solutions x''^s , $s = \overline{0, |L|}$, as follows.

Step 1. Let $x''_{ijk} := 0$, $i \in I$, $j \in J$, $k \in K$.

Step 2. For each $t = \overline{1, q}$, execute:

- If $t \in \{l_1, \dots, l_s\}$ and $p_t(x') \neq p_t(x^{*0})$, then $x''_{ijk} := 1$, $(i, j, k) \in D_t^{\overline{p_t}}$.
- Otherwise, $x''_{ijk} := 1$, $(i, j, k) \in D_t^{p_t}$.

By construction, $x''^0 = x^{*0}$, and hence the solution x''^0 is not below the straight line P_1 .

Now we demonstrate that if the solution x''^s is not below the straight lines P_t , $t = \overline{1, |L|}$, then the solution x''^{s+1} is not below the straight lines P_t , $t = \overline{1, |L|}$, as well.

For each straight line P_t , $t = \overline{1, r+1}$, two cases are possible:

1. $p_{l_{s+1}}(x') = p_{l_{s+1}}(x^{*0})$, then the solution x''^{s+1} coincides with the solution x''^s ; therefore, it is not below the straight line P_t .
2. $p_{l_{s+1}}(x') \neq p_{l_{s+1}}(x^{*0})$, then the condition that x''^s is not below P_t implies

$$D(x''^s) \geq D(x^{*t}) + tg(l_t)(C(x^{*t}) - C(x''^s)). \quad (8)$$

By construction,

$$D(x''^{s+1}) = D(x''^s) + tg(l_{s+1})(C(x''^s) - C(x''^{s+1})).$$

Applying inequality (8) yields

$$D(x''^{s+1}) \geq D(x^{*t}) + tg(l_t)(C(x^{*t}) - C(x''^s)) + tg(l_{s+1})(C(x''^s) - C(x''^{s+1}))$$

and consequently,

$$D(x''^{s+1}) \geq D(x^{*t}) + tg(l_t)(C(x^{*t}) - C(x''^{s+1})) + (tg(l_t) - tg(l_{s+1}))(C(x''^{s+1}) - C(x''^s)).$$

By construction $C(x''^{s+1}) - C(x''^s) \geq 0$ and $tg(l_t) - tg(l_{s+1}) \geq 0$ for $s+1 \geq t$. Then

$$D(x''^{s+1}) \geq D(x^{*t}) + tg(l_t)(C(x^{*t}) - C(x''^{s+1})),$$

i.e., the solution x''^{s+1} is not below the straight line P_t .

Hence, if the solution x''^s is not below the straight lines P_t , $t = \overline{1, |L|}$, then the solution x''^{s+1} is not below the straight lines P_t , $t = \overline{1, s+1}$.

By construction, $C(x''^{s+1}) \leq C(x^{*s+1})$. Then, according to Proposition 5, the solution x''^{s+1} is not below the straight lines P_t , $t = \overline{s+1, |L|}$ (here $e = s+1$). Therefore, if the solution x''^s is not below the straight lines P_t , $t = \overline{1, |L|}$, then the solution x''^{s+1} is not below the straight lines P_t , $t = \overline{1, |L|}$. By induction, we establish that $x''^{|L|}$ is not below the straight lines P_t , $t = \overline{1, |L|}$.

By construction, $x''^{|L|} = x'$, i.e., x' is not below the straight lines P_t , $t = \overline{1, |L|}$; hence, the solution x' dominates neither of the solutions $x^{*0}, \dots, x^{*|L|}$ found by Algorithm 1. This contradiction completes the proof of Theorem 1.

Theorem 2. *The solutions $x^{*0}, \dots, x^{*|L|}$ obtained using Algorithm 1 are Pareto optimal in the problem $Z_2(W(x^1, x^2))$.*

Proof. Assume on the contrary the existence of a solution $x^{*q'}$, $q' \in \{0, \dots, L\}$, that is not Pareto optimal in the problem $Z_2(W(x^1, x^2))$. Then, according to Proposition 2, there is a Pareto optimal solution x' of the problem $Z_2(W(x^1, x^2))$ dominating $x^{*q'}$:

$$C(x') < C(x^{*q'}) \quad \text{and} \quad D(x') < D(x^{*q'}).$$

However, due to Theorem 1, such a solution x' does not exist. This contradiction completes the proof of Theorem 2.

Proposition 6. *Algorithm 1 does not necessarily yield the entire set of Pareto optimal solutions of the problem $Z_2(W(x^1, x^2))$.*

Proof. We provide a corresponding numerical example. Let $n = 4$ and

$$x_{ijk}^1 = \begin{cases} 1 & \text{if } (i, j, k) \in \{(1, 1, 1), (2, 2, 2), (3, 3, 3), (4, 4, 4)\} \\ 0 & \text{otherwise,} \end{cases}$$

$$x_{ijk}^2 = \begin{cases} 1 & \text{if } (i, j, k) \in \{(1, 1, 2), (2, 2, 1), (3, 3, 4), (4, 4, 3)\} \\ 0 & \text{otherwise.} \end{cases}$$

In addition, let

$$c_{ijk} = \begin{cases} 0 & \text{if } (i, j, k) \in \{(1, 1, 1), (2, 2, 2), (3, 3, 3), (4, 4, 4), (2, 2, 1), (4, 4, 3)\} \\ 1 & \text{if } (i, j, k) = (1, 1, 2) \\ 2 & \text{if } (i, j, k) = (3, 3, 4) \\ 5 & \text{otherwise,} \end{cases}$$

$$d_{ijk} = \begin{cases} 0 & \text{if } (i, j, k) \in \{(2, 2, 2), (4, 4, 4), (1, 1, 2), (2, 2, 1), (3, 3, 4), (4, 4, 3)\} \\ 1 & \text{if } (i, j, k) = (1, 1, 1) \\ 2 & \text{if } (i, j, k) = (3, 3, 3) \\ 5 & \text{otherwise.} \end{cases}$$

Then there exist four feasible solutions of the combination problem:

$$x_{ijk}^{*1} = \begin{cases} 1 & \text{if } (i, j, k) \in \{(1, 1, 1), (2, 2, 2), (3, 3, 3), (4, 4, 4)\} \\ 0 & \text{otherwise,} \end{cases}$$

where $C(x^{*1}) = 0$ and $D(x^{*1}) = 3$;

$$x_{ijk}^{*2} = \begin{cases} 1 & \text{if } (i, j, k) \in \{(1, 1, 2), (2, 2, 1), (3, 3, 4), (4, 4, 3)\} \\ 0 & \text{otherwise,} \end{cases}$$

where $C(x^{*2}) = 3$ and $D(x^{*2}) = 0$;

$$x_{ijk}^{*3} = \begin{cases} 1 & \text{if } (i, j, k) \in \{(1, 1, 2), (2, 2, 1), (3, 3, 3), (4, 4, 4)\} \\ 0 & \text{otherwise,} \end{cases}$$

where $C(x^{*3}) = 1$ and $D(x^{*3}) = 2$; and finally,

$$x_{ijk}^{*4} = \begin{cases} 1 & \text{if } (i, j, k) \in \{(1, 1, 1), (2, 2, 2), (3, 3, 4), (4, 4, 3)\} \\ 0 & \text{otherwise,} \end{cases}$$

where $C(x^{*4}) = 2$ and $D(x^{*4}) = 1$.

Each of the four solutions constructed is Pareto optimal in the problem $Z_2(W(x^1, x^2))$. Note that $q = 2$ and $|L| = 2$; hence, Algorithm 1 will find only three solutions. The proof of this proposition is complete.

Thus, Algorithm 1 constructs Pareto optimal solutions of the problem $Z_2(W(x^1, x^2))$ without any guarantee of obtaining the entire Pareto set.

Theorem 3. *Algorithm 1 requires $O(n^2)$ computational operations.*

Proof. Let the input data of Algorithm 1 be feasible solutions x^1 and x^2 represented as a collection of triples (i, j, k) and costs c_{ijk} for which the corresponding variables are 1. According to (1)–(4), the number of such triples is n for each feasible solution. Step 1 of the algorithm is to construct a graph $G = (V, A)$, where $|V| = O(n)$ and $|A| = O(n)$. At Step 2 the graph G is partitioned into connected components. This partition is obtained using the width-first traversal of the graph, which requires $O(|V| + |A|) = O(n)$ computational operations. Step 3 serves to determine the corresponding connected components for the input triples of the algorithm; this step requires $O(n)$ computational operations. At Step 4, for each connected component, the algorithm determines from which solution the triples will be taken to the first Pareto optimal solution; this step requires $O(n)$ computational operations. Step 5 is to construct the first Pareto optimal solution; since the solution can be represented as a collection of n triples (i, j, k) , this step requires $O(n)$ computational operations. At Step 6 only the connected components matching the criterion are selected from all such components, which requires $O(n)$ computational operations. Step 7 serves to sort the subset of the connected components, which requires $O(n \log(n))$ computational operations. Finally, at Step 8, $O(n)$ solutions are constructed, each in $O(n)$ computational operations. Thus, Algorithm 1 requires $O(n^2)$ computational operations. The proof of the theorem is complete.

4. COMPUTATIONAL EXPERIMENTS

Algorithm 1 yields a subset of the Pareto set in the problem $Z_2(W(x^1, x^2))$. Let us apply this algorithm to develop heuristic approximation methods for the Pareto set of the original problem Z_2 . The effectiveness of the heuristic algorithm will be assessed via computational experiments.

Experiment 1. Construct the matrices c_{ijk} and d_{ijk} in the following way: for each index $i \in I \cup J \cup K$, generate a random point p on the XY plane so that p_x and p_y are integer and uniformly distributed on the closed interval $[0, 2^{32} - 1]$. Then $c_{ijk} = \text{dist}(i, j) + \text{dist}(j, k) + \text{dist}(i, k)$, where $\text{dist}(a, b)$ indicates the Manhattan distance between points a and b . Determine d_{ijk} by analogy.

A local optimization procedure for a feasible solution x of the problem Z_2 includes several steps as follows.

Step 1. Choose a random number $a \in [0, 1]$ equiprobably.

Step 2. Construct the three-index cost matrix $e_{ijk} = ac_{ijk} + (1 - a)d_{ijk}$.

Step 3. Apply the local optimization procedure proposed in [15] to the solution x of the original problem, but with the criterion $\sum_{i \in I} \sum_{j \in J} \sum_{k \in K} e_{ijk} x_{ijk} \rightarrow \min$.

We construct n random feasible solutions $x_1, \dots, x_n$ of the problem Z_2 , and apply the local optimization procedure to each of them until the solution stops changing. Let $x'_1, \dots, x'_n$ denote the resulting solutions. From these solutions, we select the nondominated ones. Let R be the Pareto curve approximation constructed from the latter solutions.

Next, we apply Algorithm 1 to the pairs of locally optimized solutions to solve the following problems:

$$Z_2(W(x'_1, x'_2)), \dots, Z_2(W(x'_{n-1}, x'_n)).$$

From all the solutions obtained, we choose the nondominated ones. Let Q be the Pareto curve approximation constructed from the latter solutions.

We compare the number of points not dominated by the corresponding approximation and differing by the criterion value from all points of this approximation. We introduce the number of points not dominated by the approximation A as follows: $B(A) = |\{x | x \text{ is a feasible solution of the problem } Z_2, \nexists x' \in A \text{ such that } C(x') \leq C(x) \text{ and } D(x') \leq D(x)\}|$. For each test, $B(R)$ and $B(Q)$ were computed by checking the entire set of feasible solutions for the nonexistence of a dominant or equal-criteria-value solution in the corresponding approximation (via complete enumeration). The tests were carried out on dimensions $n = 6, 7, 8$, with 50 tests for each dimension.

The results of the first computational experiment are given in Table 1. Clearly, the application of Algorithm 1 reduced the number of nondominated points by 6.57% on average.

Table 1

Problem dimension	$\frac{B(R) - B(Q)}{B(R)} 100\%$
6	2.14%
7	8.03%
8	9.54%

Experiment 2. Construct the matrices c_{ijk} and d_{ijk} in the following way: for each index $i \in I \cup J \cup K$, generate a random point p on the XY plane so that p_x and p_y are integer and uniformly distributed on the closed interval $[0, 2^{32} - 1]$. Then $c_{ijk} = \text{dist}(i, j) + \text{dist}(j, k) + \text{dist}(i, k)$, $d_{ijk} = -\max(\text{dist}(i, j), \text{dist}(j, k), \text{dist}(i, k))$, where $\text{dist}(a, b)$ indicates the Manhattan distance between points a and b . Determine d_{ijk} by analogy.

We construct n^3 random feasible solutions $x_1, \dots, x_{n^3}$ of the problem Z_2 , and apply one iteration of the local optimization procedure to each of them. Let $x'_1, x'_2, \dots, x'_{n^3}$ denote the resulting solutions. From these solutions, we select the nondominated ones. Let R be the Pareto curve approximation constructed from the latter solutions.

Next, we apply Algorithm 1 to the pairs of locally optimized solutions to solve the following problems:

$$Z_2(W(x'_1, x'_2)), \dots, Z_2(W(x'_{n^3-1}, x'_{n^3})).$$

From all the solutions obtained, we choose the nondominated ones. Let Q be the Pareto curve approximation constructed from the latter solutions.

We compare the number of points not dominated by the corresponding approximation and differing by the criterion value from all points of this approximation.

The tests were carried out on dimensions $n = 6, 7, 8$, with 50 tests for each dimension.

Table 2

Problem dimension	$\frac{B(R)-B(Q)}{B(R)}100\%$
6	11.44%
7	18.08%
8	11.71%

The results of the second computational experiment are given in Table 2. Clearly, the application of Algorithm 1 reduced the number of nondominated points by 13.74% on average.

5. CONCLUSIONS

This paper has proposed an algorithm for constructing Pareto optimal solutions in the problem of combining two feasible solutions of the two-criteria three-index assignment problem. The correctness of the algorithm has been proved, and a quadratic estimate of its complexity has been obtained. This algorithm can be an additional step in heuristic approaches for estimating the Pareto domain of the two-criteria three-index assignment problem. According to the results of computational experiments provided, post-processing by the proposed optimal combination algorithm improves the quality of approximate solutions.

REFERENCES

1. Spiessma, F.C.R., Multi Index Assignment Problems. Complexity, Approximation, Applications, in *Nonlinear Assignment Problems: Algorithms and Applications*, Pardalos, P.M. and Pitsoulis, L.S., Eds., Dordrecht: Kluwer Acad. Publishers, 2000, pp. 1–11.
2. Burkard, R., Dell'Amico, M., and Martello, S., *Assignment Problems*, PA: SIAM, 2012.
3. Kuroki, Y. and Matsui, T., An Approximation Algorithm for Multidimensional Assignment Problems Minimizing the Sum of Squared Errors, *Discret. Appl. Math.*, 2009, vol. 157, no. 9, pp. 2124–2135.
4. Poore, A.B., Multidimensional Assignment Problems Arising in Multitarget and Multisensor Tracking, in *Nonlinear Assignment Problems: Algorithms and Applications*, Pardalos, P.M. and Pitsoulis, L.S., Eds., Dordrecht: Kluwer Acad. Publishers, 2000, pp. 13–38.
5. Zhuang, Y., Zhou, Y., Hassini, E., Yuan, Y., and Hu, X., Rack Retrieval and Repositioning Optimization Problem in Robotic Mobile Fulfillment Systems, *Transportation Research Part E: Logistics and Transportation Review*, 2022, vol. 167, art. no. 102920.
6. Garey, M.R. and Johnson, D.S., *Computers and Intractability. A Guide to the Theory of NP-completeness*, San Francisco: Freeman, 1979.

7. Afraimovich, L.G., Multiindex Transportation Problems with 2-embedded Structure, *Autom. Remote Control*, 2013, vol. 74, no. 1, pp. 90–104.
8. Bandelt, H.J., Crama, Y., and Spieksma, F.C.R., Approximation Algorithms for Multidimensional Assignment Problems with Decomposable Costs, *Discret. Appl. Math.*, 1994, vol. 49, pp. 25–50.
9. Crama, Y. and Spieksma, F.C.R., Approximation Algorithms for Three-Dimensional Assignment Problems with Triangle Inequalities, *Eur. J. Oper. Res.*, 1992, vol. 60, pp. 273–279.
10. Burkard, R.E., Rudolf, R., and Woeginger, G.J., Three-Dimensional Axial Assignment Problems with Decomposable Cost Coefficients, *Discret. Appl. Math.*, 1996, vol. 65, pp. 123–139.
11. Spieksma, F. and Woeginger, G., Geometric Three-Dimensional Assignment Problems, *Eur. J. Oper. Res.*, 1996, vol. 91, pp. 611–618.
12. Ćustić, A., Klinz, B., and Woeginger, G.J., Geometric Versions of the Three-Dimensional Assignment Problem under General Norms, *Discret. Optim.*, 2015, vol. 18, pp. 38–55.
13. Balas, E. and Saltzman, M.J., An Algorithm for the Three-Index Assignment Problem, *Oper. Res.*, 1991, vol. 39, no. 1, pp. 150–161.
14. Natsu, S., Date, K., and Nagi, R., GPU-Accelerated Lagrangian Heuristic for Multidimensional Assignment Problems with Decomposable Costs, *Parallel Comput.*, 2020, vol. 97, art. no. 102666.
15. Huang, G. and Lim, A., A Hybrid Genetic Algorithm for the Three-Index Assignment Problem, *Eur. J. Oper. Res.*, 2006, vol. 172, pp. 249–257.
16. Kim, B.J., Hightower, W.L., Hahn, P.M., Zhu, Y.R., and Sun, L., Lower Bounds for the Axial Three-Index Assignment Problem, *Eur. J. Oper. Res.*, 2010, vol. 202, pp. 654–668.
17. Dichkovskaya, S.A. and Kravtsov, M.K., Investigation of Polynomial Algorithms for Solving the Multi-criteria Three-Index Planar Assignment Problem, *Comput. Math. and Math. Phys.*, 2007, vol. 47, no. 6, pp. 1029–1038.
18. Emelichev, V.A. and Perepelitsa, V.A., Complexity of Discrete Multicriteria Problems, *Discrete Math. Appl.*, 1994, vol. 4, no. 2, pp. 89–118.
19. Prilutskii, M.Kh., Multicriterial Multi-index Resource Scheduling Problems, *J. Comput. Syst. Sci. Int.*, 2007, vol. 46, no. 1, pp. 78–82.
20. Prilutskij, M.Kh., Multicriteria Distribution of a Homogeneous Resource in Hierarchical Systems, *Autom. Remote Control*, 1996, vol. 57, no. 2, part 2, pp. 266–271.
21. Afraimovich, L.G. and Emelin, M.D., Combining Solutions of the Axial Assignment Problem, *Autom. Remote Control*, 2021, vol. 82, no. 8, pp. 1418–1425.
22. Afraimovich, L.G. and Emelin, M.D., Heuristic Strategies for Combining Solutions of the Three-Index Axial Assignment Problem, *Autom. Remote Control*, 2021, vol. 82, no. 10, pp. 1635–1640.
23. Afraimovich, L.G. and Emelin, M.D., Complexity of Solutions Combination for the Three-Index Axial Assignment Problem, *Mathematics*, 2022, vol. 10, no. 7, art. no. 1062.
24. Afraimovich, L.G. and Emelin, M.D., Criteria Convolutions When Combining the Solutions of the Multicriteria Axial Assignment Problem, *Autom. Remote Control*, 2024, vol. 85, no. 8, pp. 809–818.

This paper was recommended for publication by A.A. Lazarev, a member of the Editorial Board

Statistical Study of the Quality of the Cycle Merging Algorithm for Solving the Traveling Salesman Problem at Minimum

Yu. F. Leonova^{*,a} and A. V. Panuykov^{*,b}

^{*}South Ural State University, Chelyabinsk, Russia

e-mail: ^ayuliya.igosheva@gmail.com, ^bpaniukovav@susu.ru

Received March 11, 2024

Revised February 28, 2025

Accepted March 14, 2025

Abstract—The traveling salesman problem is one of the most widely studied problems in the field of combinatorial optimization. However, the investigation of new approaches and the improvement of existing methods remain a significant area of research. This paper presents an analysis of the quality of the cycle merging algorithm used to solve the minimum traveling salesman problem. The results of a computational experiment on five families of problems are provided, and the accuracy and time complexity of the algorithm are analysed. For symmetric instances, a regression model is constructed to describe the dependence of the relative error estimate on the number of vertices. It is shown that a polynomial model provides the best approximation of the obtained data and satisfies key statistical assumptions. The results allow us to evaluate the error growth patterns and to justify the applicability of algorithms to large-scale instances of the traveling salesman problem.

Keywords: traveling salesman problem, heuristic algorithm, asymptotic accuracy of the algorithm, statistical study

DOI: 10.31857/S0005117925050067

1. INTRODUCTION

The traveling salesman problem is a classical combinatorial optimization problem and remains one of the most important and topical problems in the field of applied mathematics and computer science. Its importance is manifested in many practical spheres, such as logistics, transportation, production, route planning and others. The minimum traveling salesman problem consists in finding the Hamiltonian cycle with minimum sum of weights of edges in a complete graph.

In recent decades, a variety of methods for solving the traveling salesman problem has been widely studied. These include exact algorithms, approximations, and heuristic techniques [1, 2]. However, with the advent of new computational technologies and the constant increase in the amount of data, there is a need to develop more efficient and scalable algorithms to solve this problem.

The traveling salesman problem is NP-hard [3]. Therefore, the development of polynomial approximate algorithms is relevant for its solution.

This paper presents an empirical study and statistical analysis of the accuracy and time complexity of a heuristic algorithm for solving the traveling salesman problem — the cycle merging algorithm. Section 2 characterizes the accuracy and running time estimates of popular efficient algorithms for solving the minimum traveling salesman problem. Section 3 gives a brief description of the algorithm under study. Section 4 describes the problem generation and the computational

experiment. The results of the computational experiment on random input data are presented in Section 5: data on the accuracy and solution time of the problems are given. Section 6 provides a statistical analysis of the accuracy of the cycle merging algorithm. Section 7 summarizes the main conclusions of the experimental results.

2. STATE OF THE PROBLEM OVERVIEW

Heuristic and approximate algorithms such as nearest neighbor algorithm, greedy algorithm, nearest insertion and others are widely used in practice due to their simplicity and efficiency. They often show good results for small datasets, but may not provide the optimal solution for high dimensional problems.

Table 1 presents priori estimates of the accuracy and time of some heuristic algorithms [4] for the minimum traveling salesman problem. The upper bounds for the algorithms are calculated as the ratio $f(s)/f(s_O)$, where $f(s)$ is the obtained tour length and $f(s_O)$ is the optimal tour length.

Table 1. Upper bound of accuracy estimation and running time of the algorithms

Algorithm	Upper bound	Operation time
Immediate neighbor [5]	$0.5 [\log_2 N + 1]$	$O(N^2)$
Immediate double-ended neighbor [6]		$O(N^2) \log N$
Greedy [5]		
Nearest addition [6]	$2 - 2/N$	$O(N^2)$
Nearest insert [6, 7]		$O(N^2) \log N$
Cheapest insert [6, 7]		
Farthest insert [6, 7]		$O(N^2)$
Arbitrary insertion [7]		
Inserting the nearest segment		
Dual minimal spatial tree [8]		
Dual minimal spatial tree modified		
Christofides [9]	$3/2 - 1/N$	$O(N^3)$
Mura curve [10]	$\log N$	$O(N \log N)$
Serpinsky curve [11]		
2-Opt [12]	≈ 2	$O(N^2)$

The above algorithms for solving the traveling salesman problem have different advantages and disadvantages depending on the characteristics of the input data and the required accuracy of the result.

The following sections will analyze the accuracy and time complexity of the cycle merging algorithm.

3. THE CYCLE MERGING ALGORITHM FOR SOLVING THE TRAVELING SALESMAN PROBLEM

The authors of the study [13] investigated a heuristic algorithm for solving the traveling salesman problem — Cycle Merging Algorithm (CMA). Here is a brief description of the algorithm.

The first step of the algorithm consists of finding an extreme weight in a given graph of a 2-regular subgraph, i.e. covering this graph with cycles. This construction is commonly referred to as a 2-factor of minimum weight. The problem of finding a 2-factor of minimum weight can be reduced to an assignment problem for which exact polynomial algorithms are known.

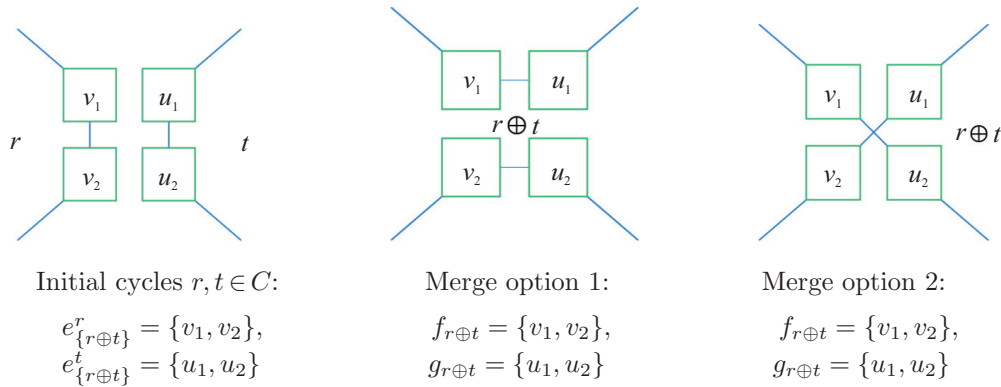


Fig. 1. Cycle merging options.

In the second phase, the condition of cycle singularity of the obtained solution is checked. If the 2-factor is represented by a single cycle, then this cycle is the solution of the problem. If the 2-factor is represented by several cycles, then different cycles r and t are searched in pairs. In each cycle of the selected pair, one edge is searched.

Let $e_{\{r \oplus t\}}^r = [v_1, v_2]$ and $e_{\{r \oplus t\}}^t = [u_1, u_2]$ be chosen, for them two pairs of conjugate edges are found ($f = [v_1, u_1]$, $g = [v_2, u_2]$) and ($f = [v_1, u_2]$, $g = [v_2, u_1]$), used to connect the cycles (Fig. 1). We search for a set $r, t, e_{\{r \oplus t\}}^r, e_{\{r \oplus t\}}^t, f, g$ of the listed elements such that the connected cycle has minimum weight.

The found pair of cycles is then replaced by the joined cycle with minimum cost. The algorithm terminates when the current 2-factor contains only one cycle.

A theorem on the computational complexity of the algorithm is formulated and proved for the described algorithm.

Theorem 1. The computational complexity of the cycle merging algorithm does not exceed $O(|V|^3)$.

A more detailed description of the cycle merging algorithm and a proof of the computational complexity theorem are given in [13].

For the metric maximal traveling salesman problem, a theoretical accuracy bound was previously established for this algorithm [13], which allows us to rigorously evaluate its quality in this class of problems. However, there are no such theoretical bounds for the minimum traveling salesman problem, so it is of interest to conduct an empirical study of the accuracy of the algorithm.

4. DESCRIPTION OF TEST PROBLEMS

To evaluate the quality of the cycle merging algorithm for the traveling salesman problem at minimum, a computational experiment was conducted on problem sets from the TSPLIB library [14], as well as on randomly generated instances with different characteristics of the cost matrix.

The performance of the algorithm was evaluated on the following families of the traveling salesman problem:

1. Asymmetric instances of the traveling salesman problem from TSPLIB. TSPLIB contains real and artificially generated problems that are widely used in heuristic and exact algorithms. The asymmetric cost of traveling between vertices makes the solution more complex, since the inherent symmetry inherent in many classical methods is broken.

2. Euclidean instances of the traveling salesman problem from TSPLIB ($n \leq 3000$). This data type models geometric versions of the problem, where vertices correspond to points in the plane and

the costs of transitions between them are determined by the Euclidean distance. Such problems are commonly found in logistics, navigation, and network planning.

3. Asymmetric cost matrices $W = [w(i, j)]$, where $w(i, j)$ are random numbers from $\{0, 1, 2, \dots, 10^5\}$. The uniform distribution of values in a fixed interval models random systems with independent transition weights. This allows to test the algorithm under full uncertainty of the structure value.

4. Asymmetric cost matrices where $w(i, j)$ are random numbers from $\{0, 1, 2, \dots, i \times j\}$. In this family, the range of possible costs increases as the vertex indices increase, which models problems where the communication between vertices becomes more expensive or complex as their numbers increase. This type of matrix allows to analyze the behavior of the algorithm on problems with non-uniform complexity of links.

5. Symmetric cost matrices W where $w(i, j)$ are random numbers from $\{0, 1, 2, \dots, 10^5\}$ for $i < j$. In contrast to the asymmetric cases, the symmetric structure limits the set of possible solutions, but leads to an increase in the number of short cycles in the original coverage. This makes it difficult to combine cycles into a single route and can negatively affect the accuracy of the heuristic algorithm.

6. Symmetric matrices where $w(i, j)$ are random numbers from $\{0, 1, 2, \dots, i \times j\}$ for $i < j$. Similar to the asymmetric case, but taking into account the symmetry of the weights. This type of problem models cases where distant vertices are associated with higher cost of transitions. However, the presence of many short cycles in the early stages of the solution makes the problem computationally complex, and may worsen the accuracy of an approximate solution.

7. Sloped plane instances of the problem where the arc weight is defined as

$$w(i, j) = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2} - \max(0, y_i - y_j) + 2 \max(0, y_j - y_i),$$

with vertex coordinates generated independently and uniformly on the interval $\{0, 1, 2, \dots, 10^5\}$. This model simulates a situation in which moving vertically involves additional costs, such as traveling on mountain roads. The difference in the cost of upward and downward movement makes the problem asymmetric and complicates its solution.

For families 3–6, the number of vertices n was varied from 100 to 3000 by increments of 100. All results are the average of 100 trials each.

The experiment uses instances that are generated in both a random and determined way. The number and variety of families used allows us to check the reliability of the algorithm under test.

5. RESULTS OF COMPUTATIONAL EXPERIMENT FOR THE TRAVELING SALESMAN PROBLEM AT MINIMUM

For the first and the second family of problem instances (asymmetric and euclidean problems from TSPLIB), the results are given in [15]. The analysis of accuracy and computational efficiency of the cycle merging algorithm for the other families are considered within the framework of this paper.

5.1. Analysis of Relative Error Estimation

Figure 2 shows the relative error estimates obtained in the course of the numerical experiment for different families of traveling salesman problems. The elements of the cost matrix of these problems were randomly generated according to the generation rules described in Section 4.

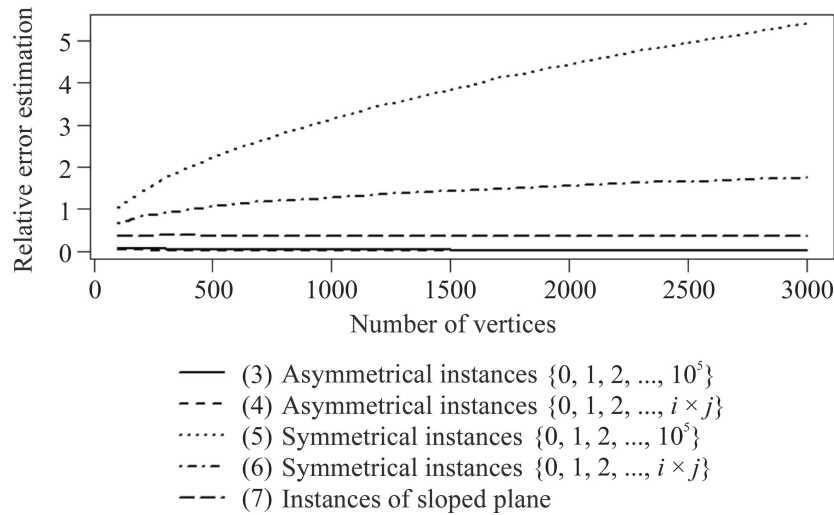


Fig. 2. Relative error estimation of the cycle merging algorithm for different families of the traveling salesman problem.

The analysis of the results shows differences in the behavior of the relative error estimation depending on the family of the considered instances.

For asymmetric instances of the traveling salesman problem with uniformly distributed elements of the weight matrix from the set $\{0, 1, 2, \dots, 10^5\}$ (family 3) relative error decreases with increasing number of vertices. At $n = 100$ its average value is 9.15%, and at $n = 3000$ it decreases to 1.97%. Analogous tendency is observed for family 4, but the error value in average remains lower, decreasing from 5.12% at $n = 100$ to 1.06% at $n = 3000$.

For symmetric instances of the traveling salesman problem, represented by families 5 and 6, a different behavior is observed. The relative error in both cases increases with increasing number of vertices. Thus, for family 5 it increases from 103% at $n = 100$ to 541% at $n = 3000$. A similar increase is characteristic for the family 6, with the error increasing from 65 to 175% for the same values of n .

For the sloped plane instances (family 7), the estimate of the relative error remains relatively stable over the entire interval of n values, fluctuating between 36.6–37.7%. This result indicates that the accuracy of the algorithm is less dependent on the dimensionality of the problem in this family of instances.

Thus, the analysis shows that:

- for asymmetric instances of the traveling salesman problem, the estimate of the relative error decreases with increasing number of vertices;
- for symmetric instances of the traveling salesman problem, the opposite trend is observed — the error increases with increasing problem dimensionality;
- the family of sloped plane demonstrates stable values of the relative error, insignificantly changing with increasing number of vertices.

5.2. Analyzing the Solution Time

Figure 3 presents the average running time of the cycle merging algorithm at solving one problem for different families of instances of the traveling salesman problem.

The analysis of time characteristics shows different behavior of time growth of the solution depending on the family of considered instances.

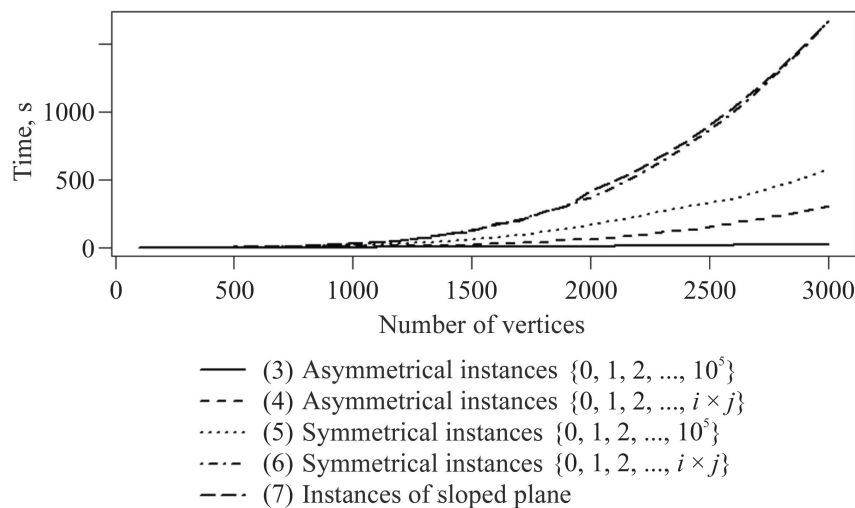


Fig. 3. Solution time per instance of the traveling salesman problem using the cycle merging algorithm for various problem families.

For asymmetric instances of the traveling salesman problem with uniformly distributed elements of the weight matrix from the set $\{0, 1, 2, \dots, 10^5\}$ (family 3) solution time increases from 0.105 s at $n = 100$ to 259.88 s at $n = 3000$. For family 4 ($W = [w(i, j)]$, where $w(i, j)$ are uniformly distributed random numbers of $\{0, 1, 2, \dots, i \times j\}$) the solution time grows faster, starting from 0.14 s at $n = 100$ and reaching 3057.04 s at $n = 3000$. This indicates the significant influence of structure of the cost matrix on the computational complexity of the algorithm.

For symmetric instances of the traveling salesman problem, a faster growth of the solution time is observed. In particular, for family 5 ($W = [w(i, j)]$, where $w(i, j)$ are uniformly distributed random numbers from $\{0, 1, 2, \dots, 10^5\}$) it increases from 0.114 s when $n = 100$ to 5760.3 s when $n = 3000$. For family 6 ($W = [w(i, j)]$, where $w(i, j)$ are uniformly distributed random numbers from $\{0, 1, 2, \dots, i \times j\}$) this growth of is even more pronounced: from 0.27 s to 16 669.63 s over the same interval of n values. Thus, the specificity of the structure of weights influences not only the accuracy, but also the computational complexity of the algorithm.

For the instances of sloped plane (family 7), a significant increase in the solution time is also observed. At $n = 100$ the average time is 0.279 s, and at $n = 3000$ it reaches 16 694.93 s, which is similar to the indicators for the sixth family of problems and significantly exceeds the data for all other families of problems. This indicates the complexity of processing such instances by the cycle merging algorithm.

In general, the analysis of temporal characteristics shows that:

- asymmetric instances of the traveling salesman problem are solved faster than symmetric ones, but the structure of the cost matrix significantly affects the growth of solution time;
- for symmetric instances there is an accelerated growth of the running time of the algorithm, especially for family 6;
- instances of sloped plane show the highest values of the solution time, which may be related to the peculiarities of the geometric structure of the problems.

The complexity of the symmetric non-Euclidean traveling salesman problem for algorithms starting from the assignment problem is due to the low accuracy of the lower bound obtained using this method. In particular, for the vast majority of symmetric instances, the assignment problem leads to a cycle coverage containing a large number of short cycles (Table 2). This significantly complicates the process of combining them into a single route, increasing both the computational cost and the resulting relative error.

Table 2. Ratio of the number of initial cycles to the number of problem vertices

Number of vertices	Asymmetrical instances	Symmetrical instances
100	0.0382	0.4920
200	0.0225	0.4768
300	0.0162	0.4771
400	0.0139	0.4892
500	0.0118	0.4848
600	0.0101	0.4867
700	0.0086	0.4896
800	0.0076	0.4902
900	0.0071	0.4890
1000	0.0070	0.4878

As can be seen from the table, for asymmetric instances the ratio of the number of initial cycles to the number of vertices decreases significantly with increasing dimensionality of problem, decreasing from 0.0382 (100 vertices) to 0.0070 (1000 vertices). This indicates a tendency to form longer cycles, which contributes to a more efficient connection of routes at subsequent stages of the algorithm.

In contrast, for symmetric instances this ratio remains practically unchanged, fluctuating in the range 0.4768–0.4920. This indicates that during the formation of the 2-factor for this family of problems, a lot of short subcycles are formed, which makes it difficult to combine them into the final route.

The growth of the solution time for symmetric instances is explained by the fact that the process of connecting a large number of short cycles requires a significantly larger number of iterations, which leads to a complication of the algorithm structure and an increase of its computational complexity. As shown in Fig. 3, the running time of the algorithm for connecting cycles for symmetric instances significantly exceeds the similar values for asymmetric problems.

Thus, in order to improve the accuracy and reduce the time to solve the symmetric non-Euclidean traveling salesman problem, further study of specialized heuristics and approximate algorithms adapted to work with this family of problems is required.

6. STATISTICAL ANALYSIS OF THE ACCURACY OF THE CYCLE MERGING ALGORITHM FOR SYMMETRIC TRAVELING SALESMAN PROBLEM

As part of the investigation of the accuracy of the cycle merging algorithm for the symmetric traveling salesman problem at minimum, a statistical analysis of empirical data obtained for instances of the problem with a weight matrix whose elements are generated uniformly from the set $\{0, 1, 2, \dots, i \times j\}$ (the fifth family of problem instances from Section 4) was carried. Since there are no theoretical bounds on the accuracy of the algorithm for the symmetric minimum comparison problem, we used regression analysis methods to predict the dependence of the relative error on the number of vertices.

Three possible regression models were considered: steppe, second-order polynomial, and logarithmic. These models were chosen on the basis of observations on the nature of the relative error growth with increasing number of vertices.

6.1. Selection of Regression Model

Exploratory analysis is a necessary step to test the assumptions [16] associated with regression models and to select the most appropriate model to approximate the data. As part of this analysis phase, visual inspections were performed and statistical characteristics were calculated for each model to compare their performance and accuracy.

Visual methods such as scatter dot plot and dot plot matrix were used to analyze the relationship between the number of vertices and relative error. Histograms and tests for normality of the distribution of residuals were also used. Statistical tests, such as the coefficient of determination R^2 , adjusted R^2 , and the Durbin–Watson and Breusch–Pagan tests, were used to assess the model's fit to the statistical assumptions.

6.1.1. Power regression. The power model is of the form

$$y = ax^b. \quad (1)$$

After logarithmization, the equation takes a linear form:

$$\log y = \log a + b \cdot \log x. \quad (2)$$

The following results were obtained for this model:

- The coefficient of determination $R^2 = 0.9676$, adjusted $R^2 = 0.9665$, which indicates high quality of data approximation.
- The significance of the model coefficients is confirmed by the p -values, which are less than $2 \cdot 10^{-16}$.
- The Durbin–Watson test showed no significant autocorrelation of the residuals ($DW = 1.9527$, $p = 0.3711$).
- The Breusch–Pagan test revealed the presence of heteroscedasticity ($p = 1.24 \cdot 10^{-6}$), indicating instability in the variance of the residuals.
- The global test for compliance with the linear regression assumptions showed violation of the assumptions ($p = 2.81 \cdot 10^{-7}$).

Thus, although the power model has a high coefficient of determination and provides a good approximation, the identified problems with heteroscedasticity and violation of linear assumptions indicate its limitations.

6.1.2. Polynomial regression. The second-order polynomial model is

$$\log y = a + b \cdot \log x + c \cdot \log^2 x. \quad (3)$$

The statistical characteristics of this model are:

- $R^2 = 0.9999$, adjusted $R^2 = 0.9999$, which confirms the almost perfect fit of the model data.
- All model coefficients are statistically significant ($p < 0.001$).
- The Durbin–Watson test showed no autocorrelation ($DW = 2.2082$, $p = 0.5771$).
- The Breusch–Pagan test showed no heteroscedasticity ($p = 0.1035$).
- The global test showed small deviations from the assumptions ($p = 0.002$).

The second-order polynomial model showed a significantly better fit to the data compared to the power model, eliminating heteroskedasticity problems. However, the test for global fit to the linearity assumptions indicates possible deviations, which nevertheless do not have a significant impact on the accuracy of the model.

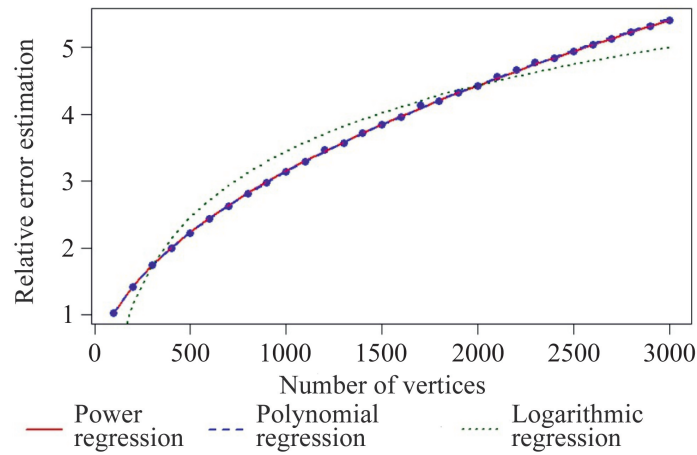


Fig. 4. Comparison of regression models for symmetric instances from the set $\{0, 1, 2, \dots, 10^5\}$.

6.1.3. Logarithmic regression. The logarithmic model is described by equation

$$y = a + b \cdot \ln x. \quad (4)$$

Statistical results:

- R^2 is slightly lower than that of the polynomial model, but remains high.
- The residuals meet the requirements of normality and homoscedasticity.
- The global test confirms satisfactory compliance with the prerequisites.

The logarithmic model, as well as the polynomial model, showed good correspondence with the data and regression assumptions with easily interpretable coefficients.

6.1.4. Model selection. Figure 4 shows plots of empirical and model values for different personal regression models.

The analysis showed that the power model has a high coefficient of determination, but violates the regression assumptions. The second-order polynomial model significantly improves the quality of approximation by eliminating heteroscedasticity. The third-order polynomial model improves the quality of predictions, but may lead to overcomplication. The logarithmic model describes the slowing growth well, but has limitations in extrapolation.

According to the test results and interpretation of the data, the second-order polynomial model and logarithmic regression are the most fitting. The polynomial model provided the best fit to the data, eliminating the major problems of power regression. The logarithmic model describes slowing growth well, but when extrapolated to values beyond the training set, it can produce bias because the logarithmic relationship assumes infinite growth at large x , which does not always fit real data. In the further analysis we will use the polynomial model of the second order as the most accurate and satisfying the key statistical requirements (Table 3).

Table 3. Global verification of polynomial regression assumptions

Test	Value	p -value
Global Stat	16.843	0.0021 (not satisfied)
Skewness	3.189	0.0742 (acceptable)
Kurtosis	2.321	0.1277 (acceptable)
Link Function	9.898	0.0017 (not satisfied)
Heteroskedasticity	1.436	0.2308 (acceptable)

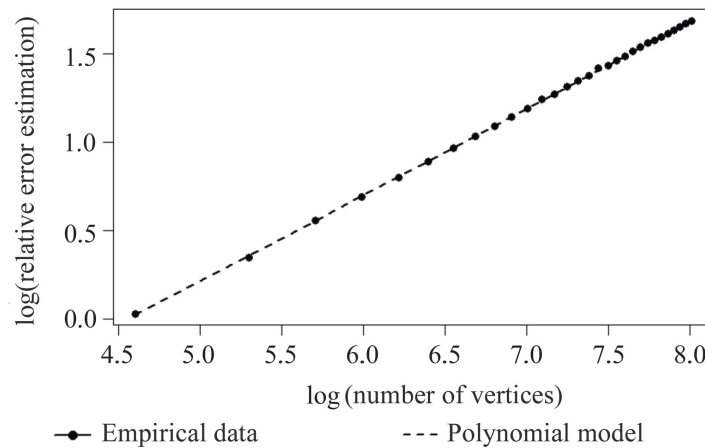


Fig. 5. Empirical and modeled values of logarithmically transformed data for polynomial regression on symmetric instances from the set $\{0, 1, 2, \dots, 10^5\}$.

The polynomial model on a logarithmic scale shows a good fit to the data, but small deviations from linearity may indicate its limitations in extrapolation. In general, the results support their applicability, although a possible bias should be considered when forecasting large values.

6.2. Evaluation and Interpretation of the Model

The constructed polynomial regression model describes the dependence of the average relative error on the number of vertices. The inclusion of a quadratic term made it possible to take into account the nonlinear nature of the error growth. The coefficient $b_1 = 0.4465$ at $\log n$ indicates an accelerating growth of the error with increasing dimensionality of the problem, and the positive value $b_2 = 0.0035$ at $(\log n)^2$ confirms a small acceleration of growth at small values of n . However, this effect is smoothed out at large problem dimensions, which indicates a tendency to stabilization of error.

Key statistical measures of the model:

- $R^2 = 0.9999$ (almost complete explanation of variance).
- F -statistic: 129 100 ($p < 2 \cdot 10^{-16}$), which confirms the significance of the model and high degree of fit to the data.
- Confidence intervals show that all model coefficients are statistically significant ($p < 0.01$).

Figure 5 presents a graphical comparison of empirical data and predicted values of the model.

Analysis of the global regression assumption test showed a violation of the link function ($p = 0.0017$). However, the absence of heteroscedasticity and autocorrelation of residuals allows us to consider the polynomial model suitable for describing the relationship, despite small deviations from the assumptions of linearity.

6.3. Prediction for Problems of High Dimensionality

On the basis of the constructed polynomial model, we extrapolated to predict the relative error for problems of up to 10 000 vertices (Fig. 6). The results show that at large n the tendency to the error growth remains, but the effect of acceleration due to the quadratic term becomes less pronounced. This confirms that the relative error tends to stabilize and its increase slows down.

Statistical analysis demonstrates that the proposed polynomial model describes well the dependence of the relative error on the dimensionality of the problem and allows us to make predictions

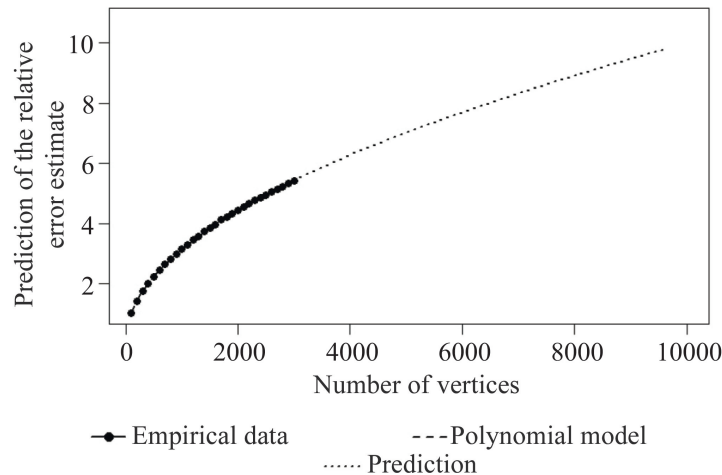


Fig. 6. Prediction of the relative error estimate for large-scale symmetric instances from the set $\{0, 1, 2, \dots, 10^5\}$.

for large n . Although the model retains high approximation accuracy, possible deviations in extrapolation indicate the need for further investigation of error behavior for problems of extremely large dimensions. Nevertheless, the obtained results confirm that the cycle merging algorithm demonstrates a stable character of change of error and can be applied to the solution of the symmetric salesman problem in the case of high dimensionality.

7. CONCLUSION

The article presents the results of an empirical study of the accuracy and time complexity of the cycle merging algorithm, as well as a statistical analysis of the accuracy of the algorithm for one of the considered families of the symmetric traveling salesman problem on the minimum.

The empirical analysis was performed on seven families of problems, of which the article presents results for five families of randomly generated instances.

The study shows that the solution quality and computational complexity of the algorithm depend on the structure of the cost matrix. For asymmetric instances the relative error estimate decreases to 1–2% at 3000 vertices, whereas for symmetric instances and sloped plane instances there is a gradual increase in relative error. The solution time for all families increases polynomially, with the solution time for symmetric and sloped plane instances being significantly higher than for the asymmetric instances.

The analysis shows that the relative error of the cycle merging algorithm for the symmetric traveling salesman problem demonstrates polynomial growth with increasing dimensionality of the problem. This is confirmed by statistical processing of experimental data, where the polynomial regression model provides the best fit to empirical observations.

In spite of the error growth, the proposed algorithm retains acceptable accuracy and polynomial running time, which makes it a promising tool for solving instances of the traveling salesman problem of large dimensionality. To further improve the quality of solutions, it is advisable to investigate hybrid approaches combining cycle merging algorithm with additional heuristics and local improvements.

REFERENCES

1. Zhang, C. and Sun, P., Heuristic methods for solving the traveling salesman problem (TSP): a comparative study, *Proc. IEEE 34th Annu. Int. Symp. on Personal, Indoor and Mobile Radio Commun. (PIMRC)*, 2023, pp. 1–6. <https://doi.org/10.1109/PIMRC56721.2023.10293957>

2. Toaza, B. and Esztergár-Kiss, D., A review of metaheuristic algorithms for solving TSP-based scheduling optimization problems, *Appl. Soft Comput.*, 2023, vol. 148, art. no. 110908. <https://doi.org/10.1016/j.asoc.2023.110908>
3. Reingold, E.M., Nievergelt, J., and Deo, N., *Combinatorial Algorithms: Theory and Practice*. Englewood Cliffs: Prentice Hall, 1977. <https://doi.org/10.2307/2987917>
4. Avdoshin, S.M. and Beresneva, E.N., Metric traveling salesman problem: experimental study of Pareto-optimal algorithms, *Proc. of ISP RAS*, 2017, vol. 29, no. 4, pp. 123–138. [https://doi.org/10.15514/ISPRAS-2017-29\(4\)-8](https://doi.org/10.15514/ISPRAS-2017-29(4)-8)
5. Flood, M.M., The traveling-salesman problem, *Oper. Res.*, 1956, vol. 4, no. 1, pp. 61–75. <https://doi.org/10.1287/opre.4.1.61>
6. Rosenkrantz, D.J., Stearns, R.E., and Lewis II, P.M., An analysis of several heuristics for the traveling salesman problem, *SIAM J. Comput.*, 1977, vol. 6, no. 3, pp. 563–581. <https://doi.org/10.1137/0206041>
7. Hahsler, M. and Hornik, K., TSP — Infrastructure for the traveling salesperson problem, *J. Statist. Softw.*, 2007, vol. 23, no. 2. <https://doi.org/10.18637/jss.v023.i02>
8. Kay, E. and Christofides, N., Graph theory: an algorithmic approach, *Oper. Res. Q.*, 1976, vol. 27, no. 4, pp. 1027. <https://doi.org/10.2307/3009122>
9. Christofides, N., Worst-case analysis of a new heuristic for the traveling salesman problem, Pittsburgh, PA: Graduate School of Industrial Administration, Carnegie Mellon Univ., 1976. <https://doi.org/10.1007/s43069-021-00101-z>
10. Buchin, K., Space-filling curves, *Organizing Point Sets: Space-Filling Curves, Delaunay Tessellations of Random Point Sets, and Flow Complexes*, Berlin: Free Univ. of Berlin, 2007, pp. 5–30.
11. Bartholdi, J.J., Platzman, L.K., Collins, R.L., and Warden, W.H., A minimal technology routing system for meals on wheels, *Interfaces*, 1983, vol. 13, no. 3, pp. 1–8. <https://doi.org/10.1287/inte.13.3.1>
12. Aarts, E.H.L. and Lenstra, J.K., *Local Search in Combinatorial Optimization*. Princeton, NJ: Princeton Univ. Press, 2003. <https://doi.org/10.1515/9780691187563>
13. Panyukov, A.V. and Leonova, Y.F., Cycle merging algorithm for the metric traveling salesman problem for maximum, *Bull. South Ural State Univ., Ser. Math. Model. Program. Comput. Software*, 2021, vol. 10, no. 4, pp. 26–36. <https://doi.org/10.14529/cmse210402>
14. Reinelt, G., TSPLIB—A traveling salesman problem library, *INFORMS J. Comput.*, 1991, vol. 3, no. 4, pp. 376–384. <https://doi.org/10.1287/ijoc.3.4.376>
15. Leonova, Y.F., Analysis of the cycle merging algorithm quality on TSPLIB instances, *Actual Problems of Applied Mathematics, Computer Science and Mechanics. Proc. Int. Sci. Conf.*, Voronezh, Dec. 13–15, 2021, Voronezh: VSU, 2022, pp. 573–579. (In Russian)
16. Heins, J., Bossek, J., Pohl, J., Seiler, M., Trautmann, H., and Kerschke, P., A study on the effects of normalized TSP features for automated algorithm selection, *Theor. Comput. Sci.*, 2023, vol. 940, pp. 123–145. <https://doi.org/10.1016/j.tcs.2023.01.045>

This paper was recommended for publication by A.A. Lazarev, a member of the Editorial Board

Relations between Average Shortest Path Length and Another Centralities in Graphs

M. A. Tuzhilin^{*,**}

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

^{**}Lomonosov Moscow State University, Moscow, Russia

e-mail: avarnavsky@hse.ru

Received December 16, 2024

Revised December 30, 2025

Accepted January 9, 2025

Abstract—Relations between average shortest path length and radiality, closeness, stress centralities and average clustering coefficient were obtained for simple connected graphs.

Keywords: networks, centralities, local and global properties of graphs, average shortest path length, Watts-Strogatz clustering coefficient

DOI: 10.31857/S0005117925050078

1. INTRODUCTION

The average shortest path length is one of the most important characteristics of real graphs (or networks). The first essential attempt for calculation this value was in 1969 year by two social psychologists Stanley Milgram and Jeffrey Travers (small-world experiment [1]). They found a phenomenon that any two people know each other by a sequence of 6 contacts. This theory is known as six degrees of separation theory. Further, this theory was proved by Duncan Watts and Steven Strogatz in [2]. It turns out, that many graphs in real applied works have the same property: they have small average shortest path length.

Another very important characteristic for real networks was found by Watts and Strogatz — big average clustering coefficient (or Watts and Strogatz clustering coefficient [2]). Graphs which satisfied these properties (small average shortest path length and big average clustering coefficient) are called small-world networks. In Social Networks science there are many other important characteristics for networks so-called centralities (for example, stress, closeness, betweenness, radiality centralities and so on). In this field they analyzed distributions of all these centralities to understand intrinsic characteristics of networks (see for example [3–7]), but the existence of relations between centralities is still not very well-known question. In these article we found relations between average shortest path length and radiality, closeness, stress centralities and also the connection between average shortest path length and average clustering coefficient for geodetic graphs.

2. MAIN DEFINITIONS

All subsequent definitions are given for a simple, connected, undirected graph G . Let's denote by

- $V(G)$ the set of vertices, $E(G)$ – the set of edges, $A = \{a_{ij}\}$ adjacency matrix of graph G ,
- neighborhood $N(v)$ – the induced subgraph in G on vertices which adjacent to the vertex v ,
- $N'(v)$ induced subgraph in G on vertices $= V(N(v)) \cup \{v\}$,

- $\bar{f}(x_1, \dots, x_k)$ for any function $f : V \times V \times \dots \times V \rightarrow \mathbb{R}$ – the restriction of this function on $N'(v)$ (for example $\bar{L}(x, y)$ will be the average shortest path between x and y with restriction to subgraph $N'(v)$),
- $d_i = \deg(v_i)$,
- $\text{dist}(s, t)$ – the shortest path length between s and t ,
- $n = |V(G)|$, $m = |E(G)|$,
- $X(i) = X(v_i)$ for any X – set or function corresponding to vertex v_i .

Let's give definitions of centralities.

(1) Diameter $\text{diam}(G) = \max_{s, t \in V(G)} \text{dist}(s, t)$.

(2) Average shortest path length

$$L(G) = \frac{1}{n(n-1)} \sum_{s, t \in V(G), s \neq t} \text{dist}(s, t).$$

(3) Local cluster coefficient

$$c_i = c(i) = \frac{\text{number of edges in } N(i)}{\text{maximum possible number of edges in } N(i)} = \frac{2|E(N(i))|}{d_i(d_i-1)}.$$

(4) Average clustering coefficient

$$C_{WS}(G) = \frac{1}{n} \sum_{i \in V(G)} c_i = \frac{1}{n} \sum_{i \in V(G)} \frac{2|E(N(i))|}{d_i(d_i-1)} = \frac{1}{n} \sum_{i \in V(G)} \frac{\sum_{j, k \in V(G)} a_{ij} a_{jk} a_{ki}}{d_i(d_i-1)}.$$

(5) Global clustering coefficient

$$C(G) = \frac{\text{number of closed triplets in } G}{\text{number of all triplets in } G} = \frac{\sum_{i, j, k \in V(G)} a_{ij} a_{jk} a_{ki}}{\sum_{i \in V(G)} d_i(d_i-1)}.$$

(6) Closeness centrality $\text{Clo}(v) = \frac{n-1}{\sum_{t \in V(G)} \text{dist}(v, t)}$.

(7) Radiality $\text{Rad}(v) = \frac{\sum_{t \in V(G), t \neq v} (\text{diam}(G)+1-\text{dist}(v, t))}{n-1}$.

(8) Stress $\text{Str}(i) = \sum_{s, t \in V(G), s \neq t \neq i} \sigma_{st}(i)$, where $\sigma_{st}(i)$ – is the total number of shortest paths from s to t which contains vertex i .

Note that all centralities are non-negative and $c_i, C_{WS}, \text{Clo}(v)$ are less or equal 1.

Also let's give the definition of geodetic graph:

Definition 1. If there exists the unique shortest path between any two vertices in G then the graph G is called **geodetic**.

This definition is equivalent to the condition then there is no even cycles in a graph.

3. MAIN RESULTS

Let's consider a induced subgraph $G' \subset G$. In general G' can be not connected graph. In this case let's define the average shortest path length for vertices of G' with relation to the distance dist in the ambient graph G . Let's call $L(N(i))$ the local average shortest path length for the vertex i .

Let's start with simple relations: let's proof the relation between local shortest path length and local clustering coefficient.

Lemma 1.

$$L(N(i)) = 2 - c_i.$$

Proof.

$$\begin{aligned}
 L(N(i)) &= \frac{1}{d_i(d_i - 1)} \sum_{s,t \in N(i), s \neq t} \text{dist}(s, t) \\
 &= \frac{1}{d_i(d_i - 1)} \sum_{(s,t) \in E(N(i))} \text{dist}(s, t) + \sum_{s,t \in N(i), (s,t) \notin E(N(i))} \text{dist}(s, t) \\
 &= \frac{1}{d_i(d_i - 1)} \left(2|E(N(i))| + \sum_{(s,i), (i,t) \in E(G), (s,t) \notin E(G)} \text{dist}(s, t) \right) \\
 &= \frac{1}{d_i(d_i - 1)} \left(2|E(N(i))| + 2(d_i(d_i - 1) - 2|E(N(i))|) \right) = 2 - c_i.
 \end{aligned}$$

Note that shortest paths for vertices in $N(i)$ are defined corresponding to whole graph G .

Averaging by i we obtain simple corollary about the relation between local shortest path length and average clustering coefficient.

Corollary 1.

$$C_{WS}(G) = 2 - \frac{1}{n} \sum_{i \in V(G)} L(N(i)).$$

Let's proof the relation between shortest path length in subgraph $N'(i)$ and shortest path length in $N(i)$.

Lemma 2.

$$L(N'(i)) = \frac{(d_i - 1)L(N(i)) + 2}{d_i + 1}.$$

Proof. By definition

$$L(N'(i)) = \frac{1}{(d_i + 1)d_i} \sum_{s,t \in V(N'(i)), s \neq t} \text{dist}(s, t) = \frac{d_i - 1}{d_i + 1} L(N(i)) + \frac{2}{d_i + 1}.$$

Let's proof the relation between shortest path length in a induced subgraph and shortest path length in ambient graph if induced subgraph is obtained from ambient graph by deleting one vertex.

Theorem 1. *Let a graph G is obtained from a connected simple graph G' by deleting one vertex and $|V(G)| = n$. Then*

$$L(G') \geq \frac{n}{n + 1} L(G),$$

where the average shortest path length $L(G)$ is defined in the ambient graph G' , if G is not connected.

Proof. Let's denote the deleted vertex by v . By the triangle inequality $\forall s, t \in V(G) : \text{dist}(s, v) + \text{dist}(v, t) \geq \text{dist}(s, t)$, here the equality holds if, there are no paths from s to t in G . Therefore,

$$\begin{aligned}
 \sum_{s,t \in V(G), s \neq t} (\text{dist}(s, v) + \text{dist}(v, t)) &\geq \sum_{s,t \in V(G), s \neq t} \text{dist}(s, t), \\
 \frac{2(n - 1)}{n(n - 1)} \sum_{t \in V(G)} \text{dist}(v, t) &\geq \frac{1}{n(n - 1)} \sum_{s,t \in V(G), s \neq t} \text{dist}(s, t), \\
 \frac{2}{n} \sum_{t \in V(G)} \text{dist}(v, t) &\geq L(G).
 \end{aligned}$$

Then,

$$\begin{aligned}
 L(G') &= \frac{1}{(n+1)n} \sum_{s,t \in V(G'), s \neq t} \text{dist}(s, t) \\
 &= \frac{1}{(n+1)n} \left(2 \sum_{t \in V(G)} \text{dist}(v, t) + \sum_{s,t \in V(G), s \neq t} \text{dist}(s, t) \right) \\
 &= \frac{1}{n+1} \frac{2}{n} \sum_{t \in V(G)} \text{dist}(v, t) + \frac{n-1}{n+1} L(G) \geq \frac{n}{n+1} L(G).
 \end{aligned}$$

Note that if G consists of n isolated vertices then the equality holds.

We obtain corollaries.

Corollary 2. *Let a graph G is obtained from a connected simple graph $G' \subset H$ by deleting one vertex, $|V(G)| = n$ and H is connected and simple. Then*

$$L(G') \geq \frac{n}{n+1} L(G),$$

where the average shortest path lengths $L(G)$ and $L(G')$ are defined in the ambient graph H , if corresponded graphs are not connected.

Proof. The proof is the same as for the previous theorem and also if G consists of n isolated vertices then the equality holds.

Corollary 3.

$$L(N'(i)) \geq \frac{d_i}{d_i + 1} L(N(i)).$$

Let's proof the relation between shortest path length in a induced subgraph and shortest path length in ambient graph.

Theorem 2. *Let's $G' \subset G$ be induced subgraph and $|V(G)| = n, |V(G')| = n + k$. Then*

$$L(G') \geq \frac{n}{n+k} L(G),$$

where the average shortest path length $L(G)$ is defined in the ambient graph G' , if G is not connected.

Proof. Let's construct the graph G' from G by adding sequentially k vertices and corresponded edges. Let's first sequentially add vertices adjacent to vertices of the graph G . Adding these vertices one by one we obtain a sequence of graphs $G_1, G_2, \dots, G_p$, where $|V(G_i)| = n + i$. Further, let's add in the same way vertices adjacent to vertices of the graph G_p and so on. In the end we obtain the graph G' . By previous corollary

$$\begin{aligned}
 L(G') &\geq \frac{n+k-1}{n+k} L(G_{k-1}) \geq \frac{n+k-1}{n+k} \frac{n-k-2}{n-k-1} L(G_{k-2}) \\
 &= \frac{n-k-2}{n+k} L(G_{k-2}) \geq \dots \geq \frac{n}{n+k} L(G).
 \end{aligned}$$

Let's proof the relation between average closeness centrality and average shortest path length.

Lemma 3.

$$L(G) \geq \frac{n}{\sum_{v \in V(G)} \text{Clo}(v)}.$$

Proof. By the inequality of harmonic mean and arithmetic mean

$$\frac{1}{n} \sum_{v \in V(G)} \text{Clo}(v) = \frac{1}{n} \sum_{v \in V(G)} \frac{n-1}{\sum_{t \in V(G)} \text{dist}(v, t)} \geq \frac{n(n-1)}{\sum_{v, t \in V(G)} \text{dist}(v, t)} = \frac{1}{L(G)}.$$

Note that an equality holds when all average shortest path lengths from any vertex to all remaining vertices are equal.

Let's proof the relation between average shortest path length and average radiality.

Lemma 4.

$$L(G) = \text{diam}(G) + 1 - \frac{1}{n} \sum_{v \in V(G)} \text{Rad}(v).$$

Proof. The proof holds from definition

$$\begin{aligned} \frac{1}{n} \sum_{v \in V(G)} \text{Rad}(v) &= \frac{1}{n} \sum_{v \in V(G)} \frac{(n-1)(\text{diam}(G) + 1) - \sum_{t \in V(G), t \neq v} \text{dist}(v, t)}{n-1} \\ &= \text{diam}(G) + 1 - L(G). \end{aligned}$$

Let's prove a theorem about a connection between the average stress centrality and average shortest path length for geodetic graphs.

Theorem 3. *If G is geodetic, then*

$$L(G) = 1 + \frac{1}{n(n-1)} \sum_{i \in V(G)} \text{Str}(i).$$

Proof. Let's define

$$\chi_{st}(i) = \begin{cases} 1, & \text{if } i \neq s \neq t - \text{is the vertex of the shortest path between } s \text{ and } t, \\ 0, & \text{otherwise.} \end{cases}$$

Thus, $\text{Str}(i) = \sum_{s, t \in V(G)} \chi_{st}(i)$. If there exists the unique shortest path between any two vertices in G , then $\text{dist}(s, t) = \sum_{i \in V(G)} \chi_{st}(i) + 1$ (otherwise, $\text{dist}(s, t) \leq \sum_{i \in V(G)} \chi_{st}(i) + 1$). Therefore, for any i

$$\begin{aligned} \sum_{s, t \in V(G)} \text{dist}(s, t) &= 2|E| + \sum_{s, t \in V(G), \text{dist}(s, t) \geq 2} \text{dist}(s, t) \\ &= 2|E| + \sum_{s, t \in V(G), \text{dist}(s, t) \geq 2} \left(\sum_{i \in V(G)} \chi_{st}(i) + 1 \right) \\ &= 2|E| + \sum_{i \in V(G)} \text{Str}(i) + n(n-1) - 2|E| = \sum_{i \in V(G)} \text{Str}(i) + n(n-1). \end{aligned}$$

Corollary 4. *For any simple connected G*

$$L(G) \leq 1 + \frac{1}{n(n-1)} \sum_{i \in V(G)} \text{Str}(i).$$

First let's proof theorem about a connection between average clustering coefficient and stress centrality and we will use it for a connection between the average shortest path length and the average local clustering coefficient further.

Theorem 4. *If there is no pendant vertices in graph, then*

$$C_{WS}(G) \geq \frac{1}{n} \sum_{i \in V(G)} \left(1 - \frac{\text{Str}(i)}{d_i(d_i - 1)} \right).$$

Proof. Note that $\forall j, k \in N(i) : (j, k) \notin E(N(i))$ the shortest path between j and k is $j \rightarrow i \rightarrow k$. Therefore,

$$\begin{aligned} \text{Str}(i) &\geq 2 \left(\frac{d_i(d_i - 1)}{2} - |E(N(i))| \right), \\ \frac{1}{d_i(d_i - 1)} \text{Str}(i) &\geq 1 - c_i. \end{aligned}$$

Averaging by i

$$C_{WS}(G) \geq \frac{1}{n} \sum_{i \in V(G)} \left(1 - \frac{\text{Str}(i)}{d_i(d_i - 1)} \right).$$

Note that for $\text{diam}(G) = 2$ holds an equality.

Corollary 5. *For any simple connected graph G*

$$C_{WS}(G) \geq \frac{1}{n} \sum_{i \in V(G)} \left(1 - \frac{\text{Str}(i)}{d_i(d_i - 1)} \right) - \frac{\text{number of pendant vertices}}{n}.$$

Proof. Let's define $\frac{\text{Str}(i)}{d_i(d_i - 1)}$ as 0 for every pendant vertex i , then in the inequality $\frac{1}{d_i(d_i - 1)} \text{Str}(i) \geq 1 - c_i$ in the right side will be 1 and in the left side 0, thus if we add in the left side 1 for every pendant vertex, will be right equality.

Now let's prove a theorem about a connection between the average shortest path length and the average local clustering coefficient for geodesic graphs.

Theorem 5. *If G is geodesic and $\forall i, j \in V(G)$ hold, if $d_i \leq d_j$ then $\text{Str}(i) \leq \text{Str}(j)$, then*

$$1 - C_{WS}(G) \leq \frac{1}{n} \sum_{i \in V(G)} \frac{(L(G) - 1)(n - 1)}{d_i(d_i - 1)} + \frac{\text{number of pendant vertices}}{n}.$$

Proof. Let's re-numerate vertices such that $\forall i \leq j : d_i \leq d_j$. Then for $i \leq j$ hold $\text{Str}(i) \leq \text{Str}(j)$ and $d_i(d_i - 1) \leq d_j(d_j - 1)$. By Theorem 3 and Corollary 5 for the case if there is no pendant vertices in graph holds

$$\begin{aligned} 1 - C_{WS}(G) &\leq \frac{1}{n} \sum_{i \in V(G)} \frac{\text{Str}(i)}{d_i(d_i - 1)} \\ &\stackrel{\text{Chebyshev's sum inequality}}{\leq} \left(\frac{1}{n} \sum_{i \in V(G)} \text{Str}(i) \right) \left(\frac{1}{n} \sum_{i \in V(G)} \frac{1}{d_i(d_i - 1)} \right) \\ &= \frac{1}{n} \sum_{i \in V(G)} \frac{(L(G) - 1)(n - 1)}{d_i(d_i - 1)}. \end{aligned}$$

For pendant vertices we should add $\frac{\text{number of pendant vertices}}{n}$ in the right side. Note that if there exist two vertices $i, j \in V(G)$ such that $d_i < d_j$ and $\text{Str}(i) < \text{Str}(j)$ then the inequality in this theorem will be strict.

Example 1. Let's consider a star G with $V(G) = n + 1$ vertices. The star is geodetic graph. The central vertex has degree n , local clustering coefficient $c_i = 0$ and stress centrality $\text{Str}(i) = n(n - 1)$. Other vertices are pendant ($d_i = 1$, $c_i = 0$, $\text{Str}(i) = 0$). Thus for this graph holds Theorem 5.

$$\begin{aligned} L(G) &= \frac{n(2n - 1) + n}{(n + 1)n} = \frac{2n}{n + 1}, \quad C_{WS} = 0. \\ 1 - C_{WS}(G) &= 1 = \frac{\frac{n-1}{n+1}n}{n(n - 1)} + \frac{n}{n + 1} \\ &= \frac{1}{n + 1} \sum_{i \in V(G)} \frac{(L(G) - 1)n}{d_i(d_i - 1)} + \frac{\text{number of pendant vertices}}{n + 1}. \end{aligned}$$

Thus, for this example holds equality.

REFERENCES

1. Travers, J. and Milgram, S., An experimental study of the small world problem, *Social networks. Academic Press*, 1977, pp. 179–197.
2. Watts, D.J. and Strogatz, S.H., Collective dynamics of ‘small-world’ networks, *Nature*, 1998, vol. 393, no. 6684, pp. 440–442.
3. Borgatti, S.P. and Everett, M.G., A graph-theoretic perspective on centrality, *Social networks*, 2006, vol. 28, no. 4, pp. 466–484.
4. Kiss, C. and Bichler, M., Identification of influencers-measuring influence in customer networks, *Decision Support Systems*, 2008, vol. 46, no. 1, pp. 233–253.
5. Lee, S.H.M., Cotte, J., and Noseworthy, T.J., The role of network centrality in the flow of consumer influence *J. Consum. Psychol*, 2010, vol. 20, no. 1, pp. 66–77.
6. You, J., Gomes-Selman, J.M., Ying, R., et al., Identity-aware graph neural networks, *Proceedings of the AAAI conference on artificial intelligence*, 2021, vol. 35, no. 12, pp. 10737–10745.
7. Yuan, M.M., Guo, X., Wu, L., et al., Climate warming enhances microbial network complexity and stability, *Nature Climate Change*, 2021, vol. 11, no. 4, pp. 343–348.

This paper was recommended for publication by A.G. Kushner, a member of the Editorial Board

A Method for Augmentation of Multidimensional Time Series in Equipment Health Monitoring Tasks

A. V. Erpalov^{*,a}, V. V. Sinitsin^{*,b}, O. L. Ibryaeva^{*,c}, and A. L. Shestakov^{*,d}

^{*}South Ural State University, Chelyabinsk, Russia

e-mail: ^aerpalovav@susu.ru, ^bsinitsinnv@susu.ru, ^cibryaevaol@susu.ru, ^dshestakoval@susu.ru

Received May 22, 2024

Revised June 21, 2024

Accepted July 25, 2024

Abstract—The paper proposes a decomposing data augmentation method. As against the available ones, it handles multidimensional time series and during the augmentation retains the general trend and the structure of multidimensional data based on plant sensor readings. The gist of the method proposed is the application of multilevel variational mode decomposition coupled with conventional augmentation techniques. The method was verified using real-life multidimensional time series for thermocouples and load cells. The augmentation results were analyzed using MAE criterion as well as stochastic process PDF.

Keywords: equipment state diagnosis, sensor readings, augmentation, multidimensional data, synthetic data, variational mode decomposition

DOI: 10.31857/S0005117925050081

1. INTRODUCTION

Data augmentation refers to the generation of new synthetic data that maintains similarity to the original dataset. Currently, the development of data augmentation techniques represents one of the most pressing challenges in measurement data preprocessing, with applications in several critical scenarios:

- (1) When obtaining a sufficiently large dataset for developing adequate mathematical models is prohibitively expensive due to high experimental costs.
- (2) When addressing missing data issues that frequently occur during practical implementation of predictive models in industrial settings, caused by system failures, production stoppages, or other disruptions.
- (3) When generating anonymized sensor data is required for model testing and public dissemination of results.

Conventional data augmentation techniques employ various manipulations of original data to create synthetic samples, including: time series noise injection [1], scaling method, magnitude warping [2], time warping, window warping [3], rotation, and permutation [2] and several other methods [4]. These approaches share a fundamental characteristic: they operate by distorting, reducing, enlarging, or otherwise modifying the original dataset. This leads to a specific challenge in their application to sensor signal processing — the potential for excessive distortion relative to the original signal, which may adversely affect subsequent model development.

Furthermore, the aforementioned methods demonstrate a crucial limitation: they cannot effectively process sets of different signals that are synchronized in time. This represents a common scenario in industrial applications where simultaneous recording from multiple sensors is required

for mathematical model development. Such signals typically exhibit shared temporal characteristics (outliers, trends, etc.), but applying independent augmentation to each signal disrupts these synchronized features, resulting in incorrect data for subsequent analysis.

Recent trends in data augmentation have shifted toward machine learning approaches, particularly:

- Generative adversarial networks (GANs) [5], which synthesize completely new datasets by learning the underlying data distribution from original samples
- Variational autoencoders (VAEs) [6], which generate data by mixing features from learned probability distributions rather than creating entirely novel samples

However, these advanced methods present two significant limitations:

- (1) They demand substantial volumes of original training data, which are often unavailable in practical applications.
- (2) The augmentation process requires considerable computational time, even for relatively small signals (e.g., 1000 data points), frequently necessitating segmented dataset processing that further complicates implementation.

We emphasize that this study does not address methods for generating stationary random processes with normal distributions, as these are well-documented in the literature [7] and typically employ inverse Fourier transform-based approaches.

This paper presents a novel decomposition-based data augmentation method for multivariate time series that combines multivariate variational mode decomposition with conventional augmentation techniques. The first section establishes the theoretical foundations of multivariate variational mode decomposition, traditional augmentation methods, and synthetic data validation criteria. The second section details the proposed decomposition-based augmentation approach. The third section demonstrates method verification using actual sensor data.

2. THEORETICAL FRAMEWORK

2.1. Multivariate Variational Mode Decomposition

The multivariate variational mode decomposition (MVMD) was developed by the authors of [8] as an extension of the variational mode decomposition algorithm (VMD) [9].

VMD represents a fully adaptive and non-recursive algorithm for time-frequency signal analysis. The fundamental assumption of this method posits that any original signal f can be decomposed into a finite number K of modes u_k (intrinsic mode functions — IMFs), each characterized by central frequencies ω_k and limited bandwidths.

The signal decomposition through VMD is formulated as a constrained variational optimization problem:

$$\min_{\{u_k\}, \{\omega_k\}} \left\{ \sum_{k=1}^K \left\| \partial_t \left[\delta(t) + \frac{j}{\pi t} \right] * u_k(t) e^{-j\omega_k t} \right\|_2^2 \right\} \quad (1)$$

subject to the constraint: $\sum_{k=1}^K u_k(t) = f$ where $\{u_k\}$ and $\{\omega_k\}$ denote the sets of modes and their corresponding central frequencies; δ represents the Dirac function; $j_2 = -1$; $\|\cdot\|_2$ indicates the vector norm; ω_k denotes the central frequency; $*$ denotes the convolution integral; each mode $u_k(t) = A_k(t) \cos(\phi_k(t))$ consists of an amplitude envelope A_k and phase ϕ_k . The unconstrained

form of Equation (1), incorporating the augmented Lagrangian method, can be expressed as:

$$L(\{u_k\}, \{\omega_k\}, \lambda) = \alpha \sum_{k=1}^K \left\| \partial_t \left[\delta(t) + \frac{j}{\pi t} \right] * u_k(t) e^{-j\omega_k t} \right\|_2^2 + \left\| f(t) - \sum_{k=1}^K u_k(t) \right\|_2^2 + \left\langle \lambda(t), f(t) - \sum_{k=1}^K u_k(t) \right\rangle, \quad (2)$$

where L represents the augmented Lagrangian function, λ denotes the Lagrange multiplier, $\langle a, b \rangle$ indicates the scalar product of a and b .

The solution is obtained through iterative optimization of u_k^{n+1} , ω_k^{n+1} , and λ_k^{n+1} using the alternating direction method of multipliers (ADMM) [10]. The final VMD formulation comprises:

$$\hat{u}_k^{n+1}(\omega) = \frac{\hat{f}(\omega) - \sum_{i < k} \hat{u}_i^{n+1}(\omega) - \sum_{i > k} \hat{u}_i^n(\omega) + \hat{\lambda}^n(\omega) / 2}{1 + 2\alpha(\omega - \omega_k^n)^2}, \quad (3)$$

$$\omega_k^{n+1} = \frac{\int_0^\infty \omega |\hat{u}_k^{n+1}(\omega)|^2 d\omega}{\int_0^\infty |\hat{u}_k^{n+1}(\omega)|^2 d\omega}, \quad (4)$$

$$\hat{\lambda}^{n+1}(\omega) = \hat{\lambda}^n(\omega) + \tau \left[\hat{f}(\omega) - \sum_k \hat{u}_k^{n+1}(\omega) \right], \quad (5)$$

where $\wedge$ denotes the Fourier transform; n represents the iteration number; α is the quadratic penalty coefficient; τ indicates the time step.

The MVMD method extends VMD by enabling the alignment of common frequencies across multiple synchronized signals (C). This capability proves particularly valuable in numerous scientific and engineering applications.

2.2. Augmentation by Magnitude Warping and Window Warping

The magnitude warping augmentation algorithm modifies signal amplitudes within a specified window through multiplication by random scalars [2]. This amplitude transformation is implemented by convolving the data window with a smooth curve varying around unity. The method's key parameters include: (σ) the root mean square value of random noise, (*knots*) the number of control points defining the smooth warping curve [11].

The window warping augmentation algorithm distorts data within a randomly selected time series window through either expansion or compression (Fig. 1) [3]. The primary parameter governing this method is the original window size.

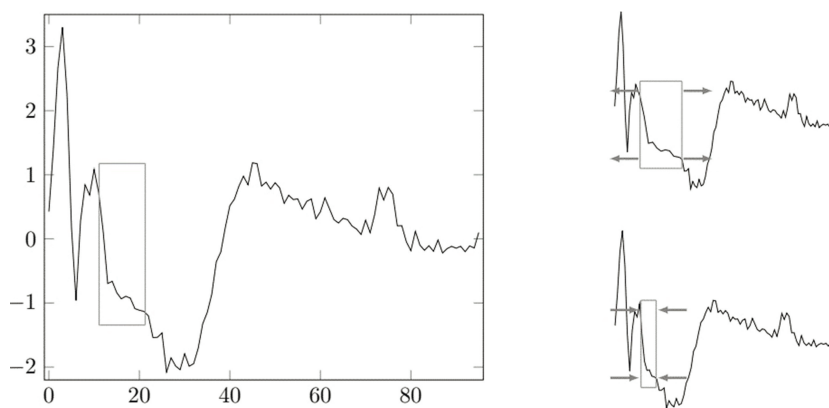


Fig. 1. Expanding and compressing data in the selected window.

2.3. Criteria for Evaluating Synthetic Data Adequacy

Currently, no universal standards exist for assessing data augmentation algorithm quality. The development of appropriate metrics for evaluating synthetic data quality and diversity remains an open research question [12].

The most widely adopted criterion for synthetic data validation is the mean absolute error (MAE):

$$MAE = \frac{\sum_{i=1}^n |y_i - x_i|}{n}, \quad (6)$$

where y_i represents the synthetic data set, x_i denotes the original dataset, and n indicates the time series length.

Alternative approaches employ neural networks for quality assessment. Yoon et al. [13] proposed evaluating synthetic data quality using a discriminative model comprising a two-layer LSTM network trained on original data. Similarly, Esteban et al. [14] developed a quality metric based on the mean absolute error of predictions from models trained on synthetic data.

For our analysis, we employ MAE while additionally comparing: probability density functions (PDFs) of original and synthetic processes; key statistical parameters (kurtosis and skewness).

3. PROPOSED METHOD FOR DATA AUGMENTATION

We present a novel decomposition-based data augmentation method designed to synthesize multivariate datasets while preserving critical time series characteristics, including trends and outliers. From an applied perspective, industrial applications typically involve synchronous data acquisition from heterogeneous sensors, which subsequently serve as input for neural network models. However, existing augmentation methods predominantly treat each signal independently, generating statistical processes that mimic individual signals while disregarding inter-signal relationships. This approach may introduce artifacts that compromise subsequent analysis.

Consider the example of process temperature sensor data (Fig. 2), where all sensors exhibit a temperature drop within the time interval $(3.8 \dots 4.3) \cdot 10^4$ minutes, corresponding to equipment shutdown. This shared temporal feature must be preserved during synthetic data generation.

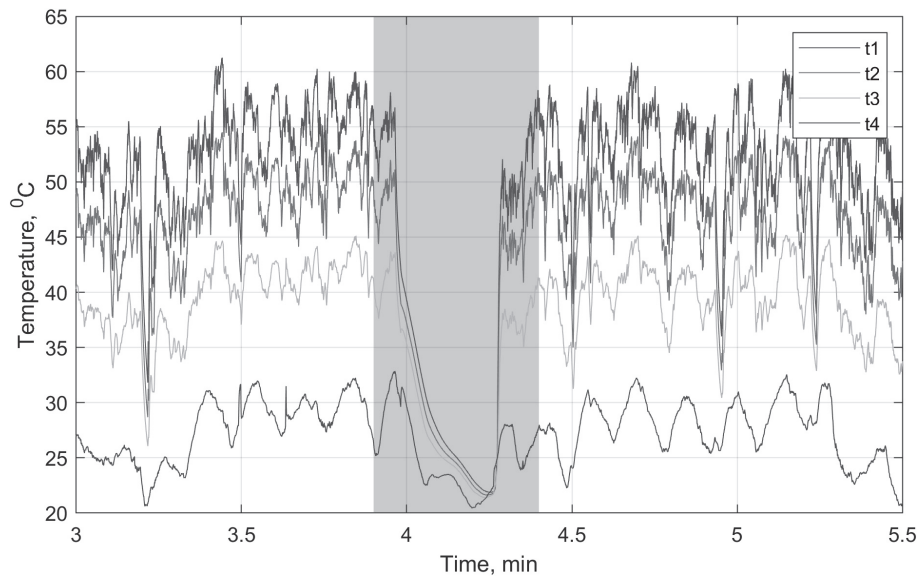


Fig. 2. Example of temperature change implementation with equipment shutdown zone.

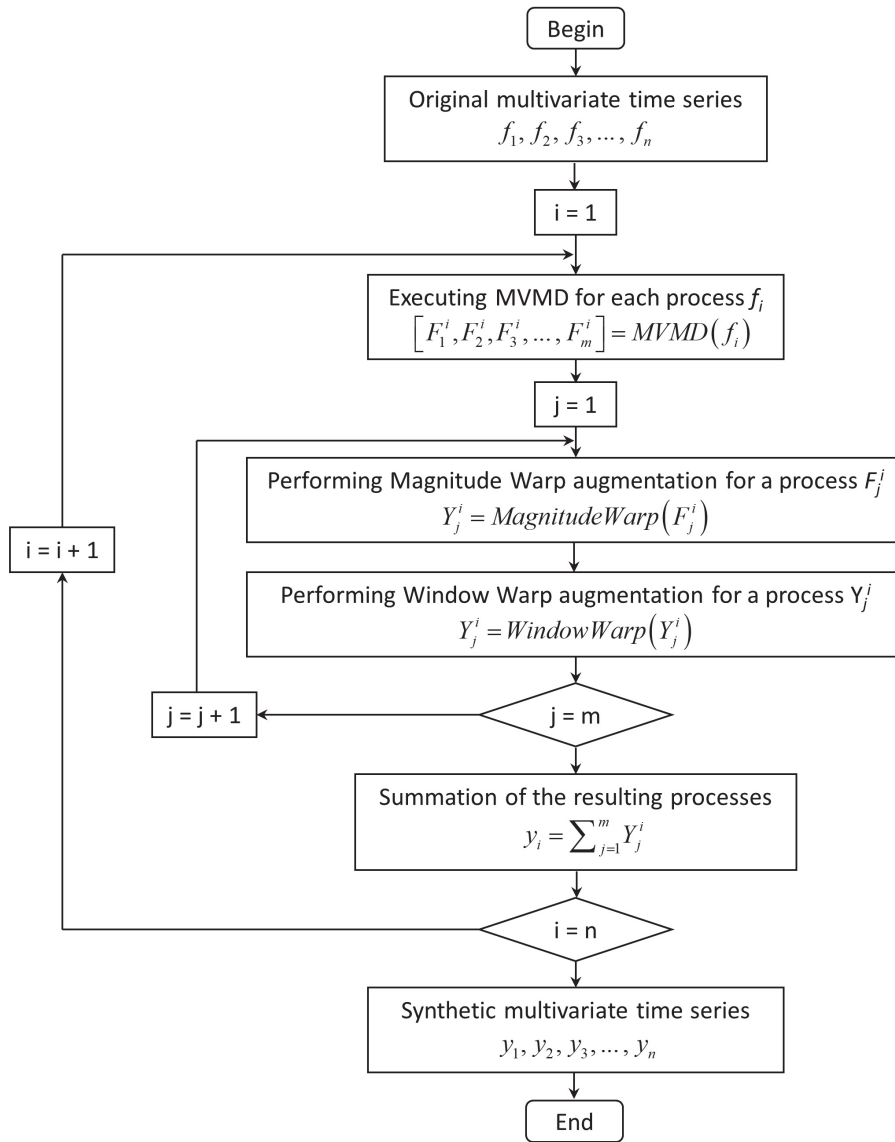


Fig. 3. Flowchart of the decomposition augmentation algorithm.

Furthermore, the relative temperature hierarchy $t1 > t2 > t3 > t4$ must be maintained throughout the augmented dataset.

Our decomposition-based augmentation method employs multivariate variational mode decomposition (MVMD) to separate each source signal into constituent modes (narrowband processes) with simpler structures. We then apply conventional augmentation techniques — magnitude warping and window warping — independently to each mode. Finally, we reconstruct the augmented signal by summing the processed modes, yielding synthetic multivariate datasets. Figure 3 presents the complete algorithm flowchart.

Like all augmentation methods, our approach incorporates several tunable parameters that significantly influence the results:

(1) MVMD Parameters:

- The number of decomposition modes (m) must be carefully selected based on signal non-stationarity and length. For typical applications, we recommend 4–6 modes.

- The first (low-frequency) mode may remain unaugmented to preserve global signal structure, or alternatively, may undergo augmentation with reduced intensity.
- (2) Conventional Augmentation Parameters:
- Magnitude Warp: (σ) Standard deviation of Gaussian noise, (*knots*) Number of control points for the smoothing curve.
 - Window Warp: (*window ratio*) Size of the original window, (*scales*) Distortion factor for window expansion/compression.

The selection of magnitude warping (for amplitude modification) and window warping (for temporal distortion) provides comprehensive signal transformation capabilities across both amplitude and time domains.

4. EVALUATION OF THE APPLICABILITY OF THE PROPOSED METHOD

4.1. Sensor Data Characteristics

The proposed decomposition-based augmentation method was validated using temperature monitoring data acquired from multiple bearing support sensors in industrial equipment [15]. As shown in Fig. 4, the dataset comprises four synchronized time series with a sampling frequency of 1 minute, spanning 27 hours of continuous operation, with each signal containing 10 000 data points. The temporal synchronization of the acquisition system ensures consistent representation of trends and behavioral patterns across all measurement channels.

4.2. Augmentation Results and Validation

Implementation of the decomposition augmentation algorithm with the parameters specified in Table 1 yielded synthetic temperature data that maintains several critical characteristics of the original dataset. By deliberately excluding augmentation of the first low-frequency MVMD mode, we preserved the fundamental structure of the time series while introducing controlled variations in other frequency components.

Table 1. Parameters of decomposition augmentation

MVMD _e	Magnitude-Warp		Window-Warp	
Number of modes, units	Sigma, units	Knots, units	Window ratio, units	Scales
6	0.2	10 000	0.5	[0.5 4.0]

The synthetic data successfully maintains both the characteristic temperature drop corresponding to equipment shutdown events and the consistent temperature hierarchy $t_1 > t_2 > t_3 > t_4$ throughout the generated dataset. Comparative analysis of the original and synthetic signals (particularly evident in Fig. 6 for channel t_1) demonstrates effective preservation of global trends while introducing natural variability in oscillation characteristics, including both amplitude and frequency components.

Statistical evaluation reveals close agreement between original and synthetic datasets. The probability density functions shown in Fig. 7 exhibit similar distributions, with root mean square values differing by less than 1%. The kurtosis and skewness parameters show deviations within 14% and 16% respectively, while maintaining the essential characteristics of the original signals. Quantitative assessment using mean absolute error yields values of 0.06 (6%) for t_1 and t_2 , 0.04 (4%) for t_3 , and 0.02 (2%) for t_4 , confirming the method's accuracy.

The critical advantage of the MVMD-based approach becomes particularly evident when comparing direct augmentation with the proposed decomposition method (Fig. 8). While conventional

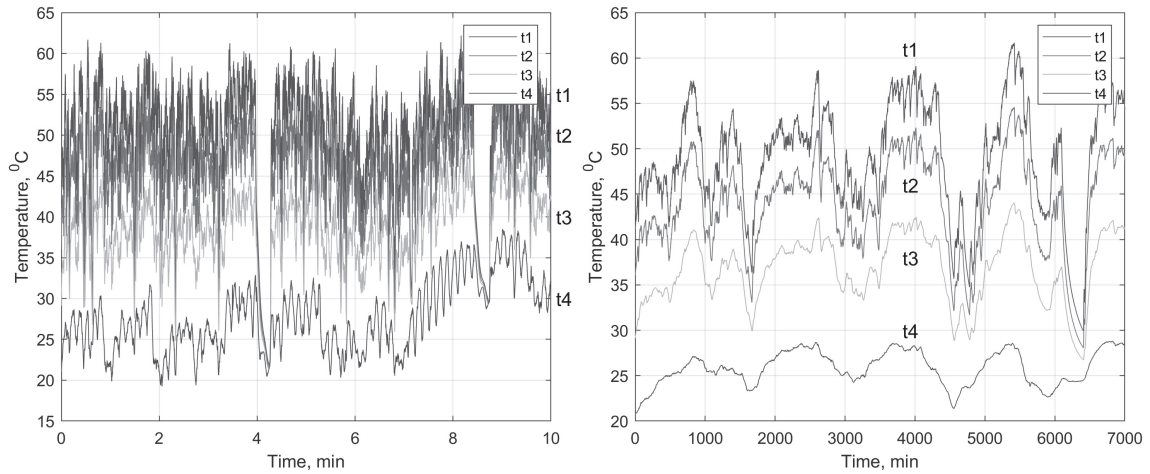


Fig. 4. Implementations of signals from temperature sensors (left) and an enlarged fragment of the implementations (right).

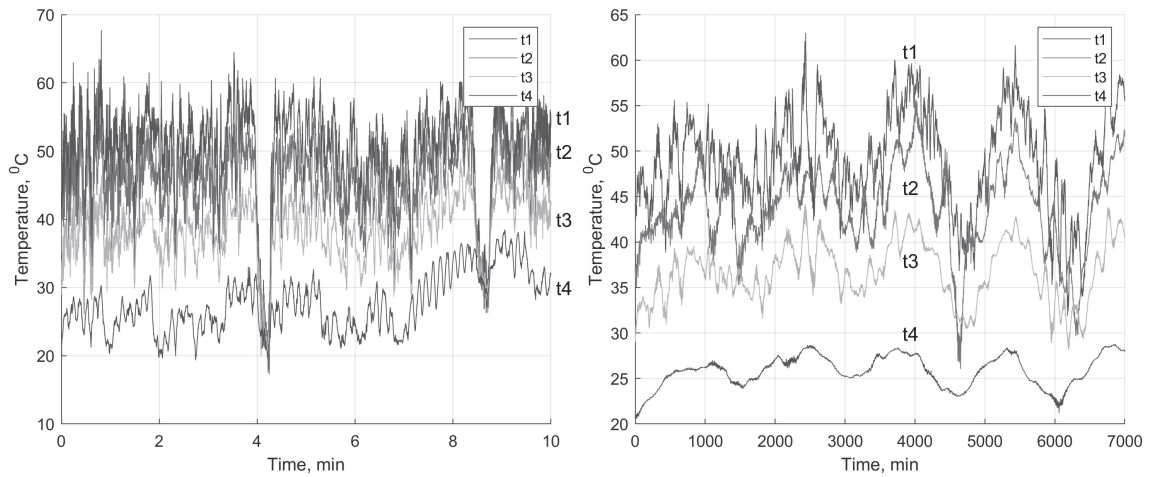


Fig. 5. Implementations of synthetic signals from temperature sensors (left) and an enlarged fragment of the implementations (right).

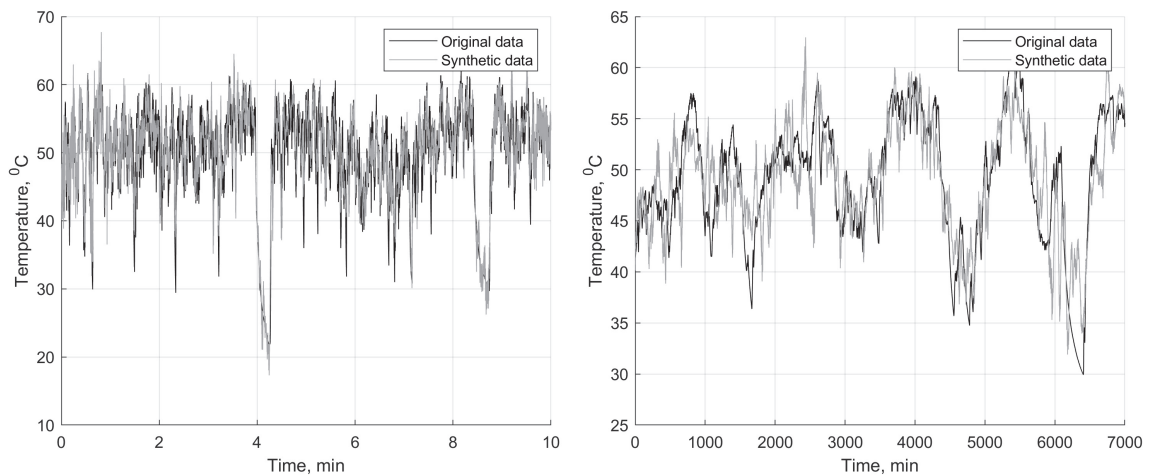


Fig. 6. Implementations of the original signal and the first synthetic signal (t1) from the multivariate data set (left) and an enlarged fragment of the implementations (right).

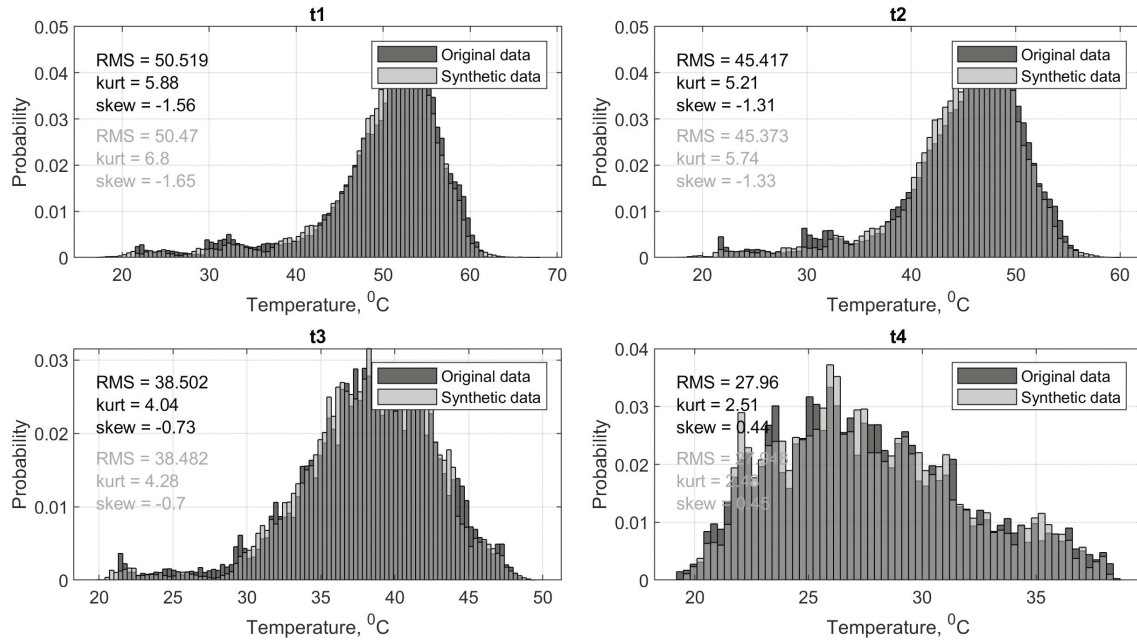


Fig. 7. Distribution densities of original and synthetic signals and their statistical characteristics.

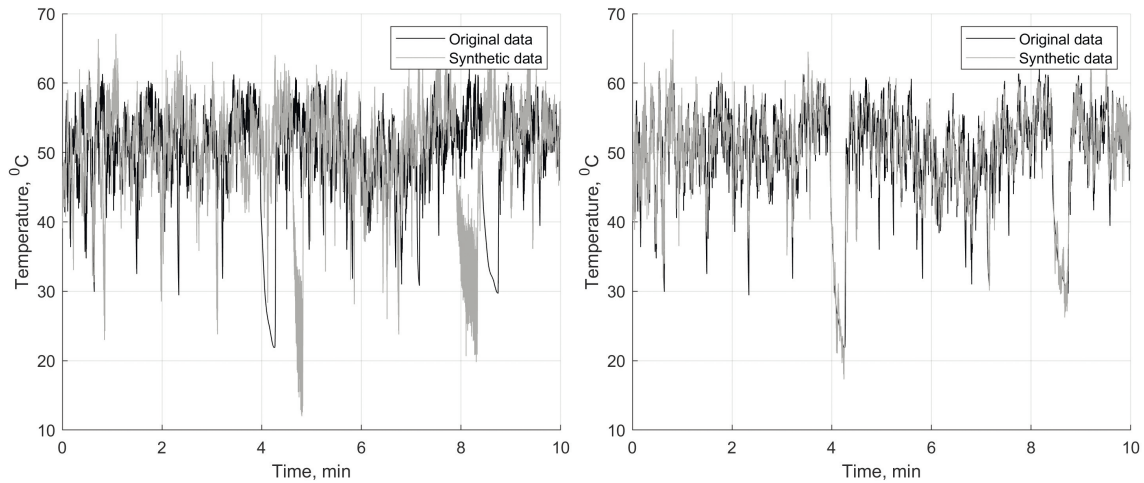


Fig. 8. The original signal and the first synthetic signal (t1) from the multivariate data set without preliminary decomposition of signals (left) and with decomposition (right).

augmentation fails to maintain temporal alignment and characteristic features, our method successfully preserves both synchronized events and global signal properties, demonstrating its superior performance for multivariate time series augmentation.

5. CONCLUSION

This study has presented an innovative decomposition-based approach for augmenting multivariate time series data, combining multivariate variational mode decomposition with conventional augmentation techniques. The developed method fundamentally advances current capabilities by preserving critical temporal characteristics while maintaining synchronization across multiple sensor signals — a crucial requirement for industrial monitoring applications where data from heterogeneous sensors must remain temporally aligned.

Through experimental validation using real-world temperature sensor data, we have demonstrated the method's ability to maintain both global data structure and precise inter-sensor relationships. The synthetic data preserves required operational characteristics such as the temperature hierarchy while introducing appropriate variability, with quantitative analysis confirming close agreement through mean absolute error metrics ranging between 2–6%. Statistical evaluation further substantiates the method's effectiveness, showing minimal deviations in key parameters including root mean square values (under 1% difference), kurtosis (within 14% variation), and skewness (below 16% deviation).

The comparative analysis provides compelling evidence for the necessity of the MVMD pre-processing stage, clearly demonstrating its superiority over direct augmentation approaches in preserving synchronized events and temporal relationships within multivariate datasets. This capability proves particularly valuable for industrial condition monitoring systems where maintaining temporal correlations between sensor signals is paramount for accurate diagnostics and predictive maintenance.

FUNDING

This work was financially supported by Ministry of Science and Higher Education of the Russian Federation (FENU-2023-0010).

REFERENCES

1. Flores, A., Tito-Chura, H., and Apaza-Alanoca, H., Data Augmentation for Short-Term Time Series Prediction with Deep Learning, *Intelligent Computing*, 2021, pp. 492–506.
2. Um, T.T., et al., Data augmentation of wearable sensor data for parkinson's disease monitoring using convolutional neural networks, *Proc. of the 19th ACM International Conf. on Multimodal Interaction*, 2017, New York, pp. 216–220.
3. Pialla, G., Devanne, M., Weber, J., Idoumghar, L., and Forestier, G., Data Augmentation for Time Series Classification using Convolutional Neural Networks, *Advanced Analytics and Learning on Temporal Data. AALTD 2022. Lecture Notes in Computer Science*, 2023, vol. 13812, pp. 117–132.
4. Efimov, A.I., Methods for improving the efficiency of training samples by supplementing them with generated graphical data (Metody povysheniya effektivnosti obuchayushchih vyborok putem dopolneniya ih generirovannyimi graficheskimi dannymi), *Automation and Remote Control (Avtomatizaciya v promyshlennosti)*, 2019, no. 4, pp. 54–57.
5. Goodfellow, I., et al., Generative adversarial networks, *Commun. ACM*, 2020, vol. 63, no. 11, pp. 139–144.
6. Kingma, D.P. and Welling, M., Auto-Encoding Variational Bayes, *2nd Int. Conf. Learn. Represent*, 2014, Canada, pp. 1–14.
7. Denisova, L.A., Modeling and optimization of NPP power unit steam generator power supply control system (Modelirovanie i optimizaciya sistemy regulirovaniya pitaniya parogeneratora energobloka AES), *Automation and Remote Control (Avtomatizaciya v promyshlennosti)*, 2013, no. 7, pp. 14–19.
8. Rehman, N.U. and Aftab, H., Multivariate Variational Mode Decomposition, *IEEE Trans. Signal Process*, 2019, vol. 67, no. 23, pp. 6039–6052.
9. Dragomiretskiy, K. and Zosso, D., Variational Mode Decomposition, *IEEE Trans. Signal Process*, 2014, vol. 62, no. 3, pp. 531–544.
10. Bertsekas, D.P., *Constrained Optimization and Lagrange Multiplier Methods*, Elsevier, 1982.
11. Iwana, B.K. and Uchida, S., An empirical survey of data augmentation for time series classification with neural networks, *PLoS ONE*, 2021, vol. 16, no. 7, pp. 1–32.

12. Iglesias, G., et al., Data Augmentation techniques in time series domain: a survey and taxonomy, *Neural Comput. Appl.*, 2023, vol. 35, no. 14. pp. 10123–10145.
13. Yoon, J., Jarrett, D., and van der Schaar M., Time-series generative adversarial networks, *Adv. Neural Inf. Process. Syst.*, 2019, vol. 32, pp. 1–11.
14. Esteban, C., Hyland, S.L., and Ratsch, G., Real-valued (Medical) Time Series Generation with Recurrent Conditional GANs, 2017, <https://arxiv.org/abs/1706.02633>
15. Shestakov, A.L., et al., Artificial intelligence methods in diagnostics of rolling production equipment. Diagnostics of tension leveler bearings, *SOFT Meas. Comput.*, 2023, vol. 10, no. 71. pp. 76–91.

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

Estimating Induction Motor State Based on Current Signals in the Conditions of Limited Data Set Using Recurring Images

A. L. Shestakov^{*,a}, D. V. Galyshev^{*,b}, O. L. Ibryaeva^{*,c}, and V. A. Ereemeeva^{*,d}

^{*}South Ural State University, Chelyabinsk, Russia

e-mail: ^aa.l.shestakov@susu.ru, ^bgalyshevdu@susu.ru, ^cibryaevaol@susu.ru, ^deremeeveva@susu.ru

Received May 20, 2024

Revised June 20, 2024

Accepted July 23, 2024

Abstract—The paper examines the diagnostics of rotor bar breakage in induction motors using electric current signals in the conditions of limited data set. The available failure detection procedure is based on the filtering of the current around the seventh harmonic of the supply voltage frequency with subsequent development of an envelope underlying the scalograms. Instead of scalograms, the paper offers recurrent graphs, which demonstrate higher computational efficiency. It also proposes the modeling of the lacking abnormal operation data that is typical for real-life operating conditions. The paper shows that a convolutional neural network (CNN) model taught on both design operation data obtained at various operation modes and the abnormal operation data from a single mode poorly detects the defect in case of operation mode changes. This is caused by the change of specific defect frequencies dependent on the operation mode. To overcome this challenge, the paper offers a data augmentation method based on spectrum restructuring. The spectrum of a real-life defective signal is reshaped in view of its structure, and synthetic data are developed, which simulate the lacking defect signals for other operation modes. Experimental testing showed that data augmentation based on spectrum reshaping significantly improves CNN model performance under insufficient volume of defect-mode training data.

Keywords: induction motor, broken rotor bar, scalogram, convolutional neural network, recurrent graphs, spectrum reshaping, Hilbert transform, data augmentation

DOI: 10.31857/S0005117925050096

1. INTRODUCTION

Lifecycle management of industrial equipment is critical for maintaining uninterrupted production processes [1]. Unplanned downtime due to equipment failures results in substantial financial losses, driving modern lifecycle strategies to prioritize machinery condition assessment. This is achieved through specialized monitoring systems that employ data processing algorithms, utilizing measurements from onboard sensors [2]. Integrating advanced analytics into existing monitoring infrastructure allows enterprises to not only improve operational safety but also optimize the performance of complex industrial systems [3].

Induction motors are among the most widespread and mission-critical industrial assets, serving as the primary driving force for the majority of industrial machinery across virtually all manufacturing sectors. However, like all electromechanical equipment, they are susceptible to various operational faults and failures. For instance, a broken rotor bar accounts for up to 10% of all failures in induction motors [4]. Early detection of this fault is crucial, as operating a motor with this defect leads to increased energy consumption, reduced torque output, and elevates the risks of rotor locking and stator winding damage [5].

Most existing diagnostic methods are based on the analysis of side harmonics arising near the power supply frequency f_s , the amplitude of which increases as the defect progresses [6]:

$$s_{sidepeaks} = (1 \pm 2s)f_s. \quad (1)$$

Here s is the motor slip. Although side harmonics are an established sign of broken bars, their amplitude may be reduced and masked by the fundamental frequency of the supply voltage, particularly during low-load operation. Partial bar failures further exacerbate this challenge by producing extremely weak diagnostic signals that are prone to noise contamination. Moreover, various operational factors interfere with reliable harmonic detection [7]. Specifically, cyclostationary loads, magnetic asymmetry, and rotor ventilation ducts generate current harmonics in the same spectral range as bar breakage signatures. These confounding factors substantially degrade the diagnostic reliability of conventional harmonic analysis, limiting its effectiveness for precise fault detection.

An alternative diagnostic approach that circumvents these masking effects involves analyzing the motor's starting current transient [8]. This method enables precise fault detection by identifying distinctive time-frequency patterns generated during motor startup. Unlike steady-state operation under cyclostationary loads, the startup transient produces a unique spectral signature that effectively isolates fault-related components from interfering harmonics, significantly improving diagnostic reliability.

However, this diagnostic approach has several important limitations. First, it cannot be applied to motors driven by frequency converters, as their low slip values prevent effective testing. Second, the method is impractical for medium- and high-power motors [9], since implementation requires repeated start-stop cycles that may compromise motor performance and operational efficiency.

When a rotor bar defect occurs, in addition to peaks (1), a number of components in the higher harmonics range also appear in the current signal spectrum [10]:

$$\begin{aligned} f_{space} &= (k(1 - s) \pm s)f_s, \\ f_{rotare} &= (k \pm 2ns)f_s, \end{aligned}$$

where $k = 5, 7$, $n = 1, 2, 3$. While cyclostationary loads may still obscure certain frequency components in this range, their masking effect is significantly less pronounced compared to the interference observed near the supply frequency's sidebands. Consequently, higher harmonic analysis of the stator current spectrum offers superior diagnostic reliability. For this study, we specifically examine signal characteristics surrounding the seventh harmonic of the supply frequency.

Beyond signal processing challenges, significant obstacles emerge during data acquisition. The development of robust fault diagnosis systems is hindered by the scarcity of failure data, as induction motors typically operate reliably for extended periods with infrequent malfunctions. This scarcity makes it particularly difficult to collect comprehensive datasets encompassing all potential operating conditions. Consequently, the resulting training data often lacks representativeness, posing substantial challenges for building reliable machine learning models capable of generalizing across diverse operational scenarios.

This paper presents a novel diagnostic method for detecting broken rotor bars in induction motors under conditions of severely limited failure data. To address the data scarcity challenge, we introduce an innovative spectrum restructuring technique for signal augmentation that preserves the inherent structure of motor current signatures. The proposed approach combines: (1) advanced analysis of higher-order harmonics in stator current signals with (2) synthetic data generation through physics-informed spectrum transformation. This dual methodology enables reliable fault detection even when training data is sparse or unrepresentative of all operational conditions. Signals are acquired from both healthy and faulty motors under varying load conditions. Each signal is filtered to extract the informative component that exhibits changes in the presence of defects. The faulty signals are then augmented to simulate modes with insufficient load. The processed signals

are then transformed into recurrent graph images and used to train a convolutional neural network (CNN) classifier. During training, the CNN is exposed to defect data from only a single load level, while during testing, it evaluates data across all load conditions. This approach validates the rotor bar break diagnosis model's capability to detect faults under varying motor operating conditions.

The paper is structured as follows: Section 2 presents the proposed diagnostic methodology. Section 3 introduces the spectrum restructuring-based augmentation technique. The experiment is described in Section 4, followed by the discussion of diagnostic model test results in Section 5. The key contribution of this work is demonstrating the efficacy of envelope spectrum distortion-based augmentation when training a classification model using higher-order stator current spatial harmonics as diagnostic features for detecting broken rotor bars in induction motors.

2. THE PROPOSED DIAGNOSTIC METHOD

Figure 1 shows the scheme of the broken rotor bar diagnostic method using stator current analysis, used in [11]. The process begins with signal preprocessing, which includes bandpass filtering around the seventh harmonic of the supply frequency. The defect frequencies are then separated from the fundamental component through the demodulation, by finding the envelope. The preprocessed signals are converted into image representations for classifier training using a convolutional neural network. Unlike the original method in [11] that employs computationally expensive scalograms, this paper proposes using more efficient recurrent graphs, achieving comparable diagnostic accuracy with significantly reduced computational requirements.

The recurrence graph is defined as a binary matrix R , where each element $R_{i,j}$ is determined by:

$$R_{i,j} = \Theta(\varepsilon - \|x_i - x_j\|), \quad i, j = 1, \dots, N,$$

where N is the number of measured points x_i , ε is a threshold distance, $\Theta(\cdot)$ is the Heaviside step function and $\|\cdot\|$ is a norm. The matrix R is symmetric across its main diagonal and highly sensitive to the ε threshold selection.

An alternative approach uses a distance matrix D with elements

$$d_{i,j} = \|x_i - x_j\|,$$

eliminating threshold dependence. To convert these distances into grayscale images, values are normalized to $[0, 255]$ by computing the matrix's minimum and maximum values. The resulting

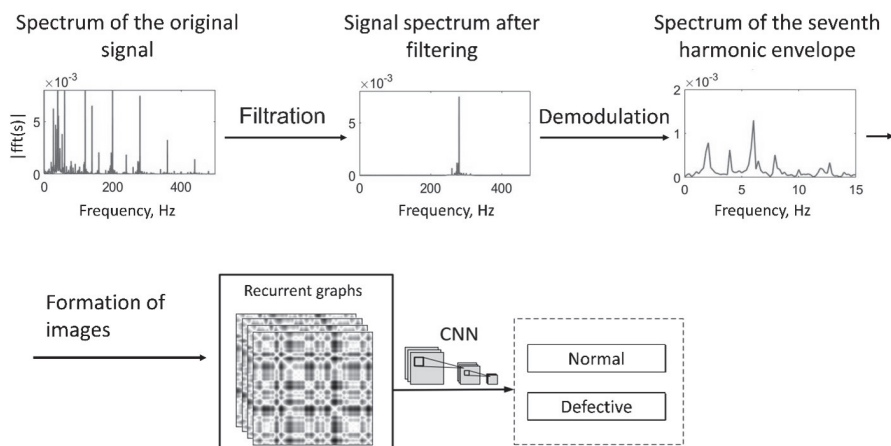


Fig. 1. Scheme of the proposed method.

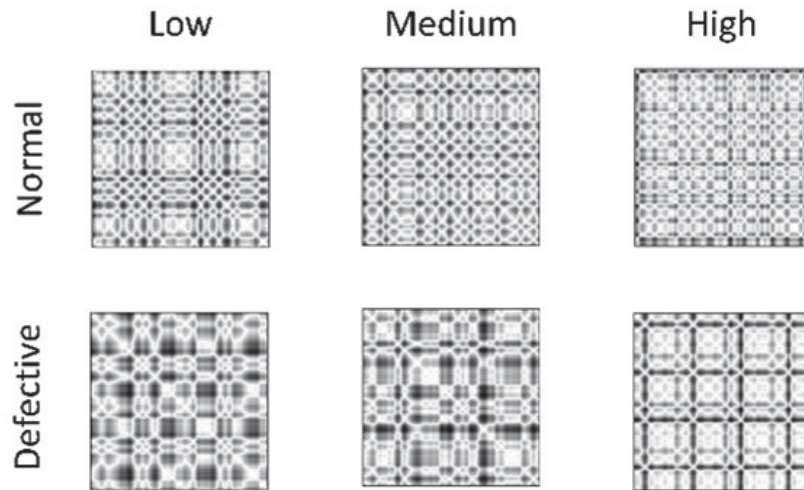


Fig. 2. Recurrence graphs of signals for healthy and rotor-bar-faulty induction motors under low, medium, and high load conditions.

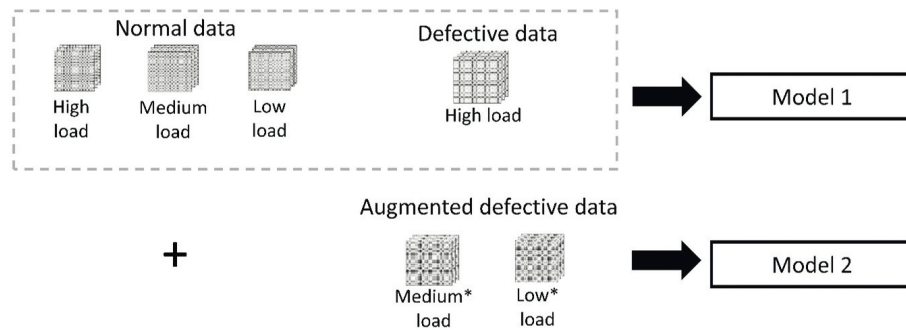


Fig. 3. Training datasets using only original data (Model 1) and augmented data (Model 2).

images are resized to 256×256 pixels using linear interpolation, which scales the original data to the target resolution while preserving value transitions.

Figure 2 displays 256×256 pixel recurrence plots generated from signals representing different motor operating conditions. The underlying dataset used to produce these images will be comprehensively detailed in Section 4. The analysis of the generated recurrence plots reveals two key findings. First, the visible dissimilarities between plots from healthy and defective motors confirm that the model can effectively differentiate between these operational states. Second, the observed variations across different operating modes suggest potential challenges for model generalization, particularly when processing data from the operating conditions that were not represented in the training dataset.

In practice, limited data availability across load conditions restricts the model's classification accuracy. While healthy motor signals can be acquired under various operating modes, faulty motor data is typically available for only one load condition. Consequently, the training set contains defect patterns from a single operating mode, constraining model performance. This scenario corresponds to Model 1 in Fig. 3.

To address this limitation, we propose augmenting the dataset through load simulation. By artificially generating signal variations that emulate different load conditions, we can significantly expand the available training data for faulty motor states. Model 2 (Fig. 3) trained on such a data

set will have greater load invariance, which will improve its accuracy in classifying different motor states. The method of current signal augmentation to expand the faulty data set will be described in the next section.

3. DFT SPECTRUM RESTRUCTURING METHOD FOR DATA AUGMENTATION

The N -point Discrete Fourier Transform (DFT) $S(k)$ of a discrete signal $s(n)$, $n = 0, \dots, N - 1$, is given by:

$$S(k) = \sum_{n=0}^{N-1} \left(s(n) e^{-j \frac{2\pi nk}{N}} \right), \quad k = 0, \dots, N - 1,$$

where j is the imaginary unit.

The signal $s(n)$ can be recovered from its DFT by taking the inverse DFT:

$$s(n) = \frac{1}{N} \sum_{k=0}^{N-1} \left(S(k) e^{j \frac{2\pi nk}{N}} \right), \quad n = 0, \dots, N - 1.$$

The DFT of a real-valued signal has the following conjugate symmetry property:

$$S(N - m) = S^*(m), \quad m = \begin{cases} 1, \dots, \frac{N}{2} - 1, & \text{if } N \text{ is even,} \\ 1, \dots, \frac{N-1}{2} - 1, & \text{if } N \text{ is odd.} \end{cases}$$

Here $*$ denotes the complex conjugation operation.

The complex-valued discrete function $S(k)$ can be written in exponential form:

$$S(k) = A(k) e^{j\Phi(k)},$$

where $A(k) = |S(k)|$, $\Phi(k) = \arg S(k)$ are known as the amplitude spectrum and the phase spectrum, respectively.

From the property of complex symmetry $S(N - m) = S^*(m)$ for real-valued signals, the symmetry properties follow:

$$A(N - m) = A(m), \quad \Phi(N - m) = -\Phi(m).$$

Our method involves stretching (or compressing) the original spectrum on the interval $[a, c]$ to a new interval $[a, c_1]$, while compressing (or stretching) the spectrum on $[c, b]$ to $[c_1, b]$. This transformation modifies the spectral segment $S(k)$ over $[a, b]$, corresponding to frequency range $\left[\frac{a \cdot Fs}{N}, \frac{b \cdot Fs}{N} \right]$, where Fs is the sampling frequency. Due to spectral symmetry requirements, it is also necessary to change the corresponding symmetric parts of the spectra.

For spectral transformation, we employ linear interpolation. Given that $S(k)$ is the complex-valued function, we can either interpolate its real and imaginary parts, or interpolate its magnitude and its argument. These two approaches can lead to different results. For example, the work [12] examines these different approaches to interpolating complex response amplitude operator data which are used in marine hydrodynamics for the prediction of the ship motions. Interpolation of complex numbers using the real and imaginary parts performed worst for most cases tested in [12]. The authors also used linear interpolation and stated that it is “safest” to use as it ensures interpolated values will always be within the ranges of neighboring values.

Our experiments have convinced us that for our problem, linear interpolation using separate interpolation of the magnitude and the argument, rather than real and imaginary parts, is preferable. This interpolation is used in the spectrum distortion algorithm proposed below.

Algorithm 1 (of spectrum distortion).

$[a, c] \rightarrow [a, c_1], [c, b] \rightarrow [c_1, b]$:

Input: $s(n), Fs, a, b, c, c_1$.

1) Calculate DFT $S(k)$ of $s(n)$.

2) $[a, c] \rightarrow [a, c_1]$:

2.1) Calculate the stretching rate: $p = \frac{c_1 - a}{c - a}$.

2.2) Old samples: $t = 0 : \frac{1}{Fs} : \frac{c - a}{Fs}$.

2.3) New samples (the interpolation nodes): $t_1 = 0 : \frac{1}{p \cdot Fs} : \frac{c - a}{Fs}$.

2.4) Part of the spectrum to be distorted: $u = S(a : c)$.

2.5) Interpolate separately the magnitude and argument of u :

$$\hat{u} = \text{interp}(t, |u|, t_1) \cdot \exp(j \cdot \text{interp}(t, \arg u, t_1))$$

3) $[c, b] \rightarrow [c_1, b]$:

3.1) Calculate the stretching rate: $p = \frac{b - c_1}{b - c}$.

3.2) Old samples: $t = 0 : \frac{1}{Fs} : \frac{b - c}{Fs}$.

3.3) New samples (the interpolation nodes): $t_1 = 0 : \frac{1}{p \cdot Fs} : \frac{b - c}{Fs}$.

3.4) Part of the spectrum to be distorted: $v = S(c : b)$.

3.5) Interpolate separately the magnitude and argument of v :

$$\hat{v} = \text{interp}(t, |v|, t_1) \cdot \exp(j \cdot \text{interp}(t, \arg v, t_1))$$

4) Concatenate:

$$W = [\hat{u}(1 : \text{end} - 1), \hat{v}(1 : \text{end})] \text{ or } W = [\hat{u}(1 : \text{end}), \hat{v}(2 : \text{end})].$$

5) Initialize $\hat{S}(k) = S(k)$.

6) Replace the desired part with the interpolated one: $\hat{S}(a, b) = W$.

7) Take into account the property of spectrum symmetry:

if $a \neq 0$

$$\hat{S}(N - b - 1 : N - a - 1) = (\text{flip}(W))^*,$$

else:

$$\hat{S}(N - b - 1 : N - a) = (\text{flip}(W(2 : \text{end})))^*,$$

Output: distorted DFT $\hat{S}(k)$

Remark 1. In step 4 of the Algorithm, it is necessary to take into account that one and the same sample c is included in both $\hat{u}$ and $\hat{v}$, so it should not be taken twice.

Remark 2. The case $a = 0$ needs to be considered separately, because $S(0)$ lacks a symmetric counterpart in the spectrum.

Remark 3. flip reverses the order of the elements. $\text{interp}(t, f, t_1)$ returns interpolated values at points t_1 of a function f initially given by its values at points t .

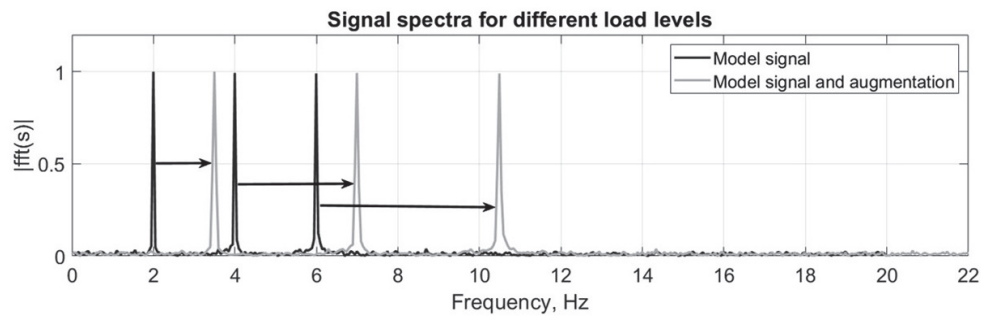


Fig. 4. Spectrum restructuring to obtain a signal with frequencies corresponding to a high load.

Let us consider the application of the algorithm using the following example. We model the envelope of the signal from a faulty rotor, containing three equally spaced frequencies of 2, 4, 6 Hz.

$$s(t) = \sum_{k=1}^3 \cos(2\pi f_1 t \cdot k) + n(t),$$

where $f_1 = 2$ Hz, and $n(t)$ is the additive Gaussian noise. The Signal-to-Noise Ratio (SNR) was measured at 16 dB, calculated as:

$$\text{SNR} = 10 \log_{10} \left(\frac{P_s}{P_n} \right),$$

where P_s and P_n represent the signal power and noise power, respectively. The signal duration was 4 seconds with a sampling frequency $F_s = 50$ Hz, yielding $N = 200$ samples.

According to the algorithm, we take: $a = 0 \cdot \frac{N}{F_s}$, $b = 20 \cdot \frac{N}{F_s}$, $c = 8 \cdot \frac{N}{F_s}$, $c_1 = 14 \cdot \frac{N}{F_s}$. Here the $\frac{N}{F_s}$ multiplier is used to go from frequency to sample number. Thus, we modify the spectrum in the section from 0 Hz to 20 Hz as follows. We stretch a part of the spectrum in the frequency range [0, 8] Hz to a new range [0, 14] Hz, and compress a part in the range [8, 20] Hz to [14, 20] Hz. Under this transformation, the original equispaced frequencies (2 Hz, 4 Hz, 6 Hz) are mapped to new equispaced frequencies (3.5 Hz, 7 Hz, 10.5 Hz). The spectra of the original and restructured signal are shown in Fig. 4. As expected, the spectra are identical above 20 Hz since this frequency region remained unmodified by the transformation.

In induction motors, the characteristic peak spacing in the envelope spectrum equals $2s$, where s is the motor slip coefficient. Our simulation of frequency transformation from (2, 4, 6) Hz to (3.5, 7, 10.5) Hz effectively models a slip change from 1 to 1.75 caused by a change in the motor load.

4. EXPERIMENTAL DATA

The effectiveness of the proposed diagnostic method was studied using data obtained from an experimental setup, which includes a 0.37 kW induction motor with one pair of poles, a bearing shaft connected to the motor through a claw coupling, a belt drive and a load. The load is a brake with permanent magnets that sets the torque on the motor shaft. A frequency converter powers the motor and sets the rotor speed.

The study utilized two identical motors: one healthy and one with artificially induced rotor bar defects (three drilled holes in the bar area). Stator current data was collected across supply frequencies from 20 Hz to 50 Hz (10 Hz increments) under three load conditions: Low (2%), Medium (20%), and High (36%) of rated torque. For each load level, four 60-seconds signals were recorded

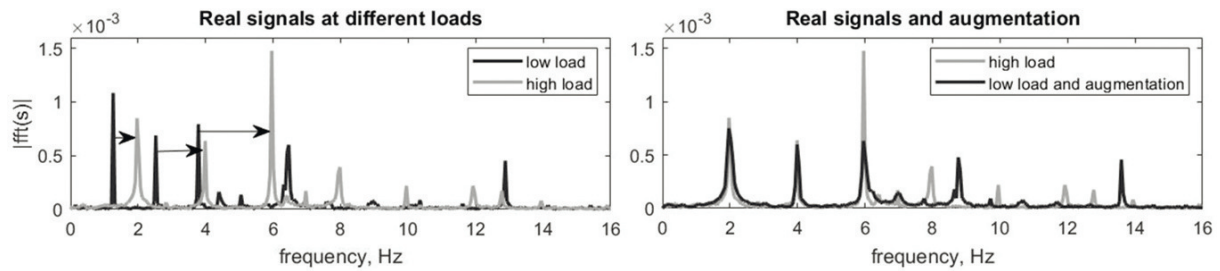


Fig. 5. Current signal spectra for low and high load levels (left — both signals are real, right — light — real, dark — synthetic).

at 50 kHz sampling frequency. The training set for Model 2 augmented real fault data using the spectral restructuring method described previously.

Figure 5 displays envelope spectra of actual current signals under two load conditions. Defect-related spectral peaks appear at: 1) low load (2%): 1.2 Hz, 2.4 Hz, 3.6 Hz; 2) high load (36%): 1.9 Hz, 3.8 Hz, 5.7 Hz. Through spectral restructuring, we transform the low-load signal by stretching the $[0, 4.8]$ Hz interval to $[0, 7.6]$ Hz. The spectrum of the new, rebuilt signal (Fig. 5, right), closely matches the actual high-load spectrum. Since there are no signals at different load levels in real conditions, our method allows compensating for the lack of data by creating them synthetically.

5. EXPERIMENT RESULTS

The training sets for Model 1 and Model 2 each contained 3000 recurrence plot representations (256×256 pixels) derived from processed current signals. The training approach differed between models: Model 1 used only high-load fault data, whereas Model 2 combined real high-load data with augmented samples (Fig. 6). For testing the trained Models 1 and 2, a set containing 1000 images for each class and for each load was used.

The training dataset consists of 1500 images for each class. The normal class contains data for all load values. In the case of Model 1, the defective class contains 1500 images of signals obtained at one load level. In the case of Model 2, the defective class contains 500 images of real signals obtained at the same load level as for Model 1 and 1000 images of synthetic signals obtained by augmentation, i.e., also 1500 images in total.

For training purposes, we utilize fault data collected under high-load conditions, as this operational regime represents the most critical and industrially relevant case, given that equipment

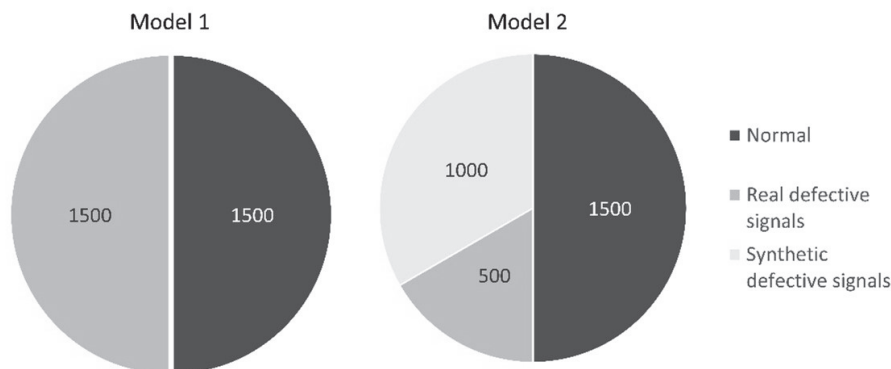


Fig. 6. Composition of the training dataset for Model 1 and Model 2.

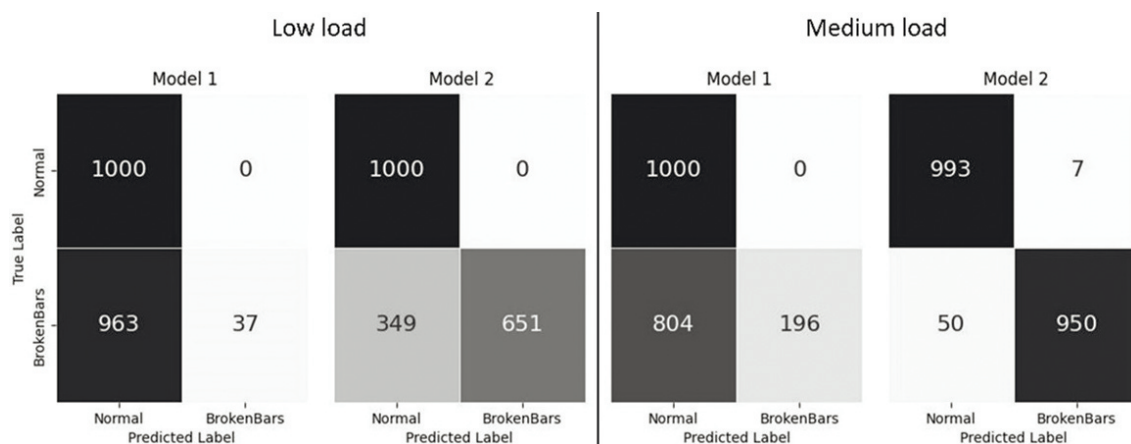


Fig. 7. Confusion matrices for Model 1 and Model 2.

typically operates under substantial loads. The model's capability to identify faults under low-load conditions demonstrates its sensitivity, as these subtle signatures are particularly challenging to detect. This example specifically evaluates the model's performance in recognizing incipient faults during their early developmental stage.

A 9-layer neural network model is used to train on the generated image dataset. The first 2D convolution layer includes 32 kernels that produce 256×256 feature maps. The next pooling layer performs the pooling operation, producing 32 maps of size 128×128 . The second convolution layer generates 64 maps of size 128×128 . This is followed by another pooling layer that produces 64 maps of size 64×64 . The third convolution layer produces 128 maps of size 64×64 , and the next pooling layer generates 128 maps of size 32×32 , which are then pulled out and fed to a fully connected layer with 2048 neurons. At the output, the data is passed through two additional fully connected layers and an output layer.

Model 1 training was completed after 100 epochs. While demonstrating perfect (100%) classification accuracy on high-load test data, its performance on low-load conditions (Fig. 7) revealed limitations: though all normal-class images were correctly identified, only 37 defective samples were properly classified. Model 2 similarly achieved perfect (100%) classification accuracy on high-load test data. For low-load conditions, the model showed significantly improved performance, correctly identifying 651 defective-class images — representing an order-of-magnitude increase in detection capability compared to Model 1. Consequently, Model 1 achieved an average accuracy of 52% on low-load conditions, while Model 2 demonstrated significantly improved performance with 83% accuracy — a 31 percentage point increase in diagnostic capability.

On medium-load test data, Model 1 achieved 60% classification accuracy compared to Model 2's 98% accuracy, though Model 2 produced seven false positives. It is important to note that in real production environments, the focus is usually on minimizing false negative cases, as they can lead to more serious consequences. Therefore, false positive classification errors are considered less critical for Model 2.

6. CONCLUSION

The paper proposes a method for diagnosing a broken rotor bar of an induction motor under realistic industrial conditions characterized by limited fault data availability. To solve the problem of insufficient training sample, we introduce a novel signal augmentation technique based on spectral restructuring. This approach allows modeling variations in the motor load, generating synthetic signals corresponding to both a decrease and an increase in load, which expands the representativeness of the data set.

This work also introduces, for the first time, the application of recurrence plots for induction motor fault diagnosis as an effective time-series visualization and analysis tool. Compared to conventional color scalograms requiring wavelet transform computation, recurrence plots offer superior computational efficiency while preserving feature discriminability. This allows reducing computational costs without losing classification accuracy.

During the experimental studies, it was found that the use of a convolutional neural network trained on recurrent graphs, in combination with the developed data augmentation method aimed at modeling reduced load conditions, led to a significant increase in the accuracy of defect classification. The achieved performance improvement validates the augmentation approach's effectiveness in developing load-invariant fault detection models.

The proposed methodology — combining spectral-based synthetic data generation with recurrence plot analysis — significantly enhances diagnostic reliability under data scarcity conditions. These findings directly support the development of robust predictive maintenance systems for industrial induction motors, particularly in scenarios with limited fault examples available for training.

FUNDING

This work was financially supported by Ministry of Science and Higher Education of the Russian Federation (FENU-2023-0010).

REFERENCES

1. Zadirán, K.S. and Shcherbakov, M.V., Forecasting Rare Events in the Evaluation of Equipment Residual Life, *Automation in Industry*, 2024, no. 10 pp. 10–15.
2. Kryukov, O.V., Intelligent Sensors for Predicting the Technical Condition of High-Voltage Electric Motors, *Automation in Industry*, 2013, vol. 10, pp. 38–41.
3. Aleksandrov, A.I. and Kvaratskhelia, N.G., Monitoring and Forecasting the Technical Condition of Electric Motors, *Automation in Industry*, 2020, vol. 10, pp. 39–43.
4. Singh, G.K. and Al Kazzaz, S.A., Induction Machine Drive Condition Monitoring and Diagnostic Research—A Survey, *Elec. Power Syst. Res.*, 2003, vol. 64, no. 2, pp. 145–158.
5. Garcia, M., Panagiotou, P.A., Antonino-Daviu, J.A., and Gyftakis, K.N., Efficiency Assessment of Induction Motors Operating Under Different Faulty Conditions, *IEEE Trans. Ind. Electron.*, 2019, vol. 66, no. 10, pp. 8072–8081.
6. Lee, S.B. et al., Condition Monitoring of Industrial Electric Machines: State of the Art and Future Challenges, *IEEE Ind. Electron. Mag.*, 2020, vol. 14, no. 4, pp. 158–167.
7. Lighthill, M.J., Contributions to the Theory of Heat Transfer Through a Laminar Boundary Layer, *Proc. R. Soc. London. Ser. A. Math. Phys. Sci.*, 1950, vol. 202, no. 1070, pp. 359–377.
8. Lee, S.B. et al., Identification of False Rotor Fault Indications Produced by Online MCSA for Medium-Voltage Induction Machines, *IEEE Trans. Ind. Appl.*, 2016, vol. 52, no. 1, pp. 729–739.
9. Zhang, P., Du, Y., Habetler, T.G., and Lu, B., A Survey of Condition Monitoring and Protection Methods for Medium-Voltage Induction Motors, *IEEE Trans. Ind. Appl.*, 2011, vol. 47, no. 1, pp. 34–46.
10. Tang, J., Yang, Y., Chen, J., Qiu, R., and Liu, Z., Characteristics Analysis and Measurement of Inverter-Fed Induction Motors for Stator and Rotor Fault Detection, *Energies*, 2019, vol. 13, no. 1, p. 101.
11. Galshev, D.V., Development of a Method for Diagnosing Broken Bars in Induction Motors Using Current Signals, *Materials of IX All-Russian Youth Scientific Forum Science of the Future — Science of the Young*, Samara, 2024, p. 150.
12. Thornhill, E., Interpolation of Complex-Valued Response Amplitude Operators: Approaches and Effects, Defence Research and Development Canada — Atlantic Research Centre, November 2019, p. 77.

Determination of Optimal Parameters of a Model Predictive Controller with Priorities of Control Objectives under Conditions of Plant Uncertainty

D. V. Shtakin^{*,a}, O. Yu. Snegirev^{*,b}, and A. Yu. Torgashov^{*,c}

^{}Institute of Automation and Control Processes, Far Eastern Branch,
Russian Academy of Sciences, Vladivostok, Russia*

e-mail: ^ashtakin@iacp.dvo.ru, ^bsnegirevoleg@iacp.dvo.ru, ^ctorgashov@iacp.dvo.ru

Received May 15, 2024

Revised June 19, 2024

Accepted July 25, 2024

Abstract—This work addresses the problem of determining the optimal parameters of a model predictive controller under conditions of plant parametric uncertainty, unmeasured disturbances, and the need to account for the prioritization of control objectives. An algorithm is proposed for determining the optimal parameters of a model predictive controller, which can be applied to both newly developed and existing control systems with predictive models, without requiring structural modifications. The algorithm incorporates a criterion that accounts for plant parametric uncertainty, the magnitude of output variable deviations beyond specified limits, and the duration of constraint violations. The effectiveness of the proposed algorithm is demonstrated through its application to an existing controller for a complex distillation column within a hydrocracking process unit. Using the algorithm, it was possible to achieve the minimum permissible initial boiling point of the kerosene fraction withdrawn from the bottom of the stripping column equipped with a thermosiphon, even under conditions of reduced coolant flow.

Keywords: model predictive control, parametric uncertainty, distillation column, hydrocracking unit

DOI: 10.31857/S0005117925050107

1. INTRODUCTION

In the oil refining industry, predictive model-based controllers are widely used due to their ability to control complex, multidimensional processes while operating under constraints on both control and controlled variables [1, 2]. A major advantage of these controllers is their capacity to optimize production performance, for example, by maximizing the yield of high-value products while ensuring compliance with quality specifications and minimizing energy consumption. Due to the multidimensional nature of process systems, predictive model-based controllers utilize weight matrices for input and output variables to achieve control objectives. However, these objectives can be compromised by plant parametric uncertainty, unmeasured external disturbances, or the use of suboptimal controller parameters [3].

Existing methods for ensuring the fulfillment of control goals can be divided into three groups. The first group includes methods that significantly change the common standard algorithms for model predictive controllers (MPCs). These methods are generally not suitable for existing systems, as they demand significant structural changes, which are often impractical in real-world applications. Thus, in the paper [4], the correct enforcement of priorities for controlled variables (CVs)

is achieved by determining the optimal changes in manipulated variables (MVs) through a lexicographic optimization approach. This involves sequentially solving multiple optimization problems, where each subsequent problem includes additional constraints that preserve the solutions obtained in the previous steps. A similar approach is presented in paper [5], which also addresses the prioritization of CV by solving a sequence of optimization problems. The method of taking into account the priorities of CV is known, taking into account the restrictions on the state of the plant by solving the problems of mixed-integer programming with additional restrictions that penalize the violation of the priority of the control objectives [6]. These works address only the prioritization of CV and do not consider the effects of plant parametric uncertainty or unmeasured disturbances. The method of handling the parametric uncertainty of the plant involves online identification of controller model parameters using the recursive least squares method [7]. However, this method requires persistent excitation of the system in a closed-loop configuration, even in the presence of unknown and unmeasured external disturbances. It is also important to note that the methods described above generally entail a high computational burden, primarily due to the repeated execution of optimization routines.

The second group includes methods for online adjustment of weight matrices of controllers that do not require significant changes to existing algorithms and are a superstructure over existing systems. The paper [8] proposes a 2-stage method for taking into account priorities with the adjustment of weight matrices online. In the first stage, using the solution of the linear programming problem, the optimal achievable target values of CV are determined, taking into account the limitations and priorities of the control goals, as well as the optimal weight matrices of the controller. In the second stage, the controller with updated parameters issues optimal MV increments that provide tracking of target CV values. The method of adjusting weight matrices using a linear approximation of the relationship between the predictions of the output variables and the values of the weight matrices, as well as the subsequent solution of the linear programming problem with constraints, is known [9]. In this paper, the criterion of the optimization problem is based on the assessment of the quality of the transient process and does not take into account the uncertainty and priorities of CV. Paper [10] is devoted to the definition of weight matrices, taking into account the uncertainty of the plant in online mode by solving the optimization problem based on the simulation of the controller and aimed at minimizing the CV mismatch and minimizing the MV increment. Paper [11] considers the application of the particle swarm optimization (PSO) method for online adjustment of weight matrices of the controller without taking into account the uncertainty of the plant, control goals, and CV priorities; a root-mean-square error is used as a criterion for the optimization problem.

The third group includes methods for offline determination of weight matrices that do not require any changes in the parameters of the controller. Thus, in paper [12], using PSO, the weight matrices of the controller and the dimensions of the prediction and control horizons are determined, taking into account the uncertainty and priorities of CV in order to minimize CV mismatch. It should be noted that in this paper, in order to take into account the priorities of CV, it is necessary to set weights in the PSO criteria, which increase the penalty for violations for higher-priority variables. It is indicated that it is necessary to accurately select the weights in the criteria in order to obtain the expected result. However, in this case, the task of determining the weights that ensure the fulfillment of the priorities of CV is “transferred” from the weight matrices of the controller to the weights of the scalarized criterion. A similar method [13] for optimizing the weight matrices of the controller using PSO is known, taking into account the uncertainty, but this method does not take into account the priorities of CV. The paper [14] proposes a method for optimizing weight matrices using a genetic algorithm with a criterion based on fuzzy logic, which allows describing CV priorities and minimizing control error, without determining the weights related to CV in the genetic algorithm criterion. A method is known for determining weight matrices without taking into

account the uncertainty and priorities of CV by solving a multi-criteria optimization problem. This approach seeks a compromise solution — known as a Pareto-optimal point — where improving one criterion is not possible without degrading the other. The criteria used are CV control error and MV increment [15]. In a similar work [16], also considering the task of determining the optimal weight matrices as multi-criteria, CV priorities are taken into account. This approach requires specifying time constant multipliers for the controller models used in the optimization criteria, ensuring that higher-priority CVs respond more quickly than lower-priority ones. However, it does not define the specific difference in the response rate needed to guarantee strict adherence to priority levels. In [17], the weight matrices of the controller are optimized in order to increase the speed of the transient process and minimize the CV error. In [17], several operating points are considered when determining the optimal weights, and the uncertainty of the plant and the priority of CV are not considered.

In contrast to the references considered above, this paper proposes an algorithm for finding the ratios of the weight matrices of the controller based on the predictive model, taking into account the control objectives (CV priority), the parametric uncertainty of the plant, and the unmeasured disturbances acting through the control channel. Also, the proposed algorithm can be used both at the stage of synthesis of new MPCs and to adjust the parameters of existing MPCs without the need to change the structure of the existing control algorithm. In the present paper, it is proposed to consider the ratios of the weight matrices, since the behavior of the MPCs is mainly determined by the ratios, and not by the absolute values of the weights [9]. The application of the proposed algorithm for determining the parameters of the existing MPCs of the complex distillation column of the hydrocracking process unit is considered. It is demonstrated that the developed algorithm for determining the optimal weight matrices of the MPC, based on the proposed criterion, enables the achievement of the minimum permissible initial boiling point of the kerosene fraction and ensures its maximum recovery from the bottom of the stripping column with a thermosiphon under conditions of low coolant flow. Additionally, the algorithm provides stable liquid accumulation at the bottom of the stripping column.

2. PLANT AND CONTROL SYSTEM DESCRIPTION

A complex distillation column of the hydrocracking process unit is considered, designed to separate the incoming stable hydrogenate into a gasoline fraction, a kerosene fraction, a middle distillate, a heavy diesel fraction, and a hydrocracking residue. Figure 1 shows a flowsheet diagram of the considered process unit.

One of the target products of this complex distillation column is the kerosene fraction. In this regard, the control task under consideration is to maximize the withdrawal of the kerosene fraction by involving lighter hydrocarbons of the gasoline fraction in it. This redistribution is achieved by minimizing the initial boiling point (FC IBP) of the kerosene fraction through three key actions: decreasing the top temperature of column K1, reducing the bottom temperature of column K2, and increasing the 98% boiling point (FC 98%) of the kerosene fraction by maximizing the flow rate of the column K2 output to the boundary limit. The bottom temperature of column K2 is regulated by adjusting the flow rate of coolant through the thermosiphon reboiler (TO2), which operates without a steam space. The coolant flow rate is controlled by valves FV113 and FV113B. It is noted that in the area of low coolant flow rates, i.e. when opening the valve FV113 less than 10%, stable accumulation of liquid in the bottom of column K2 can not be maintained. In such cases, the control mode is typically switched from automatic to manual by the operator to restore stability. Based on this operational constraint, the following control objectives are defined in descending order of priority: (1) ensure stable liquid accumulation in the bottom of column K2, (2) minimize the FC IBP of the kerosene fraction, and (3) maximize the recovery of the kerosene fraction.

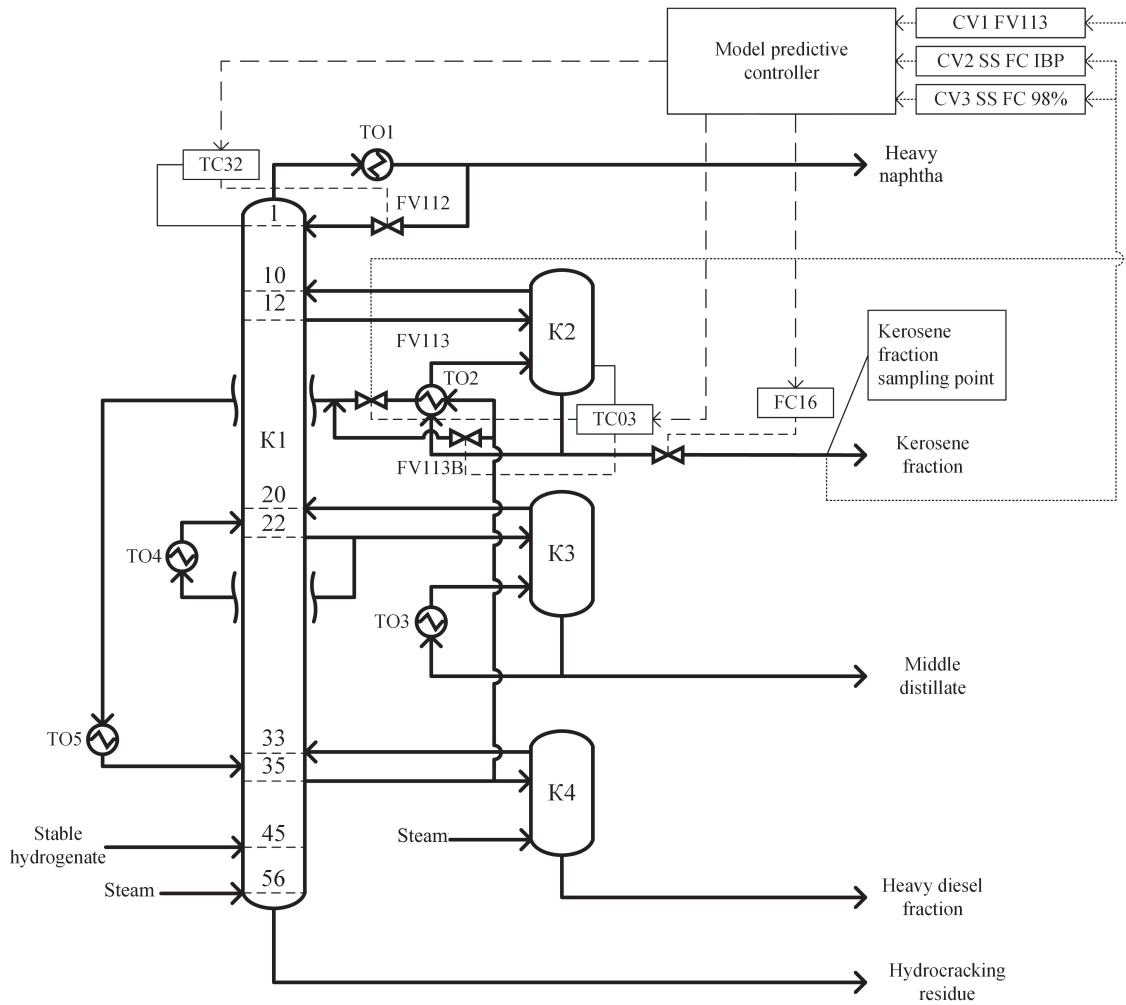


Fig. 1. Diagram of the process unit and control system: K1 – complex distillation column; K2 – kerosene fraction stripping column; K3 – diesel fraction stripping column; K4 – heavy diesel fraction stripping column; TO1 – gasoline fraction vapor condenser; TO2 – K2 column thermosiphon; TO3 – K3 column thermosiphon; TO4 – top pumparound stream cooler; TO5 – bottom pumparound stream cooler; FV112 – valve on the top reflux line of the K1 column; FV113 – valve at the outlet of the TO2 column; FV113B – valve on the TO2 bypass; SS FC IBP – soft sensor (SS) of the initial boiling point of the kerosene fraction; SS FC 98% – SS of the boiling point of the 98% kerosene fraction.

The control system consisting of n_{CV} controlled and n_{MV} control variables is considered. The MPC generates values of increments of control actions, taking into account the predicted values of the output variables by solving the quadratic optimization problem on each time period k :

$$\min_{\Delta U_k} \Phi_k = \frac{1}{2} \Delta U_k^T \left(S_f^T Q S_f + R \right) \Delta U_k + \left(\hat{E}_k^T Q S_f \right) \Delta U_k \quad (1)$$

subject to:

$$\left[\begin{array}{c|c|c} \mathbf{I}_{n_{MV}} & \dots & \mathbf{I}_{n_{MV}} \end{array} \right]^T \Delta \mathbf{u}_k^{LL} \leq \Delta \mathbf{U}_k \leq \left[\begin{array}{c|c|c} \mathbf{I}_{n_{MV}} & \dots & \mathbf{I}_{n_{MV}} \end{array} \right]^T \Delta \mathbf{u}_k^{HL}, \quad (2)$$

$n_{MV} \times n_{MV} M$ $n_{MV} \times n_{MV} M$

$$\left[\begin{array}{c|c|c} \mathbf{I}_{n_{MV}} & \dots & \mathbf{I}_{n_{MV}} \end{array} \right]^T \left(\mathbf{u}_k^{LL} - \mathbf{u}_k \right) \leq \mathbf{L} \Delta \mathbf{U}_k \leq \left[\begin{array}{c|c|c} \mathbf{I}_{n_{MV}} & \dots & \mathbf{I}_{n_{MV}} \end{array} \right]^T \left(\mathbf{u}_k^{HL} - \mathbf{u}_k \right), \quad (3)$$

$n_{MV} \times n_{MV} M$ $n_{MV} \times n_{MV} M$

where M — control horizon; $\Delta \mathbf{U}_k = \left[\Delta \mathbf{u}_k^T \mid \dots \mid \Delta \mathbf{u}_{k+l}^T \mid \dots \mid \Delta \mathbf{u}_{k+M-1}^T \right]^T$ — the vector of MV increments on the control horizon M ; $\Delta \mathbf{u}_{k+l} = (\Delta u_{k+l,1} \dots \Delta u_{k+l,r} \dots \Delta u_{k+l,n_{MV}})^T$ — the vector of MV increments for the l th step forward; $\Delta \mathbf{u}_k^{LL} = (\Delta u_{k,1}^{LL} \dots \Delta u_{k,r}^{LL} \dots \Delta u_{k,n_{MV}}^{LL})^T$ and $\Delta \mathbf{u}_k^{HL} = (\Delta u_{k,1}^{HL} \dots \Delta u_{k,r}^{HL} \dots \Delta u_{k,n_{MV}}^{HL})^T$ — the vectors of lower and upper limits on MV increments; $\mathbf{u}_k^{LL} = (u_{k,1}^{LL} \dots u_{k,r}^{LL} \dots u_{k,n_{MV}}^{LL})^T$ and $\mathbf{u}_k^{HL} = (u_{k,1}^{HL} \dots u_{k,r}^{HL} \dots u_{k,n_{MV}}^{HL})^T$ — the vectors of lower and upper limits on MV increments; $\mathbf{u}_k$ — the vector of actual MV values at a

time k ; $\mathbf{L} = \underbrace{\begin{bmatrix} \mathbf{I}_{n_{MV}} & 0 & \dots & 0 \\ \mathbf{I}_{n_{MV}} & \mathbf{I}_{n_{MV}} & \ddots & \vdots \\ \vdots & \vdots & \ddots & 0 \\ \mathbf{I}_{n_{MV}} & \mathbf{I}_{n_{MV}} & \dots & \mathbf{I}_{n_{MV}} \end{bmatrix}}_{n_{MV}M \times n_{MV}M}$ — the lower triangular block matrix consisting of diago-

nal unit matrices $\mathbf{I}_{n_{MV}}$; $\mathbf{S}_f = \begin{bmatrix} \mathbf{s}_1 & 0 & 0 & \dots & 0 \\ \mathbf{s}_2 & \mathbf{s}_1 & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \mathbf{s}_j & \mathbf{s}_{j-1} & \mathbf{s}_{j-2} & \dots & \mathbf{s}_{j-M+1} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \mathbf{s}_P & \mathbf{s}_{P-1} & \mathbf{s}_{P-2} & \dots & \mathbf{s}_{P-M+1} \end{bmatrix}$ — the matrix of plant dynamics

formed from step response (SR) coefficients; $\mathbf{s}_j = \begin{pmatrix} s_{j|1,1} & \dots & s_{j|1,r} & \dots & s_{j|1,n_{MV}} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ s_{j|q,1} & \dots & s_{j|q,r} & \dots & s_{j|q,n_{MV}} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ s_{j|n_{CV},1} & \dots & s_{j|n_{CV},r} & \dots & s_{j|n_{CV},n_{MV}} \end{pmatrix}$ —

the matrix of j th SR coefficients for CV and MV pairs; $s_{j|q,r}$ — the j th SR coefficient for q th

CV and r th MV; $\mathbf{Q} = \text{diag} \left(\underbrace{\left[\mathbf{I}_{n_{CV}} \mid \dots \mid \mathbf{I}_{n_{CV}} \right]^T}_{n_{CV} \times n_{CV}P} \mathbf{w}_{CV} \right)$ — a matrix for taking into account

CV weight coefficients when calculating (1); $\mathbf{w}_{CV} = (Q_1 \dots Q_q \dots Q_{n_{CV}})^T$ — the vector of CV weights; P — the prediction horizon; $\mathbf{R} = \text{diag} \left(\underbrace{\left[\mathbf{I}_{n_{MV}} \mid \dots \mid \mathbf{I}_{n_{MV}} \right]^T}_{n_{MV} \times n_{MV}M} \mathbf{w}_{MV} \right)$ — a matrix for tak-

ing into account MV weight coefficients when calculating (1); $\mathbf{w}_{MV} = (R_1 \dots R_r \dots R_{n_{MV}})^T$ — the vector of MV weights; $\hat{\mathbf{E}}_k = \left[\mathbf{e}_{k|1}^T \mid \dots \mid \mathbf{e}_{k|j}^T \mid \dots \mid \mathbf{e}_{k|P}^T \right]^T$ — a vector of errors of the prediction of unforced dynamics of all CV from references to the prediction horizon P in time k ; $\mathbf{e}_{k|j} = (e_{k|1,j} \dots e_{k|q,j} \dots e_{k|n_{CV},j})^T$ — a vector of errors of the prediction of unforced dynamics

of all CV to the j th step forward in time k ; $e_{k|q,j} = \begin{cases} y_q^{HL} - \tilde{y}_{k|q,j}, & \tilde{y}_{k|q,j} > y_q^{HL} \\ 0, & y_q^{LL} \leq \tilde{y}_{k|q,j} \leq y_q^{HL} \\ y_q^{LL} - \tilde{y}_{k|q,j}, & \tilde{y}_{k|q,j} < y_q^{LL} \end{cases}$ — a error

of the prediction of unforced dynamics of the q th CV from the reference for the j th step forward in time k ; y_q^{LL} — the lower limit of the q th CV; y_q^{HL} — the upper limit of the q th CV. Constraints (2) and (3) reflect the given limits for MV increments and limits for MV values.

3. STATEMENT OF THE PROBLEM

The vector of the optimized parameters $\mathbf{W}$ of the controller is as follows:

$$\mathbf{W} = \left(\mathbf{w}_{CV}^T \mid \mathbf{w}_{MV}^T \right)^T = (Q_1 \dots Q_q \dots Q_{n_{CV}} \ R_1 \dots R_r \dots R_{n_{MV}})^T.$$

In the industrial conditions, the models used in the controller may not correspond to the plant; in this regard, it is assumed that the values of the parameters of the transfer matrix of the plant model are in a known predetermined range. Therefore, in order to ensure the robustness of the MPC, it is necessary to determine the weights $\mathbf{W}$, taking into account the change in the parameters of the plant in the permissible range. Since the number of possible scenarios of variation of parameters in the range is large, in order to reduce the computational load, a limited number of scenarios N_S are used that reflect the maximum possible deviations of the parameters of the models used in the controller from the plant.

Determination of tuning parameters $\hat{\mathbf{W}}$ is carried out by solving the optimization problem with a given control performance function J :

$$\hat{\mathbf{W}} = \arg \min_{\mathbf{W} \in \mathbb{R}_{>0}} J(\mathbf{W}).$$

The following criterion is proposed as a control performance metric:

$$J = \sum_{m=1}^{N_S} \sum_{q=1}^{n_{CV}} \left(\sum_{k=1}^{N_T} D_{q,k}^{(m)} \sum_{k=1}^{N_T} B_{q,k}^{(m)} \right), \quad (4)$$

$$\text{where } D_{q,k}^{(m)} = \begin{cases} 0, & y_q^{LL} \leq y_{q,k}^{(m)}(\mathbf{W}) \leq y_q^{HL} \\ \left| y_q^{LL} - y_{q,k}^{(m)}(\mathbf{W}) \right|, & y_{q,k}^{(m)}(\mathbf{W}) < y_q^{LL} \\ \left| y_{q,k}^{(m)}(\mathbf{W}) - y_q^{HL} \right|, & y_{q,k}^{(m)}(\mathbf{W}) > y_q^{HL} \end{cases} \quad \text{— the value of exceeding the permis-}$$

sible boundaries of the q th CV at a time k ; N_T — the number of simulation time cycles; $y_{q,k}^{(m)}(\mathbf{W})$ — the value of the q th CV at a time k , obtained when used a controller with parameters $\mathbf{W}$ for sce-

nario m ; $B_{q,k}^{(m)} = \begin{cases} 0, & y_q^{LL} \leq y_{q,k}^{(m)}(\mathbf{W}) \leq y_q^{HL} \\ 1, & y_{q,k}^{(m)}(\mathbf{W}) < y_q^{LL} \vee y_{q,k}^{(m)}(\mathbf{W}) > y_q^{HL} \end{cases}$ — the flag value (discrete value) of violating the boundaries of the q th CV at a time k obtained for a controller with parameters $\mathbf{W}$ for scenario m .

The proposed criterion (4) takes into account the duration (time) and absolute values of the output of CVs beyond the boundaries. Uncertainty consideration is implemented by summarizing the values $\sum_{q=1}^{n_{CV}} \left(\sum_{k=1}^{N_T} D_{q,k}^{(m)} \sum_{k=1}^{N_T} B_{q,k}^{(m)} \right)$ for each of the N_S selected scenarios of parameters of controller model.

4. THE PROPOSED ALGORITHM FOR OBTAINING THE OPTIMAL PARAMETERS OF THE MPC

Due to the discontinuity of the proposed criterion (4) the problem of finding the optimal parameters of the controller cannot be solved by quadratic optimization methods. For this reason, the proposed algorithm for determining the optimal parameters of the controller is based on the grid search method [18, 19], which assumes that the value of the control performance function is calculated for all combinations of the values of the optimized variables within the given search space

and the combination with the lowest value of the criterion is selected. The proposed algorithm is presented in Table 1.

Table 1. Algorithm for determining the optimal parameters of the MPC

Inputs: $\mathbf{W}_{vars}, \Phi, J^*$	
Outputs: $\hat{\mathbf{W}}$	
1.	For $v = 1 \dots N_W$:
2.	Set $\tilde{J} = 0$
3.	For $m = 1 \dots N_S$:
4.	Reading $\mathbf{Y}^{m,v}$ if using $[\Phi]_m$
5.	Calculating $\tilde{J} := \tilde{J} + \sum_{q=1}^{n_{CV}} \left(\sum_{k=1}^{N_T} D_{q,k}^{(m)} \sum_{k=1}^{N_T} B_{q,k}^{(m)} \right)$
6.	If $\tilde{J} < J^*$:
7.	$\hat{\mathbf{W}} := [\mathbf{W}_{vars}]_v$
8.	$J^* := \tilde{J}$
9.	Return $\hat{\mathbf{W}}$

For the algorithm to work, it is necessary to set the matrix of combinations of possible values of the controller parameters $\mathbf{W}_{vars}$:

$$\mathbf{W}_{vars} = \begin{bmatrix} Q_{1,1} & \dots & Q_{1,q} & \dots & Q_{1,n_{CV}} & R_{1,1} & \dots & R_{1,r} & \dots & R_{1,n_{MV}} \\ \vdots & \ddots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ Q_{v,1} & \dots & Q_{v,q} & \dots & Q_{v,n_{CV}} & R_{v,1} & \dots & R_{v,r} & \dots & R_{v,n_{MV}} \\ \vdots & \ddots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ Q_{N_W,1} & \dots & Q_{N_W,q} & \dots & Q_{N_W,n_{CV}} & R_{N_W,1} & \dots & R_{N_W,r} & \dots & R_{N_W,n_{MV}} \end{bmatrix},$$

where N_W — the number of combinations of the parameters of the controller.

In this paper, first-order transfer function with time-delay is used as the transfer functions of the model of the plant $F(s) = \frac{g}{(\tau s + 1)} e^{-\theta s}$ due to its widespread in practice. To take into account the uncertainty in the algorithm for finding the optimal parameters of the controller, it is necessary to set the matrix of scenarios for the parameters of the transfer functions used:

$$\Phi = \begin{bmatrix} \mathbf{G}_1 & \mathbf{T}_1 & \mathbf{\Theta}_1 \\ \vdots & \vdots & \vdots \\ \mathbf{G}_m & \mathbf{T}_m & \mathbf{\Theta}_m \\ \vdots & \vdots & \vdots \\ \mathbf{G}_{N_S} & \mathbf{T}_{N_S} & \mathbf{\Theta}_{N_S} \end{bmatrix},$$

where $\mathbf{G}_m = (g_{i,j}^{(m)})_{1 \leq i \leq n_{CV}, 1 \leq j \leq n_{MV}}$ — the matrix of the controller's model gain coefficients for the scenario m ; $\mathbf{T}_m = (\tau_{i,j}^{(m)})_{1 \leq i \leq n_{CV}, 1 \leq j \leq n_{MV}}$ — the matrix of the controller's model time constants for the scenario m ; $\mathbf{\Theta}_m = (\theta_{i,j}^{(m)})_{1 \leq i \leq n_{CV}, 1 \leq j \leq n_{MV}}$ — the matrix of the controller's model delay values for the scenario m .

During the operation of the algorithm, for each scenario of the parameters of the transfer functions and for each combination of parameters of the controllers, the values of the measured output vector are read for the entire simulation time:

$$\mathbf{Y}^{(m,v)} = \left[\mathbf{y}_1^{(m)}([\mathbf{W}_{vars}]_v) \mid \dots \mid \mathbf{y}_k^{(m)}([\mathbf{W}_{vars}]_v) \mid \dots \mid \mathbf{y}_{N_T}^{(m)}([\mathbf{W}_{vars}]_v) \right],$$

where $\mathbf{y}_k^{(m)}([\mathbf{W}_{vars}]_v) = \left(y_{1,k}^{(m)}([\mathbf{W}_{vars}]_v) \cdots y_{q,k}^{(m)}([\mathbf{W}_{vars}]_v) \cdots y_{n_{CV},k}^{(m)}([\mathbf{W}_{vars}]_v) \right)^T$ — a vector of CV values at time k when using the scenario m of transfer functions parameters and a combination v of controller parameters; $[\cdot]_v$ — the v th row of the matrix.

At the beginning of the algorithm J^* , it is necessary to set some large threshold value of the function so that when using the combination 1 of the controller parameters, the calculated value of the function $\tilde{J}$ is less than J^* .

5. RESULTS AND DISCUSSION

Table 2 shows the transfer functions of the considered model of the plant.

Since the impact of various unmeasurable disturbances (UD) is possible on a real plant, for example, a change in the feed composition or a change in the characteristics of the coolant, additional transfer functions have been introduced into the plant model to take into account possible UD disturbances. These disturbances are used in the search for optimal parameters $\mathbf{W}$ in such a way as to create conflict situations in terms of priorities, when it is possible to keep only one CV within the boundaries.

Table 2. Coefficients of the transfer functions of the plant model

	MV1 TC32	MV2 TC03	MV3 FC16	UD1	UD2	UD3
CV1 FV113	$g_{1,1} = -6$ $\tau_{1,1} = 70$ $\theta_{1,1} = 30$	$g_{1,2} = 4$ $\tau_{1,2} = 70$ $\theta_{1,2} = 10$	$g_{1,3} = 0.3$ $\tau_{1,3} = 80$ $\theta_{1,3} = 15$	$g_{1,4} = 10$ $\tau_{1,4} = 30$ $\theta_{1,4} = 0$	$g_{1,5} = 0$ $\tau_{1,5} = 0$ $\theta_{1,5} = 0$	$g_{1,6} = 0$ $\tau_{1,6} = 0$ $\theta_{1,6} = 0$
CV2 K_IBP	$g_{2,1} = 2$ $\tau_{2,1} = 40$ $\theta_{2,1} = 20$	$g_{2,2} = 3.5$ $\tau_{2,2} = 30$ $\theta_{2,2} = 10$	$g_{2,3} = 0$ $\tau_{2,3} = 0$ $\theta_{2,3} = 0$	$g_{2,4} = 0$ $\tau_{2,4} = 0$ $\theta_{2,4} = 0$	$g_{2,5} = 10$ $\tau_{2,5} = 30$ $\theta_{2,5} = 0$	$g_{2,6} = 0$ $\tau_{2,6} = 0$ $\theta_{2,6} = 0$
CV3 K_T95	$g_{3,1} = 0.25$ $\tau_{3,1} = 55$ $\theta_{3,1} = 25$	$g_{3,2} = 0$ $\tau_{3,2} = 0$ $\theta_{3,2} = 0$	$g_{3,3} = 0.55$ $\tau_{3,3} = 15$ $\theta_{3,3} = 10$	$g_{3,4} = 0$ $\tau_{3,4} = 0$ $\theta_{3,4} = 0$	$g_{3,5} = 0$ $\tau_{3,5} = 0$ $\theta_{3,5} = 0$	$g_{3,6} = 10$ $\tau_{3,6} = 30$ $\theta_{3,6} = 0$

For the controller UD1, UD2 and UD3 are unmeasurable, therefore it is assumed that the controller matrices for the case without uncertainty coincide with the fragments of the plant model matrices:

$$\hat{\mathbf{G}} = (g_{i,j})_{1 \leq i,j \leq 3}, \quad \hat{\mathbf{T}} = (\tau_{i,j})_{1 \leq i,j \leq 3}, \quad \hat{\mathbf{\Theta}} = (\theta_{i,j})_{1 \leq i,j \leq 3}.$$

In this paper, the uncertainty, considered as a mismatch between the controller and plant parameter matrices, is set intervalized as a parameter difference of 35%. The parameters of the controller matrices with uncertainty are described as follows:

$$\begin{aligned} \overline{\mathbf{G}} &= \begin{bmatrix} -8.1 & 5.4 & 0.4 \\ 2.7 & 4.7 & 0 \\ 0.3 & 0 & 0.7 \end{bmatrix} \approx 1.35\hat{\mathbf{G}}, \quad \underline{\mathbf{G}} = \begin{bmatrix} -3.9 & 2.6 & 0.2 \\ 1.3 & 2.3 & 0 \\ 0.2 & 0 & 0.4 \end{bmatrix} \approx 0.65\hat{\mathbf{G}}, \\ \overline{\mathbf{T}} &= \begin{bmatrix} 94.5 & 94.5 & 108 \\ 54 & 40.5 & 0 \\ 74.3 & 0 & 20.3 \end{bmatrix} \approx 1.35\hat{\mathbf{T}}, \quad \underline{\mathbf{T}} = \begin{bmatrix} 45.5 & 45.5 & 52 \\ 26 & 19.5 & 0 \\ 35.8 & 0 & 9.8 \end{bmatrix} \approx 0.65\hat{\mathbf{T}}, \\ \overline{\mathbf{\Theta}} &= \begin{bmatrix} 40.5 & 13.5 & 20.3 \\ 27 & 13.5 & 0 \\ 33.8 & 0 & 13.5 \end{bmatrix} \approx 1.35\hat{\mathbf{\Theta}}, \quad \underline{\mathbf{\Theta}} = \begin{bmatrix} 19.5 & 6.5 & 9.8 \\ 13 & 6.5 & 0 \\ 16.3 & 0 & 6.5 \end{bmatrix} \approx 0.65\hat{\mathbf{\Theta}}. \end{aligned}$$

The given scenarios of the parameters of the controller matrices, considered in the search for optimal parameters $\hat{\mathbf{W}}$, are presented in Table 3.

Table 3. Options for model parameters used in the MPC

Index m of the scenario of models used in the controller	1	2	3	4	5	6	7	8	9
Parameters of models used in the controller	$\hat{\mathbf{G}}$	$\underline{\mathbf{G}}$	$\hat{\mathbf{G}}$	$\hat{\mathbf{G}}$	$\overline{\mathbf{G}}$	$\hat{\mathbf{G}}$	$\hat{\mathbf{G}}$	$\underline{\mathbf{G}}$	$\overline{\mathbf{G}}$
	$\hat{\mathbf{T}}$	$\hat{\mathbf{T}}$	$\underline{\mathbf{T}}$	$\hat{\mathbf{T}}$	$\hat{\mathbf{T}}$	$\overline{\mathbf{T}}$	$\hat{\mathbf{T}}$	$\underline{\mathbf{T}}$	$\overline{\mathbf{T}}$
	$\hat{\mathbf{\Theta}}$	$\hat{\mathbf{\Theta}}$	$\hat{\mathbf{\Theta}}$	$\underline{\mathbf{\Theta}}$	$\hat{\mathbf{\Theta}}$	$\hat{\mathbf{\Theta}}$	$\overline{\mathbf{\Theta}}$	$\underline{\mathbf{\Theta}}$	$\overline{\mathbf{\Theta}}$

The plant control simulation time is limited to $N_T = 300$ minutes. The initial values of the plant inputs and outputs are: CV1 = 116, CV2 = 205, CV3 = 75, MV1 = 12, MV2 = 145, MV3 = 241. Restrictions on increments for MV1, MV2 and MV3 are set to ± 0.25 , ± 0.25 and ± 0.33 , respectively. At the same time, the permissible limits for CV1 = [10 ... 90], CV2 = [135 ... 145], CV3 = [235 ... 245], MV1 = [112,5 ... 118], MV2 = [203 ... 210] and MV3 = [75 ... 85].

The behavior of the specified UDs is defined as follows: from the beginning of the simulation, UD 1, UD 2 and UD 3 are equal to 0, at time 50, the value of UD 1 changed by -1.5 , at time 100, the value of UD 2 and UD 3 became equal to -1.7 and 1 , respectively, at time 150, the values of UD 2 and UD 3 became equal to 0.

Since the goal is to find the optimal ratios of the weights $\mathbf{w}_{CV}$ and $\mathbf{w}_{MV}$, the search space is set as follows:

$$\mathbf{W}_{vars} \subseteq \mathbb{N}^*,$$

$$\mathbf{W}_{vars} = \left\{ (Q_1 Q_2 Q_3 R_1 R_2 R_3)^T \mid Q_1 = 8; Q_2, Q_3 \in \{1, \dots, 7\}; R_1, R_2, R_3 \in \{1, \dots, 8\} \right\}.$$

In order to reduce the time needed to iterate through possible controller parameters, the search space is limited to positive integers that are less than or equal to 8, and a fixed weight of the highest priority CV1 is set to 8. In this case, the number of possible combinations of controller parameters is $N_W = 25\,088$. If we increase the search space and limit it to 10, then the number of possible settings will increase to $N_W = 81\,000$, that is, a slight expansion of the search space significantly increases the number of combinations of controller parameters. Due to the fact that the weights of the lower priority CV are always less than the weight of the more important one, the priorities of the control goals described earlier are taken into account. It should be noted that there are no fundamental restrictions preventing the expansion of the search space to the set $\mathbb{R}_{>0}$.

To confirm the efficiency of the proposed criterion, alternative versions of the criterion that consider either only the magnitude or only the timing of CV violations of acceptable limits are considered:

$$J_D = \sum_{m=1}^{N_S} \sum_{q=1}^{n_{CV}} \sum_{k=1}^{N_T} D_{q,k}^{(m)}, \quad (5)$$

$$J_B = \sum_{m=1}^{N_S} \sum_{q=1}^{n_{CV}} \sum_{k=1}^{N_T} B_{q,k}^{(m)}. \quad (6)$$

For a comparative analysis, the optimal MPC parameters were determined for the case without taking into account the uncertainty (only the scenario $m = 1$, Table 3): when using the proposed criterion (4) — $\mathbf{W}_{opt}^{(J)} = (8 \ 2 \ 5 \ 1 \ 1 \ 7)^T$; by criterion (5) — $\mathbf{W}_{opt}^{(J_D)} = (8 \ 3 \ 7 \ 1 \ 1 \ 1)^T$; by criterion (6) — $\mathbf{W}_{opt}^{(J_B)} = (8 \ 1 \ 1 \ 1 \ 1 \ 7)^T$. Figure 2 shows graphs of the change in CV and MV when using the found optimal parameters of the controller $\mathbf{W}_{opt}^{(J)}$, $\mathbf{W}_{opt}^{(J_D)}$ and $\mathbf{W}_{opt}^{(J_B)}$.

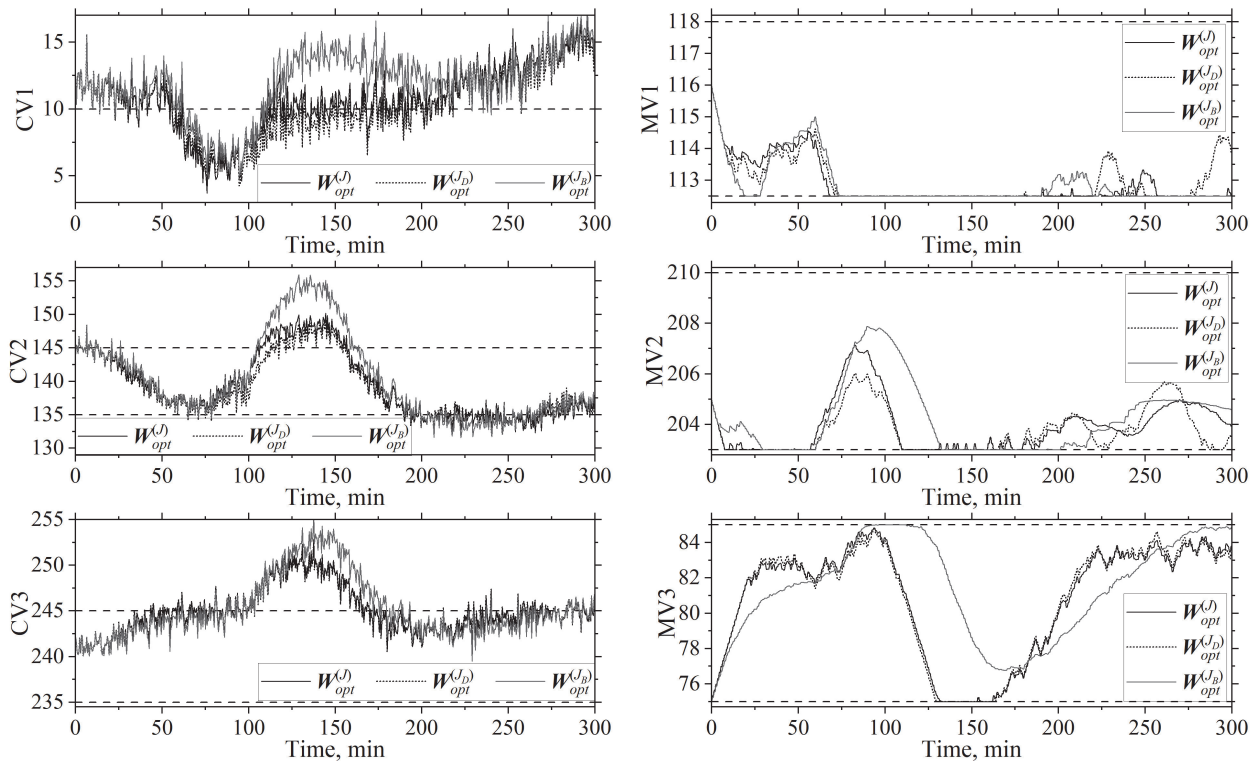


Fig. 2. The simulation results with the MPC parameters $\mathbf{W}_{opt}^{(J)}$, $\mathbf{W}_{opt}^{(J_D)}$ and $\mathbf{W}_{opt}^{(J_B)}$.

According to Fig. 2, it can be concluded that criterion J_B (6) makes it possible to find controller parameters that ensure the fulfillment of priorities, but at the same time the boundaries on CV1 and CV2 are largely violated; criterion J_D (5) makes it possible to find controller parameters that provide a minimum overshooting of boundaries on all CVs, but priorities are not always met; criterion J (4) makes it possible to find such parameters that equally take into account both priorities and values of CV overshooting.

Table 4 shows the optimal parameters (ratios between weights) found by the criterion J for each case of models used in the controller.

Table 4. MPC parameters

$\tilde{m}$ scenario index of controller models	1	2	3	4	5	6	7	8	9
J criterion optimal values of controller parameters for scenario $\tilde{m}$ — $\mathbf{W}_{opt}^{(J \tilde{m})}$	$\begin{pmatrix} 8 \\ 2 \\ 5 \\ 1 \\ 1 \\ 7 \end{pmatrix}$	$\begin{pmatrix} 8 \\ 3 \\ 1 \\ 8 \\ 1 \\ 7 \end{pmatrix}$	$\begin{pmatrix} 8 \\ 1 \\ 6 \\ 4 \\ 1 \\ 7 \end{pmatrix}$	$\begin{pmatrix} 8 \\ 3 \\ 7 \\ 1 \\ 1 \\ 2 \end{pmatrix}$	$\begin{pmatrix} 8 \\ 1 \\ 7 \\ 7 \\ 1 \\ 7 \end{pmatrix}$	$\begin{pmatrix} 8 \\ 2 \\ 7 \\ 1 \\ 1 \\ 6 \end{pmatrix}$	$\begin{pmatrix} 8 \\ 1 \\ 7 \\ 1 \\ 8 \\ 1 \end{pmatrix}$	$\begin{pmatrix} 8 \\ 3 \\ 1 \\ 1 \\ 1 \\ 7 \end{pmatrix}$	$\begin{pmatrix} 8 \\ 1 \\ 7 \\ 1 \\ 8 \\ 5 \end{pmatrix}$

Taking into account the uncertainty, the optimal parameters were found according to criterion J (4) $\mathbf{W}_{opt}^{(J)} = (8 \ 2 \ 6 \ 1 \ 1 \ 5)^T$. Figures 3 and 4 show a comparison of the simulation results using the controller parameters $\mathbf{W}_{opt}^{(J)}$ and the same weight ratios $\mathbf{W}_{std} = (1 \ 1 \ 1 \ 1 \ 1 \ 1)^T$ for the scenarios $\tilde{m} = 1$ and $\tilde{m} = 8$, accordingly.

The results presented in Figs. 3 and 4 demonstrate the efficiency of the proposed algorithm for finding the optimal parameters of the controller. When using the parameters of the controller $\mathbf{W}_{opt}^{(J)}$,

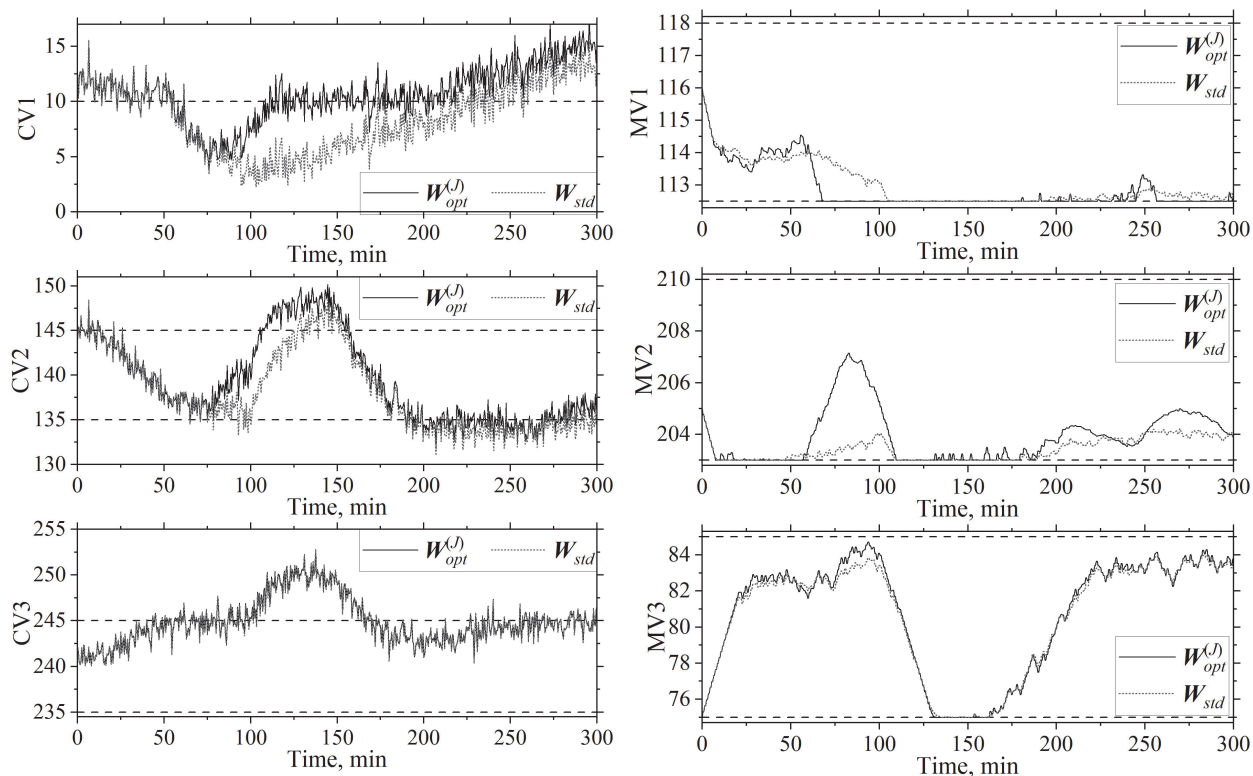


Fig. 3. Comparison of the simulation results using the controller parameters $W_{opt}^{(J)}$ and W_{std} for the scenario $\tilde{m} = 1$.

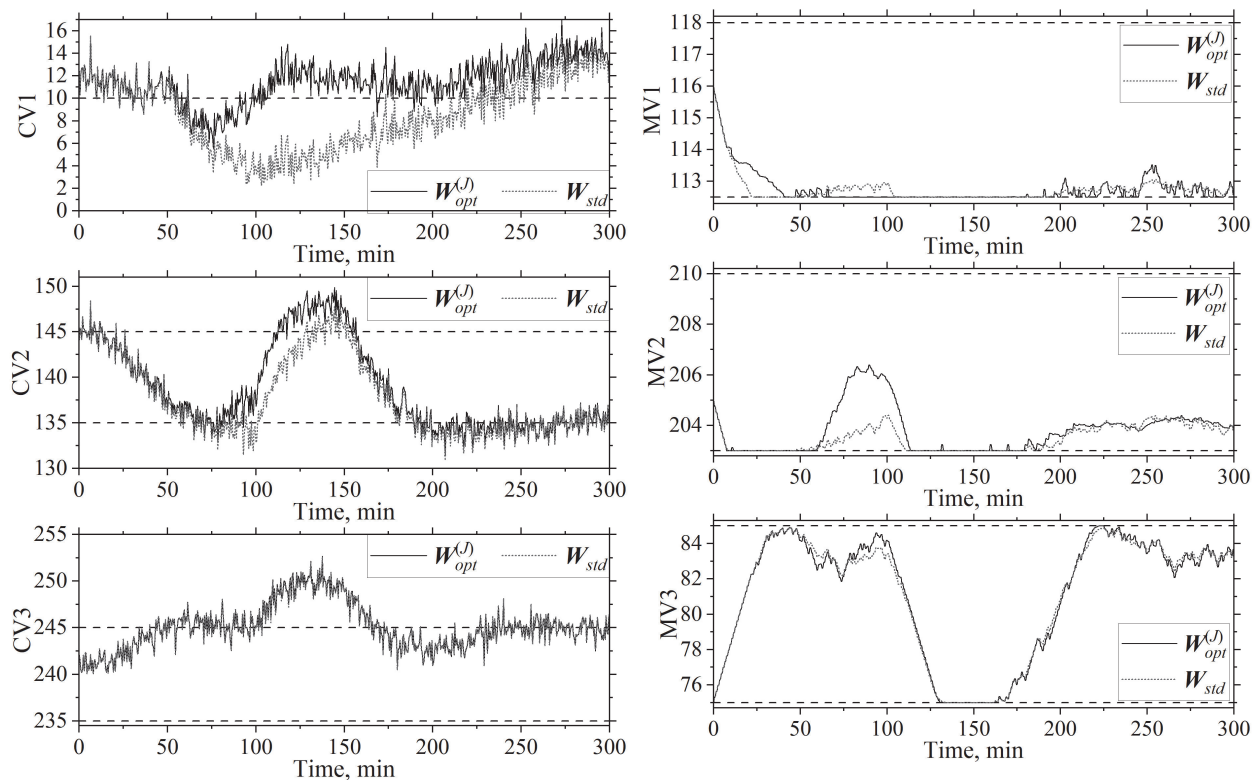


Fig. 4. Comparison of the simulation results using the controller parameters $W_{opt}^{(J)}$ and W_{std} for the scenario $\tilde{m} = 8$.

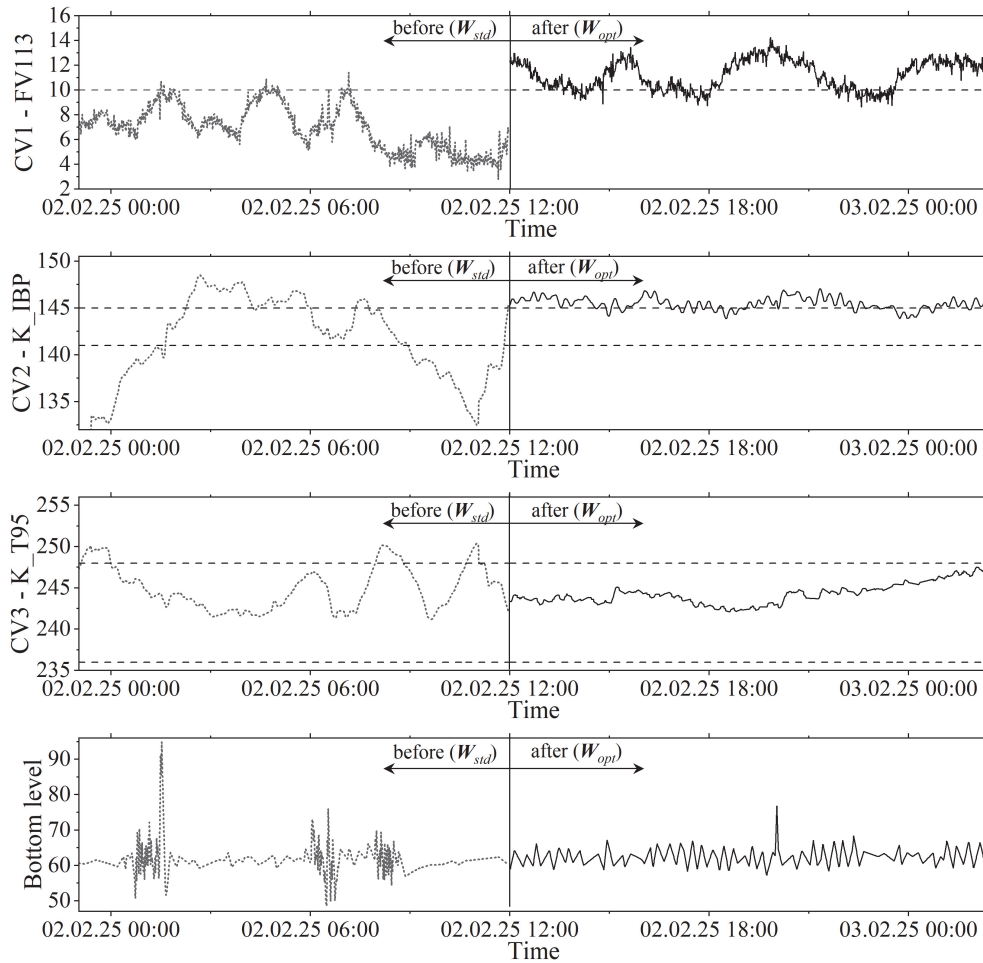


Fig. 5. Comparison of the output variables of the process unit in industrial conditions before and after determining the optimal parameters of the controller.

the time of violation of the boundaries for the most priority CV1 decreased by 42.6% (from 185 min. to 106 min.), 56.2% (from 137 min. to 60 min.) and 29.9% (from 204 min. to 143 min.) for the scenarios of the controller matrices $\tilde{m} = 1$ and $\tilde{m} = 8$, accordingly, in comparison with the use of $\mathbf{W}_{std}$. Figure 5 shows the output variables of an industrial process unit when controlled with the parameters $\mathbf{W}_{std}$ and $\mathbf{W}_{opt}^{(J)}$, respectively. The presented comparison plot makes it possible to conclude that the use of the found optimal parameters of the controller made it possible to prevent a significant violation of the boundaries CV1 — FV113, thereby ensuring a stable accumulation of liquid in the bottom of the K2 column. Also, the optimal parameters found made it possible to stabilize the fractional composition of the kerosene fraction taken from the bottom of the K2 column and to achieve the minimum possible initial boiling point of the kerosene fraction.

6. CONCLUSION

The proposed algorithm for determining the weight matrices enables the determination of the optimal MPC parameters — specifically, the relative weighting factors — under conditions of plant parametric uncertainty, unmeasured disturbances, and control objective prioritization. For the model example under consideration with the controller parameters determined using the proposed algorithm for the highest priority controlled variable, the boundary violation time decreased by an average of 42.9% in comparison with non-optimal controller parameters. Using the example of an

industrial MPC of a complex distillation column in a hydrocracking process unit, it demonstrates that the parameters determined by the proposed algorithm enable stabilization of the fractional composition and achievement of the minimum possible initial boiling point of the kerosene fraction withdrawn from the bottom of the stripping column with a thermosiphon, even under low coolant flow conditions.

FUNDING

The research was carried out within the state assignment of IACP FEB RAS on theme FW-2025-0002 (algorithm for obtaining optimal MPC parameters in section 4) and theme FFW-2021-0003 (implementation of the proposed algorithm and its testing in industrial conditions).

REFERENCES

1. Cheresenko, A.A. and Titkina, M.S., Application of associative search algorithms in model predictive control, *Automation in industry*, 2022, no. 6, pp. 58–62. <https://doi.org/10.25728/avtprom.2022.06.13>
2. Cheresenko, A.A., Bakhtadze, N.N., Kushnarev, V.N., and Elpashev, D.V., A method for proactive control of complex multi-connected plants based on digital associative models, *Automation in industry*, 2024, no. 10, pp. 16–18. <https://doi.org/10.25728/avtprom.2024.10.03>
3. Komogorov, A.V., Lavrentiev, V.A., and Sineglazov, D.S., Automation of data reconciliation and process variables selection for advanced process control systems, *Automation in industry*, 2023, no. 12, pp. 25–27. <https://doi.org/10.25728/avtprom.2023.12.04>
4. Anilkumar, M., Padhiyar, N., and Moudgalya, K., Lexicographic optimization based MPC: Simulation and experimental study, *Computers & Chemical Engineering*, 2016, vol. 88, pp. 135–144. <https://doi.org/10.1016/j.compchemeng.2016.02.002>
5. Aliyev, T.L. and Gatzke, E.P., Prioritized constraint handling NMPC using Volterra series models, *Optimal Control Applications and Methods*, 2010, vol. 31, no. 5, pp. 415–432. <https://doi.org/10.1002/oca.958>
6. Long, C.E. and Gatzke, E.P., Model Predictive control algorithm for prioritized objective inferential control of unmeasured states using propositional logic, *Industrial & Engineering Chemistry Research*, 2005, vol. 44, no. 10, pp. 3575–3584. <https://doi.org/10.1021/ie049287p>
7. Gholaminejad, T., Khaki-Sedigh, A., and Bagheri, P., Adaptive tuning of model predictive control based on analytical results, *2016 4th International Conference on Control, Instrumentation, and Automation (ICCIA)*, 2016, pp. 226–232. <https://doi.org/10.1109/ICCIAutom.2016.7483165>
8. Wojsznis, W., Mehta, A., and Wojsznis, P., Multi-objective optimization for model predictive control, *ISA Transactions*, 2007, vol. 46, no. 3, pp. 351–361. <https://doi.org/10.1016/j.isatra.2006.10.002>
9. Al-Ghazzawi, A., Ali, E., Nouh, A., and Zafirou, E., On-line tuning strategy for model predictive controllers, *Journal of Process Control*, 2001, vol. 11, no. 3, pp. 265–284. [https://doi.org/10.1016/S0959-1524\(00\)00033-0](https://doi.org/10.1016/S0959-1524(00)00033-0)
10. Fontes, R.M., Martins, M.A.F., and Odloak, D., An Automatic Tuning Method for Model Predictive Control Strategies, *Industrial & Engineering Chemistry Research*, 2019, vol. 58, no. 47, pp. 21602–21613. <https://doi.org/10.1021/acs.iecr.9b03502>
11. Kawai, F., Ito, H., and Nakazawa, C., Automatic tuning for model predictive control: Can particle swarm optimization find a better parameter?, *2007 IEEE 22nd International Symposium on Intelligent Control*, 2007, pp. 646–651. <https://doi.org/10.1109/ISIC.2007.4450961>
12. Nery Júnior, G.A., Martins, M.A.F., and Kalid, R., A PSO-based optimal tuning strategy for constrained multivariable predictive controllers with model uncertainty, *ISA Transactions*, 2014, vol. 53, no. 2, pp. 560–567. <https://doi.org/10.1016/j.isatra.2013.12.019>
13. Suzuki, R., Kawai, F., and Nakazawa, C., Parameter optimization of model predictive control by PSO, *Electrical Engineering in Japan*, 2012, vol. 178, no. 1, pp. 40–49. <https://doi.org/10.1002/EEJ.21188>

14. Van Der Lee, J.H., Svrcek, W.Y., and Young, B.R., A tuning algorithm for model predictive controllers based on genetic algorithms and fuzzy decision making, *ISA Transactions*, 2008, vol. 47, iss. 1, pp. 53–59. <https://doi.org/10.1016/j.isatra.2007.06.003>
15. Vallerio, M., Van Impe, J., and Logist, F., Tuning of NMPC controllers via multi-objective optimization, *Computers & Chemical Engineering*, 2014, vol. 61, pp. 38–50. <https://doi.org/10.1016/j.compchemeng.2013.10.003>
16. Yamashita, A.S, Zanin, A.C., and Odloak, D., Tuning the model predictive control of a crude distillation unit, *ISA Transactions*, 2016, vol. 60, pp. 178–190. <https://doi.org/10.1016/j.isatra.2015.10.017>
17. Santos, J.E.W., Trierweiler, J.O., and Farenzena, M., Robust tuning for classical MPC through the multi-scenarios approach, *Industrial & Engineering Chemistry Research*, 2019, vol. 58, no. 8, pp. 3146–3158. <https://doi.org/10.1021/acs.iecr.8b05485>
18. Snoek, J., Larochelle, H., and Adams, R.P., Practical Bayesian optimization of machine learning algorithms, *Advances in Neural Information Processing Systems*, 2012, vol. 25.
19. Bergstra, J. and Bengio, Y., Random search for hyper-parameter optimization, *The Journal of Machine Learning Research*, 2012, vol. 13, pp. 281–305.

This paper was recommended for publication by A.A. Galyaev, a member of the Editorial Board