30 січня 2025 р. виповнюється 75 років

відомому українському вченому, фахівцю в галузі кібернетики, системного аналізу, теорії прийняття рішень, тричі лауреату Державної премії України в галузі науки і техніки (1990, 1999, 2005), заслуженому діячеві науки і техніки України (2000), повному кавалеру ордена «За заслуги» (1996, 1998, 2005), лауреату премій НАН України: імені В.М. Глушкова (1995), імені В.С. Михалевича (2004) та імені С.О. Лебедєва (2019). Почесному доктору НАН України (2020), доктору технічних наук (1984), професору (1985), академіку НАН України (1995), академіку НАПН України (1995), іноземному члену-кореспонденту Австрійської академії наук (2022)

Михайлу Захаровичу Згуровському.

Михайло Захарович Згуровський народився в м. Скала-Подільська Борщівського району Тернопільської області. У 1975 р. він закінчив Київський політехнічний інститут (нині — Національний технічний університет України «Київський політехнічний інститут ім. Ігоря Сікорського» — КПІ) за спеціальністю «автоматизовані системи управління». Кандидатську дисертацію «Оптимальне дискретне управління одним класом розподілених

процесів нестаціонарного теплообміну» захистив у 1979 р., докторську «Автоматизоване проєктування та оптимальне управління нестаціонарними процесами і полями в умовах невизначеності даних» — у 1984 р. У 1987 р. став професором кафедри технічної кібернетики КПІ, у 1988—1992 рр. був проректором з навчальної, роботи, а в 1992—2024 рр. — ректором Київської політехніки. Водночас у 1996 р. він заснував і до 2015 р. очолював Інститут прикладного системного аналізу МОН України та НАН України (ІПСА), з 2015 р. і дотепер є його науковим керівником. У 1994—1999 рр. був Міністром освіти України.

М.З. Згуровський здійснює масштабну громадську діяльність Він є членом багатьох українських та закордонних наукових товариств, національним представником України в Міжнародній раді з науки (ISC, Париж, Франція), членом Поважної ради Ордена святого Пантелеймона, Головою Української ради миру. Останніми роками його громадська робота спрямована на консолідацію міжнародних організацій і видатних діячів освіти, науки і культури з різних країн світу навколо мети щодо досягнення справедливого миру для України. Зокрема, Михайло Захарович був співорганізатором і учасником семінару ООН «Екологічна безпека під час війни та збройних конфліктів» (6 листопада 2024 р., Нью-Йорк); II Саміту «Партнерство та здоров'я ветеранів» (Європарламент, 29–30 жовтня, Брюссель, Бельгія); Міжнародної конференції «Європа та Україна: спільні перспективи та цінності», яка проходила на базі Австрійського інституту європейської політики безпеки (13 грудня 2024 р., Відень, Австрія) та інших важливих заходів.

Наукова громадськість, колеги, учні
щиро вітають
Михайла Захаровича Згуровського
з ювілеєм, бажають йому активного
довголіття, невичерпного натхнення
і нових досягнень на благо Вітчизни.

НАЦІОНАЛЬНА АКАДЕМІЯ НАУК УКРАЇНИ
НАВЧАЛЬНО-НАУКОВИЙ КОМПЛЕКС
«ІНСТИТУТ ПРИКЛАДНОГО СИСТЕМНОГО АНАЛІЗУ»
НАЦІОНАЛЬНОГО ТЕХНІЧНОГО УНІВЕРСИТЕТУ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ ІМЕНІ ІГОРЯ СІКОРСЬКОГО»

СИСТЕМНІ ДОСЛІДЖЕННЯ ТА ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ

МІЖНАРОДНИЙ НАУКОВО–ТЕХНІЧНИЙ ЖУРНАЛ

№ 4 2024

ЗАСНОВАНО У ЛИПНІ 2001 р.

АДРЕСА РЕДАКЦІЇ:
03056, м. Київ,
просп. Перемоги, 37, корп. 35,
ННК «ІПСА» КПІ ім. Ігоря Сікорського
Тел.: **204-81-44**; факс: **204-81-44**
E-mail: journal.iasa@gmail.com
http://journal.iasa.kpi.ua

У номері:

NATIONAL ACADEMY OF SCIENCES OF UKRAINE
EDUCATIONAL AND SCIENTIFIC COMPLEX
«INSTITUTE FOR APPLIED SYSTEM ANALYSIS»
OF THE NATIONAL TECHNICAL UNIVERSITY OF UKRAINE
«IGOR SIKORSKY KYIV POLYTECHNIC INSTITUTE»

SYSTEM RESEARCH AND INFORMATION TECHNOLOGIES

INTERNATIONAL SCIENTIFIC AND TECHNICAL JOURNAL

№ 4 2024

IT IS FOUNDED IN JULY 2001

THE EDITION ADDRESS:

03056, Kyiv,
av. Peremogy, 37, building 35,
Institute for Applied System Analysis
at the Igor Sikorsky Kyiv Polytechnic Institute
Phone: **204-81-44**; Fax: **204-81-44**
E-mail: journal.iasa@gmail.com
http://journal.iasa.kpi.ua

In the issue:

Шановні читачі!

Навчально-науковий комплекс «Інститут прикладного системного аналізу» Національного технічного університету України «Київський політехнічний інститут імені Ігоря Сікорського» видає міжнародний науково-технічний журнал

«СИСТЕМНІ ДОСЛІДЖЕННЯ ТА ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ».

Журнал публікує праці теоретичного та прикладного характеру в широкому спектрі проблем, що стосуються системних досліджень та інформаційних технологій.

Провідні тематичні розділи журналу:

Теоретичні та прикладні проблеми і методи системного аналізу; теоретичні та прикладні проблеми інформатики; автоматизовані системи управління; прогресивні інформаційні технології, високопродуктивні комп'ютерні системи; проблеми прийняття рішень і управління в економічних, технічних, екологічних і соціальних системах; теоретичні та прикладні проблеми інтелектуальних систем підтримання прийняття рішень; проблемно і функціонально орієнтовані комп'ютерні системи та мережі; методи оптимізації, оптимальне управління і теорія ігор; математичні методи, моделі, проблеми і технології дослідження складних систем; методи аналізу та управління системами в умовах ризику і невизначеності; евристичні методи та алгоритми в системному аналізі та управлінні; нові методи в системному аналізі, інформатиці та теорії прийняття рішень; науково-методичні проблеми в освіті.

Головний редактор журналу — науковий керівник ННК «ІПСА» КПІ ім. Ігоря Сікорського, академік НАН України Михайло Захарович Згуровський.

Журнал «Системні дослідження та інформаційні технології» включено до переліку наукових фахових видань України (категорія «А»).

Журнал «Системні дослідження та інформаційні технології» входить до таких наукометричних баз даних: Scopus, EBSCO, Google Scholar, DOAJ, Index Copernicus, реферативна база даних «Україніка наукова», український реферативний журнал «Джерело», наукова періодика України.

Статті публікуються українською та англійською мовами.

Журнал рекомендовано передплатити. **Наш індекс 23918.** Якщо ви не встигли передплатити журнал, його можна придбати безпосередньо в редакції за адресою: 03056, м. Київ, просп. Перемоги, 37, корп. 35.

Завідувачка редакції **С.М. Шевченко**
Редакторка **Р.М. Шульженко**
Молодша редакторка **Л.О. Тарин**
Комп'ютерна верстка, дизайн **А.А. Патіохи**
Рішення Національної ради України з питань телебачення і радіомовлення №1794 від 21.12.2023. Ідентифікатор медіа R30-02404

Підписано до друку 25.12.2024. Формат 70×108 1/16. Папір офс. Гарнітура Times.
Спосіб друку – цифровий. Ум. друк. арк. 14,411. Обл.-вид. арк. 28,56. Наклад 106 пр. Зам. № 11/04

Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»
Свідоцтво про державну реєстрацію: ДК № 5354 від 25.05.2017 р.
просп. Перемоги, 37, м. Київ, 03056.
ФОП Пилипенко Н.М., вул. Мічуріна, б. 2/7, м. Київ, 01014. тел. (044) 361 78 68.
Виписка з Єдиного державного реєстру № 2 070 000 0000 0214697 від 17.05.2019 р.

Dear Readers!

Educational and Scientific Complex «Institute for Applied System Analysis» of the National Technical University of Ukraine «Igor Sikorsky Kyiv Polytechnic Institute» is published of the international scientific and technical journal

«SYSTEM RESEARCH AND INFORMATION TECHNOLOGIES».

The Journal is printing works of a theoretical and applied character on a wide spectrum of problems, connected with system researches and information technologies.

The main thematic sections of the Journal are the following:

Theoretical and applied problems and methods of system analysis; theoretical and applied problems of computer science; automated control systems; progressive information technologies, high-efficiency computer systems; decision making and control in economic, technical, ecological and social systems; theoretical and applied problems of intellectual systems for decision making support; problem- and function-oriented computer systems and networks; methods of optimization, optimum control and theory of games; mathematical methods, models, problems and technologies for complex systems research; methods of system analysis and control in conditions of risk and uncertainty; heuristic methods and algorithms in system analysis and control; new methods in system analysis, computer science and theory of decision making; scientific and methodical problems in education.

The editor-in-chief of the Journal is scientific director of the Institute for Applied System Analysis at the Igor Sikorsky Kyiv Polytechnic Institute, academician of the NASU Michael Zaharovich Zgurovsky.

The articles to be published in the Journal in Ukrainian and English languages are accepted. Information printed in the Journal is included in the Catalogue of periodicals of Ukraine.

СИСТЕМНІ ДОСЛІДЖЕННЯ ТА ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ

4 • 2024

ЗМІСТ

SYSTEM RESEARCH AND INFORMATION TECHNOLOGIES

4 • 2024

CONTENT

ON THE OCCASION OF THE 75th ANNIVERSARY OF NAS ACADEMICIAN M.Z. ZGUROVSKY

SCIENTIST AND ORGANIZER OF ENGINEERING EDUCATION

N.D. PANKRATOVA

On January 30, 2025, the renowned Ukrainian scientist, expert in the fields of cybernetics, systems analysis, decision theory, three-time laureate of the State Prize of Ukraine in Science and Technology (1990, 1999, 2005), Honored Scientist of Ukraine (2000), Full Cavalier of the Order of Merit (1996, 1998, 2005), laureate of NASU prizes named after V.M. Glushkov (1995), V.S. Mikhalevich (2004), and S.O. Lebedev (2019), Honorary Doctor of the NASU (2020), Doctor of Technical Sciences (1984), Professor (1985), Academician of the NASU (1995), Academician of the NAPS (1995), and Foreign Corresponding Member of the Austrian Academy of Sciences (2022), Mykhailo Zakharovych Zgurovsky, celebrates his 75th anniversary.

Mykhailo Zakharovych Zgurovsky was born in Skala-Podilska town, Borshchiv District, Ternopil Oblast. In 1975, he graduated from Kyiv Polytechnic Institute (now the National Technical University of Ukraine "Igor Sikorsky Kyiv Polytechnic Institute" — KPI) with a degree in Automated Control Systems. He defended his Candidate of Sciences dissertation, "Optimal Discrete Control of a Class of Distributed Processes of Nonstationary Heat Exchange," in 1979, and his Doctoral dissertation, "Automated Design and Optimal Control of Nonstationary Processes and Fields under Uncertainty," in 1984. In 1987, he became a Professor at the Department of Technical Cybernetics at KPI. From 1988 to 1992, he served as Vice-Rector for Academic Affairs, and from 1992 to 2024, he was the Rector of Kyiv Polytechnic.

In 1996, Zgurovsky founded and, until 2015, directed the Institute for Applied System Analysis (IASA) under the Ministry of Education and Science of Ukraine and the NASU. Since 2015, he has served as its Scientific Director. From 1994 to 1999, he was the Minister of Education of Ukraine.

By his anniversary, Mykhailo Zakharovych Zgurovsky has made a significant contribution to science, education, and public life. In the realm of science, M.Z. Zgurovsky has generalized the fundamental principles of systems analysis, laid the foundation for systemic mathematics, and proposed a new approach to the theory of extremal problems for nonlinear operator, differential-operator equations, inclusions, and variational inequalities. The most prominent applications of his scientific research are in the fields of mathematical geophysics and geoinformatics, contributing to solving the socio-economic challenges of modern society.

While serving as the Minister of Education of Ukraine, M.Z. Zgurovsky continued the work initiated by P.M. Talanchuk, the first Minister of Education of independent Ukraine. He implemented the core principles of the **State National**

Program "Education" ("Ukraine in the 21st Century"), which was approved by the First All-Ukrainian Congress of Education Workers in December 1992. Under Minister Zgurovsky leadership, a comprehensive legislative framework for Ukraine's education system was developed.

A new generation of Ukrainian-language textbooks was created, and the concept of humanitarian education in Ukraine was introduced. He established an accreditation system for higher education institutions, with its permanent collegial body—the **State Accreditation Commission of Ukraine** (a predecessor of the current **National Agency for Higher Education Quality Assurance (NAQA)**). The structure of educational fields and specialties in higher education was aligned with the needs of the economy and society of the newly independent state.

For the first time, a mandatory entrance exam in the Ukrainian language was introduced in higher education institutions. Many other significant initiatives were implemented, all aimed at developing the human capital of an independent Ukraine.

Upon assuming the role of Rector of Igor Sikorsky Kyiv Polytechnic Institute (KPI), Mykhailo Zgurovsky developed a new vision for its further development. His concept focused on transforming the institution from a large polytechnic institute of the Soviet-era centralized economy into a European-style technical university, characterized by universal and broad-oriented training to meet the evolving needs of society. Today, KPI consistently ranks among the top 4% of universities worldwide according to various international rankings.

To achieve this transformation and address the demands of the newly independent Ukraine, new faculties and institutes were established, new departments were created, and over 150 new specialties and specializations were introduced. Significant reforms were implemented to facilitate KPI's integration into the European educational and research space.

During his 32-year tenure as Rector, Mykhailo Zghurovsky was guided by core principles he articulated in an interview: "*Always remember that I come from KPI and owe my development and growth to it. Therefore, under any circumstances or position, I must work for its advancement and prestige, support and protect my colleagues at my alma mater; respect the honor and dignity of every person, regardless of their social status or position, and strive to understand the perspective of everyone who approaches the Rector; affirm respect and reverence for veterans and seniors, who created everything the university has today and are carriers of wisdom and unique experience for the new generation; inherit the best practices for KPI from my predecessors; see in every student a talented individual and in every staff member a like-minded colleague and ally; be grateful for accomplishments and capable of forgiving mistakes and weaknesses*."

Thanks to Zgurovsky's energy, his scientific talent, and his leadership as an organizer of engineering education, KPI was granted the status of a **National Technical University** in 1995. In 2007, it became the first Ukrainian university to be recognized as a **research university**, leveraging the integration of education, science, and innovation within the powerful innovation ecosystem Sikorsky Challenge, which he established.

The KPI research university model is based on refining mechanisms for modern integration of science, education, and innovation. This approach emphasizes preparing highly qualified researchers and specialists for knowledge-

intensive industries in Ukraine's economy, fostering innovation activities in market conditions through the university's scientific and technological parks, and promoting a knowledge-based economy. The foundation of this model has always been the synergy of experience and energy from both seasoned and young educators and researchers.

It is worth noting that during Ukraine's independence, KPI effectively became a laboratory for developing and drafting key documents on education. These include Ukraine's laws **"On Education," "On Higher Education," "On the Scientific Park 'Kyiv Polytechnic,'" "On Scientific Parks,"** and **"On the Basic Principles of Information Society Development in Ukraine for 2007–2015"**, along with numerous government resolutions and decisions by the Ministry of Education and Science. KPI was one of the first universities in Ukraine to sign the **Magna Charta Universitatum**. It was an active participant in introducing the Bologna Process in Ukraine and played a leading role in organizing and participating in numerous international conferences on reforming higher education and harmonizing it with the European educational system.

The transformation of Igor Sikorsky Kyiv Polytechnic Institute (KPI) into a research university was just one, albeit undoubtedly a key, project among dozens of initiatives proposed and implemented by Mykhailo Zgurovsky during his tenure as Rector of Kyiv Polytechnic. Among the achievements initiated and successfully realized under his leadership are the following:

- Establishment of 11 new faculties and educational-scientific institutes, as well as over 50 new departments to meet the modern needs of Ukraine's economy and society.
- Development and expansion of the scientific-educational information and communication network **"Uran"** and its integration into the European **GEANT** network.
- Creation of the university's supercomputer center.
- Establishment of the **World Data Center for Geoinformatics and Sustainable Development**, as part of the global network of World Data Centers.
- Creation of Ukraine's first State Polytechnic Museum.
- Establishment and development of the **Sikorsky Challenge** innovation ecosystem, based on the Law of Ukraine "On the Scientific Park 'Kyiv Polytechnic.'"
- Initiation of the educational and scientific complex **"Institute for Applied System Analysis"** (IASA) in cooperation with the National Academy of Sciences of Ukraine (NASU).
- Creation of the **Institute of Advanced Defense Technologies**.
- Establishment of the KPI Alumni Association.

Under the leadership of Rector Mykhailo Zgurovsky and with his direct involvement as the architect of every project, the university underwent significant physical and cultural transformations. These changes reflect the finest traditions of leading European and global universities. KPI restored its unique spirit of academic excellence, scientific achievements, and proud affiliation with its illustrious history.

Numerous unique symbols of the university were created or restored, becoming its pride: the Hall of the Academic Council, the Polytechnic Museum, KPI's

famous park, monuments to distinguished polytechnicians whose discoveries changed the world, the Grand Physics and Grand Chemistry Auditoriums, Knowledge Square, art galleries, commemorative plaques, the Foucault pendulum, the university church, two stadiums, numerous new parks, and the clock tower on the main building. A special place belongs to the memorial complexes dedicated to Kyiv Polytechnic students and faculty who gave their lives for Ukraine's freedom and independence. These symbolic landmarks are a profound tribute to the university's remarkable history.

Particular attention was given to the extensive reconstruction of a significant portion of the university's academic buildings and dormitories, many of which were in disrepair in the early 1990s. Thanks to meticulous efforts, these facilities were restored, revitalized, and adapted to meet modern requirements.

Today, this legacy not only reflects the grandeur of the past but also inspires new generations with pride in the great accomplishments of their predecessors. It surrounds the KPI community with a unique atmosphere: the spirit of university tradition, academic excellence, a sense of belonging to a rich history, and motivation for new achievements worthy of the outstanding contributions of previous generations.

It is important to emphasize that Mykhailo Zgurovsky has consistently prioritized the retention of experienced and highly qualified academic and scientific staff in his personnel policy while also attracting talented young professionals. This approach ensures the university's continued development and the preservation of its finest traditions.

The collaboration between Igor Sikorsky Kyiv Polytechnic Institute (KPI) and the scientific institutions of the National Academy of Sciences of Ukraine (NASU) has deep historical roots. Many of the founding scientists of KPI were directly involved in the establishment and development of the Ukrainian Academy of Sciences. For instance, Stepan Prokopovych Tymoshenko, Dean of the Mechanical and Engineering Faculties of KPI, formed the first Mechanics Department of the Ukrainian Academy of Sciences, organizing what is now the Institute of Mechanics of NASU, named in his honor. Another notable example is the distinguished KPI scientist Yevhen Paton, who founded the Institute of Electric Welding. Renowned academicians such as H.S. Pysarenko, S.V. Sorensen, K.K. Syminsky, and many others combined their work at KPI with leadership roles in academic institutes.

Mykhailo Zgurovsky actively supports these traditions. At his invitation, prominent scientists such as Academician V.G. Baryakhtar (founding Dean of the Faculty of Physics and Mathematics), Academician I.M. Kovalenko (founding Dean of the Faculty of Applied Mathematics), and other esteemed mathematicians and physicists like Academicians A.M. Samoilenko, I.V. Skrypnyk, V.M. Loktev, S.M. Dovgy, materials scientist Academician I.V. Kryvtsun (a disciple of B.Ye. Paton), renowned cyberneticist Academician V.S. Deineka, and many others have actively worked and taught at KPI.

Reflecting on the future of education, Mykhailo Zgurovsky recognized the need to reorient the university's activities toward a more fundamental approach to education, moving away from a narrowly technical focus. This fundamental approach has been inherent to KPI throughout its history, emphasizing the inseparable connection between education, scientific research, and the practical applica-

tion of results. It is this emphasis on fundamental education that once secured global leadership for the developments of former students who later became world-renowned scientists and engineers. The same approach underpins the numerous victories of current KPI students in international competitions in mathematics, programming, and cybersecurity.

Under the scientific leadership of Mykhailo Zgurovsky, the **Institute for Applied System Analysis (IASA)** of NASU and the Ministry of Education and Science of Ukraine, as well as the **World Data Center for Geoinformatics and Sustainable Development**, were established in Ukraine. The latter specializes in geoinformatics, focusing on global modeling of sustainable development processes and assessing global threats to human security and quality of life.

IASA was established in 1997 by a resolution of the Cabinet of Ministers of Ukraine as part of the Cybernetics Center of NASU. Its dual subordination reflects the implementation of the concept of integrating science and education. IASA conducts cutting-edge fundamental and applied research in developing methodologies for systems analysis of complex interconnected objects and processes of social, economic, ecological, and technological nature.

The institute develops methods for forecasting and predicting the behavior of complex systems under multi-factor risks, decision-making in such systems with conflicting objectives, and uncertainty or insufficient initial information. It also advances the theory of optimal control, differential games, nonlinear analysis, optimization, the theory of infinite-dimensional dynamic systems, methods for evaluating and managing nonlinear distributed parameter systems, and the analysis of controlled Markov processes. Furthermore, it develops the theory of information-analytical systems, methods for managing large databases, and strategies for ensuring the guaranteed functionality of cyber-physical systems, including the support of digital twins.

The platform for solving this class of problems is a new research direction proposed by Mykhailo Zgurovsky, known as **"system mathematics."** This direction represents a set of interrelated branches of mathematics (both classical and newly developed) that enable solving modern interdisciplinary problems of various natures. By applying system mathematics, the Institute, under Zgurovsky's leadership, has achieved fundamental results in the field of global modeling of sustainable development processes in the context of quality and safety of human life, analysis of global threats, and systemic studies of control problems for complex and hybrid systems.

In applied research, tools for **technology foresight** and **scenario analysis** have been developed in the form of an advanced information platform. This platform integrates mathematical, software, logical, and organizational tools, allowing for the sequencing of qualitative and quantitative analysis methods, establishing relationships between them, and ensuring the process of building future event scenarios. In practical terms, the research outcomes have been applied to address numerous challenges in the development of Ukraine's economy.

The **Institute for Applied System Analysis (IASA)** actively collaborates with NASU research departments, including the Divisions of Informatics, Mathematics, Mechanics and Machine Science, Physics and Astronomy, Materials Science, Energy and Energy Technologies, Earth Sciences, and Economics, as well as other NASU structures and many higher education institutions. It is worth

emphasizing that the fundamental and applied research and developments at IASA are conducted, as intended at its establishment, in conjunction with the educational process and the preparation of highly qualified young professionals.

In response to contemporary challenges, Mykhailo Zgurovsky proposes a new paradigm in science and education based on general trends such as interdisciplinarity, human-centered approaches, sustainability, responsibility and ethics, and the comprehensive use of Industry 4.0 core technologies.

Zgurovsky argues that the transition from Industry 4.0 to Industry 5.0 signifies not only technological shifts but also a rethinking of the human role in society and industrial production. He emphasizes sustainable human development and the creation of new societal values. This, in turn, imposes new demands on the economy, industry, and society, as well as on scientific research and the training of specialists in system analysis, intelligent service-oriented distributed computing, and artificial intelligence systems and methods.

In response to the challenges mentioned above, Mykhailo Zgurovsky initiated profound reforms in the educational and research activities of IASA, based on the following principles:

- **Human-centered innovation:** The new economic model places humans and their needs at the center of innovation processes. Industry 5.0 highlights the importance of harmonious collaboration between humans and technology, requiring greater consideration of human needs, ethical issues, and social responsibility when developing technological solutions.
- **Interdisciplinary approach:** Workforce training and scientific research must address not only the rapid development of technologies but also the complexity and multidimensionality of demands from the economy and society. Integrating an interdisciplinary approach into the system of science and education is essential to prepare specialists capable of solving complex problems.
- **Broad integration of advanced technologies:** The economy and society in the Industry 5.0 era will demand extensive integration of system analysis, artificial intelligence, distributed computing, and cyber-physical systems across all areas of life. Key aspects include technology interoperability, cybersecurity, adaptability to change, and adherence to ethical standards.
- **Competence development:** Training programs should focus on developing competencies such as the analysis of complex adaptive systems, intelligent data analysis, the design of resilient and secure cyber-physical systems, and ensuring interoperability between various technologies.
- **Scientific Achievements and Legacy:** Within Zgurovsky's scientific school, 16 doctoral and over 40 candidate theses have been completed. He is the author of 52 inventions and has authored or co-authored more than 1,000 scientific works, including 45 monographs and textbooks published in Ukraine, Japan, Poland, China, Germany, and other countries worldwide. These accomplishments highlight Mykhailo Zgurovsky's enduring contributions to science, education, and societal development, positioning IASA as a leading institution in addressing modern challenges and fostering innovation.

Under the leadership of Academician Mykhailo Zgurovsky, significant work has been carried out in the analysis and scenario planning of sustainable development for regions of Ukraine in the context of human quality of life and safety. The aim of this work is to enhance the comprehensiveness and effectiveness of

analytical and informational support for decision-making processes (in the form of medium- and long-term strategies and sequences of actions by authorities). These efforts focus on ensuring the sustainable development of individual regions and Ukraine as a whole, viewed as complex socio-economic systems. This is achieved by developing new and improving existing methods of systems analysis, big data intelligence, forecasting, scenario planning, and management.

The Institute for Applied System Analysis (IASA) publishes the international scientific journal "System Research and Information Technologies," with Mykhailo Zgurovsky serving as its Editor-in-Chief. On Zgurovsky's initiative, IASA organizes and co-organizes numerous scientific conferences and seminars. Notably, it hosts the annual International Scientific and Practical Conference on Systems Analysis and Intelligent Computing.

Mykhailo Zgurovsky is extensively involved in public activities. He is a member of numerous Ukrainian and international scientific societies, the national representative of Ukraine at the International Science Council (ISC, Paris, France), a member of the Honorary Council of the Order of Saint Panteleimon, and the Chair of the Ukrainian Peace Council. In recent years, his public efforts have focused on consolidating international organizations and prominent figures in education, science, and culture from various countries to achieve a just peace for Ukraine.

Zgurovsky has been a co-organizer and participant in key international events, including the **UN Seminar on Environmental Security During War and Armed Conflicts** (November 6, 2024, New York), the **Second Summit on Partnership and Veterans' Health** (European Parliament, October 29–30, Brussels, Belgium), and the **International Conference "Europe and Ukraine: Shared Perspectives and Values"** held at the Austrian Institute for European and Security Policy (December 13, 2024, Vienna, Austria), among others.

For his achievements in science, education, and fostering international cooperation, Mykhailo Zgurovsky has been awarded numerous state honors from Ukraine (he is a Full Cavalier of the Order of Merit), as well as from Italy, Estonia, Vietnam, Poland, France, Japan, and China.

The scientific community, colleagues, and students extend their heartfelt congratulations to Mykhailo Zgurovsky on his milestone anniversary, wishing him active longevity, boundless inspiration, and new achievements for the benefit of his homeland.

Received 20.12.2024

ПРОБЛЕМИ ПРИЙНЯТТЯ РІШЕНЬ ТА УПРАВЛІННЯ В ЕКОНОМІЧНИХ, ТЕХНІЧНИХ, ЕКОЛОГІЧНИХ І СОЦІАЛЬНИХ СИСТЕМАХ

UDC 303.732.4
DOI: 10.20535/SRIT.2308-8893.2024.4.01

SYSTEMATIC STUDIES OF CRYPTOCURRENCY USAGE TOOLS FOR FINANCIAL MARKETS

V. ROMANENKO, H. KANTSEDAL

Abstract. The paper analyzes mining processes when using cryptocurrency on financial exchanges. It considers the cognitive map (CM) of the use of cryptocurrency as a complex system. It reveals all functions of mining processes when interacting with speculative instruments under conditions of uncertainties and risks. Digital modeling of impulse processes in the CM was carried out to study the dynamic properties of the free movement of the tops of the CM under random disturbances.

Keywords: cryptocurrency, cognitive map, mining, impulse processes.

INTRODUCTION

During the 1990s and 2000s, there was a need to solve the problems of verifying and signing digital documents (or transactions) in an environment with partial trust, that is, in an environment where one or more agents may not be in good faith [1]. Bitcoin was proposed to solve this problem. In work [2], attention was focused on the exchange of monetary assets, and a workable cryptocurrency prototype was also proposed. This ensured a relatively rapid spread of the idea among cryptographic enthusiasts. In four years, the popularity of the topic of Bitcoin led to a boom in articles aimed at increasing the efficiency of its support (mining) [3], as well as the appearance of a request for specific computing systems aimed only at mining.

Miners, the backbone of the cryptocurrency ecosystem, are responsible for emissioning and maintaining cryptocurrency. Their crucial task is to ensure the recording and execution of transactions by forming digital blocks in a standard chain—blockchain. This blockchain, one of the key concepts that underpin modern cryptographic systems, is a testament to the vital role of miners in the cryptocurrency world.

Mining is a popular way to earn money through cryptocurrency. The blockchain rules require the chain of blocks to be continuous, meaning breaks in the chain are not allowed, but branching is possible. This applies even to traditional cryptocurrencies when creating new ones based on old ones or temporarily until the next block is generated. In addition, blockchain requires the instant execution of events (smart contracts) included in the block. It also prohibits changes to the results and composition of actions in each block.

In a further development, cryptocurrency led to the emergence of specialized cryptocurrencies that can solve specific tasks of cloud computing and data storage. However, Bitcoin continues to occupy a dominant position among other currencies.

In works [4–7], cognitive maps of the use of cryptocurrency in financial markets were developed. These maps were designed based on causal relationships and took into account the main financial indicators of cryptocurrency circulation. However, they did not consider the role of miners as organizers and performers of cryptocurrency exchange by performing transactions outside the banking system.

STATEMENT OF THE PROBLEM

This article examines the problems of researching the processes of using cryptocurrency in financial markets as a complex system, the model of which is developed in the form of a cognitive map (CM).

The first task is devoted to analyzing the functions of mining processes, which help perform cryptocurrency exchange operations by conducting transactions in digital form outside the banking system.

The second task is to analyze the interaction of speculative instruments with the cryptocurrency market. The third task is to analyze the uncertainties and risks of using cryptocurrencies.

In the article we simulate the processes in the CM to study the dynamic properties of the CM's vertexes under random disturbances.

PECULIARITIES OF FUNCTIONING OF DIGITAL CRYPTOCURRENCY LIKE BITCOIN

The nature of using Bitcoin for exchange

The primary purpose of using Bitcoin was to transfer funds between users during trade transactions. In recent years, Bitcoin has gained mostly speculative popularity. To reveal the specifics of using Bitcoin as a means of exchange, let us consider the traditional type of money transfer via a bank (Fig. 1).

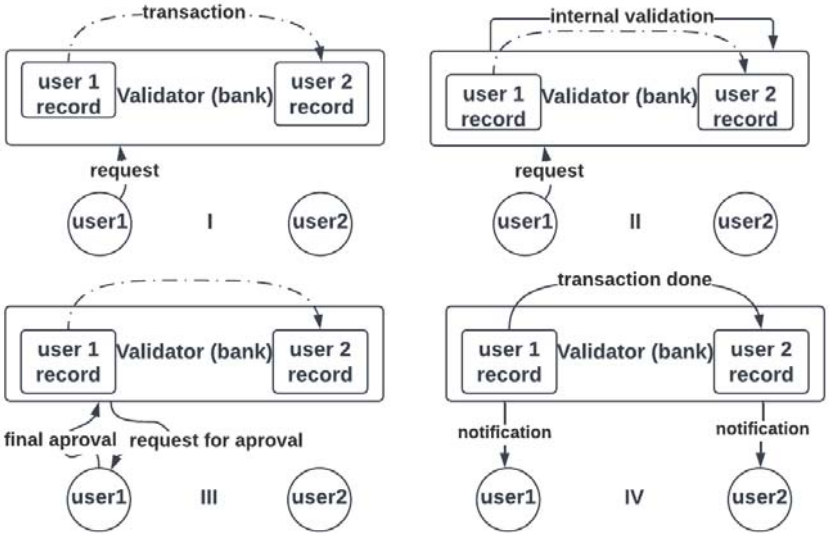


Fig. 1. The sequence of funds transfer in traditional systems is conditionally divided into four stages

There are several stages in the standard options for transferring funds:

- Authorization on the bank's server, sending a request to transfer funds.
- Internal verification of the bank (whether this transaction is possible, whether the recipient's address is valid, whether there are enough funds).
- Transaction authorization/confirmation (more a legal requirement than a necessary step).
- Physical transfer of funds (i.e., changing the balance of two users). Sending a message about a successful transaction.

The considered simple option of transferring funds between two clients of the same bank vividly illustrates the main features of the system, namely:

- The user cannot access his money and the bank's database directly.
- All operations are performed on the bank's side in its single system. The user acts as an external factor.
- The system is centralized.

This approach, of course, has its advantages, particularly for the regulation of funds; however, it also has significant disadvantages, such as limited functionality in the event of system damage (lack of Internet, light, etc.). The Bitcoin system partially solves this problem. Consider an example of Bitcoin transfer between users (Fig. 2):

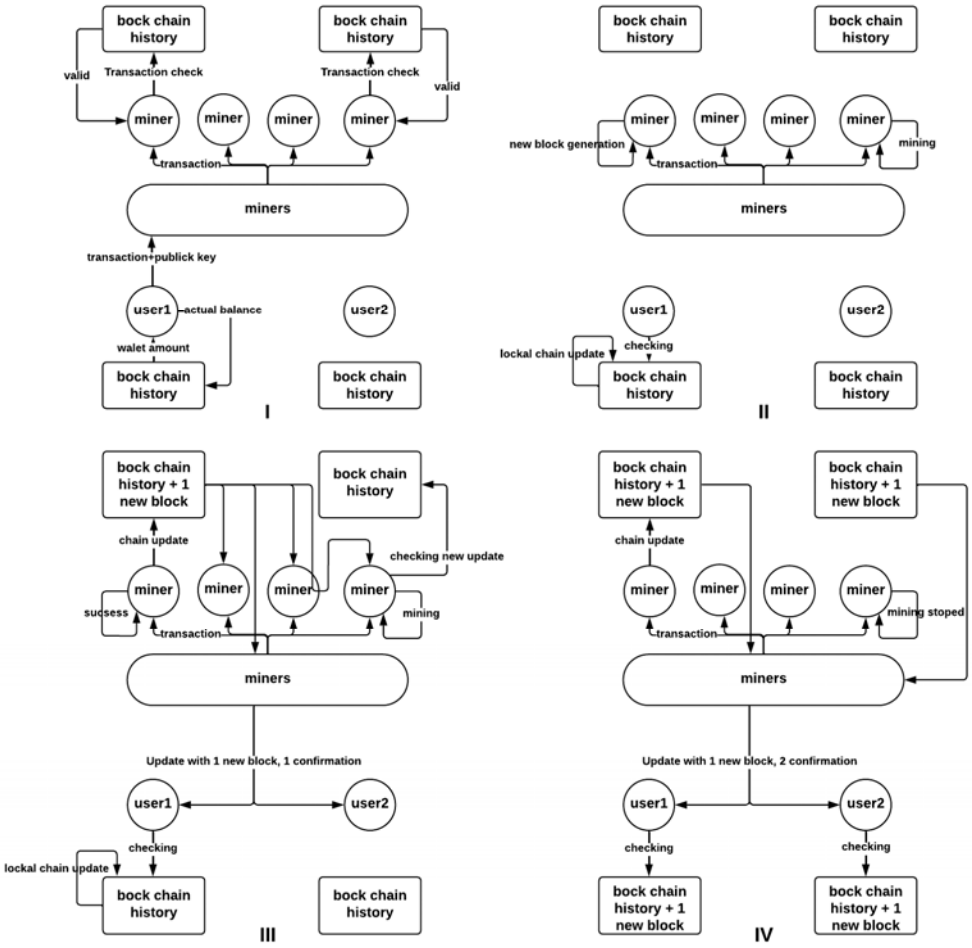


Fig. 2. The transfer of funds between two users in the Bitcoin system is conventionally divided into four sequential stages. In the figure, it is assumed that only two users are actively engaged in mining

During the transfer of bitcoins, several stages can be distinguished:

- The user checks his balance against his own copy of the blockchain (similar to a bank database). According to the verification, he requests to transfer funds to user 2. The request is coded using the user's private key and can be verified by anyone using the public key.
- The request is sent to the system and processed by one of the miners (or several). The possibility of operations is checked according to the miner's local blockchain. After verification, the operation is included in the block and the block generation process begins (that is, the selection and calculation of the hash sum that satisfies certain complexity characteristics).
- One of the miners finds the required hash sum — the new block is considered generated and is sent to other users for confirmation. The user who made the transaction receives an early warning that the transaction is already successful and agreed with one miner.
- Other crypto community users (and other miners) confirm the generation of the block (that is, it does not contradict their transaction history). This transaction is considered complete after receiving confirmation from 51% of users.

An additional requirement for users to accept the generated chain of blocks is its length. Therefore, shorter chains are rejected, and the longest chain takes priority for extension (Fig. 3).

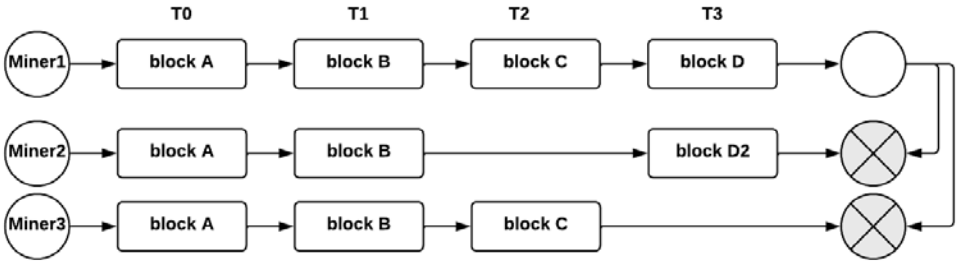


Fig. 3. Example of rejecting blocks based on chain length rule. Miner chain 1 is accepted, while chains 2 and 3 are rejected

In this system, there is no central element that can be replaced with another one. This means that changes in the miners operating the system will not be noticeable to ordinary users, and the loss of a part of the community of miners will not affect the system's operation. To ensure a smooth transition when miners' capacities change, a balancing mechanism is embedded in the blockchain. The complexity of block generation, which involves setting conditions on the selected hash sum, depends on the speed of block generation and is balanced at the level of 1 block per 10 minutes. If there is a sudden increase in the total power of the miners, the selection process may speed up. In response, the system will automatically introduce an additional complication to hash sum generation. This resistance to losing part of the mining capacity also limits the number of transactions that can be included in a block, as block sizes have physical size limits.

Anonymity in the blockchain system and benefits of use

Unlike the traditional banking system, the blockchain system does not allow for transaction cancellations or block alteration. The private key is the sole user identifier, enabling transactions from anywhere with an Internet connection. While it is possible to trace the source of funds, it is impossible to prevent their transfer.

Linking an address to a specific Bitcoin user is another concern. This problem is partly addressed through regulating cryptocurrency exchanges, which oversee real money flow and require user registration. However, a significant loophole for money laundering via cryptocurrency is the process of mining, which effectively converts electricity into cryptocurrency. This process completely anonymizes the recipient of the cryptocurrency, enabling it to be used for criminal purposes. The only way to trace the recipient of the cryptocurrency earned through mining is by locating the connection points of the mining equipment. On average, a miner can earn between $0.7 to $1.4 for every 1kWh of electricity used, depending on the exchange rate and electricity prices. This becomes incredibly profitable when access to free energy is available, such as when an enterprise or government institution covers the electricity costs. Businesses with an average power capacity of 100 kW, benefiting from discounted electricity rates, can generate up to $70.000 in monthly illegal income. This income could be undeclared or even a form of bribery.

Recent events have shown several unusual options for influencing the cryptocurrency exchange rate. Thus, countries that extract raw energy materials (gas and oil) are more prone to the mining process and, therefore, to the emission of bitcoin. As a result of the impact of the sanction, it became a source of capital outflow from the country and black purchase of goods. Bitcoin proved resistant to restrictions, but crypto exchanges, as exchangers of cryptocurrency for real money, came under restrictions and were forced to leave the market of such countries. Despite the stability of the Bitcoin system itself, such a move significantly undermined user confidence in the system. This shows that Bitcoin has not become an independent means of payment and is still perceived as a speculative asset, where the possibility of exchange for real money plays a key role.

CHARACTERISTICS OF MINING AS A DISTINCT PHENOMENON RESULTING FROM THE USE OF CRYPTOCURRENCIES

Mining is the essential link that connects the world of cryptocurrencies with the real-world environment. The concept of rewarding users for utilizing their computing power is at the core of this process. Other users repeatedly verify every action; only the result that aligns with most users (miners) is considered valid.

Initially, mining referred to creating a block in the blockchain by finding the appropriate hash and solving cryptographic tasks. Over time, the term has evolved to encompass any work for which a reward is received in the blockchain. In our context, mining encompasses any computing process that supports the operation of the blockchain system. From the network perspective, all users are integral parts of the system, acting as nodes. Mining is the only chain that connects the environment of cryptocurrencies with the real world. The idea of rewarding some users for using their computing power is fundamental. Each action is repeatedly duplicated and checked by other users. De facto, only the result that coincides with the results of most users (miners) is considered valid.

First and foremost, mining was the process of generating a block in the blockchain, that is, selecting the appropriate hash and solving cryptographic tasks. However, gradually, this term spread to any work for which a reward is received in the blockchain. In our work, mining is any computing process that supports the

operation of the blockchain system. From the network point of view, all users are parts of the system — nodes:

- Full nodes are transitional nodes of the blockchain that keep and update all information about users. They require a lot of computing resources. Usually, users who hold a full node are engaged in checking transactions (keys) and the validity of blocks received from miners. Users with full nodes are rewarded for validating transactions and maintaining the network. Direct mining may also be a secondary task for such users.
- Light nodes support only partial information about the blockchain. The majority of users use them for everyday operations—checking the balance and transferring funds. The computing load is minimal, and synchronization delays can be up to several seconds.
- Pruned full nodes — differ from full nodes by the possibility of limiting support resources. Usually, only a certain limited part of the blockchain is stored — for example, the last 10 GB of transactions. Convenient for blockchain research.
- Mining nodes — the most demanding node to support. It requires different resources – from the processor and RAM to graphics processors and special-purpose integrated circuits. Users who deploy this node on their equipment receive a reward directly from the mining process — the process of generating blocks (combining transactions into a group for calculating the cryptographic component of the system — the hash). As a result of successful block generation, a reward is received. In an unsuccessful generation, the user receives nothing (Proof of Work principal).
- Master nodes — a complete analog of the full node but with additional functions — guaranteeing anonymity in the system. Unlike users with full nodes, an additional requirement for master nodes is the presence of significant capital in cryptocurrency. When conducting an anonymous transaction with another user, the capital of the master node will participate in the transaction and dilute the anonymized transaction, among others. This mechanism will make it possible to completely exclude the possibility of tracking the communication between the sender and the recipient. Of course, from the side of ordinary users, this anonymous transaction will have an increased commission, which will go to the reward of the owners of the master node. Officially, the Bitcoin system can work without master nodes, but this mechanism is easily transferred to any cryptocurrency, so it is definitely used.
- Fast nodes (Lightning Network) are nodes whose main goal is to synchronize speed among themselves. Wines act as a kindred nervous system of the blockchain, increasing bandwidth and speeding up the number of transactions in the system. The user receives a reward for transactions and the speed of their execution.

The operation of the blockchain system requires the participation of several types of users. Mining nodes form the system's foundation, but rewards are also distributed to owners of fast nodes, master nodes, and full nodes based on similar principles. Currently, the system involves over 12.000 different nodes. The exchange of information between these nodes is a hidden centralized element of the system, as connecting to nodes requires knowledge of the API address, and the system relies on resources with a list of active nodes.

Understanding the unique aspects of Bitcoin mining emission

The actual mining process simultaneously involves issuing cryptocurrency and supporting its transfers. The miner receives a reward from two sources of income:

- User commissions, a dynamic aspect of the process, are determined by the user's need to transfer Bitcoin quickly. The higher the commission, the greater the motivation of the miner to include the transaction in the block, thus increasing the priority of the transaction. The estimated minimum reward per block is 0.8 bitcoins and depends on user demand for transfers.
- An additional block reward was staked when the cryptocurrency was created. However, the reward is halved every 210.000 blocks. So, in 2032, the additional reward will be 0.78 bitcoins per block, which will be less than the user fee (Fig. 4).

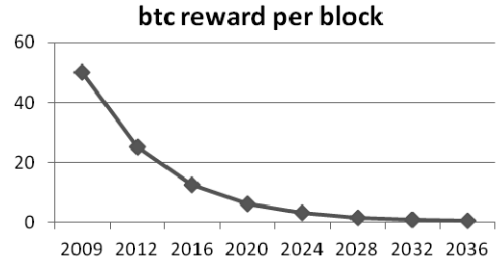


Fig. 4. Decreasing the emission (rewards for mining) of Bitcoin over time

The additional reward acts as a source of creation of new bitcoins (emission of cryptocurrency). At the same time, the commission is charged to users and is not a floating factor on the cryptocurrency rate. This means that the number of bitcoins is limited, and therefore, bitcoin is a unique resource, which gives the right to partially compare it with gold (as a limited resource of exchange). Thus, the number of bitcoins will be at most 21 million coins (Fig. 5). Therefore, considering the loss of wallets, this resource should become more and more unique over time.

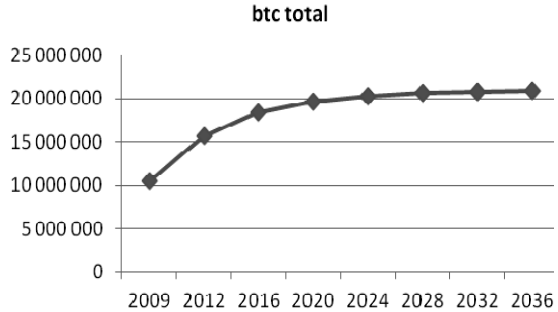


Fig. 5. The increase in the total number of bitcoins

To comprehend another mining issue, one must understand the structure of the primary mining outcome—the block (see Fig. 6).

As mentioned in 2.1, the block generation process does not depend on the number of users involved in the mining process and on their computing power. Theoretically, one user is enough for this. On the other hand, if there are several miners, only one receives a reward for mining. The computing component of min-

ing causes this circumstance. To generate a new block, we need to include user transactions and information about the previous block and generate a hash sum that would meet some complexity rules (that is, it was less than a certain number). With the help of the complexity level, the block's birth time is regulated in 10 minutes, regardless of the computing power of the miners. In the case of low computing power, the block generation time begins to increase, which leads to the inconvenience of ordinary users and the accumulation of transactions. In this case, the condition is gradually softened.

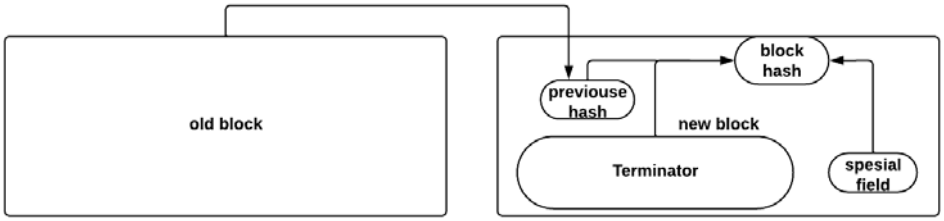


Fig. 6. A simplified Bitcoin block diagram

COGNITIVE MAP AS A MODEL FOR RESEARCHING THE USE OF CRYPTOCURRENCY

The cognitive map for researching the mining process is presented in Fig. 7. The vertices of the CM have the following names:

1. Cryptocurrency rate.
2. Number of active cryptocurrency users.
3. Number of miners.
4. Volume of processed transactions.
5. Performing miner functions.
6. Profitability of mining (profit).
7. User fees for transaction execution.
8. Issuance of cryptocurrency.
9. Amount of capitalization.
10. Indirect profit (circumvention of sanctions, economic restrictions inside of countries).
11. Funding for the development of new mining equipment.
12. Cryptocurrency supply.
13. The price of energy resources when using cryptocurrency.
14. Number of passive users (institutional investors).
15. Mining equipment price.
16. Computing complexity of the network.
17. Safety of using cryptocurrencies.
18. Legality of using cryptocurrencies.
19. Tendency to accumulate cryptocurrencies.
20. Efficiency of miners.
21. Fear of devaluation of cryptocurrency.
22. Belief in the growth of the cryptocurrency rate.
23. Informational and speculative disturbances have the following names.

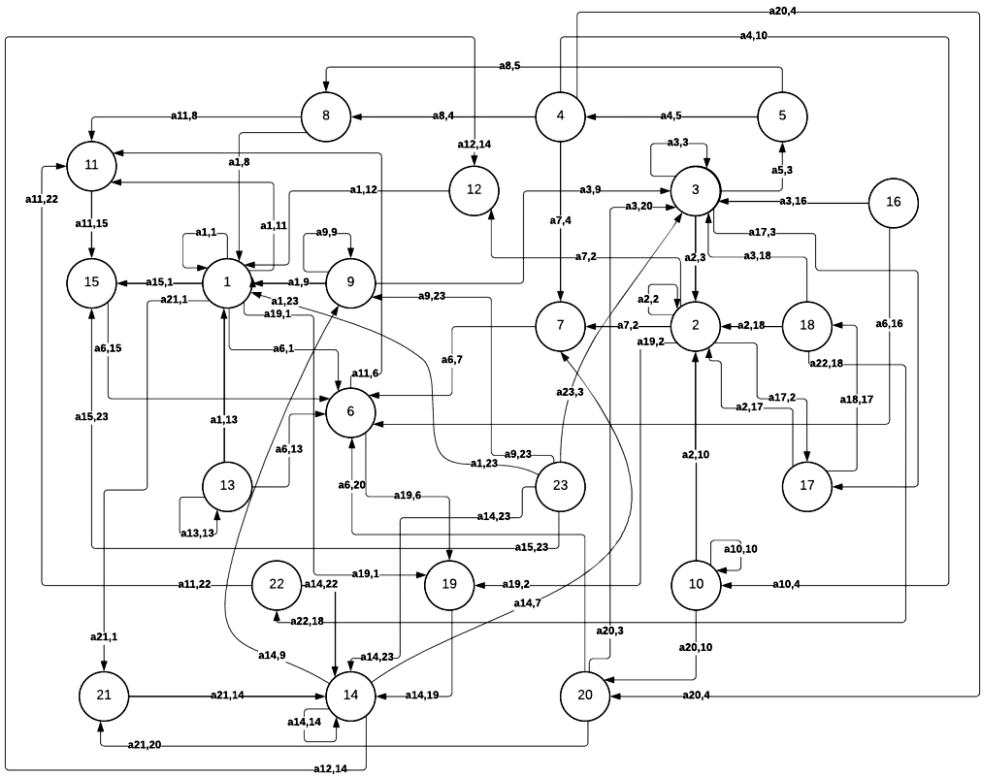


Fig. 7. Cognitive map of cryptocurrency mining

When the perturbations act on the CM vertices, a transient process occurs, which is described by the difference equation [7]:

$$\Delta y_i(k+1) = \sum_{j=1}^{n} a_{ij} \Delta y_j(k) \tag{1}$$

where $\Delta y_i(k) = y_i(k) - y_i(k-1),\ i = 1,...,n$, a_{ij} — the weight of the edges of the weighted directed graph (CM connects the j-th vertex of the CM with the i-th one. Equation (1), which describes the free movement of the i-th coordinate of the CM, can be written in vector-matrix form for the entire CM:

$$\Delta \overline{y}_i(k+1) = A \Delta \overline{y}(k), \tag{2}$$

where A is the weight matrix of the adjacency of the CM $(n \times n)$, which consists of a_{ij} CM elements.

The first 15 vertices of the CM are measurable. This means that their coordinates can be precisely measured and fixed at discrete moments of time K, starting with the period of discretization T_0.

From the 16th vertex to the 23rd, their coordinates cannot be formalized, and their change is difficult to determine on a real-time scale. To clarify this complexity, the original model of the impulse process (2) is divided into two parts. The first part is dedicated to the measured parameters of the vertices of the original CM (2), ensuring a structured approach to the research. The second part includes the non-measurable parameters of the vertices, acknowledging the inherent challenges in the field:

$$\Delta y_i(k+1) = \sum_{j=1}^{15} a_{ij}\Delta y_j(k) + \sum_{\mu=16}^{23} a_{i\mu}\Delta y_\mu(k), \quad (3)$$

where y_j — the measured coordinates of the CM; y_μ — unmeasured coordinates of the top of the original CM.

The second model is created to describe the impulse process of the non-measurable parameters of the CM vertices:

$$\Delta y_\mu(k+1) = \sum_{j=16}^{23} a_{\mu j}\Delta y_j(k) + \sum_{j=1}^{15} a_{\mu j}\Delta y_j(k). \quad (4)$$

At the same time, the unmeasured parameters of the vertices are considered disturbances in the first model (3).

Expressions (3) and (4), respectively, can be written in vector-matrix form:

$$\Delta\overline{Y}_1(k+1) = A_{11}\Delta\overline{Y}_1(k) + A_{12}\Delta\overline{Y}_2(k); \quad (5)$$

$$\Delta\overline{Y}_2(k+1) = A_{21}\Delta\overline{Y}_1(k) + A_{22}\Delta\overline{Y}_2(k), \quad (6)$$

where $\overline{Y}_1$ is a vector of measured parameters, and $\overline{Y}_2$ is a vector of non-measurable parameters, and the matrices A_{11}, A_{12}, A_{21}, A_{22} respectively have the dimension (15×15), (15×8), (8×15), (8×8) and are components of the adjacency matrix A in the model (2):

$$A = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix}. \quad (7)$$

The use of cryptocurrencies is affected by various factors that can reduce trust in their use. As a result, the following risks can occur when dealing with cryptocurrencies:

- Risk of losing users, leading to a decrease in the price of Bitcoin
- Risks associated with incorrect general expectations of many users simultaneously, created by manipulating traders on financial exchanges.
- Risk of a sudden collapse in the cryptocurrency price due to coordinated manipulations on the stock exchanges, including high-frequency trading, where assets are bought and sold at high speeds, potentially leading to investor panic.
- Risks associated with the lack of guarantees for the safety of the capital invested in cryptocurrency, leading to user anxiety when trading on the stock exchange.

It should be noted that the combination of vertices 15-11-6 has an interesting stepped nature and behavior. The process of developing new computing schemes aimed only at solving the mining problem is quite slow and requires a lot of real money. Therefore, it constantly pulls resources away from long-term Bitcoin holders. On the other hand, when the development is completed, and the new equipment is ready for serial production, it causes a significant outflow of capital invested in Bitcoin due to the need to update mining capabilities.

Vertex 13 (the price of energy resources when using cryptocurrency) closely connects the world of miners with the real world. The mining process is extremely energy-consuming, and any fluctuations in the energy market lead to sig-

nificant problems with mining. In particular, military actions near oil fields traditionally have a negative impact on cryptocurrency.

On the other hand, the presence of conflicts in other regions is a positive incentive for the growth of Bitcoin's price as a source of wealth in unstable conditions or in terms of bypassing traditional financial instruments (bypassing stations, blockades, etc.). Although mining relies increasingly on green energy, its behavior is still sensitive to daily fluctuations.

INTERACTION OF SPECULATIVE INSTRUMENTS WITH THE CRYPTOCURRENCY MARKET

Since the dawn of currency, speculation has been a fundamental part of human society. In the realm of cryptocurrencies, speculation takes on a whole new dimension. Cryptocurrencies, unlike traditional currencies, can be stored indefinitely, with the only limit being the total number of cryptocurrencies in existence. To navigate this unique landscape, many traders turn to mechanisms borrowed from the stock market, such as contracts (futures). These contracts offer the assurance of cryptocurrency redemption at any given time, a stark contrast to the typical trading of cryptocurrencies. In fact, the contract market is currently three times the trading volume of Bitcoin, making it the most popular avenue for speculation.

Contracts are divided into two types—to increase the price (long) of the asset (cryptocurrencies) and decrease it (short). To conclude a contract for an increase, it is necessary to find a counterparty for a similar contract for a price decrease. Currently, exchanges provide the opportunity to conclude contracts without being tied to a specific user and to share them between several counterparties (for ordinary users, the counterparty mechanism looks like the exchange acts as counterparty for all contracts). Thus, the mechanism of contracts on cryptocurrency exchanges is similar to the mechanism of making a bet and is a purely speculative mechanism.

The market of contracts on exchanges is divided into the following categories:

Perpetual contracts: These are derivatives that involve a commission paid depending on the duration of the contract. The commission size is directly related to the imbalance of the number of contracts on the market/exchange.

- Delivery contracts: These are futures with a fixed execution time. Until the execution time, they are similar to perpetual contracts. At the time of execution, holders of long contracts must sell the asset (cryptocurrency) for the amount specified in the contract, and holders of short contracts must buy cryptocurrency at the market price.
- Contracts for cryptocurrency in cryptocurrency: In terms of the execution mechanism, they can be similar to the first two types, but the use of only cryptocurrency expands the possibilities of contracts and allows for the addition of new conditions. These contracts can be applied in decentralized settlements.

Credit funds are widely used for all the speculative assets listed. Under very reasonable conditions, it is possible to borrow up to 124 credit dollars using just one physical dollar. The contract market is seen as a form of speculation that is a significant destabilizing factor in the financial market. Speculative assets are read-

ily accessible and fully integrated into the basic functionality of modern exchanges. Speculation can be categorized as follows:

- Inter-exchange arbitrage involves making a profit from price differences between exchanges. This involves buying and selling assets, but it carries the risk of transferring cryptocurrency between exchanges.
- Cross-currency arbitrage is about profiting from exchange rate differences between interconnected assets. This situation arises when one asset is traded in much smaller volumes than the others, leading to delayed price quotations.
- Cross-product arbitrage is an arbitrage between different products of an exchange or exchanges. It typically involves derivatives, options, and futures.
- Trading based on graphic data analysis is risky for speculation, as it involves a desire to get rich quickly rather than long-term operations. In the short term, the behavior of the cryptocurrency exchange rate is random and has a low, but not zero, chance of success.
- High-speed trading involves short-term transactions and frequent or full automation, which reduces risks due to their short duration.
- Scraping is a speculation strategy based on a stationary cryptocurrency rate assumption. The user engages in the de facto resale of the asset with a minimal markup. This is the least risky strategy among non-arbitrage options for speculation.

On the cognitive map (Fig. 7), speculation is depicted as a disruption, which is considered at the top of 23 CM. Other speculations, in various forms, stem from the above. While most arbitrage strategies enable synchronization of different parts of the system and cryptocurrency markets, scraping introduces white noise. Trading based on graphical analysis and high-speed trading with many users is entirely random.

CHARACTERISTICS OF UNCERTAINTIES WITH CRYPTOCURRENCY USAGE

Uncertainties when using Bitcoin

One of the major issues with blockchain systems in the context of using cryptocurrencies is their reliance on users. Any action or information on the blockchain is considered valid if most users accept it. This vulnerability is exploited in what is known as a “51% attack”. In this attack, the goal is to temporarily control more than 51% of the mining capacity of the entire network. Once this control is achieved, false data can be entered into the generation of the current block and validated. The false data is considered valid because the remaining 49% of users cannot reverse this. However, it is essential to note that the current power of the Bitcoin blockchain is substantial, and the network of miners is extensive, making this type of attack unlikely, given the current number of users and the advancement of computing technology.

Another drawback is the potential for the blockchain network to become overloaded due to the volume of transactions. An attack with a high volume of transactions could potentially cripple the blockchain for some time, resulting in the isolation of cryptocurrency exchanges. In theory, this could lead to a loss of user confidence and a significant drop in the price of the cryptocurrency.

Uncertainties inherent in the characteristics of Bitcoin as a digital currency

One of the significant issues with cryptocurrencies today is their reliance on the Proof of Work (PoW) algorithm to a certain extent. The global economy heavily depends on digital signature technology, with most digital signatures being created using a few simple numbers. This uneven distribution of operations reduces cryptographic stability and poses a potential challenge to economic activities in the next 50 years, not just for Bitcoin.

Another concern is the need for more decentralization and the need to store node addresses. In the event of a 51% attack on Bitcoin services, control over approximately 8.000 nodes could allow fake transactions to be introduced into the system. If most users are successfully redirected to fake nodes within 10 minutes, significant damage could be done to the system.

Additionally, the implementation of the Proof of Work protocol in mining, where the mining reward goes to only one successful miner, has led to the unification of miners in mining pools. While this makes the mining process more predictable, it makes the system less resistant to a 51% attack.

A more immediate challenge for Bitcoin is the issue of limiting both the number of coins and the number of transactions per second. As the world evolves, it's increasingly likely that cryptocurrency may no longer be a convenient medium of exchange due to its price or exchange speed. This could lead to a collapse of the industry in the future.

Furthermore, future mining limitations may lead to a significant amount of equipment being released on the market, which could be used for fraudulent actions with cryptocurrency. The same equipment used for mining can also be used to hack wallets successfully. Considering the dormant wallets with significant funds stored since the beginning of the cryptocurrency era, this may lead to a collapse of the cryptocurrency rate as early as 2040.

Uncertainties arise from speculative phenomena in the cryptocurrency market

Speculation and leverage pose potential problems for cryptocurrency markets. The most significant issue is the synchronization of speculators' actions at certain times, leading to significant market fluctuations. This is further exacerbated by the introduction of copy trading, which enables traders to share strategies with others at the exchange level, making synchronization even easier.

Another risk is the accumulation of substantial finances in speculative assets, potentially leading to a stock market crash. Currently, there is no legal regulation of cryptocurrency exchanges, and tracking the amount of collateral on the exchanges is challenging. Additionally, part of the collateral is always in cryptocurrency, the exchange rate of which can fluctuate significantly, especially when funds need to be transferred through the exchange into real money. These transaction volumes can significantly impact the cryptocurrency market and the pricing mechanism for cryptocurrency.

MODELING OF IMPULSE PROCESSES OF THE COORDINATES OF THE CM VERTICES OF THE USE OF CRYPTOCURRENCY

Based on cause-and-effect relationships, the adjacency matrix A of the impulse process of the cognitive map (7) was developed, which has the following components A_{11}, A_{12}, A_{21}, A_{22}:

$$A_{11}=\begin{bmatrix}
0.4 & 0 & 0 & 0 & 0 & 0 & 0 & -0.4 & 0.15 & 0 & 0 & 0.2 & -0.25 & 0 & 0\\
0 & 0.3 & 0.15 & 0 & 0 & 0 & 0 & 0 & 0 & 0.1 & 0 & 0 & 0 & 0 & 0\\
0 & 0 & 0.4 & 0 & 0 & 0 & 0 & 0 & 0.3 & 0 & 0 & 0 & 0 & 0 & 0\\
0 & 0 & 0 & 0 & 0.3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\
0 & 0 & 0.7 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\
0.3 & 0 & 0 & 0 & 0 & 0 & 0.6 & 0.4 & 0 & 0 & 0 & 0 & -0.4 & 0 & -0.3\\
0 & 0.5 & 0 & 0.3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -0.5 & 0\\
0 & 0 & 0 & 0.25 & 0.3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.8 & 0 & 0 & 0 & 0 & 0.7 & 0\\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.7 & 0 & 0 & 0 & 0 & 0\\
0.7 & 0 & 0 & 0 & 0 & 0.4 & 0 & 0.3 & 0 & 0 & 0 & 0.2 & 0 & 0 & 0\\
0 & 0.6 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.2 & 0\\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.8 & 0 & 0\\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.8 & 0\\
0.5 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.4 & 0 & 0 & 0 & 0
\end{bmatrix}$$

$$A_{12}=\begin{bmatrix}
0 & 0 & 0 & 0 & 0 & 0 & 0 & -0.7\\
0 & 0.5 & 0.2 & 0 & 0 & 0 & 0 & -0.3\\
-0.2 & 0 & 0.15 & 0 & 0.15 & 0 & 0 & -0.2\\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\
-0.2 & 0 & 0 & 0 & 0.3 & 0 & 0 & 0\\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\
0 & 0 & 0 & 0 & 0 & 0 & 0 & -0.2\\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\
0 & 0 & 0 & 0.4 & 0 & -0.3 & 0.15 & -0.25\\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.3
\end{bmatrix}$$

$$A_{21}=\begin{bmatrix}
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\
0 & -0.1 & 0.2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -0.2 & 0 & 0 & 0 & 0 & 0\\
0.3 & 0.1 & 0 & 0 & 0 & 0.2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\
0 & 0 & 0 & 0.5 & 0 & 0 & 0 & 0 & 0 & 0.3 & 0 & 0 & -0.15 & 0 & 0\\
-0.1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0
\end{bmatrix}$$

$$A_{22}=\begin{bmatrix}
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\
0 & 0.4 & 0 & 0 & 0 & 0 & 0 & 0\\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\
0 & 0 & 0 & 0 & 0 & 0 & 0.3 & 0\\
0 & 0 & 0 & 0 & -0.15 & 0 & 0 & 0\\
0 & 0 & 0.3 & 0 & 0 & 0 & 0 & 0\\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0
\end{bmatrix}$$

The eigenvalues of the matrix A are $0.989+0\mathrm{i}$, $0.592+0.282\mathrm{i}$, $0.489+0\mathrm{i}$, $0.22+0.333\mathrm{i}$, $0.22-0.333\mathrm{i}$, $0.08+0.336\mathrm{i}$, $-0.334+0\mathrm{i}$, $-0.188+0.13\mathrm{i}$, $-0.188-0.13\mathrm{i}$, $0.142+0\mathrm{i}$, $0.064+0.102\mathrm{i}$, $0.064-0.102\mathrm{i}$, $-0.122+0\mathrm{i}$, $0\mathrm{i}$, $0\mathrm{i}$, $0\mathrm{i}$, $0\mathrm{i}$, $0.7+0\mathrm{i}$, $0.8+0\mathrm{i}$, $0\mathrm{i}$, $0\mathrm{i}$. The eigenvalues of the matrix are smaller than one by module. Therefore, the impulse responses of the closed-loop control for models (5) and (6) are stable.

Modeling the impulse processes of the coordinates of the CM vertices is carried out to examine the movement of critical vertices 1, 2, 3, 6, and 20 under the influence of random disturbances applied to vertices 18, 22, and 23.

Figs. 8–12 display graphs of the CM coordinates' free movement, including the cryptocurrency exchange rate, the number of active cryptocurrency users, the amount of mining, and the efficiency of miners.

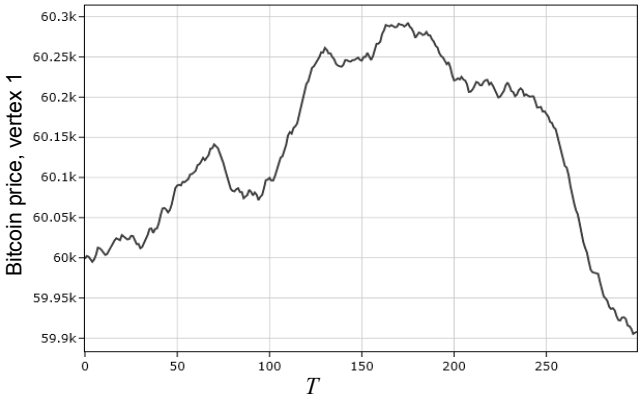


Fig. 8. Free movement of the exchange rate of cryptocurrencies (vertex 1)

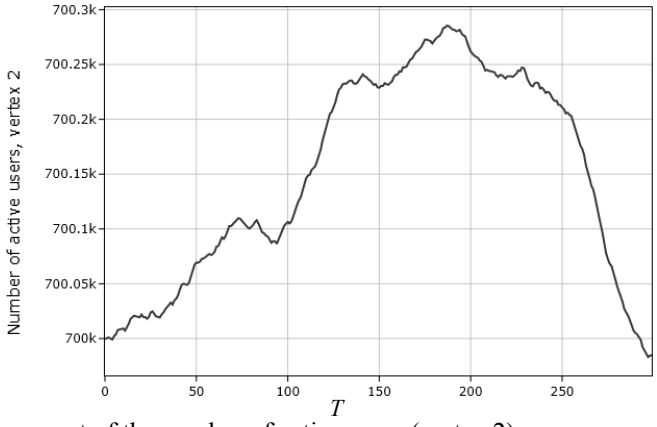


Fig. 9. Free movement of the number of active users (vertex 2)

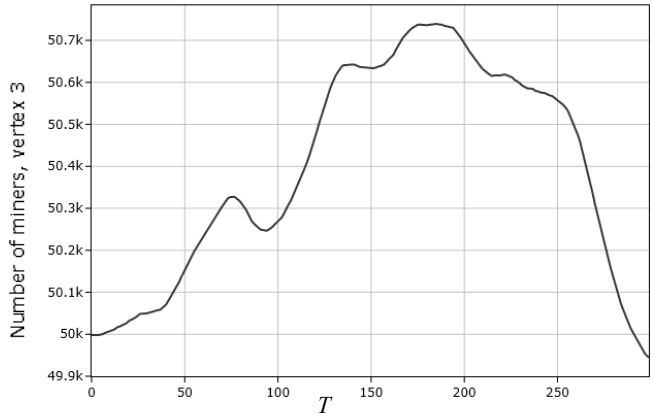


Fig. 10. Free movement of the number of miners (vertex 3)

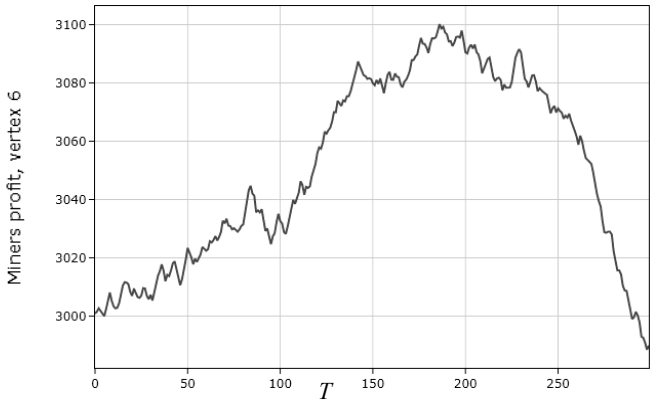


Fig. 11. Free fluctuations of mining profitability (vertex 6)

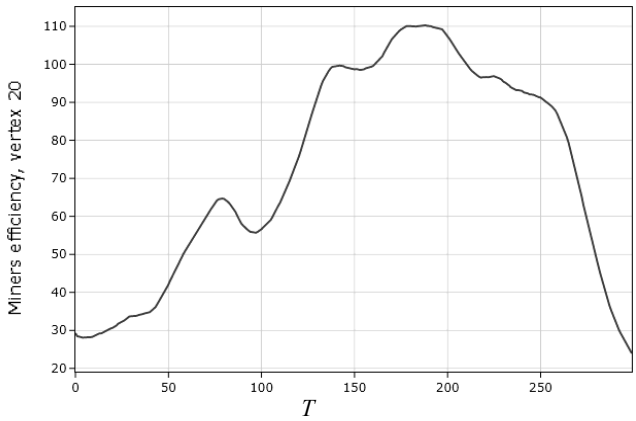


Fig. 12. Free fluctuations of mining efficiency (vertex 20)

For a clearer simulation, initial values of 60000, 700000, 50000, 3000, and 30 were set as the system's starting point for vertices 1, 2, 3, 6, and 20. The most significant changes affected the profitability of mining and cryptocurrency's rate while other changes occurred gradually and were highly correlated. There patterns closely mirror real marked behavior. Specifically, as the number of diskettes increases, the behavior of the cryptocurrency exchange rate closely resembles that of the real market (Fig. 13)

Fig. 13. The behavior of the cryptocurrency exchange rate over an extended period

Therefore, these graphs comprehensively describe the dynamic properties of the cryptocurrency circulation's CM. The model built reflects the primary interactions of cryptocurrencies in financial markets and corresponds to the simulated phenomenon.

CONCLUSION

The work performs a systematic study of the dynamic properties of mining principles when using cryptocurrency on financial exchanges, which are used to perform cryptocurrency exchange operations by conducting transactions in digital form outside the banking system. For this, a cognitive map of the use of cryptocurrency as a complex system was developed, in which all the functions of mining processes in interaction with speculative instruments in conditions of uncertainties and risks are revealed.

In the article, the digital simulation of impulse processes in the CM was carried out while investigating the dynamic properties of the free movement of the tops of the CM under the action of random disturbances.

REFERENCES

1. H. Massias, X. Serret Avila, and Quisquater Jean-Jacques, "Design Of A Secure Timestamping Service With Minimal Trust Requirement," in *20th Symposium on Information Theory in the Benelux, Citeseer, May, 1999*.
2. Satoshi Nakamoto, *Bitcoin: A peer-to-peer electronic cash system*. 2008. Accessed on: May 2, 2024. [Online]. Available: https://git.dhimmel.com/bitcoin-whitepaper/
3. Meni Rosenfeld, "Analysis of bitcoin pooled mining reward systems," *arXiv preprint, arXiv:1112.4980.* 2011. doi: https://doi.org/10.48550/arXiv.1112.4980
4. V. Romanenko, Yu. Miliavskyi, and G. Kantsedal, "An adaptive system for stabilizing unstable cryptocurrency rate based on a cognitive map impulse process model," *International Scientific Technical Journal "Problems of Control and Informatics,"* 66(2), pp. 11–23. doi: 10.34229/1028-0979-2021-2-2
5. V. Romanenko, Y. Miliavskyi, and H. Kantsedal, "Application of Impulse Process Models with Multirate Sampling in Cognitive Maps of Cryptocurrency for Dynamic Decision Making," in M. Zgurovsky and N. Pankratova (eds.) *System Analysis & Intelligent Computing: Theory and Applications*. Springer International Publishing, Cham, 2022, pp. 115–137. doi: https://doi.org/10.1007/978-3-030-94910-5_7
6. V. Romanenko, Y. Miliavskyi, and H. Kantsedal, "Stabilization of Impulse Processes of the Cognitive Map of Cryptocurrency Usage with Multirate Sampling and Coordination Between Some Nodes Parameters," in M. Zgurovsky, N. Pankratova (eds) *System Analysis and Artificial Intelligence. Studies in Computational Intelligence*, vol. 1107. Springer, Cham, 2023, pp. 83–100. doi: https://doi.org/10.1007/978-3-031-37450-0_5
7. V. Romanenko, Y. Miliavskyi, and H. Kantsedal, "Automated Control Problems for Dynamic Processes Applied to Cryptocurrency in Finansial Markets," in Y. Kondratenko, V. Kuntsevich, A.Chikrii, Y. Gubarev (eds.) *Recent Developments in Automatic Control Systems*. River Publishers, 2022, pp. 417–442. doi: 10.13140/RG.2.2.24246.42561/1
8. G. Ebrahimpour, M. Sayad Haghighi, "Is Bitcoin Future as Secure as We Think? Analysis of Bitcoin Vulnerability to Bribery Attacks Launched through Large Transactions," *ACM Trans. Priv. Secur.*, vol. 27, issue 2, Article 14, May 2024, 22 p. doi: https://doi.org/10.1145/3641546

Received 10.05.2024

INFORMATION ON THE ARTICLE

Viktor D. Romanenko, ORCID: 0000-0002-6222-3336, Educational and Research Institute for Applied System Analysis of the National Technical University of Ukraine "Igor Sikorsky Kyiv Polytechnic Institute", Ukraine, e-mail: romanenko.viktorroman@gmail.com

Heorhii O. Kantsedal, ORCID: 0000-0003-2740-2176, Educational and Research Institute for Applied System Analysis of the National Technical University of Ukraine "Igor Sikorsky Kyiv Polytechnic Institute", Ukraine, e-mail: g.kantsedal@protonmail.com

СИСТЕМАТИЧНІ ДОСЛІДЖЕННЯ ІНСТРУМЕНТІВ ВИКОРИСТАННЯ КРИПТОВАЛЮТИ НА ФІНАНСОВИХ РИНКАХ / В.Д. Романенко, Г.О. Канцедал

Анотація. Проаналізовано процеси майнінгу за використання криптовалюти на фінансових біржах. Розглянуто когнітивну карту (КК) використання криптовалюти як складної системи. Розкрито всі функції процесів майнінгу у ході взаємодії зі спекулятивними інструментами в умовах невизначеності та ризиків. Для дослідження динамічних властивостей вільного руху вершин КК під дією випадкових збурень виконано цифрове моделювання імпульсних процесів у КК.

Ключові слова: криптовалюта, когнітивна карта, майнінг, імпульсні процеси.

UDC 616-036.22.001.57:616.9-022:578.834.1
DOI: 10.20535/SRIT.2308-8893.2024.4.02

SHORT-TERM FORECASTING OF THE MAIN INDICATORS OF THE COVID-19 EPIDEMIC IN UKRAINE BASED ON THE SEASONAL CYCLE MODEL

A.B. ALYOKHIN, A.B. BRUTMAN, A.N. GRABOVOY, T.V. SHABELNYK

Abstract. The authors of this study propose a method of short-term forecasting of time series of the main indicators of the COVID-19 epidemic, which has a pronounced seasonality. This method, which has no direct analogies, provides the decomposition of a general forecasting task into several simpler tasks, such as the tasks of building a model of the seasonal cycle of a time series, aggregating the original time series, taking into account the duration of the seasonal cycle, forecasting an aggregated time series, developing an aggregated forecast into a forecast in the original time scale, using the seasonal cycle model. The solution for each task allows the usage of relatively simple methods of mathematical statistics. The article provides a formally rigorous description of all procedures of the method and illustrations of their numerical implementation on the example of a real forecasting task. The use of this method for short-term forecasting of the COVID-19 epidemic development in Ukraine has systematically demonstrated its effectiveness.

Keywords: COVID-19 epidemic, time series, short-term forecasting, seasonal cycle, indicators.

INTRODUCTION

Despite the fact that the first works on mathematical epidemiology appeared as early as the 20s of the last century [1], the COVID-19 pandemic, especially in the initial phase, created significant difficulties in building high-quality forecasts of the disease spread based on mathematical models [2–4]. This, along with the enormous scale of the pandemic, as well as its socio-economic consequences, has attracted the attention of scientists around the world to the problem of quantitative forecasting of the development of the COVID-19 epidemic.

Currently, a wide variety of mathematical tools are used to model and forecast the spread of COVID-19, among which the main place is occupied by systemic models of epidemics, including simulation, statistical models and methods for forecasting time series, methods and models of artificial intelligence (neural networks and etc.).

LITERATURE ANALYSIS AND PROBLEM STATEMENT

Taking into account the pronounced seasonality of COVID-19 epidemic development in various countries of the world, the Box–Jenkins (SARIMA) [5] and Holt-Winters [6] methods, which take into account seasonal effects, are the most widely used for statistical modelling of epidemics. SARIMA models are a linear

combination of series and seasonal profile elements, as well as past forecast errors considering autoregression. The Holt-Winters method uses the technique of triple exponential smoothing applied to the main components of a statistical time series: series level, trend, seasonal component. Both models have their own advantages and disadvantages.

This, in the absence of the only best approach or forecasting model, taking into account the principle of multiple models generally recognized in forecasting theory, leaves room for development and determines the relevance of efforts aimed at developing alternative approaches to forecasting which more accurately reflect the specifics of the forecasting object, including features of seasonal cyclicity, increasing the level of transparency, formalization and simplicity of the procedures applied.

The paper proposes a new method that decomposes the general problem of forecasting into a number of simple problems. These are the tasks of building a model of the seasonal cycle of a time series, aggregating the original time series taking into account the duration of the seasonal cycle, forecasting the aggregated time series and transforming the aggregated forecast into a forecast in the original time scale using the seasonal cycle model.

PURPOSE OF THE STUDY

The aim of the study is to develop a "direct" method for short-term forecasting of time series of the main indicators of the COVID-19 epidemic based on a seasonal cycle model.

MATERIALS AND METHODS OF THE STUDY

The study is based on statistical data from the Public Health Center of the Ministry of Health of Ukraine [7], as well as indicators derived from them, which together characterize the spread of the COVID-19 epidemic in Ukraine over the entire observation period. This statistical data is a multidimensional discrete time series with a 1-day increment, which includes univariate statistical series of three basic indicators (increase in infected, deaths, recoveries), five derived daily and cumulative indicators (increase in active cases; total number of infected, deaths, recoveries, number of active cases) and three synthetic derived indicators (disease spread and mortality, epidemic progression).

To determine the availability and duration of a cycle in the statistical series of reference indicators, autocorrelation methods were used, implemented in the statistical software IBM SPSS Statistics, STATISTICA and MS Excel 2019. For aggregating time series, the corresponding series conversion procedures of the specified application program packages were used.

The building of cycle models, as well as trend models of aggregated time series of reference indicators, was carried out using generally accepted methods of analytical alignment of time series and, in particular, curve-fitting methods using non-linear optimization methods.

To develop and evaluate indicative forecasts of reference indicators, statistical series were used that included data from the 36th to the 57th week of the

entire observation period. Forecasts were built for a period of $1+1$ weeks, where the forecast for the second week is an estimate. Statistical series of reference indicators from the 36th to the 55th week were used as training sequences, and the next two weeks were used as a verification (forecasting) sequence. The accuracy of forecasts for all forecasted indicators was assessed using the calculated (forecasting) data and data from verification sequences using the MAPE accuracy indicator.

MATHEMATICAL MODELS

In the general scheme of the developed method for short-term forecasting of the COVID-19 epidemic progress, for each one-dimensional statistical series of the reference indicator, the following procedures are performed:

– identification of the seasonal cycle in the time series and determination of its duration;

– building of a general model of the seasonal cycle (seasonal profile of the time series) or a set of models of seasonal cycles of the time series;

– removal of the seasonal component by aggregating the original time series with a step equal to the cycle duration, and building a trend model of the aggregated series;

– trend forecasting for the forecast period in the aggregated time scale;

– disaggregation of the aggregated forecast using the cycle model (cycle models), which is the decomposition of the aggregated forecast into a forecast in the original time scale.

The calculation of the forecast values of all derived indicators of the COVID-19 epidemic is based on the forecasts of the reference indicators.

For a formal description of the above stated procedures, we introduce the following notation:

$\Re_S$ — original multidimensional statistical time series;

n — dimension of the time series $\Re_S$;

I — set of indices of the components of the original time series, $I = \{1,\ldots,n\}$;

t — discrete moment of time;

T_s — length of the time series $_S$;

T_F — duration of the forecasting period;

$\Re_F$ — forecasting time series of length T_F;

Let R — n-dimensional vector of COVID-19 epidemic indicators. Then

$$R = (R^*, R', R'')\,,$$

where R^* — vector of epidemic reference indicators of dimension n^*; R' — vector of daily and cumulative derivative indicators of the epidemic of dimension n'; R'' — vector of synthetic derivative indicators of dimension n''.

In this study, the following assumptions have been made:

$$R^*\{\Delta TC, \Delta D, \Delta R\}\,,$$

where $\Delta TC, \Delta D, \Delta R$ — daily increase of infected, deaths, recovered respectively.

$$R' \in \{\Delta AC, TC, D, R, AC\},$$

where ΔAC — daily increase in the number of active cases at time *t*; TC, D, R — total number of infected, deaths and recovered; AC — number of active cases.

$$R'' \in \{R_t, I_{TC}, I_{CC}, I_P\},$$

where R_t — infection spread rate; I_{TC} — fatality rate (according to the number of infected (total cases)); I_{CC} — fatality rate (according to the number of closed cases); I_P — epidemic progress indicator.

The following notations refer to an arbitrary one-dimensional time series $\Re_S(i)$, where $i \in I^*$, where I^* — a subset of epidemic reference indices ($|I^*| = n^*$). For simplicity, the index *i*, where it does not generate ambiguity, will be omitted below:

T_C — length of the seasonal cycle time series $\Re_S(i)$;

k — number of complete cycles in time series $\Re_S(i)$;

t' — sequence number of the seasonal cycle or, what is the same, a discrete point in time in the aggregated time scale;

t'' — sequence number of the observation in the cycle;

$\Re_{SC}(t')$ — t'-th statistical time series of length T_C (series of seasonal cycle elements t').

$\Re'_S$ — aggregated statistical time series of observations;

T'_S — length of the aggregated time series $\Re'_S$ in the aggregated time scale (in units of the number of complete cycles, $T'_S = k$);

T'_F — duration of the forecast period in the aggregated time scale.

$\Re_{SCN}(t')$ — t'-th normalized seasonal cycle (normalized cycle time series);

M_{CN} — general model of normalized seasonal cycle (seasonal profile of time series $\Re_S(i)$ — model time series of length T_C).

Considering the introduced notation, the formal description of all stages of forecasting of the reference one-dimensional series $\Re_S(i)$, $i \in I^*$, is as follows.

Stage 1. Analysis of the time series cyclicity.

1.1. Determining the availability, duration of T_C and the number *k* of complete cycles in the original time series $\Re_S(i)$.

This stage is implemented by standard methods of autocorrelation analysis.

Stage 2. Building of the normalized cycle model $M_{CN}(i)$ of the time series.

2.1. Normalization of the levels of the seasonal cycle time series $\Re_{SC}(i,t')$ for all t', $t' \{1,\ldots,T'_S\}$.

Normalization is carried out according to the formula:

$$r_{CN}(i,t',t'') = \frac{r_{CS}(i,t',t'')}{\sum_{t''} r_{CS}(i,t',t'')} \quad \forall t',$$

where $r_{CS}(i,t',t'')$ — t''-th element of the t'-th cycle of the time series $\Re_S(i)$; $r_{CN}(i,t',t'')$ — t''-th element of the t'-th cycle of the time series $\Re'_S(i)$.

2.2. Building of a mathematical model of the normalized cycle $M_{CN}(i)$ based on T_S observations — the set of all cycles $\{\Re'_{SCN}(i,t')\}$.

The seasonal cycle model $M_{CN}(i)$ is developed on the elements of the normalized time series of cycles $\{\Re_{SCN}(t')\}$, the set of which acts as a repetitiveness. In the study, a degree l polynomial from the sequence number of the cycle element was used as a model of the seasonal cycle, where $l < T_C$.

$$M_{CN}(i.t'') = a_l (t'')^l + a_{l-1}(t'')^{l-1} + \ldots + a_1 t'' + a_0,$$

where $t'' \{1,\ldots,T_c\}$; A — parameter vector of the degree l polynomial, $A = (a_0,\ldots,a_l)$.

The parameters A of the $M_{CN}(i)$ model are defined as a solution to an optimization problem of the type:

$$\min_A \sum_{t',t'} \alpha_{t'} (r_{CM}(i,t'') - r_{CN}(i,t',t''))^2 \quad \forall t', t' \in \{1,\ldots,k\};$$

$$r_{CM}(i,t'') = a_l (t'')^l + a_{l-1}(t'')^{l-1} + \ldots + a_1 t'' + a_0 \quad \forall t''; \quad 0 \le r_{CM}(i,t'') \le 1 \quad \forall t'',$$

$r_{CM}(i,t'')$ — t''-th model cycle element $M_{CN}(i)$; α_t — significance coefficient of the t'-th cycle.

In the model, the weight coefficients $\{\propto_{t'}\}_{t'}$ are set using the logistic function of the cycle t' sequence number

Stage 3. Building a model of the aggregated series $\Re'_S(i)$.

3.1. Aggregation of the original one-dimensional time series $\Re_S(i)$ — formation of an aggregated time series $\Re'_S(i)$ of length T'_S.

The aggregation of the series $\Re_S(i)$ is carried out with a step equal to the duration of the cycle T_C, using the operation of summing as the aggregation operation.

3.2. Building a trend model $M_T(i)$ of the aggregated time series $\Re'_S(i)$.

It is carried out by an arbitrary method of analytical alignment of time series.

Stage 4. Forecasting of the initial time series $\Re_S(i)$.

4.1. Development of an aggregated forecast $\Re'_F(i)$ for the forecast time T_F.

It is carried out by extrapolating the trend using the $M_T(i)$ trend model for the forecast period T_F in the aggregated time scale.

4.2. Formation of the forecasting time series $\Re_F(i)$ based on the aggregated forecast $\Re'_F(i)$.

At this stage, deconvolution of each element $r'_F(i,t')$ of the forecasting series $\Re'_F(i)$ is carried out using the $M_{CN}(i)$ cycle model, as a result, each element is replaced by the time series of the corresponding cycle *t*' in the original time series scale, the elements of which are determined by the following formula:

$$r_F(i,t) = r'_F\left(i, \left[\frac{t}{T_C}\right]+1\right) \times \times M_{CN}\left(i, \left(t - \left[\frac{t}{T_C}\right] \times T_C\right)\right) \quad \forall t,\ t \in \{1,\ldots,T'_F \times k\};$$

where $[a]$ — an integer part of the number *a*.

As a result of this stage, the aggregated forecast series $\Re'_F(i)$ in the aggregated time scale is disaggregated into the forecast series $\Re_F(i)$ in the original time scale.

After forecasting of all COVID-19 epidemic reference indicators, the forecast values of the epidemic derived indicators are calculated in accordance with the formulas below.

Stage 5. Calculation of the main derivatives (synthetic) indicators of the COVID-19 epidemic

The daily increase in active cases ΔAC at each time *t* is derived from the increase in infected ΔTC, deaths ΔD and recovered ΔR and is calculated by the following formula:

$$\Delta AC_t = \Delta TC_t - \Delta D_t - \Delta R_t \quad \forall t$$

Cumulative indicators TC, D, R, AC are calculated using the following formulas:

$$P_t = P_{t-1} + \Delta P_t \quad \forall P,\ P \in \{TC,\ D,\ R,\ AC\},\ t\,.$$

The I_{TC} fatality rate is calculated using the formula:

$$I_{TC} = D_t \big/ TC_t \quad \forall t\,.$$

and the I_{CC} fatality rate — according to the formula:

$$I_{CC} = D_t \big/ (D_t - R_t) \quad \forall t$$

As one of the meaningful tools for monitoring and analysing the development of the COVID-19 epidemic, the authors propose an I_P progress indicator, which is the ratio of fatality rates:

$$I_P(t) = I_{CC}(t) \big/ I_{TC}(t) = (D_t + R_t) \big/ TC_t \quad \forall t.$$

A physical meaning of this indicator is transparent, reflected in its name and indicates the traversed path in % of the epidemic progress from the moment of its emergence to the moment of completion.

Not least informative is the statistical analogue of the reproduction coefficient R_0 — the infection spread coefficient R_t, which is proposed to be calculated taking into account the seasonality of the time series:

$$R_t = \sum_{l=t-T_c+1}^{t} R_l \Big/ \sum_{m=t-2T_c+1}^{t-T_c} R_m \quad \forall t.$$

RESULTS AND DISCUSSION

Let us illustrate the implementation of each of the above mentioned stages, using the example of short-term forecasting of the development of the COVID-19 epidemic in Ukraine.

Stage 1. Analysis of the time series cyclicity of the reference indicator.

In the study, to assess the availability and duration of seasonality in these series, the FORECAST.ETS.SEASONALITY function contained in MS Excel 2019 was used, which made it possible to detect a seasonal cycle equal to 7 days (see Fig. 1).

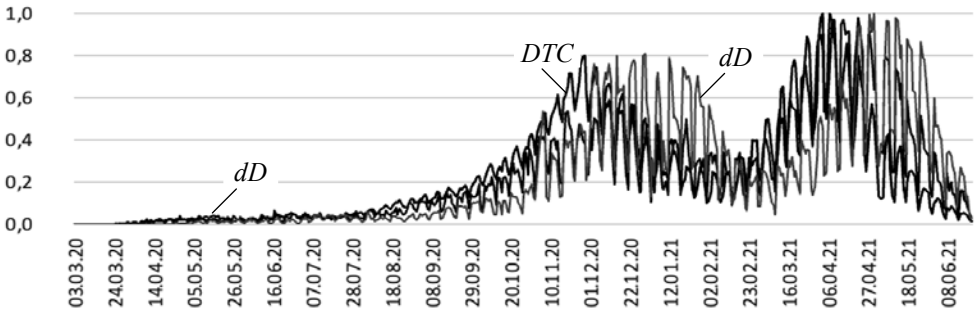


Fig. 1. Daily increase in Infected (ΔTC), Deaths (ΔD) and Recovered (ΔR) from COVID-19 in Ukraine (normalized data)

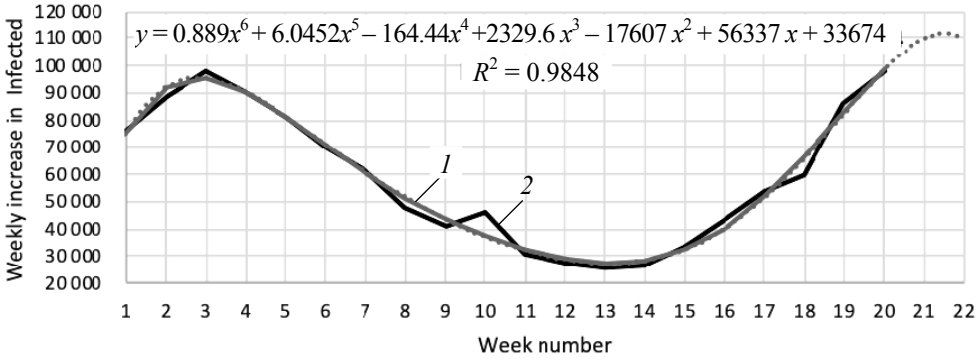
The proposed method will be further illustrated for the time series of the indicator of the daily increase in infected (ΔTC).

Stage 2. Building a seasonal cycle model.

As a model of the weekly seasonal profile, a degree 4 polynomial was used, the parameters of which are determined by solving the corresponding nonlinear optimization problem. In doing so, the weight coefficients of the actual weekly cycles were set using the logistic function of the serial number of the cycle.

Stage 3. Building a trend model of an aggregated time series.

In this study, a polynomial no higher than a degree 6 is used as a model of the aggregated series for illustrative purpose (Fig. 2).



Stage 4. Forecasting the initial time series.

Extrapolation of the aggregated time series is carried out by substituting the number of the forecast period into the equation of the $y = f(x)$ (see Fig. 2), and

disaggregation of the forecast — by deconvolution of the obtained values using the cycle model.

The results of these procedures for all reference indicators ΔTC, ΔD and ΔR, as well as the daily increase in active cases ΔAC are shown in Fig. 3.

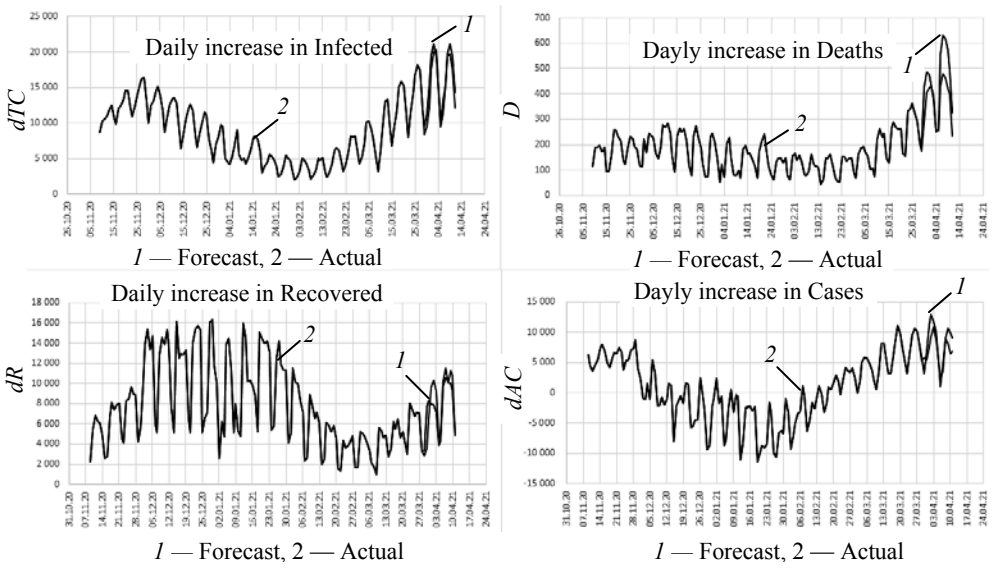


Fig. 3. Actual and forecast values of daily indicators of the COVID-19 epidemic in Ukraine

Step 5. Calculation of derived indicators of the COVID-19 epidemic.

The forecast values of derived indicators are calculated in accordance with the formulas described above. Their correspondence to the actual data is shown in Fig. 4.

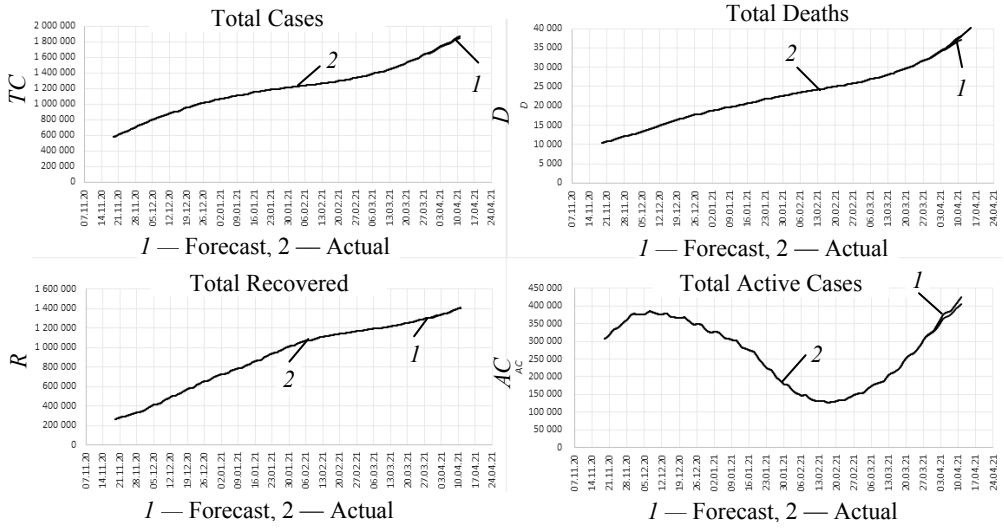


Fig. 4. Actual and forecast values of cumulative indicators of the COVID-19 epidemic in Ukraine

Evaluation of forecast accuracy. In order to evaluate the performance of the proposed method, the accuracy of forecasts has been assessed for the main indicators of the COVID-19 epidemic in Ukraine using the mean absolute percentage error MAPE, as well as a comparative analysis of the accuracy of this

method with the exponential smoothing methods and SARIMA mentioned at the beginning of the paper (see Table).

As follows from the data in Table, the estimation results confirm the fact, well known in statistical forecasting of time series, that different methods deal with different situations in different ways, and this fact does not allow us to give preference to one of them in general.

Estimates (MAPE) of the forecast accuracy of the main indicators COVID-19 epidemic in Ukraine, %

N	Indicator	Weekly forecast			2-week forecast		
		M_1	M_2	M_3	M_1	M_2	M_3
1	TC	0.56	0.17	0.14	0.60	0.30	0.10
2	D	0.66	1.01	0.74	1.27	1.17	0.92
3	R	0.21	0.48	0.31	0.22	1.13	0.65
4	AC	1.91	2.00	2.00	2.76	3.22	2.88
5	ΔTC	12.30	19.28	11.19	10.48	12.83	8.59
6	ΔD	14.51	10.66	13.07	21.03	8.44	10.02
7	ΔR	34.28	35.27	39.06	21.20	36.06	29.90
8	ΔAC	27.70	16.41	17.19	28.85	34.97	31.14
9	I_{TC}	0.16	1.02	0.88	0.70	0.96	1.00
10	I_{CC}	0.45	0.52	0.42	1.26	0.55	0.35
11	I_P	0.34	0.50	0.46	0.57	0.92	0.74

where M_1 — MAPE of proposed method; M_2 — MAPE of exponential smoothing; M_3 — MAPE of SARIMA.

As follows from the data in Table 1, the estimation results confirm the fact, well known in statistical forecasting of time series, that different methods deal with different situations in different ways, and this fact does not allow us to give preference to one of them in general.

It should also be noted that the procedures that implement these methods in the statistical software IBM SPSS Statistics provide for the generation of a series of models and the selection of the best among them. In assessing the accuracy of the method proposed in this study base forecast models were used, and the potential for improvement of forecast models, which is inherent in the method, is not utilized. The task of realizing this potential to improve the efficiency of the method is the subject of further research.

The corresponding calculations were carried out on a systematic basis within the framework of the activities of the "Working Group on Mathematical Modelling of Problems Related to the SARS-CoV-2 Coronavirus Epidemic in Ukraine" of the National Academy of Sciences of Ukraine during 2020 and confirm the method effectiveness proposed by the authors of the study.

CONCLUSIONS

The time series of daily growth indicators of the COVID-19 epidemic have a pronounced seasonal pattern.

The method proposed by the authors makes full use of this circumstance and implements the idea of decomposing the general task of developing short-term

forecasts of the main indicators of the COVID-19 epidemic into a number of particular subtasks, for which simple methods of mathematical statistics are applicable. In particular, the "direct" method of identifying the seasonal cycle makes it possible to use quite simple mathematical models of an arbitrary form to describe the seasonal profile of a time series. Aggregation of the original time series with a step equal to the seasonal cycle duration enables to eliminate the seasonal component without distorting the information, and the problem of forecasting the initial time series can be transformed into the simpler problem of extrapolating the trend model of the aggregated time series with subsequent deconvolution (using the cycle model) of its forecasting values in daily values of the corresponding indicators.

Practical use of this method for developing short-term forecasts of the COVID-19 epidemic progress in Ukraine on a systematic basis has demonstrated quite satisfactory accuracy of forecasts (MAPE estimates).

REFERENCES

1. W.O. Kermack, A.G. McKendrick, "A Contribution to the Mathematical Theory of Epidemics," *Proc. Roy. Soc. Lond. A*., no. 115, pp. 700–721, 1972.
2. S. Borenstein, C.K. Johnson, "Modeling coronavirus: 'Uncertainty is the only certainty,'" *AP News*, April 8, 2020. Accessed on: February 10, 2022. [Online]. Available: https://apnews.com/article/public-health-health-us-news-ap-top-news-virus-outbreak-88866498ff5c908e5f28f7b5b5e5b695
3. V. Romanenko, Y. Milyavsky, "Combined control of multirate impulse processes in a cognitive map of COVID-19 morbidity," *System Research and Information Technologies*, no. 3, pp. 46–56, 2022. doi: https://doi.org/10.20535/SRIT.2308-8893.2022.3.03
4. L Guryanova, R. Yatsenko, N. Dubrovina, V. Babenko, and V. Gvozditskyi, "Machine learning methods and models, predictive analytics and applications: development trends in the post-crisis syndrome caused by COVID-19," *CEUR Workshop Proceedings*, 2927, pp. 1–7, 2021.
5. G.E.P. Box, G.M. Jenkins, G.C. Reinsel, and G.M. Ljung, *Time Series Analysis: Forecasting and Control*. New Jersey: John Wiley & Sons, Inc., 2015, 712 p.
6. P.R. Winters, "Forecasting sales by exponentially weighted moving averages" *Management Science*, no. 6, pp. 324–342, 1960.
7. "Koronavirusna infektsiia COVID-19 [Corona virus infection COVID-19]," *Tsentr hromadskoho zdorovia MOZ Ukrainy*. Available: https://phc.org.ua/kontrol-zakhvoryuvan/inshi-infekciyni-zakhvoryuvannya/koronavirusna-infekciya-covid-19
8. "Prohnoz rozvytku epidemii COVID-19 v Ukraini na period 7–13 hrudnia 2020 roku vid uchenykh Instytutu problem rynku ta ekonomiko-ekolohichnykh doslidzhen NAN Ukrainy [Forecast of the development of the COVID-19 epidemic in Ukraine for the period of December 7–13, 2020 by scientists of the Institute of Market Problems and Economic and Environmental Research of the National Academy of Sciences of Ukraine]," *Natsionalna akademiia nauk Ukrainy*, December 7, 2020. Available: https://www.nas.gov.ua/ UA/Messages/Pages/View.aspx?MessageID=7237

Received 30.08.2023

INFORMATION ON THE ARTICLE

Alexei B. Alyokhin, ORCID: 0000-0001-5209-8036, Mariupol State University, Ukraine, e-mail: aba99@ukr.net

Anna B. Brutman, ORCID: 0000-0002-7774-5356, National University "Zaporizhzhia Polytechnic", Ukraine, e-mail: a_brutman@yahoo.com

Alexandr N. Grabovoy, ORCID: 0000-0001-5705-9909, Bogomolets National Medical University, Ukraine, e-mail: angrabovoy@gmail.com

Tetiana V. Shabelnyk, ORCID: 0000-0001-9798-391X, Simon Kuznets Kharkiv National University of Economics, Ukraine, e-mail: tanya.shabelnik17@gmail.com

КОРОТКОСТРОКОВЕ ПРОГНОЗУВАННЯ ОСНОВНИХ ПОКАЗНИКІВ ЕПІДЕМІЇ В УКРАЇНІ НА ОСНОВІ МОДЕЛІ СЕЗОННИХ ЦИКЛІВ / О.Б. Альохін, А.Б. Брутман, О.М. Грабовий, Т.В. Шабельник

Анотація. Запропоновано метод короткострокового прогнозування часових рядів основних показників епідемії COVID-19, яким притаманна виражена сезонність. Зазначений метод, що не має прямих аналогів, передбачає декомпозицію загального завдання прогнозування на ряд більш простих завдань, таких як побудова моделі сезонного циклу часового ряду, агрегування вихідного часового ряду з урахуванням тривалості сезонного циклу, прогнозування агрегованого часового ряду, розгортання агрегованого прогнозу в прогноз у вихідній часовій шкалі за допомогою моделі сезонного циклу, вирішення кожного з яких допускає застосування відносно простих методів математичної статистики. Наведено формально строге описання всіх процедур методу та ілюстрації їх числової реалізації на прикладі реального завдання прогнозування. Застосування зазначеного методу для розроблення короткострокових прогнозів розвитку епідемії COVID-19 в Україні на систематичній основі продемонструвало його ефективність.

Ключові слова: епідемія COVID-19, часові ряди, короткострокове прогнозування.

UDC 004.932:621.397](045)
DOI: 10.20535/SRIT.2308-8893.2024.4.03

CROWD NAVIGATION MONITORING DURING EMERGENCIES

O.L. TYMOSHCHUK, M.O. TISHKOV, V.G. BONDARENKO

Abstract. The paper considers the task of crowd navigation monitoring, which might be performed using various sensors and technologies, with surveillance cameras being the most commonly employed. These cameras provide a video stream that typically lacks supplementary information. Extracting additional data from these streams could significantly enhance pedestrian behavior modeling and the automation of the monitoring process. A critical parameter in the analysis of pedestrian movement is their speed. The analytical method and the algorithm of pedestrians' speed estimation based on the surveillance camera video are proposed. The first step of the proposed algorithm is object detection and tracking between frames. The second step is the speed estimation method, which is based on calculating the real-world distances and knowing camera parameters and distances in pixels on the resulting image. Implementation of the algorithm was tested on real videos and showed an error of about 0.04 m/s.

Keywords: computer vision, object tracking, object speed, video surveillance.

INTRODUCTION

Internal navigation monitoring in areas of large crowds during emergencies requires advanced methods of computer vision, object tracking, and estimating flow speed.

Over the past decade, there have been numerous researches aimed at creating intelligent video surveillance systems. Some typical applications are listed as follows:

- public areas such as colleges, campuses, and governmental buildings;
- traffic monitoring;
- crowd management and analysis;
- home security and intrusion detection;
- home care and safety;
- public transport areas such as airports, seaports, and bus/train terminals;
- pedestrian detection and autonomous cars;
- remote military surveillance, border monitoring, perimeter surveillance for power plants, companies, etc [1].

One of the tasks of intelligent surveillance systems is collecting and processing data about objects' behavior. One of the core characteristics of the objects is their moving speed. This paper focuses on estimating pedestrians' speed, however, the same approach could be used for the estimation of any moving objects. Considering pedestrian threads, this data can help develop more effective and safe infrastructures during urban planning and transport engineering. In computer modeling and human behavior simulation, moving speed is a critical parameter for creating realistic agents, that reproduce human behavior in different use cases, like evacuation or interaction with other pedestrians. Thereby, data about the

moving speed of pedestrians could facilitate improvements in various technologies that improve different aspects of our lives.

Problem statement

This paper aims to develop the analytical method and the algorithm for estimating pedestrian speed using a video from a regular surveillance camera.

In order to achieve the goal, the next steps need to be done:

- Analysis of existing methods and approaches of detection, tracking, and speed estimation of moving objects, particularly pedestrians.
- Development of the method for estimating real-world distance, having an image from the camera.
- Implementation of the algorithm based on the developed method that can perform detection, tracking, storing of the pedestrians' tracked data, and estimating their speed.
- Testing of the implementation on the model problems. Algorithm quality evaluation.

Related papers

The computer vision has been investigated by numerous scientists worldwide [1–3]. There are research and implementations for object detection and classification in images and videos [1]. The implementation proposed by authors in this article is based on utilizing various sensors in video surveillance systems. Infrared or thermal cameras aid in detecting people or other objects with contrasting temperatures with the surrounding environment, providing the technical capability for object detection [1].

Radars and lidars also provide technical ability for object detection and accurate speed measurement. However, such sensors are expensive and rarely used. For instance, the number of surveillance cameras in London is about 127.000 [4], but the number of cameras with radar for speed estimation is only about 1000 speed cameras with radar [5]. Therefore, during further research of the methods and their implementations, we will rely on the existing radarless surveillance systems.

An alternative approach utilizing object tracking [2] proposes the extraction and tracking of objects in a video with a stationary background based on motion analysis within frames and representation of images and objects as sets of structural elements. Evaluation of this approach, according to the provided experimental data, reveals that the method does not consider a potential merging and loss of one of the objects, leading to data distortion.

There are numerous studies on object detection and tracking based on background separation with some specific improvements [6–9]. However, this approach has essential drawbacks as explained above.

The methods based on spatial filtering look promising for contour detection [3]. The implemented solution enables image adjustment for object contour detection. One of the advantages of this approach is the ability to simplify data processing having comparably low technical complexity.

Shvandt and Moroz consider a specific case of recognition and tracking of laboratory animals, mice, and fish specifically. The authors provide numerous methods of pattern recognition, including background subtraction and filtering

methods. Also, trainable models are considered, which are used for face recognition, for instance. Such models are more generic, meaning they can work in a wider range of situations, although, they are not ideal as well and can make mistakes or lose tracking objects. It is worth mentioning, that cases reviewed by the authors, — laboratory animal recognition and tracking, are specific and facilitate working conditions for algorithms like background separation since in laboratory conditions background might be homogenous and also there is a possibility to obtain its image without tracking objects [10].

Khan proposes a method of pedestrian detection based on foreground segmentation and calculating the speed on pixels [11]. The author focuses on pedestrian detection on the crosswalks and aims to detect slow speed movements. This approach has similar limitations to the previous author's. Also, the calculation speed in pixels on the image plane is the first step of estimating the real-world speed, which is explained further in this article.

Zhao and Li present a pedestrian tracking method combining a Histogram of Oriented Gradients detection and particle filter and a method for the detection of abnormal crowd activity [12].

In this article, on the stage of people recognition and tracking a well-known model YOLOv7 is used [13]. This model is rapidly developed. It is built on a complex deep-learning model based on a convolutional neural network, which consists of several convolutional and pulling blocks, that subtract characteristics of the image on a different scale level. The determined characteristics are further used to predict the position, size, and class of the object in the image.

Despite the breadth of previous research, pedestrian flow speed measurement has received comparatively little attention, despite its significance. Understanding the flow speed of pedestrians can help to detect unusual events like congestion or emergencies. Additionally, it is vital information for developing pedestrian traffic models, which are instrumental in architectural design and evacuation planning.

Teknomo et al. introduced a data collection framework for analyzing pedestrian flow in [14]. Their system autonomously identifies moving objects, tracks them, and records their positions along with timestamps. By leveraging this collected data on individual movements, the system could discern various characteristics of pedestrian traffic flow, including individual speeds, flow rates, average speeds, and directions. However, it's worth noting that their approach is constrained to video sequences captured from a top-view camera and does not ensure accurate results for videos recorded with lower angles.

Tordeux et al. use an artificial neural network for predicting pedestrian speed in different situations like corridors or bottlenecks [15]. The result of the implementation is prediction, but not the estimation of the physical movements which is not the same and could be used in different use cases and for different goals.

There are studies aimed at estimating pedestrian flow speed using portable devices like phones. Guo et al. propose a method of pedestrian speed estimation based on the human pose detected by sensors of smartphones [16]. Huang et al. use Wi-Fi sniffers to catch Wi-Fi probe requests from mobile devices to estimate pedestrian flow speed and number of people in the crowd [17].

The approach by Lee et al. [18] is closely related to the problem discussed in this research. They proposed a speed estimation algorithm based on calculating conversion factors between the angles on the camera image and real-world motion

vectors. Also, they track the heads of the people for the algorithm. However all people have different heights and the vertical distance from the camera to the tracking object is different for each pedestrian, so they propose a way to predict this distance based on statistical data and crowd density. It adds additional complexity and might be a root for inaccuracy in calculations. Additionally, when calculating conversion factors the authors assume that the rectangle on the camera image projects into a rectangle on the pedestrian movement plane although this is wrong since during central projection, a rectangle is projected into a trapezoid, which changes the provided calculations.

In the considered resources different approaches for moving object detection are proposed. One of them satisfies our needs so it will be used without any changes. However, the main focus of our research is the speed estimation. Method and implementation that cover all our needs and requirements have not been found. The method suggested by Teknomo et al. [14] doesn't work with cameras in their real-life positions, Lee et al. [18] propose a similar approach to ours, but simplifies projection which increases the error of calculations, Tordeux et al. [15] try to predict the speed instead of estimation, and Guo et al. [16] propose an interesting, but completely different approach using the radio signals detected from the pedestrians' phones.

The suggested analytical method allows pedestrians' speed estimation from the video of the camera located above the pedestrians, knowing the camera parameters.

PEDESTRIAN SPEED ESTIMATION METHOD

In this section, the proposed method for pedestrian speed estimation will be explained. The method is based on the distance estimation between two points on a real-world plane based on the points on the image plane. Knowing the distance and time between estimations the moving speed could be calculated.

Experiment model

Fig. 1 shows the pedestrian's movement from point A to point B. To calculate the pedestrian speed between these two positions we need to know the distance and time elapsed. Time could be defined from the video. The real-world distance we aim to estimate.

Fig. 1. Pedestrian movement

Using Cartesian coordinates and knowing the resolution of the image we can calculate the distance in pixels. In the Fig. 2 rectangle $A_0B_0C_0D_0$ is the full view of the camera, $\overrightarrow{BD}$ — the vector of the movement and the diagonal of the rectangle $ABCD$.

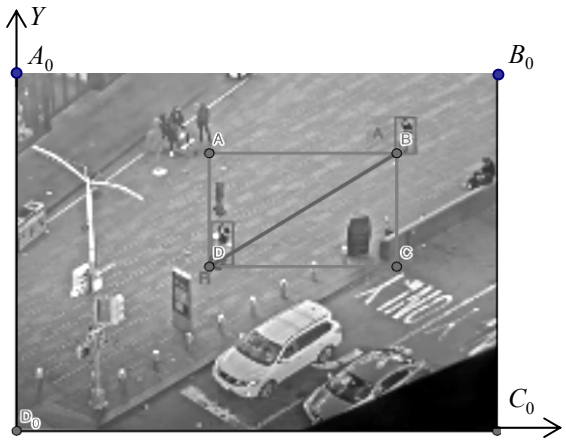


Fig. 2. Pedestrian movement formalization

Fig. 3 shows how the rectangle from Fig. 2 is projected to the real-world plane from the camera point located above the image plane.

In the result we need to estimate size of the vector $\overrightarrow{B_pD_p}$ on the real-world plane which is a projection of the vector of the pedestrian movement $\overrightarrow{BD}$ from the image plane.

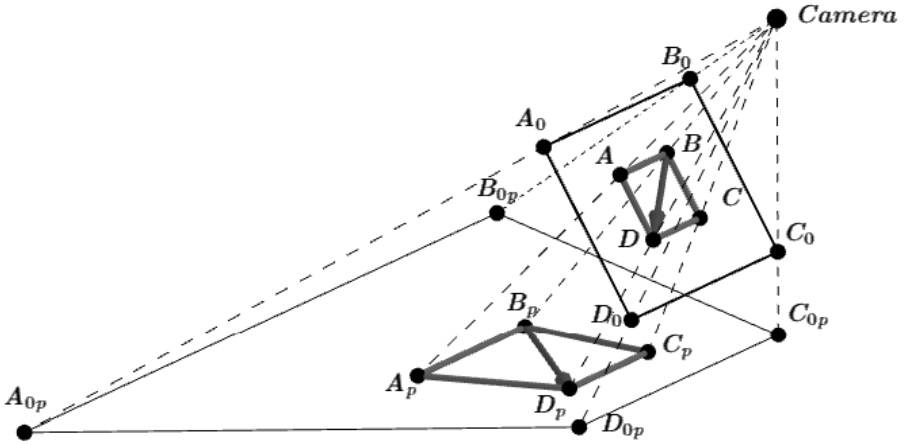


Fig. 3. Projection from image plane to a real-world plane

Mathematical formalization

The first thing that is worth mentioning is that the rectangle on the image plane, located at an angle relative to a real-world plane, is being projected from a point above the image plane to a trapeze on a real-world plane — Fig. 4, as shown in Fig. 1.

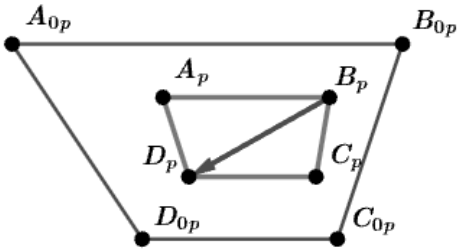


Fig. 4. Camera image and movement vector projection to a real-world plane

We aim to calculate the size of the vector *s* — the diagonal of the trapeze.

For better understanding, we will use simplified figures — Fig. 5.

According to a known formula, denoting the bigger base *AD* as a, we have for both cases of Fig. 5:

$$d = \sqrt{h^2 + (a + h\,ctg\,(\alpha))^2}\ . \qquad (1)$$

In order to find the required diagonal we need to find the height of the trapeze — h, its larger base — a, and the angle near the smaller base from which the diagonal extends — α.

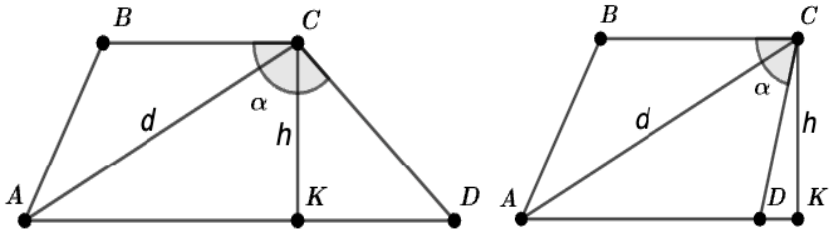


Fig. 5: Trapeze diagonal

The height of the trapeze

Depending on the location of points on the image plane, the calculation of height will vary. It depends on the vertical position of each point relative to the center of the image plane. There are 3 possible cases (Fig. 6).

Here:

C — camera position;

H — height of the camera position;

P_i, P_j — points of the vector on the image plane;

M — middle of the image plane;

Θ — angle of deviation of the camera from the vertical;

Θ_i — angle of deviation of the point P_i from image plane vertical;

Θ_j — angle of deviation of the point P_j from image plane vertical.

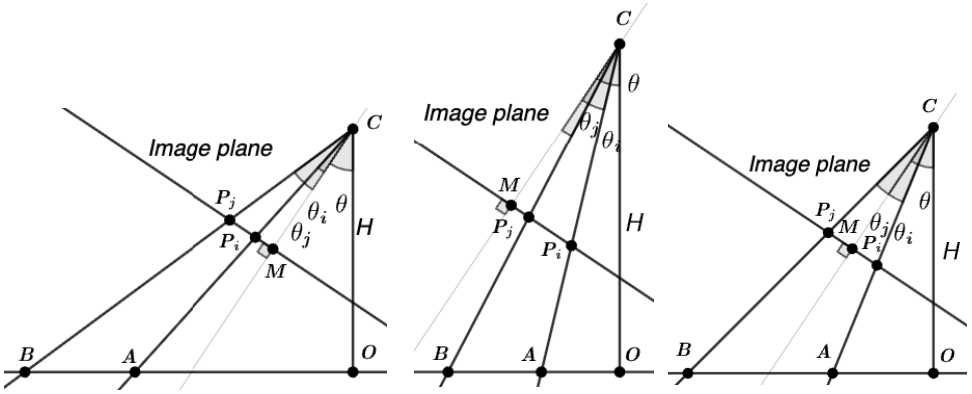


Fig. 6: Trapeze height projection

Segment AB would be the height of the projected trapeze on the real-world plane. Having all described input data the height of the trapeze for cases *a*, *b*, and *c* could be calculated by the next formulas:

$$h = H(\mathrm{tg}(\Theta + \Theta_j) - \mathrm{tg}(\Theta + \Theta_i)) \quad (2)$$

$$h = H(\mathrm{tg}(\Theta - \Theta_j) - \mathrm{tg}(\Theta - \Theta_i)) \quad (3)$$

$$h = H(\mathrm{tg}(\Theta + \Theta_j) - \mathrm{tg}(\Theta - \Theta_i)) \quad (4)$$

$$\Theta_i = \mathrm{arctg}\left(\frac{|P_{iy} - M_y|}{f}\right) = \mathrm{arctg}\left(\frac{\Delta P_i}{f}\right), \quad (5)$$

where ΔP_i — deviation of point P_i from the center of the image plane along the *y*-axis, f — focal length of the sensor.

Knowing the matrix dimensions, we can calculate the metric size of the pixel:

$$k = \frac{Res_{length}}{Matrix_{length}}. \tag{6}$$

We can measure the deviation $\Delta P_{i,j}$ in pixels. By multiplying the vector's size in pixels by the size of the pixel k we can calculate the metric size of the vector $\Delta P_{i,j}$.

The larger base of the trapeze

A larger base of a trapeze will always be the projection of the upper side of the rectangle on the image plane. In Fig. 4 A_1B_1 is the top side of the rectangle on the image plane; $a = AB$ — the larger side of the trapeze; $CO = H$ — the height of the camera position, $B_1O_1 \perp CO$ and $A_1O_1 \perp CO$.

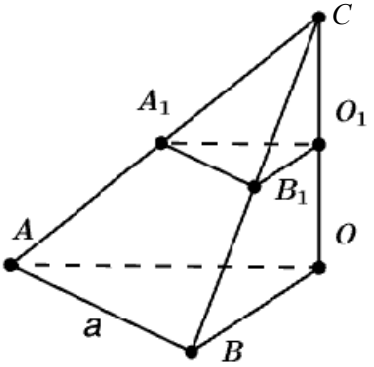


Fig. 7. Larger base projection

Since the triangles ΔAOB and $\Delta A_1O_1B_1$ on the Fig. 7 are similar:

$$a = \frac{CO\,A_1B_1}{CO_1} = \frac{H\,|\,P_{i_x} - P_{j_x}\,|k}{h} \tag{7}$$

Depending on the position of the point on the image plane regarding the center of the plane there are 2 possible cases: when the point is above or below the center of the image — Fig. 8.

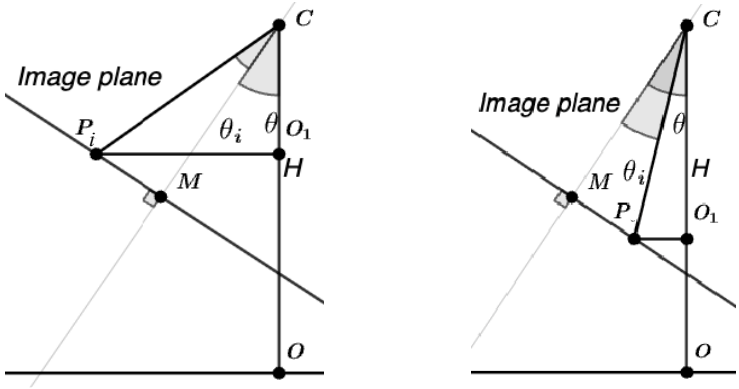


Fig. 8. Point position on the image plane

According to Fig. 8, h could be found using the following equation:

$$h = CP_i \cos(\Theta \pm \Theta_i) = \frac{f \cos\left(\Theta \pm \operatorname{arctg}\left(\frac{\Delta P_i}{f}\right)\right)}{\cos\left(\operatorname{arctg}\left(\frac{\Delta P_i}{f}\right)\right)} \tag{8}$$

Equations (7) and (8) give us the total equation for calculating the larger base of the trapeze:

$$a = \frac{H\,|P_{i_x} - P_{j_x}|\,k\cos\left(\operatorname{arctg}\left(\frac{\Delta P_i}{f}\right)\right)}{f\cos\left(\Theta \pm \operatorname{arctg}\left(\frac{\Delta P_i}{f}\right)\right)}, \tag{9}$$

α — the angle near the small base of the trapeze

Fig. 9 shows the same projection as Fig. 3 but with required additions.

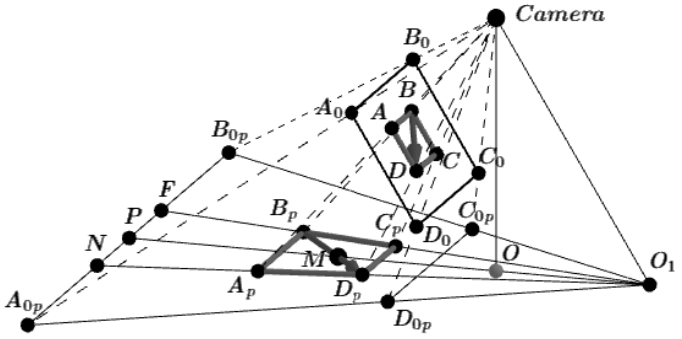


Fig. 9. Extended projection

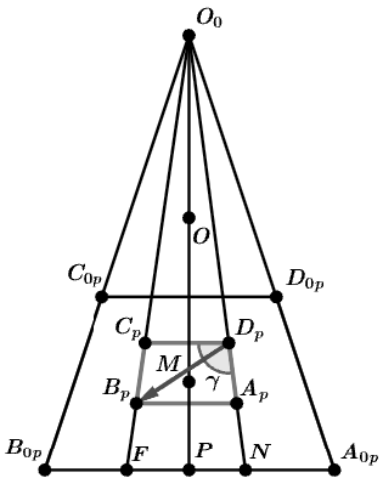


Fig. 10. Projection plane

In this section, we will consider triangle $\Delta A_{0p}B_{0p}O_1$ from Fig. 9 in more detail — Fig. 10.

$\overrightarrow{QS}$ — projection of some vector on the image plane. We aim to find γ — the angle near a small base of the trapeze.

$$\alpha = \pi - \angle O_1NP = \operatorname{arctg}\left(\frac{O_1P}{NP}\right), \tag{10}$$

NP could be calculated using (9), O_1P consists of three parts

$$O_1P = PM + MO + OO_1, \tag{11}$$

PM could be calculated using (2)

$$MO = H\,\mathrm{tg}(\Theta), \tag{12}$$

$$OO_1 = H\,\mathrm{tg}\left(\frac{\pi}{2} - \Theta\right) = H\,\mathrm{ctg}(\Theta). \tag{13}$$

Considering equations above:

$$\alpha = \pi - \operatorname{arctg}\left(\frac{H\,(\mathrm{tg}(\Theta + \Theta_{\max}) + \mathrm{ctg}(\Theta\Theta)\cos(\Theta + \Theta_{\max})}{H|N_x - P_x|\,k\cos(\Theta)}\right), \tag{14}$$

where $\Theta_{\max}$ — maximum vertical deflection angle or half of the vertical viewing angle of the camera.

All required components for (1) could be found using the equations provided above. The required input parameters are:

H — the height of the camera position; f — the focal length of the camera; Θ — the tilt angle of the camera; β — the vertical viewing angle of the camera.

ALGORITHM IMPLEMENTATION

Detection and tracking

The first step of the implementation of pedestrians' speed estimation is pedestrians' identification and tracking. The YOLOv7 object detection model variant by Rizwan Munawar [19] is used in the proposed implementation.

Pedestrians are detected on each frame of the video and data is saved to the MySQL database for further processing. After the video file is processed, the speed estimation step can be executed. It updates the database with speed data, calculated using the provided algorithm, foreach $N's$ frame. As a result, we have data that could be displayed on the video or used for further research, for instance, for estimating the pedestrians' average speed.

The algorithm requires two input parameters, H and Θ, that can be measured manually and the measurement error can significantly impact the performance. To improve the accuracy, a calibration step is implemented that can be executed between detection and speed estimation steps. It requires an array of vectors on the image plane and the distance of their projections on the real-world plane. On the calibration step parameters are tuned to decrease the root-mean-square deviation of calculated results of provided vectors and their real distance.

Final algorithm

At the estimation step, we have all the required data:

- Coordinates of pedestrians at every frame;
- Camera parameters:
  – height of the camera position — h;
  – the tilt angle of the camera — Θ;
  – the focal length of the camera in millimeters;
  – camera resolution;
  – sensor size
  – vertical viewing angle of the camera — β.

The estimation algorithm calculates the speed between every N frames, where N could be set arbitrarily. Knowing the FPS of the video the time between frames could be calculated and the distance traveled by the pedestrian in N frames could be calculated using equation (1), substituting parameters from equations (2–4), (7), (14). Dividing the founded distance by time between N frames gives us speed. Operation is repeated for every N frames till the end of the video for each tracked pedestrian on the video and the results are stored in the database.

Results and model problems testing

Testing the algorithm performance on real videos is a problem since the actual speed of pedestrians is not known. Testing videos were recorded with known distances and times to evaluate the algorithm's performance. The results of the estimation and real manually measured values are presented in Table.

Test video results

Test #	Real average speed	Estimated average speed	Error
1	1.17 m/s	1.25 m/s	6.8%
2	1.1 m/s	1.09 m/s	1%
3	1.19 m/s	1.22 m/s	2.5%
4a	0.99 m/s	0.99 m/s	0%
4b	0.7 m/s	0.77 m/s	11%

In Fig. 11 the estimation process is shown. As we can see, due to the changing size of the detected object, the tracked trajectory for a certain perspective acquires a stepped form. It might increase the error, however, as we can see from the results, this error is compensated through the entire path.

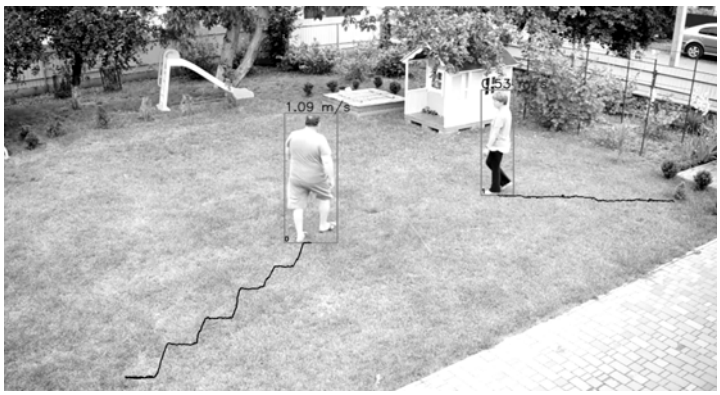


Fig. 11: Estimation process

As mentioned above, the proposed algorithm requires a set of parameters of the processed video to be set. Without them, it will work but the results might differ from real speeds significantly. An example of such a case is shown on a video from the Internet where the parameters of the camera are unknown, which is shown in Fig. 12.

Fig. 12. Speed detection with unknown parameters

Also, for videos with known parameters, we can expect realistic estimations only for pedestrians moving on one plane, which is used for measuring the height

of the camera position. The results for other detected pedestrians will be wrong. The detected pedestrian on the balcony at the left top corner of Fig. 12 can be used as an example of such a case. He is standing above the plain where other people walking, so even having the correct parameters of the camera won't allow us to estimate his speed.

CONCLUSION

In this paper, the method for pedestrian speed estimation based on the video from a surveillance camera, knowing camera parameters is proposed. The proposed method estimates distance on the real-world plane by visible movement vectors on the image plane. In the proposed implementation, the existing model for pedestrian detection and tracking is used. Unlike the previous methods, the proposed method doesn't have an issue with different heights of the tracked objects, since it uses points on the surface for tracking and estimation.

To evaluate the algorithm's accuracy, it was run on testing videos with known pedestrian speed. The resulting relative error is 0–11%, while in absolute terms, it ranges from 0 to 0.08 m/s.

Next steps. Knowing the parameters of the surveillance camera and having access to its video proposed algorithm could be used to retrieve data for pedestrians' behavior analysis. This information could be used during crowd modeling in decision support systems. Also, by improving performance or running on more powerful hardware proposed algorithm may work in a real time. In this case, it could be used for detecting unusual behavior in the crowd.

The current implementation can be found on GitHub [20].

REFERENCES

1. S.W. Ibrahim, "A comprehensive review on intelligent surveillance systems," *CST*, vol. 1, no. 1, pp 7–14, May 2016.
2. O.V. Aharkov, "Object detection and tracking by motion analysis," *Shtuchnyi intelekt*, no. 1-2, pp. 28–35, 2015.
3. O.V. Bahatskyi, "The instrumental complex for research algorithms for object boundaries," *Computers means, networks and systems*, no. 15, pp. 74–78, 2016.
4. P. Bischoff, *Surveillance camera statistics: which cities have the most CCTV cameras?* Accessed on: May 23, 2023. [Online]. Available: https://www.comparitech. com/vpn-privacy/the-worlds-most-surveilled-cities/
5. A. Moody, "Speed Camera Hotspots," *GoShorty*. Accessed on: Jan 17, 2022. [Online]. Available: https://goshorty.co.uk/blog/speed-camera-hotspots/
6. R.D. Sharma, S.K. Gupta, "A Survey on Moving Object Detection and Tracking Based On Background Subtraction," *Oxford Journal of Intelligent Decision and Data Science*, vol. 2018, pp. 55–62, 2018. doi: https://doi.org/10.5899/2018/ojids-00041
7. P. Spagnolo, T. Orazio, M. Leo, and A. Distante, "Moving object segmentation by background subtraction and temporal analysis," *Image and Vision Computing*, vol. 24, no. 5, pp. 411–423, 2006. doi: https://doi.org/10.1016/j.imavis.2006.01.001
8. A. Hardas, B.M. Dattatray, and M. Vibha, "Moving Object Detection using Background Subtraction Shadow Removal and Post Processing," *International Journal of Computer Applications, ICCT 2015*, pp. 1–5, 2015.
9. P. Kalaivani, S. Vimala, "Human action recognition using background subtraction method," *International Research Journal of Engineering and Technology*, vol. 2, pp. 1032–1035, 2015.
10. M.A. Shvandt, V.V. Moroz, "Overview of the detection and tracking methods of the lab animals," *System Research and Information Technologies*, no. 1, pp. 124–148, Apr 2022. doi: https://doi.org/10.20535/SRIT.2308-8893.2022.1.10

11. D. Khan, “Estimating Speeds and Directions of Pedestrians in Real-Time Videos: A solution to Road-Safety Problem” in *AgeingAI 2013 The Challenge of Ageing Society: Technological Roles and Opportunities for Artificial Intelligence, Turin, Italy, 201*3.
12. F. Zhao, J. Li, “Pedestrian Motion Tracking and Crowd Abnormal Behavior Detection Based on Intelligent Video Surveillance,” *Journal of Networks*, no. 10, pp. 2598–2605, 2014. doi: https://doi.org/10.4304/jnw.9.10.2598-2605
13. B.Y. Wang, A. Bochkovskiy, and H.Y.M. Liao, “YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors,” in *CVPR*, New Orleans, LA, USA, 2022, pp. 7464–7475. doi: https://doi.org/10.48550/arXiv.2207.02696
14. K. Teknomo, “Microscopic Pedestrian Flow Characteristics: Development of an Image Processing Data Collection and Simulation Model,” Ph.D. dissertation, Department of Human Social Information Sciences, Graduate School of Information Sciences, Tohoku University, Japan, 2002.
15. A. Tordeux, M. Chraibi, A. Seyfried, and A. Schadschneider, “Prediction of Pedestrian Speed with Artificial Neural Networks,” in *Proc. Int. Conf. Traffic Granular Flow*, pp. 327–335, 2017. doi: https://doi.org/10.1007/978-3-030-11440-4_36
16. G. Guo et al., “A Pose Awareness Solution for Estimating Pedestrian Walking Speed,” *Remote Sensing*, vol. 11, no. 1, pp. 55–73, 2019. doi: https://doi.org/10.3390/rs11010055
17. B. Huang, G. Mao, Y. Qin, and Y. Wei, “Pedestrian flow estimation through passive WiFi sensing,” *IEEE Trans. on Mobile Computing*, vol. 20, no. 4, pp. 1529–1542, 2021. doi: https://doi.org/10.1109/TMC. 2019.2959610
18. G.G. Lee, K.H. Ka, and W.Y. Kim, “Estimation of pedestrian flow speed in surveillance videos,” in *Conf. Korean Society of Broadcast Engineers Conference, Seoul, Republic of Korea, 2009*, pp. 330–333.
19. R. Munawar, *YOLOv7 object tracking*. 2023. [Online]. Available: https://github.com/RizwanMunawar/yolov7-object-tracking
20. M. Tishkov, *Pedestrian speed estimation*. 2024. [Online]. Available: https://github.com/MaksymTishkov/pedestrian_speed_estimation

Received 26.04.2024

INFORMATION ON THE ARTICLE

Oksana L. Tymoshchuk, ORCID: 0000-0003-1863-3095, Educational and Research Institute for Applied System Analysis of the National Technical University of Ukraine “Igor Sikorsky Kyiv Polytechnic Institute”, Ukraine, e-mail: oxana.tim@gmail.com

Maksym O. Tishkov, Educational and Research Institute for Applied System Analysis of the National Technical University of Ukraine “Igor Sikorsky Kyiv Polytechnic Institute”, Ukraine, e-mail: maksym.tishkov@gmail.com

Victor G. Bondarenko, ORCID: 0000-0003-1663-4799, Educational and Research Institute for Applied System Analysis of the National Technical University of Ukraine “Igor Sikorsky Kyiv Polytechnic Institute”, Ukraine, e-mail: bondarenvg@gmail.com

МОНІТОРИНГ НАВІГАЦІЇ НАТОВПУ ПІД ЧАС НАДЗВИЧАЙНИХ СИТУАЦІЙ / О.Л. Тимощук, М.О. Тішков, В.Г. Бондаренко

Анотація. Розглянуто задачу моніторингу навігації натовпу, що може здійснюватися за допомогою різних сенсорів і технологій, причому найчастіше використовуються камери спостереження. Ці камери забезпечують відеопотік, який зазвичай не містить додаткової інформації. Отримання додаткових даних з цих відеопотоків може значно покращити моделювання поведінки пішоходів та автоматизацію процесу моніторингу. Критичним параметром в аналізі руху пішоходів є їх швидкість. Запропоновано аналітичний метод та алгоритм оцінювання швидкості пішоходів на основі відео з камер спостереження. Першим кроком запропонованого алгоритму є розпізнавання об’єктів та їхній трекінг між феймами відео. Наступний крок — оцінювання швидкості руху об’єктів, що базується на розрахунку реальних відстаней з відомими параметрами камери та відстанями в пікселах на результуючому зображенні. Додатково пропонується алгоритм калібрування для вирівнювання параметрів камери з метою забезпечення найточніших результатів. Реалізацію алгоритму протестовано на реальних відео, похибка — близько 0,04 м/с.

Ключові слова: комп’ютерний зір, трекінг об’єктів, швидкість руху об’єктів, відеоспостереження.

МАТЕМАТИЧНІ МЕТОДИ, МОДЕЛІ,
ПРОБЛЕМИ І ТЕХНОЛОГІЇ ДОСЛІДЖЕННЯ
СКЛАДНИХ СИСТЕМ

UDC 519.004.942
DOI: 10.20535/SRIT.2308-8893.2024.4.04

QUANTUM MECHANICS APPROXIMATION APPROACH TO INVESTIGATE MOLECULAR BEHAVIOR IN NITROGEN BINDING TO ENZYMES AND PROTEINS: IMPLICATIONS FOR BIOFUEL PRODUCTION

YOSHIO MATSUKI, PETRO BIDYUK

Abstract. This research delves into the essential mechanisms underlying the binding of Nitrogen (N) atoms to enzyme molecules and their implications for protein formation in food crops and biogas production. Nitrogen (N), along with Phosphorus (P) and Potassium (K), plays a pivotal role in soil fertility and crop growth. The study explores the interactions between atoms through various mechanisms, such as catalysts, photosynthesis, and adiabatic reactions, to comprehend their roles in facilitating organic molecule formation. Additionally, the research examines the influence of enzymes on amino acids and their contributions to protein structure. The simulation process employs the Hamiltonian equation to quantify energy intensities and explore the effectiveness of adiabatic reactions in organic transformations. By investigating the molecular interactions in enzyme-catalyzed processes, this research aims to enhance protein formation in crops and optimize biogas production.

Keywords: nitrogen binding, enzyme molecules, protein formation, adiabatic reactions, biogas production, organic molecule formation.

INTRODUCTION AND BACKGROUND

Phosphorus (P), Potassium (K), and Nitrogen (N) are key elements in soil for growing food crops. Besides the other nutrients like calcium, magnesium, and sulfur, they form the foundation of soil and crop cultivation. Among them, Nitrogen is a fundamental building block for amino acids, proteins, and chlorophyll, which are necessary for plant growth and photosynthesis, and it is a key component of DNA, RNA, and other essential plant molecules [1].

Many organic reactions require an input of energy to overcome activation barriers and facilitate the conversion of inorganic molecules into organic compounds. This energy can be provided through various ways, including heating, irradiation with light, or the presence of catalysts.

Proton capture is an adiabatic reaction where a proton (H^+) is incorporated into a molecule, leading to the formation of a new compound. In the process of organic molecule formation, proton capture can occur when inorganic molecules react with protons to form organic molecules. While proton capture can contribute

to the adiabatic process of organic molecule formation, it is just one of several mechanisms in the formation of organic molecule formation from inorganic precursors such as Nitrogen in inorganic form [2].

The specifics of the reactions involved in proton capture processes can vary depending on the particular inorganic molecules and the conditions under which they occur. Additionally, other factors such as the presence of catalysts or energy sources can influence the efficiency and outcomes of proton capture. It is noted that organic molecule formation is a complex and diverse field of study, and the processes involved can be influenced by numerous factors.

Photosynthesis contributes to proton capture in amino acids indirectly by producing NADPH (nicotinamide adenine dinucleotide phosphate oxidase) during the light-dependent reactions. The NADPH, in turn, supplies the necessary reducing power for the Calvin cycle, where carbon dioxide is converted into carbohydrates, including the building block for amino acid synthesis [3].

Enzymes influence proton capture in target amino acids as catalysts. Enzymes influence the formation of protein structure, but not by consuming itself. The interaction happens through their electrostatic field made by the electrostatic potentials of the atoms held by the enzyme and the targeted amino acids of the protein [4].

Fermentation process in biogas production from wheat and maize involves a series of physical reactions facilitated by various groups of microorganisms, leading to the generation of biogas, which is primarily composed of methane (CH_4) and carbon dioxide (CO_2) [5]. In the fermentation process of wheat/maize, the primary microorganism involved is yeast, and the key enzyme responsible for the conversion of sugars into ethanol (alcohol) and carbon dioxide is called "zymase". The typical atoms included in this enzyme are as same as in the other enzymes' atoms shown in Table 1 [6].

The formation of organic molecules from inorganic molecules can occur through various processes, including biological and non-biological pathways. While it is possible some organic reactions to occur at room temperature or under normal conditions without external heating, the generalization that all organic reactions can proceed adiabatically (without heat exchange with the surroundings) or at ambient temperature is not accurate. With this research, we will estimate the degree of the contribution to the organic molecule formation by adiabatic process and by other processes.

RESEARCH OBJECTIVES

The primary goal of this research is to investigate the fundamental mechanism of binding a Nitrogen atom to an enzyme molecule. The study aims to enhance protein formation in food crops and biogas production, focusing on Nitrogen in soil, wheat, and maize crops, and the enzymes involved in their production, namely Glutamate and Nitrate Reductase. Additionally, the research will explore the composition of biogas production during this process, primarily consisting of methane and carbon dioxide.

The main simulated physical reactions include:

1. Adiabatic Perturbation: This aims to understand the interactions between protons and targeted atoms of the amino acids, which play a crucial role in the binding process.

2. Electrostatic Perturbation: This analysis focuses on the influence of enzymes in accelerating the formation of proteins in specific targeted amino acids.

3. Photon Absorption and Electron Discharge: This part examines how photon absorption leads to electron discharge from the target atom.

To quantify the energy intensities of these physical reactions in the protein formation process involving the selected objects, the Hamiltonian equation will be utilized.

Furthermore, this research explores the effectiveness of adiabatic reactions in organic transformations. Specifically, it investigates whether certain organic reactions can occur at room temperature or under normal conditions without external heating, as this aspect is not yet clearly proven. The approximation method in quantum mechanics will be employed to assess the possibility of adiabatic reactions in comparison with photo synthesis and electrostatic energy fields produced by the catalyst (enzyme) in this context.

In conclusion, this research aims to provide essential insights into the binding of Nitrogen to enzyme molecules, which can contribute to improved protein formation in food crops and enhance biogas production.

METHODOLOGY

Selecting typical atoms in plant tissues and enzymes. A simplified system is considered, including typical atoms observed in wheat and maize as well as the atoms in enzymes in these crops, as shown in Table 1.

Calculating the probability of the reactions. Each of the adiabatic perturbation, electrostatic interaction, and the photon absorption, is calculated on each atom listed in Table 1. The algorithm of the calculation is shown in the latter section.

T a b l e 1 . Typical atoms observed in wheat and maize tissues

N	Typical atoms in tissue of the crops	Typical atoms in enzymes	Atomic number (Number of electrons)	Mass number	Empirically measured radius (re) in pico-meters [8]	Calculated diameter in pico-meters
1	Carbon (C)	Carbon (C)	6	12	70	140
2	Hydrogen (H)	Hydrogen (H)	1	1	25	50
3	Oxygen (O)	Oxygen (O)	8	16	60	120
4	Nitrogen (N)	Nitrogen (N)	7	14	65	130
5	Phosphorus (P)	Phosphorus (P)	15	31*	100	200
6	Potassium (K)	–	19	39*	220	440
7	Calcium (Ca)	–	20	40*	180	360
8	Magnesium (Mg)	–	12	24*	160	320
9	Sulfur (S)	Sulfur (S)	16	32*	100	200

* (the most common isotope, but there are others)

Comparing the result of the atomic level calculation with the molecular structure. Discussion will be made on the consistency of the calculated result with the actual molecular structure of glutamate synthetase (GS), which is essential for nitrogen uptake and assimilation from the soil. It helps convert inorganic nitrogen compounds, such as ammonium (NH_4^+) and nitrate (NO_3^-), into organic nitrogen forms like glutamate, with the following reaction [7]:

Glutamate + NH_4^+ + ATP → Glutamine + Pi.

Here, glutamate is an amino acid and serves as the precursor for glutamine synthesis. By incorporating ammonia, wheat utilizes GS to convert inorganic nitrogen (ammonium) into the organic nitrogen compound glutamine. “Pi” is an abbreviation for “inorganic phosphate”. It refers to a form of phosphorus, an essential element for life, which exists in the inorganic state in chemical compounds.

Molecular form of Glutamate. The molecular formula of glutamate is $C_5H_9NO_4$. It is an organic compound composed of carbon (C), hydrogen (N), and Oxygen (O) atoms.

ALGORITHM (HAMILTONIAN EQUATION)

Hamiltonian equation consists of 6 terms: The kinetic energy of the target proton of amino acid, the potential of elastic electron scattering, the potential of electron capture, the electrostatic energy of catalyst (enzyme) to influence the targeted amino acid, and the photon absorption to drop an electron from the target amino acid. The first term, the kinetic energy of the target proton, is set as unity, which enables calculating relative probabilities of the occurrences of those terms.

Capture of an electron by a proton (charge exchange) in adiabatic process. The algorithm to calculate the probability of electron capture was taken from [9]. A case was considered, in which a proton of an atom, for example Oxygen, captured an electron of another atom, for example Hydrogen, which passed by the proton of the atom. Fig. 1 shows the coordinates of two protons and an electron. Two protons are symmetrically located on both sides of the origin O of the coordinate. $(1/2)R$ and $-(1/2)R$ are the coordinates (geometric positions) of two protons that will capture the electron of another atom. R is the distance between two protons, and the positions of these protons are fixed. On the other hand, r is the position of electron in a plane polar coordinate system and it changes as a function of geometric coordinate x, where $-(1/2)R \le x \le (1/2)R$.

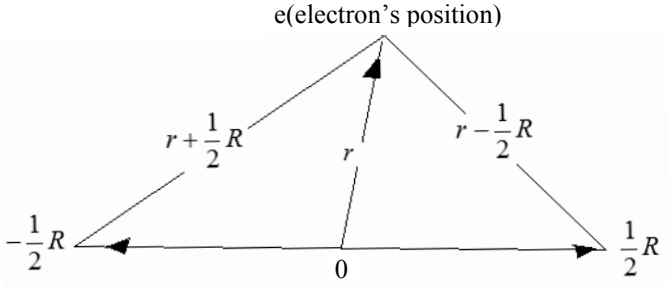


Fig. 1. The coordinates of the proton and the electron (adapted from [9] p. 89, Fig. 21)

Then it is assumed that the electron is initially attached to the proton at the coordinate of $(1/2)R$; then the initial state of the electron had the form, $\varphi = (r-(1/2)R)\cos wx$, where $\cos wx$ is the Eigen-wave function, w is the frequency of the oscillation of the electron, representing its energy level. Then, the probability of electron capture is calculated by the Hamiltonian equation shown below:

$$H = T_e - C_1\left(\frac{1}{r+(1/2)R}\right)\cos wx - C_2\left(\frac{1}{r-(1/2)R}\right)\cos wx, \qquad (1)$$

where T_e is the kinetic energy of the electron. Here, it was assumed that the relative speed of proton was much slower than the electron's speed. Therefore, the geometry of an electron and protons was the main focus, not time dependency of the system. And, T_e was set as a unity (one).

When the electron is attached to one proton at the coordinate of $+(1/2)R$ as its initial state, the wave function is $(r-(1/2)R)\cos wx$; but, it changes to

$(r+(1/2)R)\cos wx$, when the electron is transferred to another proton located at $-(1/2)R$, as the process of the charge exchange.

The influence of enzyme to the amino acid by the electrostatic energy [10; 11] is described by the following equation:

$$E_{elec} = \sum_{i,j} \frac{q_i q_j}{4\pi\pi_0 \cdot r_{ij}} \approx \frac{q_i q_j}{r_{ij}},$$

where E_{elec} is the electrostatic energy; q_i and q_j are the charges of the atoms; r_{ij} is the distance between the charges; and ε_0 is the vacuum permittivity.

The interaction with photon (from the interaction of a particle with the electromagnetic field, p. 45, [9]). Under the influence of the electromagnetic field, the momentum of a particle, p, becomes $p - eA/c$, where e is an electron charge, A is a vector potential of the electromagnetic field. Therefore the term of the Hamiltonian becomes:

$$\frac{p^2}{2m} - \frac{e}{mc} pA + \frac{e^2 A^2}{2mc^2}.$$

The first term, $p^2/2m$, is T_e of (1). The second term is described by only angular coordinates, therefore we replace this term with an oscillation function $\cos\varphi$, where the frequency of φ is set higher than wx in order to simulate the light (photon). The third term is negligibly small.

By including the terms of kinetic energy, the elastic scattering, the electron capture, the electrostatic energy of the enzyme, and the photon absorption effect, the Hamiltonian becomes as the follow:

$$H = T_e - C_1\left(\frac{1}{r+(1/2)R}\right)\cos wx - C_2\left(\frac{1}{r-(1/2)R}\right)\cos wx -$$
$$- C_3 \frac{1}{r_{i,j}}\cos wx - C_4 \cos\varphi \cos wx .$$

When the reference [9] was published in 1969, a personal computers was not available, therefore the reference [9] further described the algorithm in mathematical forms with calculus, and predicted that the squared module of the coefficients, C_1, C_2, C_3 and C_4, gave the probability of charge exchange (the electron capture). However, in this research a personal computer was used to calculated the coefficients, C_1, C_2, C_3 and C_4 with the following algorithm of matrix algebra:

$$H = T_e - X \quad c,$$

where X is made of four vectors, $-\left(\frac{1}{r+(1/2)R}\right)\cos wx$, $-\left(\frac{1}{r-(1/2)R}\right)\cos wx$, $1/(r_{ij})\cos wx$, $\cos\varphi \cdot \cos wx$. And c is the four column vector:

$$c = \begin{bmatrix} C_1 \\ C_2 \\ C_3 \\ C_4 \end{bmatrix}.$$

Then a constraint was set

$$X'H = 0,$$

so

$$X'(T_e - Xc) = 0,$$

where X' is transpose matrix of X. Then,

$$X'Xc = X'T_e, \quad c = (X'X)^{-1}X'T_e.$$

SIMULATION PROCESS

For this simulation the values of R were assigned as the diameters of Hydrogen, Carbon, Nitrogen, Oxygen, Phosphorus, Potassium, Calcium, Magnesium, and Sulfur, as shown in Table 1, and r is given by (11), where x is the distance from the origin O toward $(-(1/2)R)$ and toward $(+(1/2)R)$ in Fig. 1, while the origin O is located at $x = 13$: and $(-(1/2)R)$ is at $x = 1$, and $(1/2)R$ is at $x = 25$.

Note: According to [9], p. 84, "Capture of an Electron by a Proton (Charge Exchange)" of the Chap. 2.4 "Adiabatic Perturbations", R is the distance between the two protons of Fig. 1. Here an assumption was made as if two same atoms, which were centered by each of two protons, were located next to each other: therefore, $R = 2 \times r$, where r is empirically measured radius of Table 1.

In our simulation for the adiabatic process, a symmetric geometry of two atoms was assumed as the mirror images on the both-sides of the origin O, as shown in Fig. 2. We assign the value of R by the empirically measured radius, 25 pico-meters, for example of hydrogen-atom [10]. Because two hydrogen-atoms are placed next to each other in Fig. 2, we assign 50 to the value of R. If charge exchange happens, the electron's plane polar coordinate, r, changes its position from the initial position, $+R/2$ of x-coordinates, to the position of the charge exchange, $-(1/2)R$. Then the relation between r and R is:

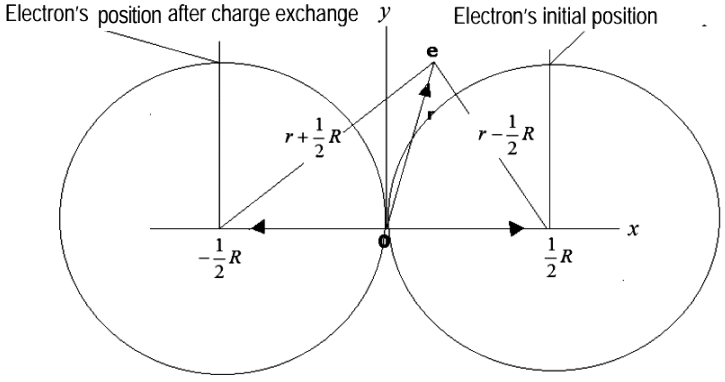
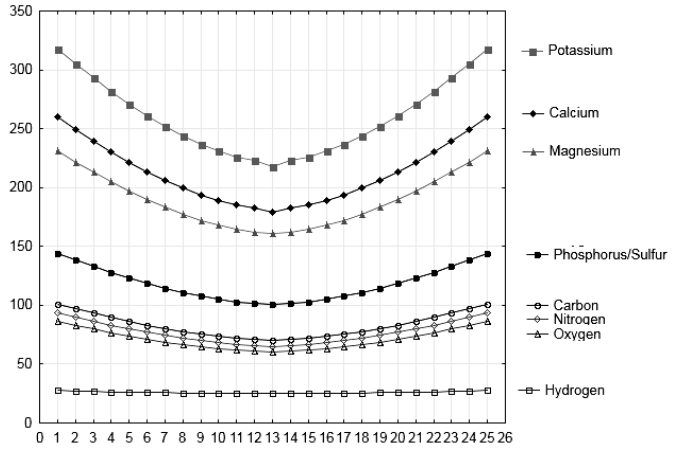


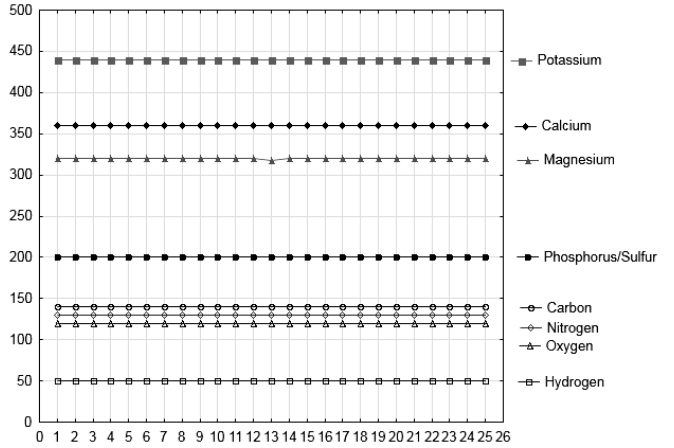
Fig.2. Position of the electron and its coordinate r

$$r = \sqrt{\left(\frac{R}{2}\right) + x^2}, \tag{2}$$

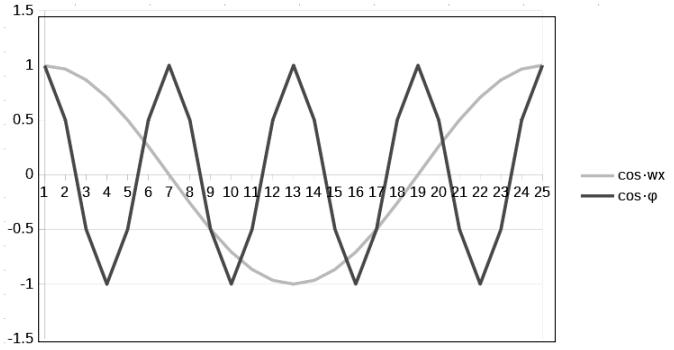
where x is the distance from the origin O toward $-(1/2)R$ and toward $+(1/2)R$, and the origin O is at 0 on the x-axis, $-(1/2)R$ is at –12 on the x-axis; and, $+(1/2)R$ is at +12 on the x-axis. The electron is initially attached to the proton at $+(1/2)R$ of x-coordinates; and then it will be attached to the proton at $-(1/2)R$ of x-coor di-nates after the charge exchange.



(1) r: distance of an electron from the origin



(2) R: distance between two protons



(3) cos·wx and cos·φ

Fig. 3 Input data for the simulation: R, r, cos·wx and cos·φ

When the nucleons are far apart, the electron will be localized near one or the other proton. However, it doesn't mean that R in this simulation should be far apart to infinity, but it only justifies the wave functions of hydrogen-atom that distinguish the initial state of the wave function $(r-(1/2)R)\cos wx$ and the wave function $(r+(1/2)R)\cos wx$ after the charge exchange.

Then we set cosine curves as the wave functions $\cos wx$ for the Hamiltonian equation (4). Also, we set $\cos\varphi$ to model the photon's wave function, where frequency of $\cos\varphi$ is higher than of $\cos wx$ as shown in Fig. 3.

Input data for the numeric simulation. Then the input data were made on R, r, $\cos wx$ and $\sin wx$ as shown in Fig. 3. The case of $\sin wx$ was also calculated during this research, but it was eliminated from this report due to the less significance of the calculated standard error of the coefficient.

RESULTS AND DISCUSSION

Fig. 4 and Fig. 5 illustrate the outcomes of the calculations that uncover the probabilities of energy transitions between atoms through distinct mechanisms: elastic scattering, charge exchange, catalysts, and photosynthesis. We will focus on the latter three mechanisms, as elastic scattering involving protons is found to be negligible.

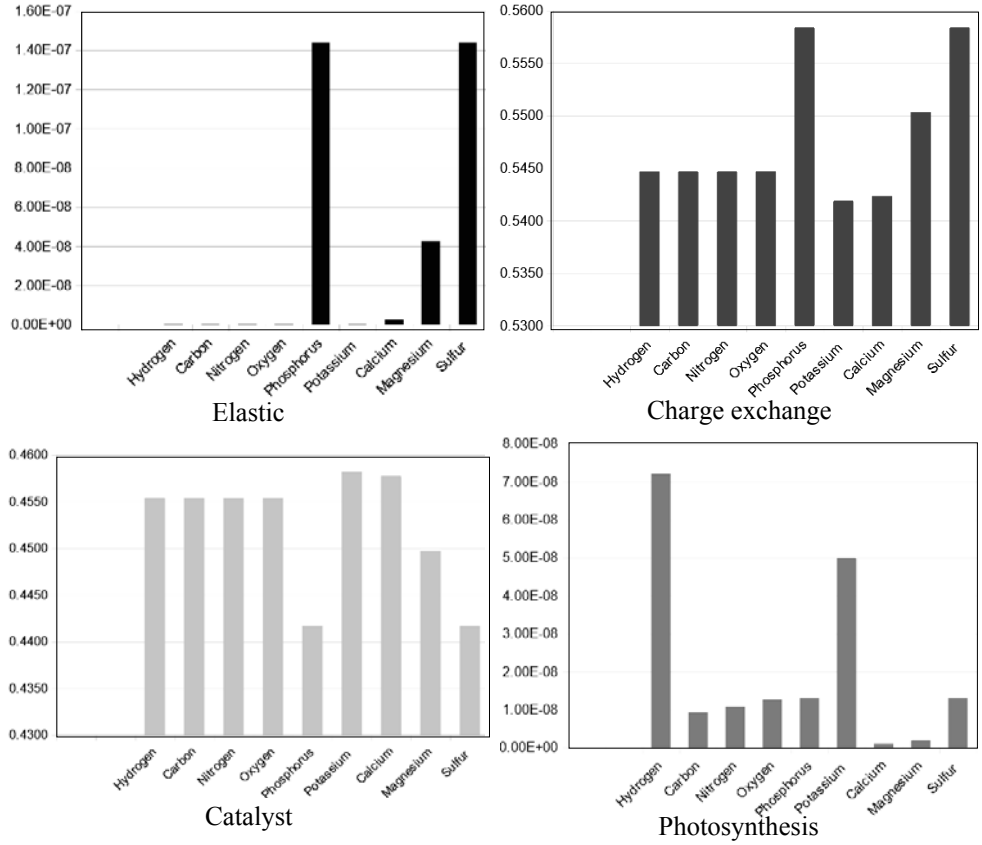


Fig. 4. Probabilities of Elastic Scattering, Charge Exchange, Catalyst Perturbation, and Photosynthesis

Early Stage of Wheat/Maize Growth. The insights from Fig. 4 are interesting. They suggest that Nitrogen (N), Hydrogen (H), and Oxygen (O) atoms have a unique connection with enzymes, driven by electrostatic energy. This observation holds significance during the initial stage of wheat and maize growth. At this juncture, enzymes like glutamate synthetase (GS), a pivotal role in converting inorganic compounds such as ammonium (NH_4^+) and nitrate (NO_3^-), into organic compounds like glutamate ($C_5H_9NO_4$). These compounds lay the foundation for the plant's structural development.

Latter Stage of Wheat/Maize Growth by Photosynthesis. Advancing with Fig. 4, we find Nitrogen (N) taking a central role in the adiabatic charge exchange process, shaping the plant's main body. Enzymes also contribute to this process, collaborating with Nitrogen (N). Photosynthesis, however, appears to have a gentler effect on Nitrogen (N), but exerts a more prominent influence on non-enzyme atoms like Potassium (K). This balance reflects the rhythms of nature: photosynthesis becomes significant after the plant has established its protein structure, while the enzyme-driven charge exchange process takes precedence in earlier stages.

Fermentation Process for Biogas Production. Moving beyond growth stages to the biogas production, Fig. 4 extends its narrative. It emphasizes the significant roles of adiabatic charge exchange and enzyme activity in transforming substances into methane (CH_4) and carbon dioxide (CO_2). Interestingly, photosynthesis, which often occupies a prominent position in energy discussions, appears to exert a subtler influence in the context of fermentation. These insights deepen our understanding of how plants harness and covert energy.

Common Patterns. Revealing broader patterns, this study consistently emphasizes enzyme atoms as key players in the adiabatic charge exchange process. This observation waves a coherent thread through the narrative: the orchestrated absorption of Nitrogen (N) from the soil, contributing to the formation of plant structures. Additionally, the impact extends beyond enzymes. Catalysts, which are enzymes, also affect non-enzyme atoms like Potassium (K), Calcium (Ca), and Magnesium (Mg), showing nature's synchronized efforts to assimilate Nitrogen (N) into plant bodies. For example, the calculated probabilities of the adiabatic perturbation and the electrostatic perturbation on the atoms of Nitrogen (N), Hydrogen (H), and Oxygen (O) inspire possibilities of applying this result for further making other enzymes that may enhance the nitrogen assimilation process.

T a b l e 2 . Calculated results and the facts (discussions)

Stage of growth	Calculated probabilities in Fig. 4	Facts
The early stage of wheat/ maize growth	Each of Nitrogen (N), Hydrogen (H), and Oxygen (O) have the high probability of catalyst perturbation. It means the enzyme works well with these atoms, of the electrostatic energy.	During the early stage of wheat/maize growth, one of the enzymes, glutamate synthetase (GS), helps convert inorganic compounds, such as ammonium (NH_4^+) and nitrate (NO_3^-), into organic nitrogen forms like glutamate ($C_5H_9NO_4$), which will be the main body of the plant.
Latter stage of wheat/ maize growth by photo synthesis	1. Nitrogen (N) has the high probabilities of the adiabatic charge exchange and the enzyme-catalyst reactions. 2. However, the probability of photosynthesis is negligible. 3. On the other hand, one of the non-enzyme atoms (in the plant body protein), Potassium (K), has the high probability of photosynthesis.	1. The main body of the plant is formed with the help of enzymes. 2. The photosynthesis should be effective only after the plant forms its protein structure. [Biol Res 43: 99-111, 2010 BR] 3. This result is also consistent with the fact that the photosynthesis should be effective in forming the plant's protein structure, while the enzyme is to enhance the photosynthesis of the plant proteins.
Fermentation	The probability of the catalyst perturbation is high on Carbon (C), Hydrogen (H), and Oxygen (O), which form methane (CH_3) and carbon dioxide (CO_2). The probability of adiabatic charge exchange is also large on these atoms	Fermentation process in biogas production from wheat and maize involves a series of physical reactions facilitated by various groups of microorganisms, leading to the generation of biogas, which is primarily composed of methane (CH_4) and carbon dioxide (CO_2).
Conclusion	The calculated results suggest that Nitrogen is notably influenced by adiabatic charge exchange and the enzyme-catalyzed reactions.	"Nitrogen fixation" is facilitated by enzymes and proteins through an adiabatic process without the need for heating, accompanied by photon absorption, which enhances the formation of robust protein structures.

CONCLUSION AND RECOMMENDATION

In light of the comprehensive investigation into the mechanisms governing the binding of Nitrogen (N) atoms to enzyme molecules and their implications for protein formation in food crops and biogas production, several key insights emerge.

Enhancing Protein Formation in Food Crops: The study's findings shed light on the intricate interactions occurring during the early and latter stages of wheat and maize growth. The high probabilities of catalyst perturbation, as indicated by the analysis, suggest that enzymes effectively collaborate with Nitrogen (N), Hydrogen (H), and Oxygen (O) atoms. Particularly during the initial growth stages, enzymes such as glutamate synthetase (GS) facilitate the conversion of inorganic compounds into organic nitrogen forms like glutamate, which contributes to the formation of the plant's structural components. These insights not only deepen our understanding of protein formation but also hold potential for optimizing crop growth strategies.

Biogas Production Optimization: For the fermentation process aimed at biogas production, the observation highlights the important role of enzyme atoms in both the adiabatic charge exchange process and the catalytic effects of enzymes in promoting methane (CH_4) and carbon dioxide (CO_2) production. These factors are crucial contributors to efficient biogas generation. Importantly, the limited influence of photosynthesis on the fermentation process suggests that the latter stages of plant growth are more relevant to biogas production. This understanding could aid in refining biogas production processes, potentially leading to increased energy yield from agricultural products.

Adiabatic Charge Exchange and Enzyme-Catalyst Dynamics: The research consistently highlights the role of the adiabatic charge exchange process, particularly concerning Nitrogen (N) assimilation into plant structures. This corresponds to the process of constructing the fundamental elements of plants by absorbing Nitrogen (N) from the soil. Additionally, the interplay between catalysts (enzymes) and non-enzyme elements such as Potassium (K), Calcium (Ca), and Magnesium (Mg) highlights how enzymes play a significant role in aiding the incorporation of Nitrogen (N) into plant structures, aligning with anticipated outcomes.

Ethical Consideration and Future Implications: Beyond the calculated results, these findings encourage us to ponder ethical dimensions. The concept of genetically enhancing enzymes for improved biogas production emerges as a guiding principle, as indicated by the Quantum Mechanics Approximation Approach. Unlike the heated debates surrounding genetic engineering in food crops this pathway appears to encounter fewer ethical challenges. The fusion of scientific understanding and ethical contemplation points to a fresh direction in optimizing biogas production.

Recommendation: As each calculated probability unveils the narrative of atomic interactions, the study's conclusions prompt us to take actionable steps. Guided by these insights, it is essential for the scientific community to initiate practical applications. Validating these findings through experiments in actual agricultural and biogas production settings could pave the way for transformative breakthroughs. Collaboration among mathematicians, molecular biologists, agricultural experts, and careful environmental management opens avenues to a future where mathematical analysis, innovation, and ethical considerations converge to address urgent needs in sustainable agriculture and renewable energy. In concluding this mathematical exploration, the symphony of intricate atomic interactions resonates. Beyond mathematical harmonic, we catch a glimpse of a world where ethical considerations intertwine harmoniously with exploration, guiding us toward a future enriched by mathematical inquiry, innovation and ethical guidance.

REFERENCES

1. S.J. Parikh, B.R. James, "Soil: The Foundation of Agriculture," *Nature Education Knowledge*, 3(10):2, 2012. Available: https://www.nature.com/scitable/knowledge/ library/soil-the-foundation-of-agriculture-84224268/
2. Agostino Migliore, Nicholas F. Polizzi, Michael J. Therien, and David N. Beratan, "Biochemistry and Theory of Proton-Coupled Electron Transfer," *Chem. Rev.*, 2014, 114, pp. 3381−3465. Available: https://pubs.acs.org/doi/pdf/10.1021/cr4006654
3. B. Alberts et al., *Molecular Biology of the Cell*; 4th edition. New York: Garland Science, 2002. Available: https://www.ncbi.nlm.nih.gov/books/NBK26819/
4. António J.M. Ribeiro, Jonathan D. Tyzack, Neera Borkakoti, Gemma L. Holliday, and Janet M. Thornton, "A global analysis of function and conservation of catalytic residues in enzymes," *J. Biol. Chem.*, 295(2), pp. 314–324, 2019. doi: 10.1074/jbc.REV119.006289
5. P. Weiland, "Biogas production: current state and perspectives," *Appl. Microbiol. Biotechnol.*, 85, pp. 849–860, 2010. doi: 10.1007/s00253-009-2246-7
6. D.C. Phillips, "The Three-dimensional Structure of an Enzyme Molecule," *Scientific American*, November 1966. Available: https://web.ics.purdue.edu/~gchopra/class/ public/readings/Introduction_Lecture1/Phillips_SCIAM_66_Lysozyme_ structure.pdf
7. M. Spodenkiewicz, C. Diez-Fernandez, V. Rüfenacht, C. Gemperle-Britschgi, and J. Häberle, "Minireview on Glutamine Synthetase Deficiency, an Ultra-Rare Inborn Error of Amino Acid Biosynthesis," *Biology* 5(4), 40, 2016. doi:10.3390/biology5040040
8. J.C. Slater, "Atomic Radii in Crystals," *J. Chem. Phys.*, 41, 3199, 1964. doi: https://doi.org/10.1063/1.1725697
9. A.B. Migdal, V. Krainov, *Approximation Methods in Quantum Mechanics*; translated by Anthony J. Leggett, New York: W.A. Benjamin Inc., 1969, 146 p.
10. Vinicius de Godoi Contessoto et al., "Electrostatic interaction optimization improves catalytic rates and thermotolerance on xylanases," *Biophys. J.*, 120(11), pp. 2172–2180, 2021. doi: 10.1016/j.bpj.2021.03.036
11. R.P. Feynman, R.B. Leighton, and M. Sands, *Thy Feynman Lectures on Physics*; Vol. II Ch. 4: Electrostatic Energy. New Millennium edition, 2011, 1552 p. Available: https://www.feynmanlectures.caltech.edu/II_08.html

Received 23.08.2023

INFORMATION ON THE ARTICLE

Yoshio Matsuki, ORCID: 0000-0002-5917-8263, World Data Center for Geoinformatics and Sustainable Development of the National Technical University of Ukraine "Igor Sikorsky Kyiv Polytechnic Institute", Ukraine, e-mail: matsuki@wdc.org.ua

Petro I. Bidyuk, ORCID: 0000-0002-7421-3565, Educational and Research Institute for Applied System Analysis of the National Technical University of Ukraine "Igor Sikorsky Kyiv Polytechnic Institute", Ukraine, e-mail: pbidyuke_00@ukr.net

АПРОКСИМАЦІЙНИЙ ПІДХІД КВАНТОВОЇ МЕХАНІКИ ДЛЯ ДОСЛІДЖЕННЯ МОЛЕКУЛЯРНОЇ ПОВЕДІНКИ ЗВ'ЯЗУВАННЯ АЗОТУ З ФЕРМЕНТАМИ ТА БІЛКАМИ: ЗНАЧЕННЯ ДЛЯ ВИРОБНИЦТВА БІОПАЛИВА / Й. Мацукі, П.І. Бідюк

Анотація. Це дослідження заглиблюється у суттєві механізми зв'язування атомів азоту (N) з молекулами ферментів та їх наслідки для утворення білків у харчових культурах та виробництві біогазу. Азот, разом з фосфором (P) та калієм (K), відіграє важливу роль у родючості ґрунту та рості врожаю. Досліджено взаємодії між атомами за допомогою різних механізмів, таких як каталізатори, фотосинтез та адіабатичні реакції, для розуміння їх ролей у полегшенні утворення органічних молекул. Додатково досліджено вплив ферментів на амінокислоти та їх внесок у структуру білків. Процес симуляції використовує рівняння Гамільтона для кількісної оцінки інтенсивності енергії та дослідження ефективності адіабатичних реакцій у органічних перетвореннях. Через дослідження молекулярних взаємодій у фермент-каталізованих процесах це дослідження спрямоване на поліпшення утворення білків у врожаях та оптимізацію виробництва біогазу.

Ключові слова: зв'язування азоту, молекули ферментів, утворення білків, адіабатичні реакції, виробництво біогазу, утворення органічних молекул.

UDC 517.925
DOI: 10.20535/SRIT.2308-8893.2024.4.05

IDENTIFICATION OF NONLINEAR SYSTEMS WITH PERIODIC EXTERNAL ACTIONS (Part II)

V. GORODETSKYI

Abstract. The article presents the results of the study, which is a continuation of the author's previous research. This paper considers more complex problems in identifying nonlinear systems with periodic external actions. The article shows that the previously proposed method is applicable when the periods of external actions in the same differential equation may differ. At the same time, the ratio between the values of the periods can be both integer and fractional. The conditions under which this is possible are formulated. These conditions are based on the theorem proved in the previous work. Part of this study is devoted to the problem of identification of a chaotic system with an external non-sinusoidal action. To create such an external action, a function with three harmonic components was used. A numerical experiment confirmed the effectiveness of the algorithm in this case as well.

Keywords: identification, ordinary differential equation, external action, periodic coefficient, constant coefficient.

INTRODUCTION

As is known, non-autonomous mathematical models are widely used to describe various physical processes [1; 2]. The construction of such models can be reduced to the so-called inverse problem [3]. In this case, the model is built on the basis of information about the output of the system, that is, the problem of system identification is solved. In this case, the usual formulation of the problem assumes the presence in the system equations of additive periodic actions and information about the structure of the system [4–6]. The task becomes more complex when the structure of the system is unknown and external actions can be either additive or multiplicative. At the same time, the number of such actions may be not limited. The solution of the mentioned problem was proposed in [7]. This study is a development of the author's previous work and demonstrates additional capabilities of the method introduced in [7].

NOTATIONS AND SOME PREVIOUS RESULTS

Using the notations from [7], we consider a system of ordinary differential equations (ODE) of the form

$$\dot{x}_i = \sum_{j=0}^{m} c_{ij}(t) f_j(\mathbf{x}), \tag{1}$$

$$i = 1,...,n; \quad \mathbf{x} = \{x_1(t),...,x_n(t)\}. \quad t \in [0; t_e], \quad t_e > 0.$$

In equation (1), we consider the time functions $x_i(t)$ to be known and the coefficients $c_{ij}(t)$ to be unknown. In this case, any of the coefficients can be either constant or a continuous time function of a period T.

If in equation (1) all the coefficients $c_{ij}(t) = const$, then to find them we can apply the well-known relation:

$$\mathbf{C} = \mathbf{A}^{-1}\mathbf{B}, \tag{2}$$

where $\mathbf{C}$ is the vector of the required coefficients of equation (1), and $\mathbf{B}$ is the vector of values $\dot{x}_i(t_k)$, $k = 0,...,m$, $\mathbf{A}$ is the matrix of function $f_j(\mathbf{x}(t_k))$ values, $j = 0,...,m$. In [7], a theorem was proven according to which relation (2) can be used to calculate the coefficients of equation (1) if the moments of time t_j are subject to the relations:

$$t_1 = t_0 + \tau,\ t_2 = t_0 + 2\tau,\ ...,\ t_m = t_0 + m\tau;\ t_0 \geq 0,\ \tau > 0,\ t_m \leq t_e \tag{3}$$

and wherein $\tau = T$. Based on this theorem, the algorithm described in [7] was constructed. That is, if conditions (3) are met and $\tau = T$, then applying formula (2) for any t_0, we will obtain the exact values of all constant coefficients of the identified equation. Therefore, if $\tau = T$, then for two different t_0: t_{01} and t_{02} the relation

$$\delta_j = | c_{ij}^1 - c_{ij}^2 | \to \min, \tag{4}$$

must be satisfied, where c_{ij}^1 and c_{ij}^2 are the values of the coefficient $c_{ij}(t)$ obtained for t_{01} and t_{02}, respectively. In order to avoid errors possible for a specific value of t_0, intervals of t_{01} and t_{02} values are used in calculations. The application of the algorithm is illustrated in [7] using the example of identifying an equation with additive and multiplicative periodic actions having the same period.

GENERALIZATION OF THE PROPOSED METHOD

It is easy to show that the method can be effective for solving more complex problems. Let an equation of the form (1) have two external periodic actions with periods T_1 and T_2, respectively. Let also there exist q_1 and q_2, $q_1, q_2 \in 1,2,3,...$ and $T < t_e$, such that

$$q_1 T_1 = q_2 T_2 = T, \tag{5}$$

where T is the least common multiple of T_1 and T_2. That is, we have period T, which is common for both external actions. Therefore, the condition of the theorem from [7] is met.

Identification of equations with an integer ratio of periods of external actions

Let us consider the case when relation (4) is satisfied, and at the same time $T_1/T_2 \in 2,3,...$ or $T_2/T_1 \in 2,3,...$. As an example of using the method, consider identification of a system

$$\begin{cases} \dot{x}_1 = -x_2 - x_3, \\ \dot{x}_2 = x_1 - dx_2, \\ \dot{x}_3 = c_{30}(t) + c_{33}(t)x_3 + c_{36}(t)x_1x_3, \end{cases} \tag{6}$$

obtained on the basis of the well-known Rössler system [8]. The system has the following coefficients:

$$d = 0.15,\ c_{30}(t) = 0.5 + 0.4\sin\left(\frac{2\pi t}{T_0}\right),$$

$$c_{33}(t) = -20,\ c_{36}(t) = 5 + 2.5\sin\left(\frac{2\pi t}{T_6} - \frac{\pi}{2}\right),\ T_0 = 2s,\ T_6 = 4s.$$

It is obvious that for the external actions of the system (6), we have $T_6/T_0 = 2$. That is, condition (5) is satisfied with $T_6 = T$. Fig. 1 shows the time series of the variables of the system (6) and Fig. 2 shows its phase trajectories.

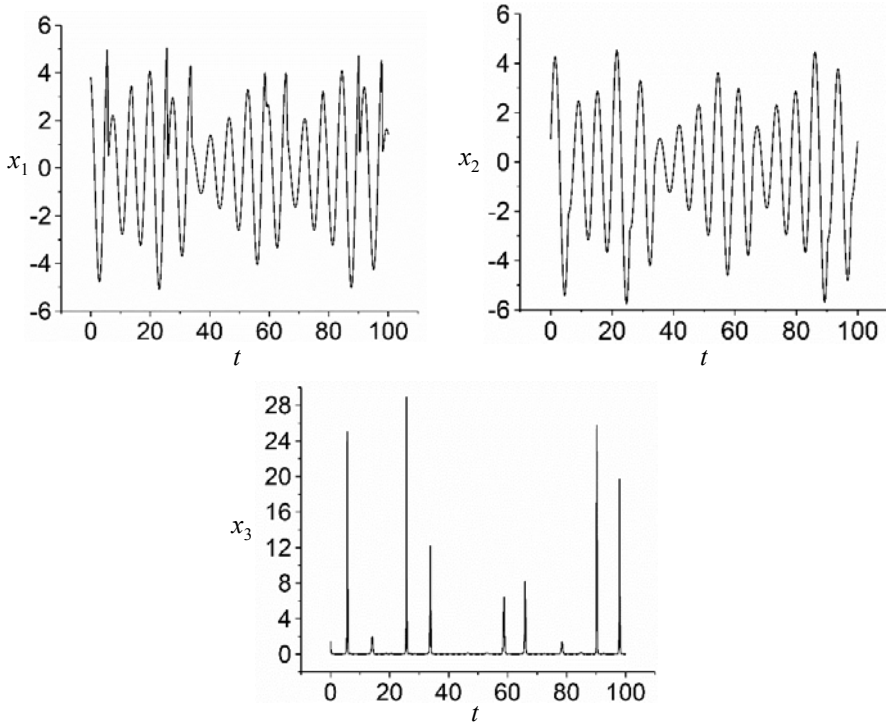


Fig. 1. Time series of system (6) variables

To identify the third equation of system (6), its general structure was chosen in the form of a polynomial of the second degree:

$$\dot{x}_3 = c_{30}(t) + c_{31}(t)x_1 + c_{32}(t)x_2 + c_{33}(t)x_3 + c_{34}(t)x_1^2 + c_{35}(t)x_1x_2 +$$

$$+ c_{36}(t)x_1x_3 + c_{37}(t)x_2^2 + c_{38}(t)x_2x_3 + c_{39}(t)x_3^2. \tag{7}$$

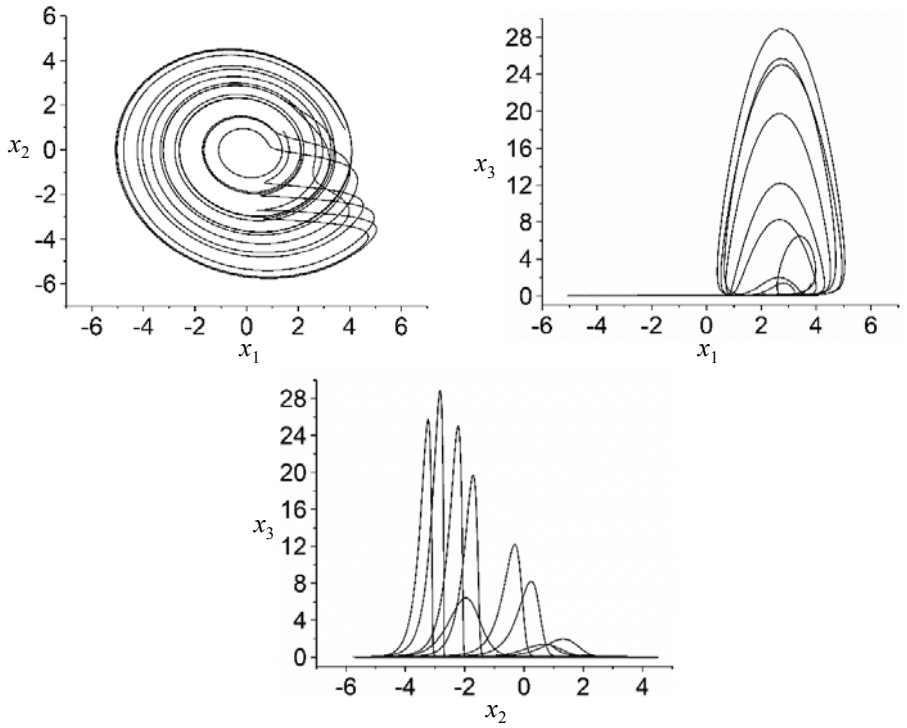


Fig. 2. Phase trajectories of system (6)

The results of applying the algorithm [7] are presented in Table 1.

Table 1. The τ values for which the δ_j value is closest to zero. The first row shows the τ values at which δ_j takes on the least values. As the row number increases, the δ_j value also increases

№	**The τ values calculated for the coefficients of equation (7) at $\delta \to \min$**									
	$c_{30}(t)$	$c_{31}(t)$	$c_{32}(t)$	$c_{33}(t)$	$c_{34}(t)$	$c_{35}(t)$	$c_{36}(t)$	$c_{37}(t)$	$c_{38}(t)$	$c_{39}(t)$
1	2.18	4.75	**8.00**	9.10	**4.00**	**8.00**	6.18	**4.00**	**8.00**	2.71
2	4.31	**4.00**	**4.00**	**8.00**	**8.00**	**4.00**	0.58	**8.00**	**4.00**	**8.00**
3	1.88	**8.00**	1.81	5.30	3.91	6.25	6.16	10.33	4.44	8.33
4	1.26	3.41	7.98	4.89	9.76	3.95	4.09	7.99	4.01	5.39
5	7.99	1.06	3.82	3.20	5.78	6.07	6.19	3.98	7.98	6.48

In Table 1, the τ values that are repeated or multiples are highlighted in bold. The theorem in article [7] suggests that they can correspond to the real value of the period. It also follows from the theorem that the presence of such values in a certain column of the table indicates that this coefficient is constant.

Note that when $\tau = T$, relation (4) must be satisfied. That is, the τ values highlighted in bold (which correspond to the real value of the period) must be located in the top row of the table. However, this is not observed for some coefficients, which is explained by the possible presence of computational errors [7].

As can be seen from Table 1, the least period obtained as a result of the calculation is $T = 4s$. The value of $8s$ from the table obviously corresponds to $2T$. Now that the period T of the external actions is known, it is possible to deter-

mine the form of all functions $c_{3j}(t)$ using the final part of the algorithm [7]. To do this, we form matrix **A** and vector **B** for system (2) taking into account relations (3) with $\tau = T$ and solve system (2) for t_0 values from a certain range.

Thus, we obtain the values of the functions $c_{3j}(t)$ at all points in this range. Fig. 3 shows the time series $c_{30}^c(t)$, $c_{31}^c(t)$, $c_{33}^c(t)$, $c_{36}^c(t)$, obtained as a result of the calculation at the interval $t_0 \in 0,...,20\,s$. The figure shows time series of coefficients of different types: variables $c_{30}^c(t)$ and $c_{36}^c(t)$, constant zero $c_{31}^c(t)$, and constant non-zero $c_{33}^c(t)$. The numerical values of the constant coefficients can be estimated from the form of the obtained time series $c_{3j}(t)$. More accurate values can be obtained using their values obtained by solving system (2) for t_{01} and t_{02}, for which $\tau = T$ or τ is a multiple of T, see Table 2 in [7].

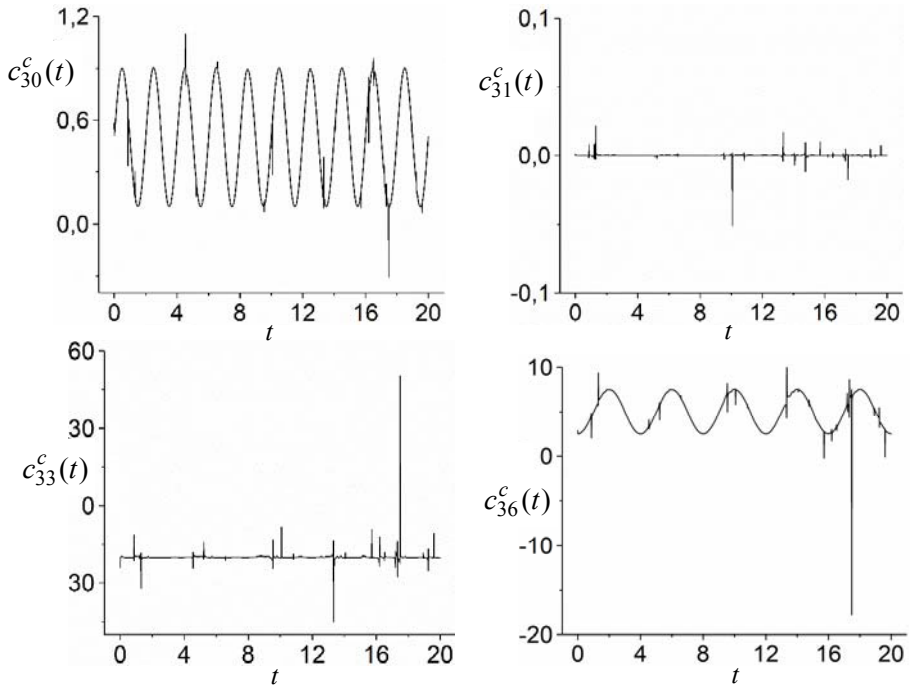


Fig. 3. Time series of calculated coefficients of the third equation of system (6). Calculation was performed using structure (7) of equation

Based on the graphs and estimated values of the constant coefficients, the zero coefficients can be eliminated. As a result, the structure of the general equation (7) is reduced to the structure of the third equation of system (6). Since we know T value and have a simplified equation structure, we can re-identify the equation. Time series of the obtained coefficients are presented in Fig. 4.

Note that according to the calculations (Table 1), condition (3) was satisfied for $\tau = 4\,s$. This value obviously corresponds to the period of external action T_6. At the same time, as expected, the value of T_0 was not determined as a result of the calculation. However, after comparing the form of the external action $c_{30}^c(t)$ with $c_{36}^c(t)$ in Fig. 3 or 4, it can be argued that $T_0 = T_6/2 = 2\ s$.

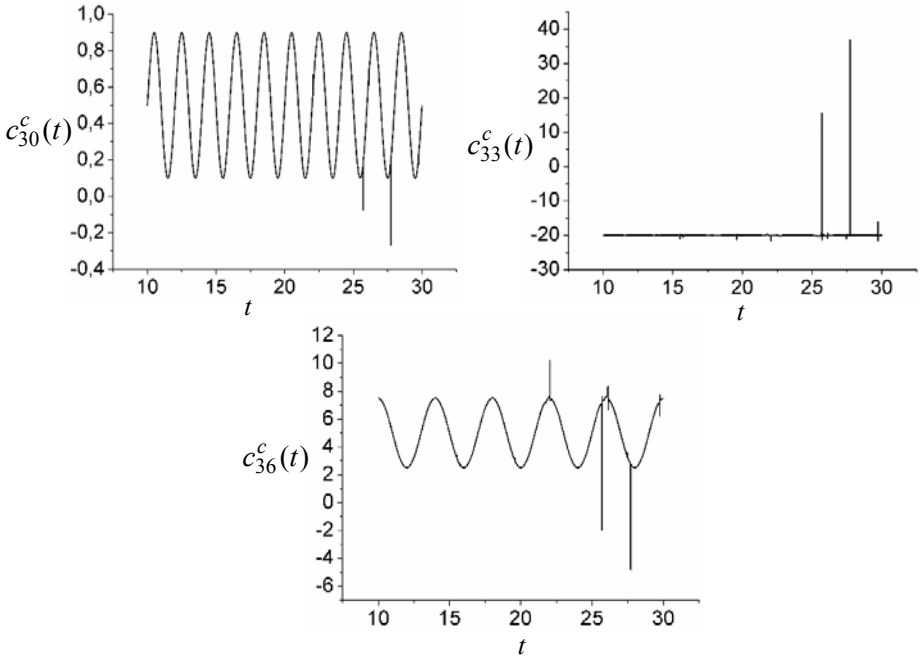


Fig. 4. Time series of calculated coefficients of the third equation of system (6). Calculation was performed using simplified structure of the equation

Identification of equations with a fractional ratio of periods of external actions

Let's consider a more general case when the periods of actions T_1 and T_2 in the equation are subject to condition (5) and in this case T_1/T_2, $T_2/T_1 \notin 2,3,\ldots$. Let the external actions in the third equation of the system (6) have the form:

$$c_{30}(t) = 0.5 + 0.4\sin\left(\frac{2\pi t}{T_0}\right),\ c_{36}(t) = 5 + 2.5\sin\left(\frac{2\pi t}{T_6} - \frac{\pi}{4}\right),\ T_0 = 3s,\ T_6 = 2s. \quad (8)$$

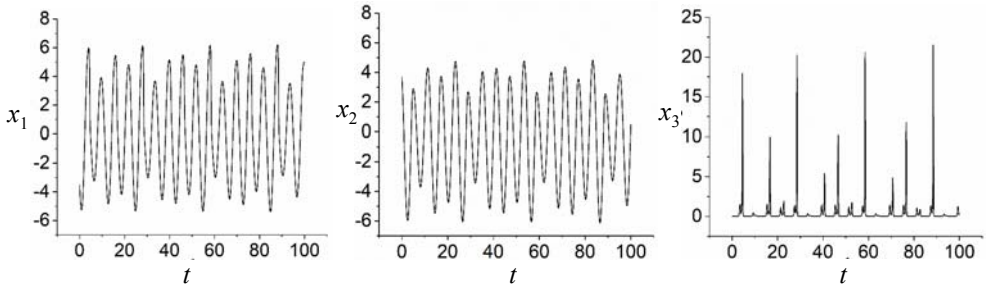


Fig. 5. Time series of variables of system (6) with external actions (8)

Time series and phase trajectories of system (6) under input actions (8) are presented in Fig. 5 and 6, respectively.

Obviously, with $q_1 = 2$ and $q_2 = 3$ we get $q_1T_0 = q_2T_6 = T$, $T = 6s$. That is, relation (5) is satisfied. Then, considering T to be the only period of external actions in the equation, we can apply the theorem and algorithm from [7]. System (6) with external actions (8) was solved on an interval of $100s$ with a step of $\Delta t = 0.01s$. According to the algorithm, the initial times $t_{01} = 0.15s$, $t_{02} = 0.4s$ were selected. The τ value was chosen from the interval $[\tau_b;\tau_e]$, $\tau_b = 1s$,

$\tau_e = 11s$. Table 2 presents the τ values for which relation (4) is satisfied. As can be seen from the table, in most of its columns the value $\tau = 6s$ is found (shown in bold in the table).

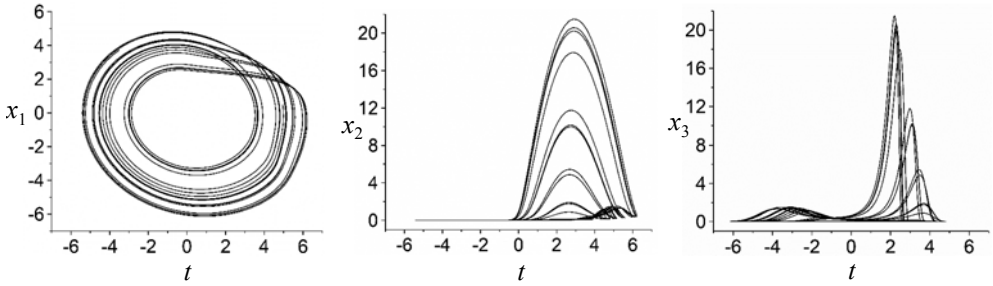


Fig. 6. Phase trajectories of system (6) with external actions (8)

Table 2. The same as in Table 1, for system (6) with external actions (8)

№	The τ values calculated for the coefficients of equation (7) at $\delta \to \min$									
	$c_{30}(t)$	$c_{31}(t)$	$c_{32}(t)$	$c_{33}(t)$	$c_{34}(t)$	$c_{35}(t)$	$c_{36}(t)$	$c_{37}(t)$	$c_{38}(t)$	$c_{39}(t)$
1	9.81	2.91	2.00	10.02	**6.00**	**6.00**	6.32	**6.00**	6.36	2.90
2	2.69	**6.00**	8.03	7.29	1.36	2.70	5.94	9.98	9.99	1.86
3	9.48	4.89	7.62	8.18	8.05	1.43	7.99	2.59	**6.00**	4.49
4	9.68	9.12	8.00	1.25	2.55	6.91	6.55	1.47	2.61	**6.00**
5	8.91	2.90	**6.00**	1.27	10.11	7.19	1.96	6.01	5.94	3.00

After an analysis similar to that carried out in the previous section and elimination of zero coefficients, we obtain time series of constant and variable coefficients presented in Fig. 7.

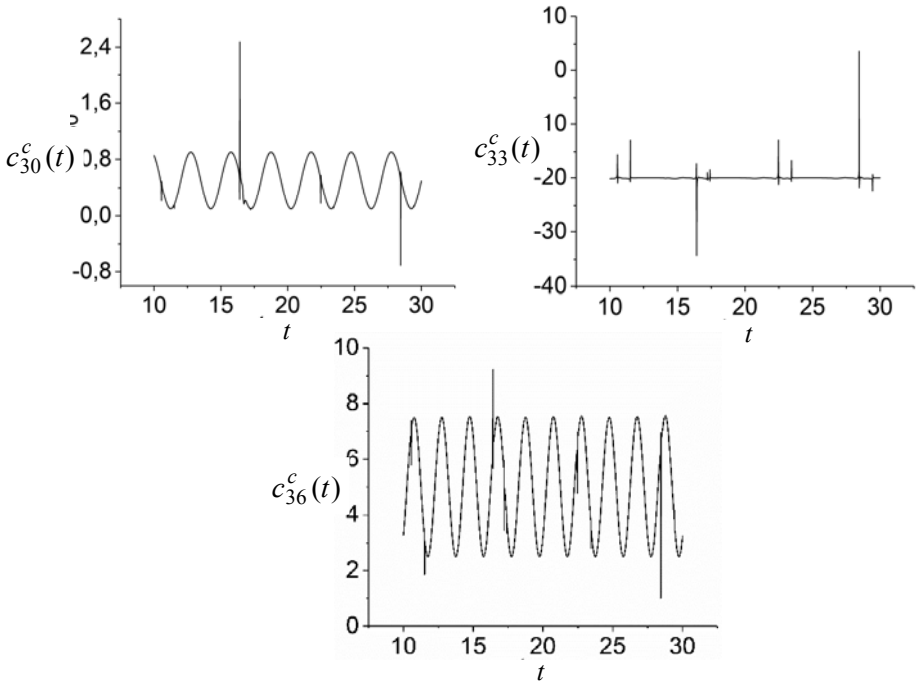


Fig. 7. Time series of calculated coefficients of the third equation of system (6) with external actions (8). Calculation was performed using simplified structure of the equation

If the data in Table 2 are not informative enough to confidently determine periods of external actions, one can repeat the numerical experiment on a larger time interval or/and use the results for more than five least δ values when creating the table. Such a numerical experiment was carried out for system (6) with external actions (8), which was solved over an interval of $200s$. As a result of applying the algorithm, the values given in Table 3 were obtained.

As one can see, the data in Table 3 confirm the correctness of the relationship $\tau = T = 6s$ obtained as a result of the analysis of the data in Table 2. The values of $\tau = 12s$ and $\tau = 18s$ in Table 3 are additional arguments for such a conclusion, since these values are obviously the multiples of $\tau = 6s$.

Table 3. The same as in Table 2, for system (6) with external actions (8). We used a time interval of $200s$ and ten τ values for which $\delta \to \min$

№	**The τ values calculated for the coefficients of equation (7) at $\delta \to \min$**									
	$c_{30}(t)$	$c_{31}(t)$	$c_{32}(t)$	$c_{33}(t)$	$c_{34}(t)$	$c_{35}(t)$	$c_{36}(t)$	$c_{37}(t)$	$c_{38}(t)$	$c_{39}(t)$
1	9.81	2.91	2.00	16.51	**6.00**	**18.00**	6.32	**6.00**	14.13	2.90
2	2.69	**6.00**	8.03	10.02	**18.00**	**6.00**	5.94	**18.00**	6.36	19.48
3	9.48	**18.00**	7.62	7.29	1.36	2.70	7.99	**12.00**	16.95	17.79
4	16.82	4.89	8.00	8.18	8.05	17.06	6.55	9.98	**12.00**	1.86
5	9.68	17.29	18.74	11.40	12.31	**12.00**	12.90	14.05	9.99	14.91
6	8.91	9.12	**18.00**	19.25	2.55	14.14	12.89	2.59	17.45	4.49
7	1.96	2.90	**6.00**	15.50	10.11	1.43	10.88	1.47	14.97	15.00
8	11.29	15.96	**12.00**	13.56	1.02	14.18	18.50	12.16	**6.00**	16.69
9	5.20	1.37	7.33	15.70	19.09	18.68	19.65	16.21	**18.00**	**6.00**
10	2.87	9.72	6.72	12.50	19.39	17.94	12.32	6.01	17.98	3.00

IDENTIFICATION OF THE EQUATION UNDER NON-SINUSOIDAL PERIODIC EXTERNAL ACTION

In this section we investigate system (6) with such coefficients:

$$d = 0.15,\ \ c_{30}(t) = 0.5,\ \ c_{33}(t) = -20,$$

$$c_{36}(t) = 5 + 2\sin\left(\frac{2\pi t}{T} - \frac{\pi}{2}\right) + 1.25\sin\left(\frac{4\pi t}{T}\right) + 0.8\sin\left(\frac{8\pi t}{T} + \frac{\pi}{2}\right),\ \ T = 2s. \quad (9)$$

System (6) with coefficients (9) was solved on the interval of $100s$ with a step $\Delta t = 0.01s$. Fig. 8 shows the time series of system under study and its external action $c_{36}(t)$. Fig. 9 shows the phase trajectories of this system.

Our goal was to identify the third equation of system (6) with coefficients (9). For this purpose, the general structure of an equation of the form (7) was used. As a result of applying the algorithm, the data presented in Table 4 were obtained. As can be seen from this table, the smallest possible value of the period is $T = 2s$.

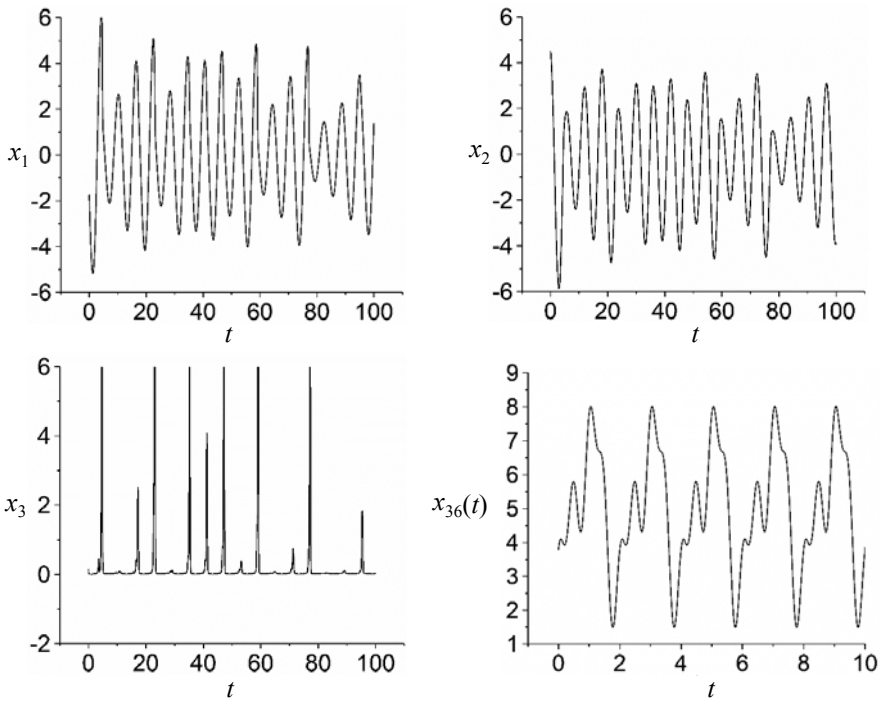


Fig. 8. Time series of system (6) with coefficients (9) and its external action $c_{36}(t)$

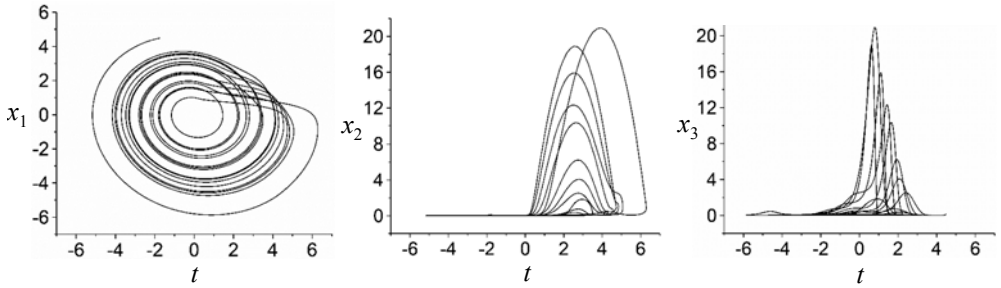


Fig. 9. Phase trajectories of system (6) with coefficients (9)

Fig. 10 shows the calculated time series of coefficients of different types: constant non-zero $c_{30}^{c}(t)$, $c_{33}^{c}(t)$, variable $c_{36}^{c}(t)$ and constant zero $c_{31}^{c}(t)$. The rest calculated coefficients of this equation have time series similar to $c_{31}^{c}(t)$, that is, they are zero.

Table 4. The same as in Table 3, for system (6) with coefficients (9)

№	The τ values calculated for the coefficients of equation (7) at $\delta \to \min$									
	$c_{30}(t)$	$c_{31}(t)$	$c_{32}(t)$	$c_{33}(t)$	$c_{34}(t)$	$c_{35}(t)$	$c_{36}(t)$	$c_{37}(t)$	$c_{38}(t)$	$c_{39}(t)$
1	**8.00**	**4.00**	1.49	**8.00**	**10.00**	**10.00**	2.11	**10.00**	**8.00**	10.57
2	9.23	**2.00**	**10.00**	**10.00**	**8.00**	**8.00**	2.75	**8.00**	**10.00**	7.56
3	**10.00**	9.24	**8.00**	6.62	**4.00**	4.54	2.77	**2.00**	**2.00**	5.78
4	10.31	**10.00**	10.67	5.67	**2.00**	**2.00**	8.71	5.60	**4.00**	4.50
5	**4.00**	**8.00**	**2.00**	**4.00**	10.97	**4.00**	1.63	**4.00**	7.60	**2.00**

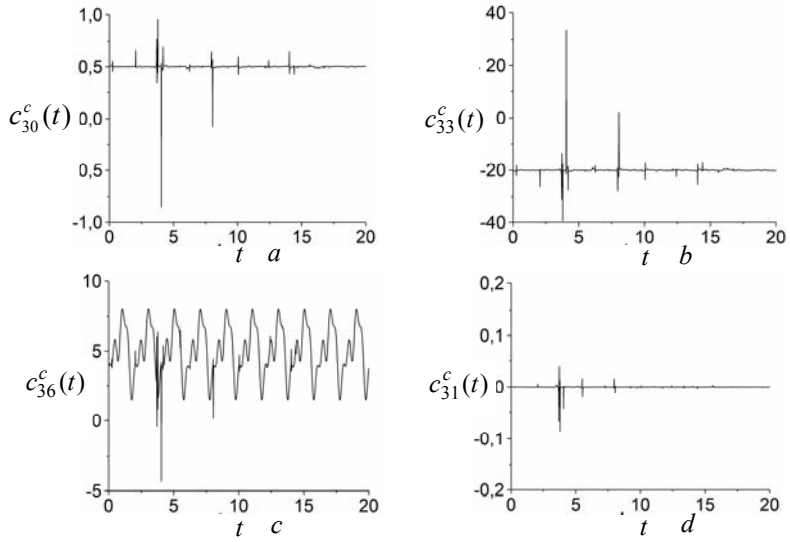


Fig. 10. Time series of calculated coefficients of the third equation of system (6) with coefficients (9)

After simplifying the structure, re-identification was carried out and the time series presented in Fig. 11 were obtained. Fig. 11, *d* shows the calculated time series of external action $c^c_{36}(t)$ (line 1) and the original one $c_{36}(t)$ (line 2). As we can see, these time series practically coincide, with the exception of points with singularity.

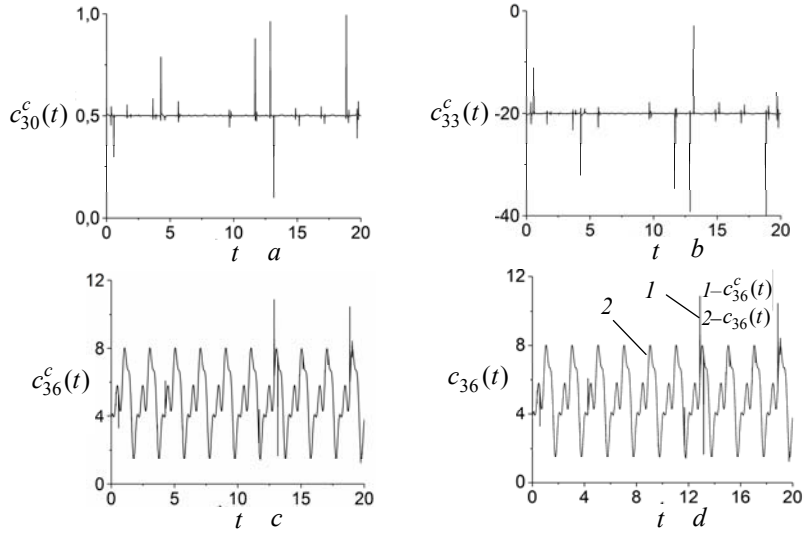


Fig. 11. Time series of calculated coefficients of the third equation of system (6) with coefficients (9). Calculation was performed using simplified structure of the equation

CONCLUSION

This study examines a number of special cases of using the proposed method for identifying nonlinear oscillatory systems with external periodic actions. The most complex part of the identification problem in this case is finding the periods of external actions. The first part of the algorithm is devoted to solving this problem. It can be noted that this method makes it relatively easy to find periods of external actions when identifying systems with an integer value of the ratio T_1/T_2 or

T_2/T_1 . If one of these conditions is met, estimating the values of T_1 and T_2 using this algorithm does not differ from the case $T_1 = T_2$ considered in [7].

With a fractional ratio T_1/T_2 or T_2/T_1, much longer observations of system's functioning may be required in order to make an estimation.

The study of the system with non-sinusoidal periodic external action demonstrates that the proposed method is as effective as in the case of sinusoidal action.

A possible prospect for further development of the method could be, for example, a study of the dependence of the magnitude of the algorithm error on various parameters of the identified equations. It is also of interest to assess the influence of noise on the result of applying the algorithm.

REFERENCES

1. P.E. Kloeden, M. Rasmussen, "Nonautonomous dynamical systems," *Mathematical Surveys and Monographs*, vol. 176. Providence, Rhode Island: American Mathematical Society, 2011. doi: 10.1090/surv/176
2. P.E. Kloeden, C. Potzsche, "Nonautonomous dynamical systems in the life sciences," *Lecture Notes in Mathematics*, vol. 2102, pp. 3–39, 2013. doi: 10.1007/978-3-319-03080-7_1
3. A. Tarantola, *Inverse problem theory and methods for model parameter estimation*. Philadelphia: Society for Industrial and Applied Mathematics, 2005. doi: 10.1137/1.9780898717921
4. B.P. Bezruchko, D.A. Smirnov, "Constructing nonautonomous differential equations from a time series," *Phys. Rev. E.*, vol. 63, art. no. 016207, 2001. doi: 10.1103/PhysRevE.63.016207
5. T. Sauer, "Observing periodically forced systems of difference equations," *J. Difference Equ. Appl.*, vol. 16, no. 2-3, pp. 269–273, 2010. doi: 10.1080/10236190902870439
6. V.G. Gorodetskyi, "Identification of nonlinear systems with additive external action," *J. Aut. & Inf. Sci.*, vol. 50, no. 4, pp. 13–24, 2018. doi: 10.1615/JAutomatInfScien.v50.i4.20
7. V.G. Gorodetskyi, "Identification of nonlinear systems with periodic external actions (Part 1)," *System Research and Information Technologies*, no. 3, pp. 93–106, 2024. doi: 10.20535/SRIT.2308-8893.2024.3.06
8. O.E. Rössler, "An equation for continuous chaos," *Phys. Lett. A.*, vol. 57, no. 5, pp. 397–398, 1976. doi: 10.1016/0375-9601(76)90101-8

Received 12.09.2024

INFORMATION ON THE ARTICLE

Viktor G. Gorodetskyi, ORCID: 0000-0003-4642-3060, National Technical Universityof Ukraine "Igor Sikorsky Kyiv Polytechnic Institute", Ukraine, e-mail: v.gorodetskyi@ukr.net

ІДЕНТИФІКАЦІЯ НЕЛІНІЙНИХ СИСТЕМ З ПЕРІОДИЧНИМИ ЗОВНІШНІМИ ДІЯМИ (Частина ІІ) / В.Г. Городецький

Анотація. Подано результати дослідження, яке є продовженням попередніх досліджень автора. Розглянуто більш складні задачі ідентифікації нелінійних систем з періодичними зовнішніми впливами. Показано, що запропонований раніше метод також застосовний, коли періоди зовнішніх дій в одному диференціальному рівнянні можуть відрізнятися. При цьому співвідношення між значеннями періодів може бути як цілим, так і дробовим. Сформульовано умови, за яких це можливо, і які базуються на теоремі, доведеній у попередній роботі. Частину цього дослідження присвячено проблемі ідентифікації хаотичної системи з вхідною несинусоїдальною дією. Для створення такої зовнішньої дії використано функцію з трьома гармонічними складовими. Чисельний експеримент підтвердив ефективність алгоритму і в цьому випадку.

Ключові слова: ідентифікація, звичайне диференціальне рівняння, зовнішня дія, періодичний коефіцієнт, сталий коефіцієнт.

UDC 004.8
DOI: 10.20535/SRIT.2308-8893.2024.4.06

ON THE EVOLUTION OF RECURRENT NEURAL SYSTEMS

G.S. ABRAMOV. I.V. GUSHCHIN, T.O. SIRENKA

Abstract. The evolution of neural network architectures, first of the recurrent type and then with the use of attention technology, is considered. It shows how the approaches changed and how the developers' experience was enriched. It is important that the neural networks themselves learn to understand the developers' intentions and actually correct errors and flaws in technologies and architectures. Using new active elements instead of neurons expanded the scope of connectionist networks. It led to the emergence of new structures — Kolmogorov–Arnold Networks (KANs), which may become serious competitors to networks with artificial neurons.

Keywords: recurrent neural networks, transformer technology, KANs.

INTRODUCTION

In modern programming there are three degrees of formalization. The first is codes, the second is languages, for example, the most common language for neural network developers is Python. Third, these are libraries that, in addition to data and dictionaries, have a large range of technologies. You actually turn to such technology, mark it in the code, enter the necessary data and it does everything itself. Instead of hundreds of lines of the program, there are dozens left. Moreover, the complexity of the program only increases taking into account libraries. Humanity is increasingly moving into the category of users, since less and less attention is paid to basic formal primary knowledge and descriptions, and people are already using derivatives, such auxiliary structures for describing knowledge. Only a few people are interested in the basics of science, but without such people progress will stop. This article is an example of the efforts of such smart and ambitious people, inventors looking for new opportunities.

For example, this is a matrix recording, a vector with a large number of components is immediately supplied to the network input — a whole set of queries, transformations in the network array occur in matrix form, while fortunately it is linear. The next aspect is the use of the attention mechanism that arose in developed recurrent networks.

Parallel calculations of vectors and matrices due to advanced CUDA technologies on video cards, as well as the use of matrix notation itself, are essential. speeds up calculations. Each of these methods: 1) matrix notation; 2) parallel computing on video cards; 3) the attention mechanism speeds up the work of

modern artificial neural networks by approximately an order of magnitude. It is not surprising that all this has made it possible to move from human-controlled machine learning to deep learning, which is implemented by the network itself, which has acquired new qualities.

However, the very structure of connectionist networks, that is, networks consisting of many active elements with a set of free parameters that allow it to learn or learn, can develop towards a computation graph in the most general sense. These are KAN networks (Kolmogorov–Arnold Networks), and other networks of this type may yet appear.

But it is interesting to consider how people thought when creating language models. These models first took the form of recurrent networks, and then, when the attention mechanism appeared there, networks with "transformer" technology. But the most important are the methods and methods of creating new devices and technologies. *This work is dedicated to this urgent problem*.

Usually, people have a set of data — the history of processes and want to predict the future. Formally, we are talking about a known distribution of probabilities $P(x_t \mid x_{t-1},...x_1)$ on these data, an estimate of the conditional expectation $\mathrm{E}[(x_t \mid x_{t-1},...x_1)]$ (here conditional means that the expectation of the value $x_{t-1},...x_{t-\tau}$) should be found if the conditions for the appearance of the values before) are met. Linear regression is usually used for this. usually only an undetermined number of these previous quantities is confusing, although this problem can be solved by choosing a window τ — the length of this sequence of data (this is attributed to Markov models, for example — orders that take into account the sequence $x_{t-1},...x_{t-\tau}$). If it is possible to somehow summarize the previous data and this summary is marked as $h_i = g(h_{i-1}, x_{t-1})$, then it is possible to enter into the previous forms of description of the forecast $\hat{x}_t$, i.e. $\hat{x}_t = P(x_t \mid h_i)$. Here, a summary appears in the description, which in recurrent models of neural systems is formed by the network itself, and therefore this summary is often called a hidden description (probably because the network does not give users a clear view).

CLASSIC RECURRENT NETWORKS

The idea of the original linguistic classical recurrent networks[1] (RNN) (their recurrent nature is that they constantly use what was known before) is to step by step supplement the text with the most likely next word. At each step, the output h_t, which depends on the inputs x_{t-1} x_{t-2}, … x_{t-m} at the previous steps[2], is calculated. Since it is not desirable for users to find an explicit view h_t, it is easier to call it a hidden description. Later data values in these models depend on earlier ones. Architecturally, a recurrent neural network is a chain of repeating modules. Dictionaries began to be used — embeddings, which represent words in vector

[1] The active use of such architectures tends to be attributed to S. Hochreiter and his colleagues in the early 90s of the last century.

[2] To select the influence of previous words on the following ones (selection of internal memory with a state vector s_t), gate architectures are used, for example, Long Short-Term Memory (LSTM) and gate recurrent unit (Gated Recurrent Unit - GRU). In the practice of creating recurrent LSTM networks [1; 2], blocks were used to improve the transfer of information from previous iterations of recurrent RNN networks.

form, and the distance between vectors depends on how often these words correlate with each other in texts (sets of which are often called corpora).

In recurrent networks of the classical type, the probability and frequency of a pair-triple of neighbouring words from the dictionary was found, and the integral probability of the sentence or phrase maximized $p(x_1,...x_T)$ by the network was formed, by expanding it into a product of conditional densities from left to right, applying the chain rule of probability:

$$P(x_1,...x_T) = P(x_1)\prod_{t=2}^{T} P(x_t \mid x_{t-1},...x_1). \quad (1)$$

You can find the conditional probability for the entire depth of memory $P(x_t \mid x_{t-1},...x_{\tau>1})$, or $P(x_t \mid x_{t-1})$ the length of the corpus of words (1), as well as Markov approximations of different depths of memory , and even , then the probability of the entire sentence, phrase, or corpus will be according to the choice of Markov models, for example τ — orders that take into account the sequence $x_{t-1},...x_{t-\tau}$):

$$P(x_1,...x_T) = P(x_1)\prod_{t=2}^{T} P(x_t \mid x_{t-1},...x_1),$$

$$P(x_1,...x_T) = P(x_1)\prod_{t=2}^{T} P(x_t \mid x_{t-1},...x_{\tau>1}), \quad (2)$$

and even

$$P(x_1,...x_T) = P(x_1)\prod_{t=2}^{T} P(x_t \mid x_{t-1}).$$

Here, respectively, the length of the data sequence T, τ and the length equal to one are chosen.

By selecting words from the dictionary, the network searches for the maximum conditional probability of individual parts of the corpus and the entire corpus. The main task of the classical recurrent network is to generate text, to select words that are most similar to previous phrases and sentences. Further development takes into account a certain summary of past calculations, replacing

$$P(x_t \mid x_{t-1},...x_1) \to P(x_t \mid h_t),$$

and updating the form of these hidden (from developers) states $h_i = g(h_{i-1}, x_{t-1})$. These models were also called latent autoregressive models (see, for example, [3]).

RECURRENT NETWORKS WITH ENCODER AND DECODER FOR TRANSLATION

More modern recurrent networks are even bidirectional (they remove the problem of using only previous data), still form sentences sequentially word by word with the maximum of first local (digram probability, for example), then integral maximum conditional probability (1) or (2), but now they have an encoder and a decoder, which are capable of forming an initially poor-quality translation between language A (encoder) and language B (decoder) when using dictionaries. A phrase in language A is presented to the encoder. A phrase in language B is formed from the dictionary in the decoder.

The translation procedure: 1. The probability of finding a pair of words next to each other is estimated (from previous training of the network). 2. The frequency of appearance of this pair in the studied samples is estimated. 3. The overall probability of a phrase or text fragment is formed.

In the encoder, sequences of hidden states $\vec{h}_t = f(\vec{x}_t, \vec{h}_{t-1})$ are formed, hidden from people, and collected into a context vector $\vec{c} = q(\vec{h}_1,...,\vec{h}_T)$ for language A, which is presented in the encoder. In the decoder for language B, the output sequence $(y_1,...,y_T)$ for each time step t' (we use *t'* in contrast to the time steps of the input sequence of the encoder *t*), the decoder assigns a predicted probability to each possible word (token) occurring at step *t'*+1 determined by the previous tokens in the target object $(y_1,...,y_{t'})$ for language B and adds the context vector, i.e.

$$P(y_{t'+1} \mid y_1,..., y_{t'}, \vec{c}) . \quad (4)$$

Prediction of the next lexeme t'+1 in the target sequence: the RNN decoder takes the target marker (the marker here is the *y* value *y*) of the previous step, the hidden state of the RNN from the previous time step $h_{t'-1}$, the context vector $\vec{c}$ as input, and translates them into the hidden state at the current time step $h_{t'}$. Already in this description, it is clear that even the developers do not know how exactly this was done, but they almost understand the structure and nature of the transformations. The revolutionary action at this step of evolution was the use of two networks — encoder and decoder, which are respectively connected to dictionaries of different languages.

RECURRENT NETWORKS WITH ATTENTION MECHANISM

Using the Bahdanau attention mechanism (https://d21.ai/) allows you to use the information obtained not only from the last hidden state, but also from any hidden state h_t of the encoder for any iteration *t* (runs from 1 to m). With the help of the attention mechanism (accepts the hidden states of the encoder h_i and the hidden state of the decoder $\vec{s}_{t'-1}$ creates a weighted estimate *s* from the sum of the states of the encoder) "focusing" of the decoder on certain hidden states of the encoder is achieved. In cases of machine translation, this capability helps the decoder predict which hidden states of the encoder, given the output of certain words in language A, should be paid more attention to when translating this word into language B.

The attention mechanism was first used in the Seq2seq[3] (sequence-to-sequence) network for machine translation (Machine Translation — MT) [4].

The layer of the attention mechanism is a single-layer neural network, which is fed not the final hidden value, but all such values h_i (t =1, ...*m*), as well as the hidden value of the decoder $\vec{s}_{t'-1}$ at its previous step (iteration). The output of the attention layer is the value of the vector ***s*** (score). This will actually be the weight of the hidden value h_i. Softmax is used to normalize *s*. Then $e = \text{soft}\max(\boldsymbol{s})$. Now the context vector takes the form

[3] In fact, Seq2seq technology has changed the nature of the recurrent network of the previous type.

$$c = \sum_{i=1}^{m} e_i h_i .$$

Thus, the result of the work of the attention layer is the context vector ***c***, which is constantly changing during the calculation process and includes information about all hidden states of the encoder weighted by attention. Transferring a constantly corrected context vector to the decoder improves, as practice has shown, the quality of translation due to changing the context of the encoder and decoder. The main idea of the attention mechanism is that instead of storing a state that summarizes the encoder's original sentence, the network dynamically updates it as a function of the original text (encoder hidden states h_i) as well as the translation text that has already been generated (hidden states decoder $\vec{s}_{t'-1}$). This gives a new context vector ***c***, which is updated after any decoding step t'. The main thing is that already at this stage of the development of neural language models, researchers stop understanding the meaning of transformations of vectors describing sequences. It is argued that "models of attention provide "interpretability", although what exactly the weights of attention mean ... remains a nebulous topic of research".

Then they found an opportunity to use new developments of the attention mechanism used in the "Transformer" technology to form the context vector. In this variant, the context vector c is the result of the combination of attention (the layer of the attention mechanism is then not needed):

$$c_{t'} = \sum_{t=1}^{T} \alpha(s_{t'-1}, h_i) h_i ,$$

used here as a query $S_{t'-1}$, and h_i as a key, and as a value in the terms and designations of the "Transformer" technology.

"TRANSFORMER" TECHNOLOGY

In the "Transformer" technology (see, for example, [5–7]), which already replaces all previous translation systems, the attention mechanism allows you to abandon the recurrent mechanism of forming phrases and corpora "from word to word" and exclude LSTM and GRU blocks, now the sentence is considered all at once. Therefore, the principle of recurrence is no longer needed.

Now all the hidden states of the encoder $h(t)$ are passed to the decoder, which forms the attention weights for the initial sequence. During token prediction, if not all input tokens are relevant (fit), the model considers more of the input sequence that is considered relevant to the current prediction. So to speak, he focuses his attention on them.

With the emergence of the attention mechanism, there is a need for coding, which replaces the numbering of words (tokens). It is possible to enter trigonometric functions — modes (for example, with the wave number $k = 2\pi / L$) on the body L (the number of sentences and words), the multiplication of which does not yield the absolute value of vectors from the interval (0–1). The matrix that numbers the tokens is added to the matrices used in calculations. The need to use this matrix is due to the fact that it is necessary to restore the sequence formation procedure, which was previously automatically implemented in the recurrent scheme.

Next, we denote $D \overset{def}{=} \{(\vec{k}_1, \vec{v}_1),...,(\vec{k}_m, \vec{v}_m)\}$ the database *m* of tuples of keys $\vec{k}_i$ and values $\vec{v}_i$. In addition, we denote $\vec{q}$ as request[4] . This approach, it is believed, helps to form the principles of creating the attention mechanism of the "Transformer" technology

$$Attention\ (\vec{q}, D) \overset{def}{=} \sum_{s=1}^{m} \alpha(\vec{q}, \vec{k}_i)\vec{v}_i,$$

where $\alpha(\vec{q}, \vec{k}_i)$ are the scalar weights of attention. All values of hidden states are multiplied by these weights, and form a weighted sum of values. Note that the scalar weights are chosen quite phenomenologically, using, for example, the most famous Gaussian kernel [8], which describes the characteristic dimensions of the distances between words

$$\alpha(\vec{q}, \vec{k}_i) = \vec{q}^T \vec{k}_i / d\ . \tag{3}$$

Note that attention weights still need to be normalized. We can simplify this with the softmax operation:

$$\alpha(\vec{q}, \vec{k}_i) \to \text{soft}\max(\alpha(\vec{q}, \vec{k}_i)) = \frac{\exp(\vec{q}^T \vec{k}_i / d)}{\sum_j \exp(\vec{q}^T \vec{k}_j / d)}.$$

In this way, it was possible to move to a more effective analysis of sentences both individually and in texts (corpus). However, some problems remained, the solution of which led to the appearance of important mechanisms not only for coding, but also, importantly, for improving the style and quality of translation.

AUXILIARY MECHANISMS OF TRANSLATION

Construction of multi-head attention

Vectors corresponding to text elements are divided into several fragments, which are treated in the same way as whole vectors. This approach, where each of the H_i outputs of the attention pool is a head, was made largely to use parallel computing, which was considered more productive. So far, mathematicians are thinking about the correctness of such an approach, practical results have already shown its effectiveness.

In practice, with the same set of requests, keys and values, it is possible to divide different ranges of changes and enter different subspaces of the representation of requests, keys and values. Actually divide the vectors into parts. To this end, instead of performing a single attention merge, queries, keys, and values can be transformed into a set of queries, keys, and values served in parallel. Such a design is called multi-headed, where each of the H_i outputs of the attention pool is a head [7].

In addition, the researchers discovered that, just as in the case of using several encoders and decoders, each such calculation channel is independently filled with a different meaning, and the network creates these so-called ranges and subspaces in a form that is sometimes incomprehensible to developers.

[4] Key, request, value — this is the structure that seemed more understandable to developers. It is not a fact that such a representation will be preserved in the future.

Given a query $\vec{q} \in \mathrm{R}^{d_q}$, a key $\vec{k} \in \mathrm{R}^{d_k}$, and a value $\vec{v} \in \mathrm{R}^{d_v}$, each head with attention H_i is calculated as

$$\vec{H}_i = f(\vec{W}_i^q \vec{q}, \vec{W}_i^k \vec{k}, \vec{W}_i^v \vec{v}) \in \mathrm{R}^{p_v},$$

where $\vec{W}_i^q \in \mathrm{R}^{p_q \times d_q}$, $\vec{W}_i^k \in \mathrm{R}^{p_k \times d_k}$, $\vec{W}_i^v \in \mathrm{R}^{p_v \times d_v}$ are input parameters R^p, R^d — show the dimensionality of vectors and matrices and *f* is an attention pool, such as additive attention. It is surprising, but such an action, initiated by too determined developers of neural networks, does not lead to nonsense, but gives quite reasonable results. How it works out in the network still needs serious research.

Self-attention

In addition to the attention used between the encoder and the decoder, each of them needs so-called self-attention, or internal attention. This is practically the same as the classic recurrent network, but in a form that has already become the basis for the Transformer technology. This self-attention now works differently [7], and elsewhere is described as a model of internal attention [9].

The same elements of input or output sequences alternately play the role of queries, keys, and values.

The authors of many works give an example.

Thus, when translating the sentence “Student is studying a transformer”, the word “Student” is the first query, and the key is “studying”. The scalar product of the corresponding vectors of the hidden representation gives the attention score of this pair, which will then be multiplied by the value, i.e., the vector representation of the word “learns”. In the next passage, the query will be the word “learns”, and the key may be the word “transformer”.

As a result, according to expression (3), an attention score will be formed for all request/key pairs, by which all value vectors of the input sequence will be multiplied. The encoder context vector will now be first multiplied by these self-attention weights, and then sent to the decoder. And in the decoder, even after all transformations, it is rational to use self-discipline to avoid inconsistency of the translation with the basics of this language. Self-attention allows you to rework a faithful but not very literary text into a rather attractive and more acceptable one for the reader.

In this mechanism, a query is the name of what needs to be found. Keys are signatures on folders and blocks in the middle of the filing cabinet. Having found the appropriate folder, we can get it and find out the content — the value vector. But in the case of internal attention, one must look for not one value, but a significant number of values from a set of folders. Multiplying the query vector by each of the key vectors will give us the coefficients for each folder (technically: a scalar value followed by a softmax function, i.e. converting this value into a unit interval that makes sense of probability). By adding up all the values with their coefficients, you can get the result of internalattention.

CONCLUSIONS

The process of developing neural networks continues and looks like a strange search method, more intuitive than strictly logical. If at first neural networks were

created by neurophysiologists who understood how the brain works, then mathematicians joined this process, but they were not really listened to.

However, the rapid development of computing systems, advances in parallel computing (see, for example, [10]), and a significant amount of memory have made it possible to more boldly form neural network architectures and technologies. And technologies appeared on the scene, people who were more focused on the technical development of networks. They created such complex and large systems that a different approach to their understanding and presentation was needed. It turned out that the initiative in the development of neural networks is already moving to the neural networks themselves, which are capable of correcting the defects and weaknesses of people's technological innovations, independently finding methods of correcting weak human decisions. An illustration is the creation of the Transformer technology, which is quite inaccurately made by humans, but the neural network itself found methods to correct inaccuracies and inaccuracies and demonstrated a remarkable ability to present users with the result they desired.

The development of neural and similar networks with active elements did not stop there. In fact, Tsybenko's theorem (Universal approximation theorem), which allows approximating any continuous function with a set of neurons with activation functions and a significant number of inputs and outputs, can be used for more general networks with active elements. The main thing is to be able to make the necessary functional connection between the inputs and outputs of the network, which is possible if there is an opportunity to teach the network.

Therefore, it is not surprising that the idea and the first attempts to create a network appeared, where the active elements are spline functions (multiple-polynomial functions that can consist of different polynomials at different segments) [11]. For each spline, more polynomial coefficients need to be introduced, so the new network created from them — KANs (Kolmogorov-Arnold Networks), which very boldly uses the theorem of these famous mathematicians, needs more parameters than exist in networks based on artificial neurons (there the parameters are weights and displacement).

However, it turned out that much fewer layers could then be used. You will have to teach these polynomial functions and this seems to be easier, but it will take longer, and increasing the coefficients of the polynomials will even improve the capabilities of such a network. Such networks are more suitable for solving problems in mathematics. Against the background of such innovations, the achievements of the developers of "transformer" technology no longer seem so significant, especially since mathematicians did not see the mathematical rigor in its architecture.

Even the limitation associated with the problem of using processing on GPUs also turned out to be a solvable problem. Such modified networks were called ReLU-KAN [11], they turned out to be faster than expected and more accurate, which was a pleasant surprise. All these hopes of the developers were confirmed by the practice of using these networks.

In conclusion, it can be noted that in general, the creation of networks with active elements, with customizable connections — such a computational graph — can be implemented in different ways, the main thing is that there are free parameters for its appropriate optimization and the possibility of using parallel computing to speed up learning and use. Although it should be understood that the

amount of internal memory and the complexity of the tasks will still require a large number of active elements and network parameters.

REFERENCES

1. S. Hochreiter, Y. Bengio, P. Frasconi, and J. Schmidhuber, "Gradient flow in recurrent nets: the difficulty of learning long-term dependencies," *A Field Guide to Dynamical Recurrent Neural Networks*. IEEE Press, 2001.
2. J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, *Empirical evaluation of gated recurrent neural networks on sequence modeling.* 2014. Available: https://arxiv.org/pdf/1412.3555
3. I.V. Gushchin, O.V. Kirychok, and V.M. Kuklin, *Introduction to the methods of organization and optimization of neural networks: a study guide.* Kh.: KhNU named after V. N. Karazin, 2021, 152 p.
4. E. Charniak, *Introduction to deep learning*. Massachusetts: The MIT Press Cambridge, 2019, 192 p.
5. D. Bahdanau, K. Cho, and Y. Bengio, *Neural machine translation by joint learning to align and translate.* 2014. Available: https://arxiv.org/abs/1409.0473
6. I. Sutskever, J. Martens, G. Dahl, and G. Hinton, "On the importance of initialization and momentum in deep learning," *International Conference on Machine Learning*, pp. 1139–1147, 2013.
7. A. Vaswani et al., "Attention is all you need," *Advances in Neural Information Processing Systems*, pp. 5998–6008, 2017.
8. E.A. Nadaraya, "On estimating regression," *Theory of Probability & its Applications*, 9(1), pp. 141–142, 1964. doi: https://doi.org/10.1137/1109020
9. A.P. Parikh, O. Täckström, D. Das, and J. Uszkoreit, *A decomposable attention model for natural language inference*. 2016. Available: https://arxiv.org/ pdf/1606.01933
10. V. Gushchin, V.M. Kuklin, O.V. Mishin, and O.V. Pryimak, *Modeling of physical processes using CUDA technology*. Kh.: V.N. Karazin KhNU, 2017, 116 p.
11. Z. Liu et al., *KAN: Kolmogorov-Arnold Networks*. 2024. doi: https://doi.org/ 10.48550/arXiv.2404.19756

Received 01.03.2024

INFORMATION ON THE ARTICLE

Gennadii S. Abramov, ORCID: 0000-0003-0333-8819, Kherson State Maritime Academy, Ukraine, e-mail: gennadabra@gmail.com

Ivan V. Gushchin, ORCID: 0000-0002-1917-716X, Kharkiv National University named after V.N. Karazin, Ukraine, e-mail: i.v.gushchin@karazin.ua

Tetiana O. Sirenka, Kharkiv National University named after V.N. Karazin, Ukraine

ПРО ЕВОЛЮЦІЮ РЕКУРЕНТНИХ НЕЙРОННИХ СИСТЕМ / Г.С. Абрамов, І.В. Гущин, Т.О. Сіренька

Анотація. Розглянуто еволюцію нейромережевих архітектур, спочатку рекурентного типу, а потім із використанням технології уваги. Показано, як змінювалися підходи та збагачувався досвід розробників. Важливо, що нейронні мережі самі навчилися розуміти наміри розробників і фактично виправляли помилки та недоліки в технологіях і архітектурах. Використання нових активних елементів замість нейронів розширило сферу застосування конекціоністських мереж і призвело до появи нових структур — мережі Колмогорова–Арнольда (KAN), які можуть стати серйозними конкурентами мереж зі штучними нейронами.

Ключові слова: рекурентні нейронні мережі, технологія трансформер, KAN.

UDC 004.852
DOI: 10.20535/SRIT.2308-8893.2024.4.07

DETECTING UNSAFE BEHAVIOR IN NEURAL NETWORK IMITATION POLICIES FOR CAREGIVING ROBOTICS

A. TYTARENKO

Abstract. This paper explores the application of imitation learning in caregiving robotics, aiming at addressing the increasing demand for automated assistance in caring for the elderly and disabled. While leveraging advancements in deep learning and control algorithms, the study focuses on training neural network policies using offline demonstrations. A key challenge addressed is the "Policy Stopping" problem, which is crucial for enhancing safety in imitation learning-based policies, particularly diffusion policies. Novel solutions proposed include ensemble predictors and adaptations of the normalizing flow-based algorithm for early anomaly detection. Comparative evaluations against anomaly detection methods like VAE and Tran-AD demonstrate superior performance on assistive robotics benchmarks. The paper concludes by discussing further research in integrating safety models into policy training, which is crucial for the reliable deployment of neural network policies in caregiving robotics.

Keywords: assistive robotics, reinforcement learning, diffusion models, imitation learning, anomaly detection.

INTRODUCTION

In recent years the fields of robotics and AI attracted lots of interest. The advances in deep learning, robotics hardware, deep reinforcement learning, and imitation learning made it possible to solve complex control problems by training a neural network policy from mere hundreds of demonstrations.

In this paper caregiving robotics is considered. Given the growing numbers of elderly and disabled people who need daily physical care [1; 2], the importance of automation rapidly increases. Caregiving (or assistive) robotics has a promise of addressing this problem, especially in the light of advances in control algorithms and hardware.

As in most human-robot interaction scenarios, one of the biggest concerns in caregiving control algorithms is safety. This concern is especially important with neural network-based policies, which lack interpretability and are known to become unstable on out-of-distribution data [3].

For the case of imitation learning, this problem is visualized on Fig. 1.

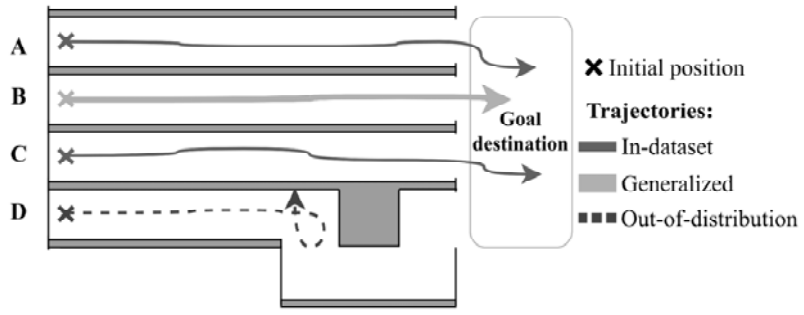


Fig. 1. Out-of-distribution data may lead to failures of a policy

There are 4 episodes: *A*, *B*, *C*, *D* visualized as trajectories from an initial position marked as *X* to goal region. *A* and *C* are present in dataset. *B* is not present, but since it does not differ much from *A* and *C*, the algorithm is able to generalize. The episode *D*, however, is significantly different, and thus, a policy makes unexpected wrong decisions, failing the task.

The progress in the field is nevertheless vast. [4] proposes a method for robotic arm for assistive manipulation tasks. It is a learning-based system, capable of learning from demonstration, based on Dynamic Movement Primitives (DMP) [5]. DMP is a vast framework that includes many instances. Although those methods give a potential for lifelong/incremental learning, they also rely careful modelling and are more difficult to implement and deploy.

Paper [6] introduced simulation software for assistive manipulation tasks, named AssistiveGym. It comes with multiple predefined tasks (feeding, drinking, arm manipulation, etc.) and robots (Jaco, PR1, etc.) to pick. For the study, this simulator is chosen for its versatility, simplicity, and speed. The simulator also comes with a Proximal Policy Optimization-based (PPO, [7; 8]) baseline. In this work, an imitation learning-based approach is used for training a neural network policy. Imitation learning [9–11] allows to avoid the necessity of learning from interaction, by instead leveraging the offline data (demonstrations) collected using an existing policy or via teleoperation.

The uncertainty estimation problem for Reinforcement Learning algorithms is studied in [12]. Although applied to a different task, the authors show that the uncertainty can be estimated using the log-likelihood and the variance of the model. The problem is, DDPMs in general, and Diffusion Policy specifically, is a generative model, for which calculating a likelihood for the generated plan is difficult [13], making the proposed approach hardly applicable for the considered problem. Other methods include [14–17].

In the following sections the "Policy Stopping problem" is studied and solutions are proposed. These solutions are compared to the application of out-of-box anomaly detection and uncertainty detection methods, proved to be successful in other domains. A system with a safety model and an imitation policy is developed and demonstrated. Lastly, the paper concludes with the discussion of the results and further research.

PRELIMINARIES

Markov Decision Process (MDP) is a collection (S, A, r, T) with S — state space, A — action space, $r(s,a)$ – reward function and $T = P(s_{t+1} \mid s_t, a_t)$ — dynamics. In this paper the reward is not assumed to be defined for full trajectories, classifying them as either "success" or "failure".

Reinforcement Learning (RL) algorithms optimize a policy π, which maximizes the expected total reward of the MDP:

$$\pi^* = \arg\max_\pi \mathrm{E}_{\tau \sim p(\cdot)} \sum_{t=0}^{\infty} \gamma r(s_t, a_t),$$

where τ is a trajectory $(s_0, a_0, s_1, a_1, \ldots s_T)$ sampled by applying a policy π.

In offline setting (offline RL) an access to environment for collecting more interactions is assumed to be absent, and the whole training is conducted using only pre-collected demonstrations.

Diffusion Policy is essentially a Denoising Diffusion Probabilistic Model (DDPM) which models a distribution $p(A \mid O)$, where O is a subset of prior ob-

servations, and A is a limited sequence of further actions, i.e. a short-horizon plan.

Normalizing flow-based methods [18] estimate the data likelihood explicitly, by using a reversible block of various kinds. A trained network maps the input data x to latent space Z, such that the inverse mapping $f^{-1}(f(x))$ is trivially computable.

METHOD

Data collection. In this work imitation learning techniques are used to train a neural network policy. Imitation learning methods as a rule require pre-recorded trajectories, e.g. a dataset with sequences of a form:

$$D = \{(s_0, a_0, s_1, a_1, ...s_T)_i,\ i = 1,..., N\}.$$

Here N — is the number of trajectories and T — is a length of a trajectory. For collecting the trajectories, two methods are used — teleoperation and online reinforcement learning algorithms.

Teleoperation is a fairly difficult task when it comes to robotic arm manipulation problems, especially in simulation. A keyboard-based teleoperation feature from the original AssistiveGym implementation is adapted for the task. The modified version is available via GitHub [19].

Online reinforcement learning algorithms allow training a policy neural network by interacting with an environment. They are usually way less sample-efficient, i.e. it takes much more data and training steps to learn a useful behaviour. Nevertheless, it is convenient in case of AssistiveGym, since some tasks are very difficult to teleoperate. Proximal Policy Optimization [7] algorithm is used, which is a well-established baseline Reinforcement Learning method, to collect useful trajectories for some of the tasks.

Diffusion Policy for Assistive Robotics. Recent advances brought much more efficient imitation learning methods, such as Diffusion Policy [10] and Action Chunk Transformer [20]. Diffusion Policy, for instance, allows to train a relatively small neural network policy from up to 200–300 demonstrations in some cases [10].

Diffusion Policy fits a network capable of producing a plan of actions A from $P(A|S)$ without explicitly learning it. More precisely,

$$S = (s_{k-T_O}, ..., s_k), \quad A = (a_{k-T_O}, ..., a_k, ..., a_{k+T_A}),$$

where k is a current time step, T_A — action plan horizon, and T_O — state (observation) horizon. In this work, S is a concatenation of previous states, each of which is represented as a vector of real numbers, i.e. $s_t \in \Re^{N_S}$. In the current study N_S is a relatively small number (<100), although the method allows working with larger-dimensional state spaces. This description also applies to the action plan A: $a_t \in \Re^{N_a}$.

The problem, however, is that it is difficult to compute a likelihood of a sample given a model only, which means that there is only a short-horizon plan **A** without any additional information.

Although the method is known to be sample-efficient, it still highly depends on the quality of the dataset, i.e. state-space coverage, trajectory optimality, etc. See Fig. 1.

Therefore, there are almost none guarantees that a deployed robotic policy won't fail in unexpected ways, potentially damaging the hardware. Moreover, since the Caregiving Robotics deals with human-robot interaction, this may make the robot dangerous to a human, which is a critical in this domain.

Policy stopping problem. In this study approaches to the "Policy Stopping" problem are proposed and compared. In it, an algorithm must decide whether a policy execution must be stopped immediately. This problem can be also viewed as an early anomaly detection problem. However, there is one important difference. The stopping algorithm must be trained on offline data, generated by a behavioural policy (a human demonstrator, a scripted policy, arbitrary neural network policy, or a mix), but tested on a data, generated by a different policy trained on that data (e.g. imitation learning algorithm).

The key difference from traditional unsupervised anomaly detection is that an algorithm is conditioned on a dataset, generated by a distribution different from the test one. Therefore, such algorithm must balance the similarity of test trajectory and train trajectories, distinguishing between a good plan executed successfully but in unusual way and a bad plan that ends up in failure.

State-prediction approach. The first approach considered is inspired by MBPO [14] and widely used in Reinforcement Learning algorithms for different purposes [21–23]. This approach uses a "disagreement" of an ensemble of next state prediction neural networks. The idea is that the next state prediction will be accurate and won't vary much between networks in the ensemble if the input is in-distribution (familiar to the model). At the same time, a state-action pair may not be known. The reason may be that it was not present in a dataset or that a dataset does not contain enough data for a predictor to generalize successfully to this state-action pair. Then, the next state predictors will "disagree", which can be measured as a variance of some kind.

Based on that principle, a network is trained, approximating a function

$$f_\phi(S \mid A) = S',$$

which predicts a vector of T_{out} future states.

For training, inputs and outputs are sampled from a collection of trajectories and a neural network is fit in a simple supervised way, minimizing the MSE (Mean Squared Error) objective:

$$L_{MSE}(S, A, S', \phi) = \| f_\phi(S, A) - S' \|_2^2 .$$

Sampling is executed in a following way:

$$(s_0, ..., s_T) \sim D_{demon}, \quad k \sim \{k, ..., T-k\}, \quad S = (s_{k-T_{in}}, ..., s_k),$$

$$A = (a_{k-T_{in}}, ..., a_k), \quad S = (s_{k+1}, ..., s_{k+T_{out}}),$$

An ensemble of K models is trained, by initializing and fitting them independently on the same data. For estimating the level of uncertainty, a standard deviation between state predictions is computed by the following formula:

$$U^l(S,A)=\frac{1}{K-1}\sqrt{\sum_{i=1..K}\left[f_{\phi_i}(S,A)^l-\frac{1}{K}\sum_{j=1..K}f_{\phi_j}(S,A)^l\right]^2},$$

$$U_{ESP}(S,A)=\sum_{l=1..|S|N_S}U^l(S,A),$$

where $\phi_i\,(i=1..K)$ — parameters of neural networks in an ensemble.

After computing the uncertainty level U_{ESP}, an algorithm compares it to a manually tuned threshold and returns a decision for whether to stop an episode or not. See the Pseudocode 1 for details.

Pseudocode 1. Training an ensemble state prediction model.

1. Input: Dataset D_{demon}.
2. Initialize hyperparameters T_{in}, T_{out}, K.
3. For N_E epochs:
4. For $j=1..K$:
5. Sample S, A, S'.
6. Compute the MSE loss $L_{MSE}(S,A,S',\phi)$.
7. Compute the gradients w.r.t. ϕ_j, update the weights ϕ_j.
8. End for.
9. End for.
10. Return: $\phi_1...\phi_K$.

The considered approach follows [14] with a difference that the input to the state prediction function is not necessarily a single state-action pair, but a chunk, or an entire sub-trajectory. Although excessive due to the assumed Markovianess of the MDP, this allows to incorporate correlations between earlier states and decisions made by an agent, such the resulting neural network ensemble shall disagree when there are longer-term non-immediate anomalies in entire sub-trajectories and not only a single state-action pair.

In other words, single state-action version computes $U_{ESP}(s,a)$, while the proposed one computes $U_{ESP}(S,A)$.

In this study a simple MLP (multi-layered perceptron) architecture is used for a single state-action version, and a CNN (convolutional neural network) is used for the proposed sub-trajectory version.

Adapting anomaly detection methods based on normalizing flows. A promising approach in unsupervised anomaly detection is normalizing flows.

In this paper, a method named MVT Flow [18] is considered. MVT Flow is designed for unsupervised anomaly detection in time series in a robotics domain. Using a convolutional neural network as a backbone, it is trained to estimate the likelihood of normal data. The anomaly score is then computed as a loss function of a test data w.r.t. the trained model.

MVT Flow can't be successfully applied to the presented problem out of box. Although [18] provides a method for credit assignment of elements of the series, it still requires a network to process the entire time series first. Thus, to adapt MVT Flow to early anomaly detection setting the following modification is proposed.

Masking augmentation and sample weighting. The anomaly detection method MVT-Flow is (i) unsupervised and (ii) assigns an anomaly score to the entire input sequence. Therefore, applying it to a not finished sequence may be problematic. The neural net directly maximizes the likelihood of training data, so a previously unseen sequence will get a low likelihood score and will be considered anomaly.

First, unfinished sequences are added to the training data, by randomly choosing a sub-episode length and removing all following elements from the episode. The problem, however, is that the actual abnormal trajectory may start as a normal one with only minor differences. Resulting model does not distinguish between a beginning of a normal trajectory and a fully normal trajectory, where clearly the likelihood should be different.

So, second, a sample weighting is introduced to compensate for that effect:

$$w = \max\left(\sqrt{\frac{K - K_{\min}}{K_{\max} - K_{\min}}}, w_0\right),$$

where $K, K_{\min}, K_{\max}$ are respectively a sub-episode length, a minimum sub-episode length and a full episode length.

Intuitively, the ratio under the square root is a value which is 0 when the sub-episode is minimal and 1 when the sub-episode is full. The square root is applied to smooth the weights, making the difference between the full episode and minimal one smaller.

Full algorithm. Pseudocode 2. Training an early-detection MVT-Flow model.

1. Input: Dataset D_{demon}.
2. Initialize hyperparameters $\gamma, N_E, K_{\min}, K_{\max}, w_0$.
3. For N_E epochs:
4. Sample $S_r, A_r \sim D_{demon}$.
5. Sample random sub-episode length $K \sim \{K_{\min}, ..., K_{\max}\}$.
6. Compute masked data S, A:

$$S^i = I_{i<K}, i = 1...|S|, \quad A^i = I_{i<K}, i = 1...|A|$$

7. Compute the MVT-Flow loss $L_{MVT}(S, A, \theta)$.
8. Compute the sample weight: $w = \max\left(\sqrt{\frac{K - K_{\min}}{K_{\max} - K_{\min}}}, w_0\right)$.
9. Update weights: $\theta := \theta - \gamma w \nabla_\theta L_{MVT}(S, A, \theta)$.
10. End for.
11. End for.
12. Return: weights θ.

EXPERIMENTAL VALIDATION

In this section the results of the study on several benchmarks of Caregiving Robotics are provided. All benchmarks are conducted using environments from the modified version of the AssistiveGym suit, available via GitHub [19].

A simulated Jaco robotic arm is used, the following assistive tasks are considered: Assistive Feeding (250 teleoperation deomnstrations), Assistive Bed Bathing (1000, PPO), Arm Manipulation (1000, PPO), and Scratch Itch (1000, PPO).

First, a policy network is trained using a diffusion policy algorithm on each collected dataset with trajectories.

Next, on each dataset, the models of weighted-masked (WM) MVT-Flow, original MVT-Flow, ensemble state predictors (single state-action and sub-trajectory based), Variational Autoencoder (VAE) and Tran-AD.

Weighted-masked MVT-Flow is trained with $K_{\min} = 1$, $K_{\max} = 200$, $w_0 = 0.1$, $\gamma = 8 \cdot 10^{-4}$, $N_E = 85$. Other hyperparameters are kept in sync with [18].

Ensemble state predictors with $K = 5$. For single state-action version take $T_{in} = T_{out} = 1$, and for sub-trajectory based, take $T_{in} = t$, $T_{out} = 1$. Here t means that all observations and actions observed up to a moment t are considered. Single state-action predictor is applied sequentially, and a maximum uncertainty score is taken as a resulting anomaly score.

Variational Autoencoder has a small CNN backbone and KL penalty is set to 1. The anomaly score is set to the value of reconstruction loss of the input sub-episode.

For Tran-AD the window size is set to 20. For evaluation, a Tran-AD network is inferred on all windows contained within the sub-episode and the resulting anomaly score is set to maximum anomaly score of every window.

Every other hyperparameter remains unchanged from the original papers.

To evaluate the quality of the proposed models, two kinds of metrics are reported: AUROC and FPR@TPR95. The former one is defined as an area under the Receiver Operating Characteristic curve. The later one is defined as the False Positive Rate on a threshold corresponding to 0.95 True Positive Rate. Both are common metrics in anomaly detection literature [24].

However, since the goal is to evaluate the early anomaly detection property, the metrics are reported for partial trajectories of various maximum lengths, namely 10%, 20%, 30%, 50%, 75%, and 100% of the maximum episode length. Better metrics on smaller percentages correspond to better earlier detection ability of an evaluated method.

Tables 1–5 contain metrics reported when evaluated of each assistive environment datasets. Note, that for AUROC larger is better, while for FPR@TPR95 lower is better.

Table 1. Evaluation on Assistive Feeding

Method	Metric	10%	20%	30%	50%	75%	100%
Single SP	FPR@TPR95	0.81	0.73	0.77	0.76	0.58	0.34
	AUROC	**0.79**	0.80	0.79	0.81	0.90	0.92
VAE	FPR@TPR95	0.70	0.76	0.73	0.45	0.18	**0.001**
	AUROC	0.70	0.71	0.77	0.86	0.96	**1.00**
MVT-Flow	FPR@TPR95	0.72	0.37	0.55	**0.23**	**0.06**	**0.02**
	AUROC	0.70	0.80	0.80	**0.94**	**0.98**	**0.99**
Tran-AD	FPR@TPR95	0.74	0.74	0.73	0.63	0.64	0.40
	AUROC	0.65	0.70	0.69	0.79	0.89	0.92
Sub-trajectory SP*	FPR@TPR95	**0.51**	**0.63**	0.60	0.33	**0.07**	**0.001**
	AUROC	**0.79**	**0.83**	**0.83**	**0.92**	**0.97**	**1.00**
WM MVT-Flow*	FPR@TPR95	0.64	**0.61**	**0.50**	**0.21**	**0.06**	**0.001**
	AUROC	**0.77**	**0.83**	**0.84**	**0.94**	**0.98**	**1.00**

Assistive feeding is a simpler task, so most normal trajectories have a relatively short length. Therefore, it is expected that a good method gets maximum score on 100% of the environment length.

Table 2. Evaluation on Arm Manipulation

Method	Metric	10%	20%	30%	50%	75%	100%
Single SP	FPR@TPR95	0.82	0.80	0.77	0.82	0.40	0.20
	AUROC	0.73	0.72	0.79	0.78	0.89	0.95
VAE	FPR@TPR95	0.82	0.80	0.73	0.37	0.26	**0.02**
	AUROC	0.82	0.80	0.77	0.87	0.92	**0.99**
MVT-Flow	FPR@TPR95	0.84	0.76	0.43	**0.16**	**0.08**	**0.01**
	AUROC	0.72	0.75	0.88	**0.95**	**0.97**	**0.99**
Tran-AD	FPR@TPR95	0.97	0.96	0.97	0.90	0.83	0.78
	AUROC	0.43	0.43	0.45	0.51	0.57	0.65
Sub-trajectory SP*	FPR@TPR95	0.84	0.80	0.85	0.41	**0.10**	**0.02**
	AUROC	0.72	0.74	0.68	0.88	**0.96**	**0.99**
WM MVT-Flow*	FPR@TPR95	**0.72**	**0.71**	**0.67**	**0.16**	**0.07**	**0.03**
	AUROC	**0.88**	**0.88**	**0.89**	**0.96**	**0.98**	**0.99**

Table 3. Evaluation Assistive Bed Bathing

Method	Metric	10%	20%	30%	50%	75%	100%
Single SP	FPR@TPR95	0.88	0.91	0.94	0.95	0.90	0.87
	AUROC	**0.79**	0.65	0.66	0.64	0.66	0.67
VAE	FPR@TPR95	**0.84**	0.79	0.79	0.76	0.54	0.85
	AUROC	0.68	0.63	0.63	0.70	0.81	**0.97**
MVT-Flow	FPR@TPR95	1.00	0.83	0.83	0.55	0.44	0.28
	AUROC	0.40	0.66	0.71	0.77	0.82	0.82
Tran-AD	FPR@TPR95	1.00	0.88	1.00	0.94	0.89	0.89
	AUROC	0.50	0.53	0.51	0.51	0.52	0.54
Sub-trajectory SP*	FPR@TPR95	0.88	0.87	0.80	0.82	0.72	**0.001**
	AUROC	**0.77**	**0.74**	**0.74**	0.66	0.71	**1.00**
WM MVT-Flow*	FPR@TPR95	0.87	**0.69**	**0.67**	**0.50**	**0.40**	0.22
	AUROC	0.72	**0.77**	**0.77**	**0.81**	**0.86**	0.94

Bed bathing dataset is challenging due to the low success rate of the demonstration policy. Therefore, the distribution of input trajectories may not cover most scenarios, limiting an imitation learning policy's performance.

Table 4. Scratch Itch

Method	Metric	10%	20%	30%	50%	75%	100%
Single SP	FPR@TPR95	0.84	0.89	0.79	0.73	0.70	0.56
	AUROC	0.60	0.63	0.66	0.69	0.80	0.82
VAE	FPR@TPR95	0.95	0.89	0.84	0.77	**0.29**	0.17
	AUROC	0.60	0.61	0.66	0.75	**0.88**	**0.92**
MVT-Flow	FPR@TPR95	0.85	0.89	0.84	**0.67**	0.45	0.30
	AUROC	0.56	0.65	0.70	0.75	0.84	**0.90**
Tran-AD	FPR@TPR95	**0.72**	**0.60**	**0.70**	0.81	0.67	0.55
	AUROC	**0.77**	**0.79**	0.75	0.71	0.79	0.83
Sub-trajectory SP*	FPR@TPR95	0.88	0.84	0.82	**0.68**	0.38	**0.07**
	AUROC	0.56	0.60	**0.77**	**0.80**	**0.87**	**0.93**
WM MVT-Flow*	FPR@TPR95	0.81	0.83	0.84	**0.65**	0.41	0.30
	AUROC	0.74	**0.78**	**0.79**	**0.79**	**0.86**	**0.91**

From Tables 1–4, on can make the following observations.

First, Weighted-Masked MVT-Flow consistently outperforms raw MVT-Flow. For higher % of maximum length the raw version usually performs on par with the proposed modification, which is expected. This is because anomalous episodes take 100% of maximum length time, while most normal episodes are up to 50–75% of time.

Second, Sub-trajectory SP performs on par with WM MVT-Flow on simpler environments, such as Feeding. It also outperforms single step predictors, especially on larger time periods.

Tran-AD models perform the worst on most datasets due to its windowed inputs. The only exception is Scratch Itch (Table 4). It is hypothesized that the reason for this is the smaller-scale nature of anomalies in the test trajectories.

Now, a demonstration of a system with a diffusion policy deployed with a safety model is provided (see Fig. 2). In practice, a set of thresholds for each time period is selected, since the anomaly score for applied methods is non-decreasing.

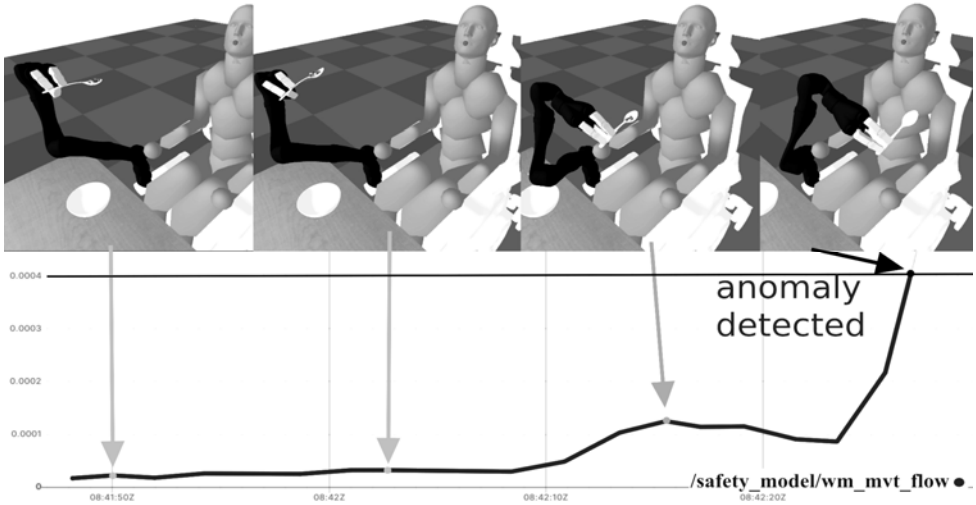


Fig. 2. Demonstration of the proposed approach on the Assistive Feeding environment

The lower part of the diagram shows a plot of the anomaly score (normalized), and arrows matching the upper images with corresponding time steps. Most of the time, the score is low, since the arm performs usual moves. The end of the plot shows a spike in anomaly score, resulting in the system halt. The anomaly is that the arm drops food and spins itself in unusual way. In the remaining of this episode, the arm would twist itself dangerously, potentially damaging hardware.

CONCLUSION

In this paper a challenging “Policy Stopping problem” is introduced and studied. This problem is important for improving safety of imitation learning-based neural network policies, specifically diffusion policies.

The solutions specific to the introduced problem are proposed: ensemble of sub-trajectory-based state predictors and a modification of a recent MVT-Flow algorithm for early anomaly detection.

The algorithms are evaluated and compared against ablated original unmodified versions and known anomaly detection approaches, such as VAE and Tran-

AD. The proposed solutions are shown to be more suitable for the introduced problem and tend to outperform other methods on assistive robotics benchmarks. For the evaluation of early-detection capabilities the usual metrics have been adapted. Lastly, a system with a safety model and an imitation policy is developed and demonstrated.

The interesting future work directions include integration of the proposed safety models to training of imitation policies (e.g. [21]), safe data collection for model finetuning, and adaptation of safety models to vision-based tasks. This may bring the safe and robust deployment of neural network policies, so important for caregiving robotics domain.

REFERENCES

1. J. Broekens, M. Heerink, and H. Rosendal, "Assistive social robots in elderly care: A review," *Gerontechnology*, vol. 8, no. 2, pp. 94–103, 2009. doi: https://doi.org/10.4017/gt.2009.08.02.002.00
2. D.M. Taylor, "Americans with disabilities: 2014," *US Census Bureau*, pp. 1–32, 2018.
3. Dan Hendrycks et al., "The many faces of robustness: A critical analysis of out-of-distribution generalization," *Proceedings of the IEEE/CVF international conference on computer vision, 2021*. doi: 10.1109/ICCV48922.2021.00823
4. Clemente Lauretti et al., "Learning by demonstration for planning activities of daily living in rehabilitation and assistive robotics," *IEEE Robotics and Automation Letters*, vol. 2, issue 3, pp. 1375–1382, 2017. doi: 10.1109/LRA.2017.2669369
5. Matteo Saveriano, Fares J. Abu-Dakka, Aljaz Kramberger, and Luka Peternel, "Dynamic movement primitives in robotics: A tutorial survey," *The International Journal of Robotics Research*, vol. 42, issue 13, pp. 1133–1184, 2023.
6. Z. Erickson, V. Gangaram, A. Kapusta, C.K. Liu, and C.C. Kemp, "Assistive gym: A physics simulation framework for assistive robotics," in *2020 IEEE International Conference on Robotics and Automation (ICRA), IEEE, 2020*, pp. 10169–10176.
7. J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," *arXiv preprint*, 2017. doi: https://doi.org/10.48550/arXiv.1707.06347
8. Jakhotiya Yash, Iman Haque, "Improving Assistive Robotics with Deep Reinforcement Learning," *arXiv preprint*, 2022. doi: https://doi.org/10.48550/arXiv.2209.02160
9. Maryam Zare, Parham M. Kebria, Abbas Khosravi, and Saeid Nahavandi, "A survey of imitation learning: Algorithms, recent developments, and challenges," *arXiv preprint*, 2023. doi: https://doi.org/10.48550/arXiv.2309.02473
10. Chi Cheng et al., "Diffusion policy: Visuomotor policy learning via action diffusion," *arXiv preprint*, 2023. doi: https://doi.org/10.48550/arXiv.2303.04137
11. J. Ho, A. Jain, and P. Abbeel, "Denoising diffusion probabilistic models," *arXiv preprint*, 2020. doi: https://doi.org/10.48550/arXiv.2006.11239
12. Vincent Mai, Mani Kaustubh, and Paull Liam, "Sample efficient deep reinforcement learning via uncertainty estimation," *arXiv preprint*, 2022. doi: https://doi.org/10.48550/arXiv.2201.01666
13. Yang Song, Jascha Sohl-Dickstein, Diederik P. Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole, "Score-based generative modeling through stochastic differential equations," *arXiv preprint*, 2020. doi: https://doi.org/10.48550/arXiv.2011.13456
14. Michael Janner, Justin Fu, Marvin Zhang, and Sergey Levine, "When to trust your model: Model-based policy optimization," *Advances in Neural Information Processing Systems* 32, 2019. doi: 10.48550/arXiv.1906.08253
15. Shunan Guo, Zhuochen Jin, Qing Chen, David Gotz, Hongyuan Zha, and Nan Cao, "Visual anomaly detection in event sequence data," *2019 IEEE International Conference on Big Data* (*Big Data*). doi: 10.1109/BigData47090.2019.9005687
16. Diederik P. Kingma, Max Welling, "Auto-encoding variational bayes," *arXiv preprint*, 2013. doi: https://doi.org/10.48550/arXiv.1312.6114

17. Shreshth Tuli, Giuliano Casale, and Nicholas R. Jennings, "TranAD: Deep transformer networks for anomaly detection in multivariate time series data," *arXiv preprint*, 2022. doi: https://doi.org/10.48550/arXiv.2201.07284
18. Jan Thieß Brockmann, Marco Rudolph, Bodo Rosenhahn, and Bastian Wandt, "The voraus-AD Dataset for Anomaly Detection in Robot Applications," *IEEE Transactions on Robotics, 2023*. doi: 10.1109/TRO.2023.3332224
19. A. Tytarenko, *Assistive Gym Fork*. 2024. Accessed on June 19, 2024. [Online]. Available: https://github.com/titardrew/assistive-gym
20. Tony Z. Zhao, Vikash Kumar, Sergey Levine, and Chelsea Finn, "Learning fine-grained bimanual manipulation with low-cost hardware," *arXiv preprint*, 2023. doi: https://doi.org/10.48550/arXiv.2304.13705
21. Tianhe Yu et al., "Mopo: Model-based offline policy optimization," *Advances in Neural Information Processing Systems 33*, pp. 14129–14142, 2020. Available: https:// proceedings.nips.cc/paper/2020/file/a322852ce0df73e204b7e67cbbef0d0a-Paper.pdf
22. Rahul Kidambi, Aravind Rajeswaran, Praneeth Netrapalli, and Thorsten Joachims, "MOReL: Model-based offline reinforcement learning," *Advances in Neural Information Processing Systems 33*, pp. 21810–21823, 2020. Available: https://proceedings. neurips.cc/paper_files/paper/2020/file/f7efa4f864ae9b88d43527f4b14f750f-Paper.pdf
23. Laura Smith, Yunhao Cao, and Sergey Levine, "Grow your limits: Continuous Improvement with Real-World RL for Robotic Locomotion," *arXiv preprint*, 2023. doi: https://doi.org/10.48550/arXiv.2310.17634
24. Weitang Liu, Xiaoyun Wang, John D. Owens, and Yixuan Li, "Energy-based out-of-distribution detection," *Advances in Neural Information Processing Systems 33*, pp. 21464–21475, 2020. Available: https://proceedings.neurips.cc/paper/2020/file/f5496252609c43eb8a3d147ab9b9c006-Paper.pdf

Received 11.07.2024

INFORMATION ON THE ARTICLE

Andrii M. Tytarenko, ORCID: 0000-0002-8265-642X, Educational and Research Institute for Applied System Analysis of the National Technical University of Ukraine "Igor Sikorsky Kyiv Polytechnic Institute", Ukraine, e-mail: titarenkoan@gmail.com

ВИЯВЛЕННЯ НЕБЕЗПЕЧНОЇ ПОВЕДІНКИ В ПОЛІТИКАХ ІМІТАЦІЇ НЕЙРОМЕРЕЖІ ДЛЯ РОБОТОТЕХНІКИ ДЛЯ ДОГЛЯДУ / А.М. Титаренко

Анотація. Досліджено застосування навчання за імітацією в задачах робототехніки для догляду, спрямоване на вирішення зростаючого попиту на автоматизовану допомогу в обслуговуванні літніх людей і людей з інвалідністю. На підставі досягнень у глибокому навчанні та керуванні дослідження зосереджено на навчанні стратегій, представлених нейронними мережами за допомогою попередньо зібраних демонстрацій. Однією з ключових проблем, яку вирішується, є проблема «зупинки стратегії», що є важливою для підвищення безпеки в стратегіях, заснованих на навчанні імітацією, таких як дифузійні стратегії. Пропонуються рішення проблеми на базі ансамблів прогнозів стану та адаптації алгоритму на основі нормалізаційного потоку для виявлення аномалій на ранніх стадіях виконання. Порівняльний аналіз з методами виявлення аномалій, такими як VAE та Tran-AD, демонструє перевагу в ефективності методів у задачах робототехніки для догляду. Запропоновано подальші напрями досліджень з інтеграції моделей безпеки в навчання нейромережевих стратегій, що є важливим для надійного впровадження нейромережевих рішень у робототехніку для догляду за людьми.

Ключові слова: допоміжна робототехніка, навчання з підкріпленням, дифузійні моделі, навчання імітацією, виявлення аномалій.

UDC 004.8.032.26
DOI: 10.20535/SRIT.2308-8893.2024.4.08

APPLICATION OF NEURAL NETWORK TECHNOLOGY FOR PUBLIC OPINION ANALYSIS

K.M. PEREVOZNIK, Y.V. PARZHYN

Abstract. The research is devoted to studying and using neural network technologies, in particular algorithms and methods of natural language processing, to increase the efficiency of studying and analyzing public opinion of Ukraine's partner countries regarding the war in Ukraine. The research involved analyzing and processing databases consisting of messages about the war in Ukraine on the social network Twitter. The resulting datasets were used to train several neural network models. The best classification results were obtained with the GPT-3.5-turbo model. For a deeper understanding of the results of the public opinion analysis, we created their visualization. The results of the study have shown the high efficiency of the selected solutions. They may be of great practical importance for improving methods of analyzing public opinion and making informed decisions based on a deep understanding of global feedback.

Keywords: public opinion, neural networks, natural language processing, large language models, social networks, classification.

INTRODUCTION

In the context of war, the Ukrainian government has to make prompt decisions on how to interact with the governments of partner countries. These decisions should take into account the public opinion of partner countries in the course of the war in Ukraine. Therefore, monitoring changes in public opinion and the attitudes of people in different countries towards this war is a key aspect for further analysis of the prospects of these countries' assistance to our country, because public opinion can influence both the official positions of partner countries and their leaders. Of course, public opinion is reflected in social media, which can be used as an information base for this analysis.

To improve the efficiency of public opinion analysis, artificial neural networks can be used, so the purpose of this paper is to study the use of algorithms and methods of natural language processing based on artificial neural network models to improve the efficiency of studying and analyzing public opinion in Ukraine's partner countries.

To achieve this goal, the following tasks were solved: search, analysis, and preparation of datasets; determination of the dependence of the target variable on the input data for dataset annotation; dataset markup, cleaning, and verification; tokenization and vectorization of text; creation of training and test samples; determination of the effectiveness of solving the text classification problem using a Gaussian naive Bayesian classifier model and a multilayer perceptron to determine further the effectiveness of using large language models.

The peculiarity of solving the classification problem in this paper is to identify the types of classes and data that most affect the classification process. That is, it is necessary to analyze not the mood of people in its classical psychological sense based on the color of the message, but to determine which posts and messages on social media can be classified as useful or not useful for helping Ukraine. For example, messages about support, assistance, and texts about the success of our troops will be classified as positive, while posts approving of the Russian government's statements, which often turn out to be fake news and disinformation, and texts supporting the war or Russia will be labeled negative. The issues and problems of combating disinformation in social networks and news are discussed in Bergstrom C.T., West J.D. [1].

LITERATURE REVIEW

Literature analysis and selection of technological solutions. Many scientific works are devoted to the problem of using neural networks for analyzing public opinion. Thus, the work of Wordliczek Ł. [2] points out the need to use neural networks for sociological research. The works [3–5] discuss the technical aspects of using neural networks for analyzing public opinion. Gong M. [3] proposes a deep neural network (DNN) model for text sentiment classification, which, according to the author, makes it possible to better extract local and contextual information from the text. The article by Yang S. [4] proposes to use of a genetic algorithm to train a multilayer perceptron to increase the sensitivity of public opinion forecasting. The article by Chen X. et al [5] proposes the use of hybrid fuzzy neural networks. It should be said that the use of large language models is limited by their high computational complexity, which is a common disadvantage of connectionist neural networks [6].

The software model was developed in Python. All parts of the project, namely data aggregation, processing, analysis, model development, training, and testing, were carried out in Jupyter Notebook. The most common Python libraries were used in the work, such as Tensorflow, Keras, Pandas, Sklearn, RegEx, NumPy, Matplotlib, Seaborn, WordCloud, as well as the Gensim package for text processing and topic modeling. In this work, we used the Word2Vec model, which is one of the most popular models for vectorization and contextual word association.

We also conducted an experiment using the GPT-3.5-turbo model for text annotation and classification [7]. The choice of these technological solutions is due to their popularity, wide functionality, convenient interface, availability of multifunctional libraries, and support of the developer community. To work with this model we used the API of the OpenAI platform [8–10]. The use of OpenAI's GPT models is governed by license agreements, which may differ depending on whether the intended use is commercial or non-commercial. Users must also abide by certain terms of use and ethical considerations, such as avoiding harmful or malicious content and complying with privacy laws. However, these restrictions do not, in our opinion, prevent this task from being accomplished. It is likely that using a commercial version of GPT-4 would improve the quality of content analysis but would incur additional financial costs. There are also open-source models, such as LLaMA, whose use for this task also requires further research.

Data selection. Open datasets were chosen to analyze public opinion on the war in Ukraine: Unveiling Global Narratives: A Multilingual Twitter Dataset of News Media on the Russo–Ukrainian Conflict (in the future — Zenodo) [11], which contains 1.5 million tweets in 60 languages, and the dataset from the Kaggle platform — Ukraine Conflict Twitter Dataset (in the future — Tweets) [12], which contains 44 million tweets until June 2023 related to the war between Ukraine and Russia.

The Zenodo dataset was selected due to the presence of columns generated using the RoBERTa model [13]. These columns reflect the essence of the text and indicate its tendency towards being "in support" or "against" Ukraine, Russia, and the war in general. Each record in this dataset is represented in JSONL format. It was decided to use the Tweets dataset for data analysis and classification for the last two months using the GPT-3 model.

MATERIALS AND METHODS

Project structure and implementation. The project consists of three program code files. The purpose of each of the development files is shown in detail in Table.

Purpose of development files

File	File purpose
PartnerSentiment.ipynb	Data preparation and data analysis.
ClassifySentiment.ipynb	Processing text data and creating a process for using the OpenAI API
ModelingSentiment.ipynb	Text data processing, text vectorization, and modeling.

Data preparation. The project implementation begins with the creation of the PartnerSentiment.ipnb file, the main parts of which are provided on the GitHub platform [20]. All the necessary modules were imported and the functionality that iteratively transforms Zenodo data from JSONL to dataframe was implemented. Tweets data was also processed, a data frame was created using Pandas, and functionality was developed to concatenate certain columns, filter only English texts, and extract hashtags from the text using a Regex pattern.

Exploratory Data Analysis. After the data preparation was completed, we started analyzing it. To do this, we created a new column with a shortened date to the year-month format and grouped the data by this criterion to see how many unique text messages there were in each month from the beginning of the war until June 2023. The resulting distribution is shown in Fig. 1.

The next step was to analyze the number of tweets from each country per month. To do this, we used Regex to reduce all the locations of tweet authors to the country name, and the rest of the records were removed. The processing results are shown in Fig. 2.

Next, we analyzed the hashtags that users added to their messages. Using the WordCloud library, we created word distribution graphs for each month.

The analysis of the information shows that in the first months of the war, the country received significant support and the war in Ukraine became the most relevant topic of communication on the world's largest online platforms. It is worth noting that all war crimes were also noticed and brought to the public.

A more complete picture of the history of changes in the most popular topics of discussion can be obtained by creating a hashtag timeline. We also analyzed the hashtags of users with more than 500 thousand followers. This was done to see what topics are discussed by popular figures and whether they are in favor of Ukraine.

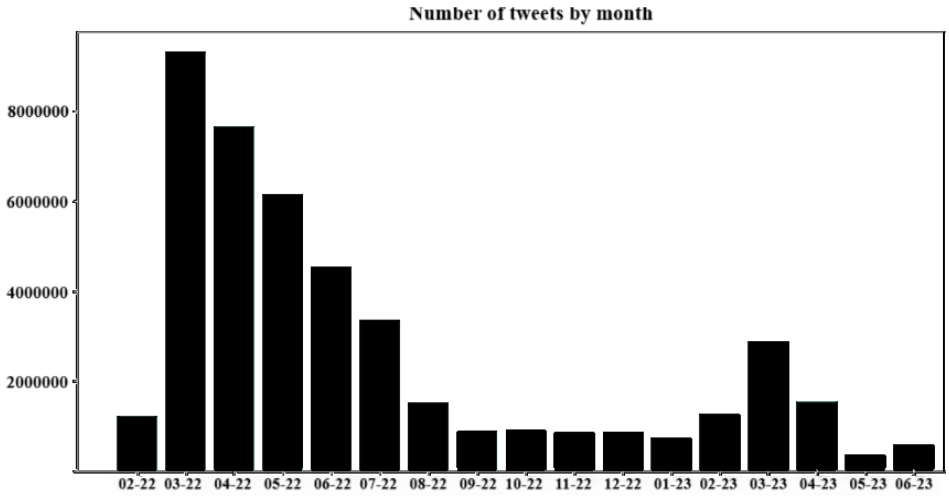


Fig. 1. Number of tweets per month

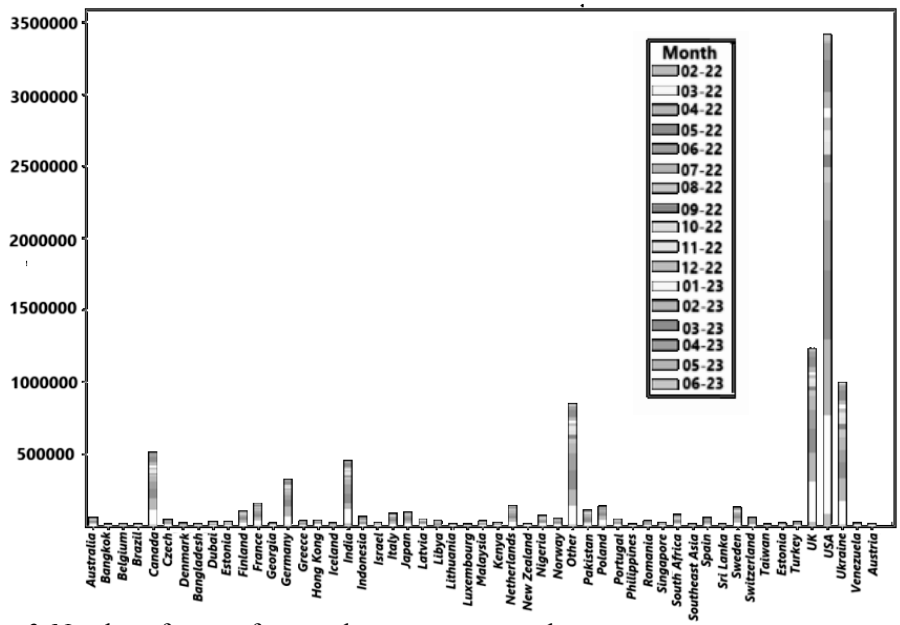


Fig. 2. Number of tweets from each country per month

EXPERIMENTS

Preparation of datasets. To experiment, we prepared the Zenodo dataset for training. The entire algorithm was implemented in the ModelingSentiment.ipynb file, the contents of which are available on GitHub [14]. To determine the target variables in the datasets that are necessary to solve the classification problem, the following rules were empirically derived:

IF $(A_1+A_2) > 2{,}5(B_1+B_2) \rightarrow$ Cl (True);

IF $1{,}7(A_1+A_2) < (B_1+B_2) \rightarrow$C2 (False);

Else → C3 (Unknown).

Where: A1 is the column "This statement is against russia" in the Zenodo dataset; B1 is the opposite column in terms of content; A2 — is the column "This statement is in favor of Ukraine" in the Zenodo dataset; B2 — is the opposite column in terms of content; C1 is a positive class; C2 — negative class; C3 — neutral (undefined) class.

Using the identified class labels, we checked the dataset markup. The markup results are shown in Fig. 3. Next, we created a function to clean the text from unwanted characters, and links, and convert it to lowercase. The text processing function is shown in Fig. 4.

```
[8]: df.mySentiment.value_counts()
```

```
[8]: mySentiment
     Neutral     1194233
     Positive     197985
     Negative     132614
     Name: count, dtype: int64
```

Fig. 3. Distribution of class labels

The text was vectorized using the Word2Vec model, which was trained on the full corpus of words after processing. Before converting the data to a numerical form, it is worth looking at the distribution of text lengths. To do this, we created a process for counting the number of tokens in a message. The process itself and the graph showing the distribution of text lengths are shown in Fig. 5.

```
def preprocess(text):
    text = str(text).lower()
    text = re.sub(r"< user_mention_1 >|< url_1 >",'', text)
    text = re.sub(r"\n|\r|\d", '', text)
    text = re.sub('https:[/a-z0-9.]+|&[a-z;]+|@[a-z]+','',text)
    return text
```

Fig. 4. Function for text processing

It was found that almost 900 entries were less than 5 in length and did not contain much meaning, so they were excluded from the dataset.

```
13 df['len_text'] = df['clean_text'].apply(lambda x: len(x.split()))
   df.len_text.value_counts().sort_index().plot()
13 <Axes: xlabel='len_text'>
```

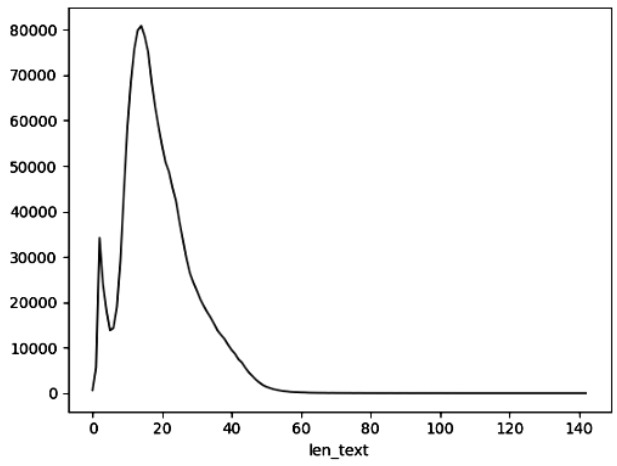


Fig. 5. Distribution of text lengths

The next step was to create a Word2Vec model by splitting all the values of the processed text into tokens and feeding them to the model for training. As a result, the model has a vocabulary of 44351 unique tokens and each vector has 100 values. The training result is shown in Fig. 6.

```
[21]: model = Word2Vec(sentences=sentences.values,
                       sg=1,
                       workers=4)

      model.wv.vector_size, len(model.wv.index_to_key)

[21]: (100, 44351)
```

Fig. 6. Word2vec model

The resulting dataset was divided into training and validation samples. The validation data accounted for 20% of the entire corpus. It was also foreseen that words not contained in the model's dictionary would appear. In such cases, instead of these words, a hundred-dimensional vector filled with values of 0.5 is added to the sum of vectors. The code for this function is shown in Fig. 7.

```
[23]: import numpy as np

      def text_to_vector(text):
          words = text.split()
          vectors = []
          for word in words:
              try:
                  vector = model.wv.get_vector(word)
                  vectors.append(vector)
              except KeyError:
                  vectors.append([0.5 for i in range(100)])
          if not vectors:
              return None
          return np.mean(vectors, axis=0)
```

Fig. 7. The function of converting texts to vectors

Modeling. We first solved the classification problem using a Gaussian naive Bayesian classifier to evaluate the effectiveness of using large language models. The classification results are shown in Fig. 8.

In multiclass classification, accuracy is calculated by summing up the number of correct predictions for each class and dividing it by the total number of predictions made across all the classes. It is a commonly used metric to evaluate the performance of models in multiclass problems. In this study, the model's overall accuracy was calculated using this metric, and it was found to be 45%. The confusion matrix was also analyzed to understand the model's performance for each class. It was observed that the model performed well for the positive class, but it had a high false positive rate for the neutral and negative classes. A new model with four layers of neurons was developed, which improved the accuracy to 57%. However, even with this improvement, the model's accuracy was still not sufficient for practical applications.

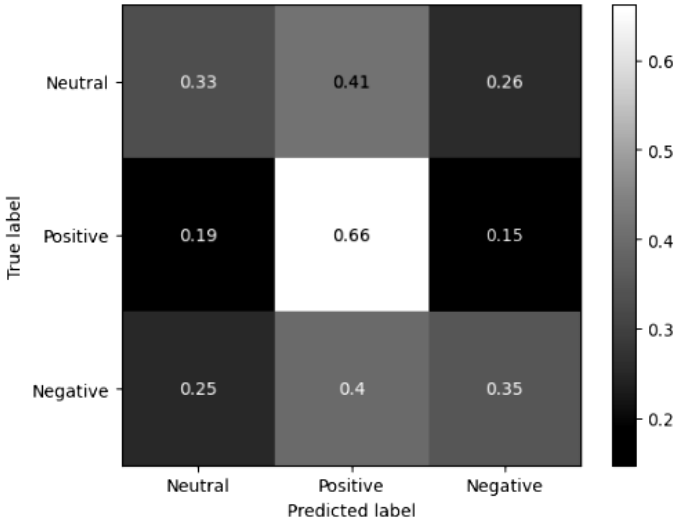


Fig. 8. Confusion matrix of Gaussian naive Bayesian classifier

Use of large language models for annotation. To work with large language models, we created the ClassifySentiment.ipynb file [14]. For the experiment, we chose the GPT-3.5-turbo model developed by OpenAI. It can be used using the API. The following restrictions were also used: data were taken only for May and June of 2023; duplicates were removed; location values were processed and tweets from Ukraine were removed.

Next, we created a dictionary called mySentiment and the functionality of working with the OpenAI API, the implementation of which is also given in [14]. The next step was to create functionality for sending queries to the GPT-3.5-turbo model, as well as the process of updating the dictionary. As a result, we obtained a dictionary with 59470 entries annotated with a large language model. They were saved and compared with the results of the multilayer perceptron. The results of the comparisons are presented in the file "ModelingSentiment.ipynb" on GitHub [14].

The GPT-3.5-turbo model demonstrates the best result of all the models studied. This is an expected result, but it demonstrates that the large language model GPT-3.5-turbo can effectively solve many problems with a fairly simple functionality using the OpenAI API.

RESULTS

The final stage of the experiment was the analysis of the results of public opinion classification based on the GPT-3.5-turbo model. First, the change in the distribution of forecasts for May and June 2023 was demonstrated. The results are shown in Fig. 9.

The public opinion score for Ukraine has both positive and negative ratings. The positive assessments are shown in Fig. 10. The negative, neutral, and positive grades are labeled as 0, 1, and 2, respectively. Thus, the average score from a country can range from 0 to 2.

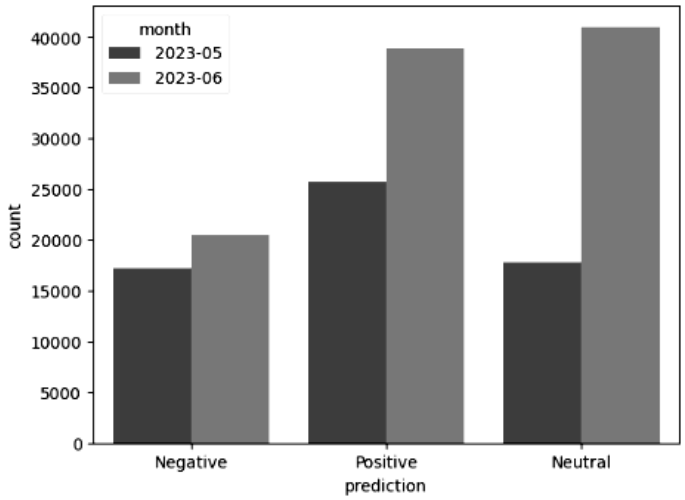


Fig. 9. Classification results

To improve the overall sentiment towards Ukraine, we need to analyze the factors contributing to negative public opinion and eliminate them. We can also highlight the positive aspects of the country, such as its rich culture, history, picturesque nature, and tourist attractions. By promoting these aspects, we can create a positive image of the country and improve its sentiment.

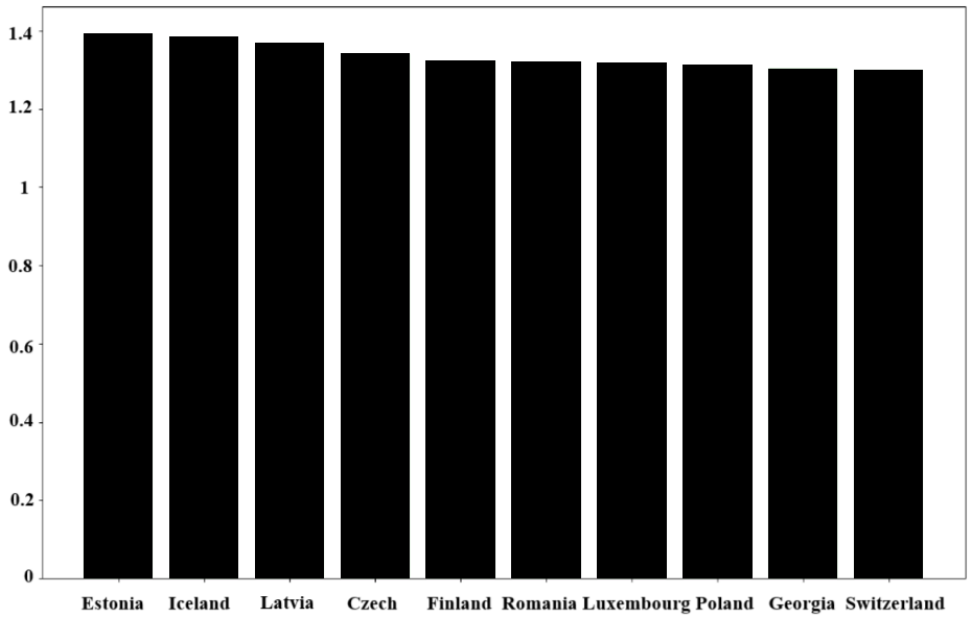


Fig. 10. Countries with the most positive attitude toward Ukraine

DISCUSSION

The results of classification using fairly simple models - Gaussian naive Bayesian classifier and multilayer perceptron - do not allow us to conclude that they can be practically used to process large data sets required for public opinion analysis. At the same time, the results of our work show that large language models with an

open API can be effectively used both for analyzing public opinion and for solving other applied problems. To obtain qualitative results, an important step was taken in collecting and preparing a large amount of data, which amounted to an impressive 44 million tweets from the social network Twitter (X). This amount of data was one of the key indicators for determining an objective assessment of public opinion in many partner countries regarding the war in Ukraine.

As part of this work, a visualization of the results of the analysis of public opinion and sentiment regarding the war in Ukraine was created for a better and deeper understanding.

CONCLUSIONS

The results of the work show that its goal of studying the effectiveness of using algorithms and methods of natural language processing based on artificial neural network models to improve the efficiency of studying and analyzing public opinion in Ukraine's partner countries has been achieved. All the research and development tasks were solved, namely: a large amount of data was collected and prepared, which amounted to 44 million messages from the social network Twitter (X); two models were used to analyze the data: a Gaussian naive Bayesian classifier and a multilayer perceptron, and their comparative analysis was conducted, which made it possible to find out the advantages and features of each model; an experiment was conducted using a large language model GPT-3.5-turbo, which demonstrated the high efficiency of its extraction. In addition, to improve the evaluation of the analysis results, we created their visualization.

REFERENCES

1. C. Bergstrom, J. West, *Calling Bullshit: The Art of Skepticism in a Data-Driven World*. Random House, 2020, 336 p.
2. Ł. Wordliczek, "Neural Networks and Political Science: Testing the Methodological Frontiers," *Revista de Metodología de Ciencias Sociales*, no. 57, pp. 37–62, 2023. Accessed on: January 3, 2024. [Online]. Available: https://revistas.uned.es/ index.php/empiria/article/view/36429/27239
3. M. Gong, "Analysis of Internet Public Opinion Popularity Trend Based on a Deep Neural Network," *Computational Intelligence and Neuroscience*, vol. 2022, article ID 9034773, 2022, 11 p. Accessed on: January 3, 2024. [Online]. Available: https://www.hindawi.com/journals/cin/2022/9034773/
4. S. Yang, "Analysis of Network Public Opinion in New Media Based on BP Neural Network Algorithm," *Mobile Information Systems*, vol. 2022, article ID 3202099, 2022, 9 p. Accessed on: January 3, 2024. [Online]. Available: https://www.hindawi.com/journals/misy/2022/3202099/
5. X. Chen, S. Duan, S. Li, et al., "A method of network public opinion prediction based on the model of grey forecasting and hybrid fuzzy neural network," *Neural Comput & Applic*, vol. 35, pp. 24681–24700, 2023. doi: https://doi.org/10.1007/ s00521-023-08205-9
6. Y. Parzhin, V. Kosenko, A. Podorozhniak, et al., "Detector neural network vs connectionist ANNs," *Neurocomputing, Elsevier*, vol. 414, pp. 191–203, 2020. doi: 10.1016/j.neucom.2020.07.025
7. R. Kysliy, "Artificial intelligence in natural language: language models, low-resource languages and dehumanization in texts," *Forum of GlobalLogic company*, *2023*. Accessed on: January 3, 2024. [Online]. Available: https://dou.ua/forums/topic/45583/

8. *OpenAI: Home page of the openAI platform*. 2023. Accessed on: January 3, 2024. [Online]. Available: https://platform.openai.com/overview
9. A. Amri, "OpenAI GPT For Python Developers: The art and science of developing intelligent apps with OpenAI GPT-3, DALL·E 2, CLIP, and Whisper - Suitable for learners of all levels," *Leanpub*, 2023, 236 p. Accessed on: January 3, 2024. [Online]. Available: https://leanpub.com/openaigptforpythondevelopers/
10. L. Tunstall, L. Werra, and T. Wolf, *Natural Language Processing with Transformers: Building Language Applications with Hugging Face*. O'Reilly Media, Inc., 2022, 406 p.
11. S. Hakimov, G. Cheema, *Unveiling Global Narratives: A Multilingual Twitter Dataset of News Media on the Russo-Ukrainian Conflict*. 2023. Accessed on: January 3, 2024. [Online]. Available: https://arxiv.org/pdf/2306.12886.pdf
12. "Ukraine Conflict Twitter Dataset," *Kaggle*, 2023. Accessed on: January 3, 2024. [Online]. Available: https://www.kaggle.com/datasets/bwandowando/ukraine-russian-crisis-twitter-dataset-1-2-m-rows/data
13. Y. Liu, M. Ott, N. Goyal, et al., "RoBERTa: A Robustly Optimized BERT Pretraining Approach," *T*he *Hugging Face Community*. Accessed on: January 3, 2024. [Online]. Available: https://huggingface.co/docs/transformers/model_doc/roberta
14. K. Perevoznyk, "Social Sentiment Analysis," *GitHub*, 2023. Accessed on: January 3, 2024. [Online]. Available: https://github.com/Kyrylopara/Social-Sentiment-Analysis

Received 05.02.2024

INFORMATION ON THE ARTICLE

Kyrylo M. Perevoznyk, ORCID: 0009-0009-2327-1501, Educational and Scientific Institute of Computer Science and Information Technology of the National Technical University "Kharkiv Polytechnic Institute", Ukraine, e-mail: kyrylo.perevoznyk@cs.khpi.edu.ua

Yurii V. Parzhyn, ORCID: 0000-0001-5727-1918, Educational and Scientific Institute of Computer Science and Information Technology of the National Technical University "Kharkiv Polytechnic Institute", Ukraine, e-mail: yurii.parzhyn@khpi.edu.ua

ЗАСТОСУВАННЯ ТЕХНОЛОГІЇ НЕЙРОННИХ МЕРЕЖ ДЛЯ АНАЛІЗУ СУСПІЛЬНОЇ ДУМКИ / К.М. Перевозник, Ю.В. Паржин

Анотація. Присвячено дослідженню та використанню технологій нейронних мереж, зокрема алгоритмів та методів оброблення природної мови, для підвищення ефективності вивчення та аналізу суспільної думки країн-партнерів України щодо війни в Україні. У ході виконання досліджень проаналізовано й опрацьовано бази даних, що складалися з повідомлень стосовно війни в Україні у соціальній мережі Twitter. Отримані датасети використано для навчання декількох моделей нейронних мереж. Найкращі результати класифікації отримано на моделі GPT-3.5-turbo. Для більш глибокого розуміння результатів аналізу суспільної думки створено їх візуалізацію. Результати дослідження показали високу ефективність обраних рішень і можуть мати важливу практичну значущість для поліпшення методів аналізу суспільної думки та прийняття обґрунтованих рішень на основі глибокого розуміння глобальних відгуків.

Ключові слова: суспільна думка, нейронні мережі, оброблення природної мови, великі мовні моделі, соціальні мережі, класифікація.

Received 05.02.2024

UDC 621.372.061:391.266
DOI: 10.20535/SRIT.2308-8893.2024.4.09

NEW APPROACH TO FINDING EIGENVECTORS FOR REPEATED EIGENVALUES OF A MATRIX

A.I. PETRENKO

Abstract. An efficient method of calculating eigenvectors for multiple eigenvalues of a matrix is proposed. This method is based on a formalized transformation of the problem of solving degenerate systems of equations into a regular problem by "repairing" their matrices and correspondingly correcting the right-hand sides of the equations, as well as "exclusion" during calculations from the spectrum eigenvalues of the matrix of one of the multiple values. In the case of non-defective multiples of the matrix, orthogonal eigenvectors are formed in contrast to the results obtained using the Mathematica program.

Keywords: eigenvectors, multiples of eigenvalues, algebraic and geometric multiplicity, solutions of degenerate systems, change of spectrum of a matrix, defective and non-defective multiples of a matrix.

INTRODUCTION

Finding the eigenvectors x_i for multiple eigenvalues λ_i of the matrix A is the least formalized task of modern Linear Algebra, as it is related to the solution of homogeneous (degenerate) systems of equations that have an infinite number of solutions:

$$B_i x_i = (A - \lambda_i E) x_i = 0 \quad \text{for} \quad i = 1, \ldots, n, \tag{1}$$

since by definition the eigenvectors cannot be zero even for zero eigenvalues. An eigenvector corresponding to an eigenvalue creates an eigenspace associated with it. The set of all eigenvectors for different eigenvalues forms the vector space of the matrix.

It is well known that equation (1) has nonzero solutions for the vector x_i if and only if the matrix $(A - \lambda_i E)$ has a zero determinant, which determines the characteristic polynomial of the matrix [1–3]. It can be used to find the eigenvalues of the matrix (for small order tasks) together with a more powerful QR algorithm with orthogonal Householder or Givens rotation matrices, which reduce the original matrix to a triangular matrix, on the diagonal of which there are real eigenvalues or 2×2 blocks of eigenvalues (for large-scale tasks).

The article attempts to formalize the procedure for solving equation (1), excluding the traditional manual selection of individual components of the solution vector x_i in order to eliminate the degeneracy of the problem. Special attention is paid to the case of multiple eigenvalues, for which equation (1) has the same form, but there can be both different and coincident solutions. In the latter case, the rank of the matrix, which is determined by the number of independent eigenvectors, is lower than its order, so such multiples of eigenvalues are called "defective" [4–6].

The analysis of publications showed that there is great uncertainty in the issue of finding eigenvectors for multiples of the matrix, the Internet is full of requests from specialists of different countries for help and consultations [7–11] and educational materials [12–14]. This motivated the conduct of own research, the results of which formed the basis of this article.

STATE OF AFFAIRS

The existing method of solving the problem of finding eigenvectors x_i for multiple eigenvalues λ_i of the matrix A is best considered using some examples, say, the matrix

$$\begin{bmatrix} 5 & 4 & 2 \\ 4 & 5 & 2 \\ 2 & 2 & 2 \end{bmatrix}, \tag{2}$$

which has a spectrum of eigenvalues $\lambda_1 = 10$ and $\lambda_2 = \lambda_3 = 1$.

When choosing an eigenvalue $\lambda_2 = 1$, the degenerate system of equations (1) is reduced to the form that demonstrates the relationship of all three equations:

$$B_2 x_2 = \begin{bmatrix} 4 & 4 & 2 \\ 4 & 4 & 2 \\ 2 & 2 & 1 \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \\ v_3 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}. \tag{3}$$

The system of equations (2) is transformed into the following form by the "row-reduction" procedure (i.e., reduction to a normal trapezoidal form):

$$\begin{bmatrix} 1 & 1 & 0.5 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \\ v_3 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}, \tag{4}$$

from which the following expression is formed

$$\begin{bmatrix} v_1 \\ v_2 \\ v_3 \end{bmatrix} = \begin{bmatrix} -v_2 - 0.5v_3 \\ v_2 \\ v_3 \end{bmatrix} = v_2 \begin{bmatrix} -1 \\ 1 \\ 0 \end{bmatrix} + v_3 \begin{bmatrix} -0.5 \\ 0 \\ 1 \end{bmatrix}.$$

Then the eigenvectors corresponding to the eigenvalues $\lambda_2 = \lambda_3 = 1$ have the form:

$$x_2 = \alpha \begin{bmatrix} -1 \\ 1 \\ 0 \end{bmatrix} + \beta \begin{bmatrix} -0.5 \\ 0 \\ 1 \end{bmatrix}, \quad |\alpha| + |\beta| \neq 0. \tag{5}$$

The condition $|\alpha|+|\beta|\neq 0$ excludes the selection of a zero eigenvector. At the same time, α and β can take any values, since the degenerate equations (3) have many solutions. By choosing different α and β, using (5), different values of x_2 are obtained and checked whether they satisfy the basic equation

$$Ax_i = \lambda_i x_i . \tag{6}$$

For example, choosing, say, $\alpha = 0$ and $\beta = 1$, we get an eigenvector

$$x_2 = \{-0.5, 0, 1\}, \tag{7}$$

which satisfies (6). On the contrary, choosing $\alpha = 1$ and $\beta = 0$, we get a solution

$$x_3 = \{-1, 1, 0\}, \tag{8}$$

which also satisfies the basic equation (6).

It is interesting that when choosing $\alpha = 1$ and $\beta = 1$, we get according to (5), an eigenvector

$$x_4 = \{-1.5,\ 1,\ 1\},$$

which also satisfies equation (6) and is independent with respect to vectors x_2 and x_3. But they are not all orthogonal because $x_2 x_3 = 0.5$, $x_3 x_4 = 2.5$ and $x_2 x_4 = 1.75$.

In the case of multiple eigenvalues of the matrix, two of them are chosen from the set of independent solutions obtainable from (5) (eg, x_2 and x_3) as the corresponding eigenvectors for the multiple eigenvalues. It is clear that when the selected eigenvectors are multiplied by an arbitrary scale coefficient *m*, the basic equation (6) continues to be fulfilled.

For the completeness of the picture, one more method should be mentioned, which recommends, in the case of repeated eigenvalues, to use instead of equation (1) its modification [1, 2]

$$B_i^k x_i = (A - \lambda_i E)^k x_i = 0,$$

where k is an indicator of the algebraic multiplicity of an eigenvalue, and the set of solutions x_i for $k > 1$ corresponds to the so-called root eigenvectors. But, based on equation (3), it can be shown that this is rather a delusion. Indeed, for $k = 2$ we get instead of (3) the expression:

$$B_2^2 x_2 = \begin{bmatrix} 36 & 36 & 18 \\ 36 & 36 & 18 \\ 18 & 18 & 9 \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \\ v_3 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix},$$

which is transformed by the "row-reduction" procedure to the already known equation (4)

$$\begin{bmatrix} 1 & 1 & 0.5 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \\ v_3 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix},$$

what indicates that the eigenvectors for multiple eigenvalues can be chosen from the set of solutions of equation (4), as was done above. A similar result is maintained if the multiplicity index k is increased.

Most likely, the considered selection of solutions is implemented in the well-known Mathematica program (the algorithmic support of which, unfortunately, is not described in detail in its documentation), because with its help (Fig. 1) for multiples of $\lambda_2 = \lambda_3 = 1$ of the matrix (2) eigenvectors $x_2 = \{-1, 0, 2\}$, $x_3 = \{-1, 1, 0\}$ can be obtained, which coincide with the accuracy of the coefficient with the values (7) and (8), which were previously manually selected when solving the system (4).

```
A={{5,4,2},{4,5,2},{2,2,2}};
{vals, vecs} = Eigensystem[A]
{{10, 1, 1}, {{2, 2, 1}, {-1, 0, 2}, {-1, 1, 0}}
```

Fig. 1. A fragment of the Mathematica code

It is interesting to compare the results of calculations obtained with the help of the Mathematica for two matrices that have the same spectrum of eigenvalues, but the multiples of the second one are defective (Fig. 2).

```
A1={{3,0,1},{0,3,2},{0,0,1}};
{vals, vecs} = Eigensystem[A1]
{{3, 3, 1}, {{0, 1, 0}, {1, 0, 0}, {-1, -2, 2}}}
```

a

```
A2={{3,1,1},{0 ,3,2}, {0,0,1}};
{vals, vecs} = Eigensystem[A2]
{{3, 3, 1}, {{1, 0, 0}, {0, 0, 0}, {0, -1, 1}}}
```

b

Fig. 2. Eigenvectors of two matrices with the same eigenvalues

The defect of a multiple eigenvalues matrix A2 is reflected in the Mathematica results by generation of a **zero eigenvector** (Fig. 2, *b*), what can mislead beginners who suspect an error in the program's operation.

But the Mathematica, unfortunately, sometimes contradicts itself, because it is enough to use the another its operator *JordanDecomposition*[A], related also to the calculation of eigenvectors and eigenvalues, and to find out with surprise that the same matrix A2 now has different eigenvectors $x_2 = \{1, 0, 0\}$ and $x_3 = \{0, 1, 0\}$ for the same multiple $\lambda_2 = \lambda_3 = 3$ (Fig. 3).

```
A2={{3,1,1},{0 ,3,2}, {0,0,1}};
JordanDecomposition [A2]
{{{0, 1, 0}, {-1, 0, 1}, {1, 0, 0}}, {{1, 0, 0}, {0, 3, 1}, {0, 0, 3}}}
```

/MatrixForm = $\begin{pmatrix} 0 & -1 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix} \quad \begin{pmatrix} 1 & 0 & 0 \\ 0 & 3 & 1 \\ 0 & 1 & 3 \end{pmatrix}$

Fig. 3. The Jordanian normal form of the matrix A2

But the obtained value of x_2 does not satisfy the basic equation (6). In addition, the eigenvector $x_1 = \{0, 1, 0\}$ for $\lambda_1 = 1$ differs from its value $x_1 = \{0, -1, 1\}$ shown before in Fig. 2, and also does not satisfy equation (6).

Since the algorithmic core of the Mathematica is also used in other well-known calculation programs (Matlab, Mathcad, Maple), the results of their application to calculating eigenvectors for multiple eigenvalues will be similar.

THE PROPOSED METHOD

The paper contains a procedure for generating **orthogonal vectors** for multiple non-defective eigenvalues, what does not interfere with Mathematica, and two depended vectors for defective multiples. In this case, the solution of degenerate systems of type (3) is formalized by diagonal correction of the systems matrix after "row-reduction" (4) with a simultaneous correction of the zero vector of the right side of the system.

The system's degeneration (4) is manifested by zero *k*-th diagonal elements in its matrix. Similar to the method of diagonal correction [3], this matrix is "repaired" (so that degeneracy is eliminated) by replacing zero diagonal elements with a number equal to one or by some constant *g*, which is chosen to be equal to the middle value of the elements of the matrix B row. Then the solution is ongoing with the already ingenerated matrix and the new right-hand side $b_1 = \{0,...1,...0\}$, represented by a transposed vector of dimension n, such that all elements are zero and only *k*-th elements are equal to one.

However, if at the same time there is a zero column and row in the matrix of equations (4), then only the position of the diagonal element of the column is adjusted in the vector of the right part.

Let us illustrate what has been said with the example of the degenerate system of equations (4):

$$\begin{bmatrix} 1 & 1 & 0.5 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \\ v_3 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \Rightarrow \begin{bmatrix} 1 & 1 & 0.5 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \\ v_3 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \\ 1 \end{bmatrix}. \quad (9)$$

The zero second and third diagonal elements of the zero rows of the original matrix are corrected by introducing the constant $g_2 = g_3 = 1$ and an additional vector of the right part b_2 is formed. From the solution of the adjusted system, we get

$$x_2 = \{-0.5, 0, 1\} \quad (10)$$

or normalized value

$$X2 = Normalize[\mathrm{N}[x_2]] = \{-0.447214, 0., 0.894427\}. \quad (11)$$

The eigenvector x_1 of the matrix for $\lambda_1 = 10$ is found quite similarly. In this case, the system of equations (1) looks like:

$$B_1 x_1 = \begin{bmatrix} -5 & 4 & 2 \\ 4 & 5 & 2 \\ 2 & 2 & -8 \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \\ v_3 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

and by the "row-reduction" procedure it is transformed into the following form:

$$\begin{bmatrix} 1 & 0 & 2 \\ 0 & 1 & -2 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \\ v_3 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}. \quad (12)$$

Unlike the matrix of the system of equations (4), the matrix of the system (12) has only one zero row, so its correction is performed differently:

$$\begin{bmatrix} 1 & 0 & 2 \\ 0 & 1 & -2 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \\ v_3 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \Rightarrow \begin{bmatrix} 1 & 0 & 2 \\ 0 & 1 & -2 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \\ v_3 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}. \quad (13)$$

As a result, we obtain the solution of the adjusted system (13)

$$x_1 = \{2, 2, 1\} \quad (14)$$

or normalized value

$$X1 = Normalize[N[x_1]] = \{0.666667,\ 0.666667,\ 0.333333\}. \quad (15)$$

Let us now consider an innovative procedure for finding the eigenvector for the second multiple of the eigenvalue $\lambda_2 = 1$ of the matrix. For this purpose, it is proposed to apply the following transformation of the matrix A, in which one of its multiple roots is excluded (zeroed), and then the problem is reduced to the previous one, when all the eigenvalues of the new matrix A1 are different. Such a transformation is performed according to the formula [3]:

$$\mathrm{A1} = A - \lambda_2 X_2 \otimes X_2 = A - \lambda_2 \cdot \mathrm{KroneckerProduct}[X_2, X_2], \quad (16)$$

where vector multiplication according to Kronecker, which results in a matrix, and the normalized eigenvector $X2$ (11) are used.

According to the formula (16) taking into account (11), we build the matrix

$$\mathrm{A1} = \begin{bmatrix} 4.8 & 4 & 2.4 \\ 4 & 5 & 2 \\ 2.4 & 2 & 1.2 \end{bmatrix},$$

for which the spectrum of eigenvalues $\lambda = \{10., 1., 3.8613110^{-16}\}$ does not contain multiples.

According to (1), we obtain a homogeneous system of equations

$$B_3 x_3 = \begin{bmatrix} 3.8 & 4 & 2.4 \\ 4 & 4 & 2 \\ 2.4 & 2 & 0.2 \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \\ v_3 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}. \quad (17)$$

Using the "row-reduction" procedure, the system of equations (17) is transformed and then is being "repaired" taking into account the fact that after the "row-reduction" procedure, only one zero row is formed in the matrix:

$$B_3 x_3 =$$

$$= \begin{bmatrix} 3.8 & 4 & 2.4 \\ 4 & 4 & 2 \\ 2.4 & 2 & 0.2 \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \\ v_3 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \Rightarrow \begin{bmatrix} 1 & 0 & -2 \\ 0 & 1 & 2.5 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \\ v_3 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \Rightarrow \begin{bmatrix} 1 & 0 & -2 \\ 0 & 1 & 2.5 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \\ v_3 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}. \quad (18)$$

As a result of the solution (18), we obtain the value of the second eigenvector

$$x_3 = \{2, -2.5, 1\} \tag{19}$$

or in normalized form

$$X3 = Normalize[N[x_3]] = \{0.596285, -0.745356, 0.298142\} \tag{20}$$

Thus, for the matrix (2) the following Eigensystem[A] is obtained by the proposed method

$$\{vals, vecs\} = \{\{10,1,1\}, \{\{2,2,1\}, \{-0.5,0,1\}, \{\mathbf{2}, -\mathbf{2.5}, \mathbf{1}\}\}\} \tag{21}$$

which differs from the results of Mathematica

$$\{vals, vecs\} = \{\{10, 1, 1\}, \{\{2, 2, 1\}, \{-1, 0, 2\}, \{-\mathbf{1}, \mathbf{1}, \mathbf{0}\}\}\} \tag{22}$$

presented in Fig. 1, by the value of the eigenvector for the second multiple eigenvalue $\lambda_3 = 1$.

The obtained eigenvectors given (10), (14) and (19) are **orthogona**l, since

$$x_1 x_3 = 0, \;\; x_1 x_2 = 0 \;\; \text{and} \;\; x_2 x_3 = 0.$$

If in the Mathematica's solution (22) we denote different components as $y_1 = \{2, 2, 1\}$, $y_2 = \{-1, 0, 2\}$ and $y_3 = \{-1, 1, 0\}$, then we can make sure, what

$$y_1 y_3 = 0, \;\; y_1 y_2 = 0, \;\; \text{but} \;\; y_2 y_3 = 1.$$

This means that these vectors y_i although they satisfy the corresponding basic equations (6), are **not orthogonal** and therefore, unlike the set of eigenvectors x_i from (21), cannot ensure unmistakably the canonical *JordanDecomposition*[A] operation for the matrix *A*, when

$$A = TDT^t, \tag{23}$$

where T is the orthogonal matrix of eigenvectors, and *D* is the diagonal matrix of all eigenvalues, including multiples. Indeed, using the normalized values of the obtained eigenvectors X1, X2 and X3 from the corresponding formulas (15), (11) and (20), it is possible to build

$$T = \begin{bmatrix} 0.666667 & -0.447214 & 0.596285 \\ 0.666667 & 0 & -0.745356 \\ 0.333333 & 0.894427 & 0.298142 \end{bmatrix}, \quad D = \begin{bmatrix} 10 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

and make sure that according to (23)

$$TDT^t = \{\{5.00001, 4., 2.\}, \{4., 5., 2.\}, \{2., 2., 2.\}\} = A.$$

For comparison, if you normalize the eigenvectors of the matrix *A* obtained with the help of Mathematica (22), you can get:

$$Y1 = Normalize[N[y_1]] = \{0.666667, 0.666667, 0.333333\},$$

$$Y2 = Normalize[N[y_2]] = \{-0.447214, 0., 0.894427\},$$

$$Y3 = Normalize[N[y_3]] = \{-0.707107, 0.707107, 0.\}.$$

and instead of the orthogonal matrix T construct another matrix

$$T_1 = \begin{bmatrix} 0.666667 & -0.447214 & -0.707107 \\ 0.666667 & 0 & -0.707107 \\ 0.333333 & 0.894427 & 0 \end{bmatrix},$$

with which we can check whether equation (23) is satisfied:

$$A^* = T_1 D T_1^t = \begin{bmatrix} 5.14445 & 3.94445 & 1.82222 \\ 3.94445 & 4.94445 & 2.22222 \\ 1.82222 & 2.22222 & 1.91111 \end{bmatrix} \neq A, \ .$$

while $\mathrm{Det}[T_1] = \mathrm{Det}[T_1^t] = -0.948684$ and $\mathrm{Det}[A^*] = 9.00001$ instead of 10.

The same erroneous Mathematica's result may be obtained by applying the standard JordanDecomposition[A] operator.

The obtained result calls into question the existing lemma that orthogonal eigenvectors correspond only to different simple eigenvalues [14], which was formulated, most likely, on the basis of practical results obtained with the help of the traditional selection of solutions of a homogeneous system of equations, considered above using the example of the system (3). A new approach with the exclusion of multiples and consideration of two homogeneous systems of equations provides new opportunities.

It seems interesting, using the method described above, to find the eigenvectors of the matrix A2 for its eigenvalue's spectrum $\lambda = \{3,3,1\}$ for which the Mathematica generates a solution with a *zero eigenvector* (Fig. 2).

For the first multiple eigenvalue $\lambda_1 = 3$, by analogy with the above example, instead of equation (9), we obtain the following expression

$$\begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} v_1 \\ v_1 \\ v_3 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \Rightarrow \begin{bmatrix} 1 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \\ v_3 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix},$$

from which we find $x_1 = \{1, 0, 0\}$.

Using the obtained value of x_1, which coincides with its normalized value of X1, to exclude, according to (16), one of the multiples of the matrix A2, we find the matrix A3

$$A3 = \begin{bmatrix} 0 & 1 & 1 \\ 0 & 3 & 2 \\ 0 & 0 & 1 \end{bmatrix},$$

which has a modified spectrum of eigenvalues $\lambda = \{0,3,1\}$ and for which, by analogy with (18), we construct an equation for finding the eigenvector of the second multiple $\lambda_2 = 3$ of the matrix A2:

$$\begin{bmatrix} 1 & -1/3 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \\ v_3 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \Rightarrow \begin{bmatrix} 1 & -1/3 & 0 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \\ v_3 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix},$$

from which we get $x_2 = \{1, 0, 0\}$.

As you can see, the values of x_1 and x_2 are linearly dependent (they just coincide), which indicates a defect in the multiple eigenvalues of the matrix.

CONCLUSIONS

One of the most important tasks of computational mathematics is the creation of effective and stable algorithms for finding the eigenvalues and vectors of a matrix [1]. They are a powerful tool that provides a deep understanding of matrix properties and opens wide perspectives for its application. Possession of this tool opens up opportunities for research and innovation in various fields of science and technology (for example, for identifying the main components and clustering of data during their analysis, for filtering signals and extracting a useful signal, for clustering and pattern recognition, etc.).

The state of affairs with the formalization of finding the eigenvectors of a matrix in general and for multiple eigenvalues in particular requires better. The article takes a certain step in this direction and proposes an innovative method of calculating eigenvectors for multiple eigenvalues of a matrix, which is based on the formalized transformation of the problem of solving degenerate systems of equations into a regular problem by "repairing" their matrices and by correspondingly correcting the right-hand sides of the equations, as well as "exclusion" of one of the multiple values from the spectrum of eigenvalues of the matrix during calculations of eigenvectors for multiples eigenvalues. In the case of non-defective multiples eigenvalues of the matrix, this method allows you to form orthogonal eigenvectors in contrast to the results obtained using the Mathematica.

REFERENCES

1. *Mathematics: Finding Eigenvectors with repeated Eigenvalues*. Available: https://math.stackexchange.com/questions/144798/finding-eigenvectors-with-repeated-eigenvalues
2. *Repeated Eigenvalues*. Available: https://ocw.mit.edu/courses/18-03sc-differential-equations-fall-2011/051316d5fa93f560934d3e410f8d153d_MIT18_03SCF11_s33_8text.pdf
3. L.P. Feldman, A.I. Petrenko, and O.A. Dmitrieva, *Numerical Methods in Computer Science: Textbook (in Ukrainian)*. Kyiv: BHV Publishing Group, 2006, 480 p. Available: https://library.kre.dp.ua/Books/2-4%20kurs/%D0%90%D0%BB%D0%B3%D0%BE%D1%80%D0%B8%D1%82%D0%BC%D0%B8%20%D1%96%20%D0%BC%D0%B5%D1%82%D0%BE%D0%B4%D0%B8%20%D0%BE%D0%B1%D1%87%D0%B8%D1%81%D0%BB%D0%B5%D0%BD%D1%8C/%D0%A4%D0%B5%D0%BB%D1%8C%D0%B4%D0%BC%D0%B0%D0%BD_%D0%A7%D0%B8%D1%81%D0%B5%D0%BB%D1%8C%D0%BD%D1%96_%D0%BC%D0%B5%D1%82%D0%BE%D0%B4%D0%B8_%D0%B2_%D1%96%D0%BD%D1%84%D0%BE%D1%80%D0%BC%D0%B0%D1%82%D0%B8%D1%86%D1%96_2007.pdf
4. *Real, Repeated Eigenvalues (Sect. 5.9) Review*. Michigan State University. Available: https://users.math.msu.edu/users/gnagy/teaching/13-summer/mth235/l31-235.pdf
5. *Repeated Eigenvalues*. Available: https://ocw.mit.edu/courses/18-03sc-differential-equations-fall-2011/051316d5fa93f560934d3e410f8d153d_MIT18_03SCF11_s33_8text.pdf

6. Jiry Lebl, *Multiple Eigenvalues*. Available: https://math.libretexts.org/Bookshelves/Differential_Equations/Differential_Equations_for_Engineers_(Lebl)/3%3A_Systems_of_ODEs/3.7%3A_Multiple_Eigenvalues
7. *How do I find an eigenvector matrix when eigenvalues are repeated?* Available: https://www.quora.com/How-do-I-find-an-eigenvector-matrix-when-eigenvalues-are-repeated
8. *Eigenvalues and Eigenvectors*. Available: https://sites.calvin.edu/scofield/courses/m256/materials/eigenstuff.pdf
9. Marco Taboga, "Linear independence of eigenvectors," *Lectures on matrix algebra*, 2021. Available: https://www.statlect.com/matrix-algebra/linear-independence-of-eigenvectors
10. *MATLAB Answers: How to identify repeated eigenvalues of a matrix?* Available: https://www.mathworks.com/matlabcentral/answers/395353-how-to-identify-repeated-eigenvalues-of-a-matrix
11. *Repeated eigenvalues -> crazy eigenvectors?* Available: https://www.reddit.com/r/matlab/comments/lsk3v7/repeated_eigenvalues_crazy_ eigenvectors/?rdt=37293
12. *Eigenvalues and Eigenvectors*. Available: https://personal.math.ubc.ca/~tbjw/ila/eigenvectors.html
13. *Eigenvalues and Eigenvectors*. Available: https://www.math.hkust.edu.hk/~mabfchen/Math111/Week11-12.pdf
14. *Eigenvectors, Eigenvalues, and Diagonalization* (*solutions*). Available: https://math.berkeley.edu/~mcivor/math54s11/worksheet2.28soln.pdf

Received 08.01.2024

INFORMATION ON THE ARTICLE

Anatolii I. Petrenko, ORCID: 0000-0001-6712-7792, Educational and Research Institute for Applied System Analysis of the National Technical University of Ukraine "Igor Sikorsky Kyiv Polytechnic Institute", Ukraine, e-mail: tolja.petrenko@gmail.com

НОВИЙ ПІДХІД ДО ПОШУКУ ВЛАСНИХ ВЕКТОРІВ ДЛЯ КРАТНИХ ВЛАСНИХ ЧИСЕЛ МАТРИЦІ / А.І. Петренко

Анотація. Запропоновано ефективний метод обчислення власних векторів для кратних власних чисел матриці, який базується на формалізованому перетворенню задачі розв'язання вироджених систем рівнянь у звичайну задачу шляхом «ремонту» їх матриць і відповідного корегування правих частин рівнянь, а також «вилучення» під час обчислень зі спектра власних чисел матриці одного з кратних значень. У випадку недефектних кратних чисел матриці формуються ортогональні власні вектори на відміну від результатів, отриманих за допомогою програми Mathematica.

Ключові слова: власні вектори, кратні власні числа, алгебрична і геометрична кратність, розв'язання вироджених систем, зміна спектра матриці, дефектні і недефектні кратні числа матриці.

UDC 517.58 + 512.643
DOI: 10.20535/SRIT.2308-8893.2024.4.10

CLASSICAL SPECIAL FUNCTIONS OF MATRIX ARGUMENTS

D.O. SHUTIAK, G.B. PODKOLZIN, V.G. BONDARENKO, Y.A. CHAPOVSKY

Abstract. This article focuses on a few of the most commonly used special functions and their key properties and defines an analytical approach to building their matrix-variate counterparts. To achieve this, we refrain from using any numerical approximation algorithms and instead rely on properties of matrices, the matrix exponential, and the Jordan normal form for matrix representation. We focus on the following functions: the Gamma function as an example of a univariate function with a large number of properties and applications; the Beta function to highlight the similarities and differences from adding a second variable to a matrix-variate function; and the Jacobi Theta function. We construct explicit function views and prove a few key properties for these functions. In the comparison section, we highlight and contrast other approaches that have been used in the past to tackle this problem.

Keywords: matrix, special function, matrix function, gamma function, beta function, Jacobi theta function, Jordan normal form.

INTRODUCTION

Data with matrix responses for each experiment are increasingly common in modern statistical problems. For example, observations over a time period can be viewed holistically as a matrix variable, labeling the rows and columns as time and actual measurements respectively. Temporal and spatial data, multivariate growth curve data, imaging data, and data from cross-sectional designs also generate matrix-valued responses. On the other hand, many of these phenomena are still often built on generalized cases of classical problems, many of which are solved, or at least interpreted or simplified, by special functions. Therefore, the motivation of the study was to combine these two parts and to do so as generally as possible analytically, without relying on a specific problem or purely numerical methods. There were earlier studies on this topic, but they were aimed at either generalizing a specific concept (Mitra S. 1970 [1]), or calculating values for certain classes of matrices needed for further calculations (Kishka Z., Saleem M. 2019 [2]). In this article, based on the theory of matrices, matrix exponents and using the Jordanian canonical form of matrices, we formulate a basic toolkit of definitions and key properties of special matrix-variate functions. These properties are applicable to the widest range of matrices and have an explicit form, that is, they can be used for further research with minimal changes.

First, the definition and key properties for the matrix Gamma function will be given, as an example of a univariate special function, followed by a series of two-variable special functions such as the Beta function and the Jacobi Theta function. A comparison of the obtained results with the existing methods mentioned above will also be made

UNIVARIATE SPECIAL FUNCTIONS

1 GAMMA FUNCTION

1.1 the definition and the general form of the matrix-variate gamma function

For the Gamma function and all subsequent special functions, the integral definition of functions was taken as the basis of the study. Specifically for the Gamma function and the Beta function, the following shift was also made to simplify the calculations:

Definition. For an arbitrary matrix A, we define

$$\Gamma(A)=\int_0^\infty x^{A-I}e^{-x}dx=\begin{matrix}\text{to simplify further}\\ \text{calculations replace } A-I=B\end{matrix}=\int_0^\infty x^{B}e^{-x}dx. \quad (2.1)$$

From this definition using the matrix, we get the following form for the matrix-variate Gamma function:

For an arbitrary matrix $A\in\mathbb{R}^{(k\times k)}$, which has the Jordanian canonical form $J=U^{-1}AU$ its' Gamma function will have the form:

$$\Gamma(A)=U\Gamma(J)U^{-1};$$

$$\Gamma(J)=\begin{pmatrix}\Gamma(J_{r_1}(\lambda_1)) & & & \\ & \Gamma(J_{r_2}(\lambda_2)) & & \\ & & \ddots & \\ & & & \Gamma(J_{r_l}(\lambda_l))\end{pmatrix} \text{ — block matrix,}$$

where

$$\Gamma(J_{r_j}(\lambda_j))=$$

$$=\begin{pmatrix}\Gamma(\lambda_j+1) & \frac{\Gamma'(\lambda_j+1)}{1!} & \frac{\Gamma''(\lambda_j+1)}{2!} & \cdots & \frac{\Gamma^{(r_j-2)}(\lambda_j+1)}{(r_j-2)!} & \frac{\Gamma^{(r_j-1)}(\lambda_j+1)}{(r_j-1)!}\\ 0 & \Gamma(\lambda_j+1) & \frac{\Gamma'(\lambda_j+1)}{1!} & & \cdots & \frac{\Gamma^{(r_j-2)}(\lambda_j+1)}{(r_j-2)!}\\ 0 & 0 & \Gamma(\lambda_j+1) & \ddots & \ddots & \vdots\\ \vdots & \vdots & \vdots & \ddots & \frac{\Gamma'(\lambda_j+1)}{1!} & \frac{\Gamma''(\lambda_j+1))}{2!}\\ 0 & 0 & 0 & \cdots & \Gamma(\lambda_j+1) & \frac{\Gamma'(\lambda_j+1)}{1!}\\ 0 & 0 & 0 & \cdots & 0 & \Gamma(\lambda_j+1)\end{pmatrix} \quad (2.2)$$

The blocks of the resulting matrix correspond to the blocks of each of the eigenvalues of the Jordan matrix J of the matrix A and have the corresponding dimensions $(r_i\times r_i)$. It should also be noted that these matrices are upper-triangular, that is, they have zero-values below the main diagonal. This fact will also be important for the subsequent special functions.

1.2 Main functional equation

One of the most important and used properties of the Gamma function is the functional equation, as several other properties of the Gamma function are based on it. Also, this property allows you to recursively find the values of the function, thereby significantly simplifying calculations.

For a scalar argument, the identity has the following form:

$$\Gamma(\lambda+1)=\lambda*\Gamma(\lambda). \tag{2.3}$$

To prove this statement in the matrix case, we first consider the following auxiliary equality:

$$=\begin{pmatrix} \lambda+1 & 1 & 0 & \dots & 0 \\ 0 & \lambda+1 & 1 & \ddots & \vdots \\ \vdots & 0 & \ddots & \ddots & 0 \\ \vdots & \vdots & \ddots & \lambda+1 & 1 \\ 0 & \dots & \dots & 0 & \lambda+1 \end{pmatrix} = J_{r_i}(\lambda+1). \tag{2.4}$$

Let us now use this and definition (2.2) to generalize identity (2.3):

$$\Gamma(J_{r_i}(\lambda_i)+I)=\underset{(2.4)}{\overset{using}{=}}\Gamma\left(J_{r_i}(\lambda_i+1)\right)=\underset{(2.2)}{\overset{using}{=}}=$$

$$=\begin{pmatrix} \Gamma(\lambda_j+2) & \frac{\Gamma'(\lambda_j+1)}{1!} & \frac{\Gamma''(\lambda_j+1)}{2!} & \dots & \frac{\Gamma^{(r_j-2)}(\lambda_j+1)}{(r_j-2)!} & \frac{\Gamma^{(r_j-1)}(\lambda_j+2)}{(r_j-1)!} \\ 0 & \Gamma(\lambda_j+2) & \frac{\Gamma'(\lambda_j+2)}{1!} & & \dots & \frac{\Gamma^{(r_j-2)}(\lambda_j+1)}{(r_j-2)!} \\ 0 & 0 & \Gamma(\lambda_j+2) & \ddots & \ddots & \vdots \\ \vdots & \vdots & \vdots & \ddots & \frac{\Gamma'(\lambda_j+2)}{1!} & \frac{\Gamma''(\lambda_j+2))}{2!} \\ 0 & 0 & 0 & \dots & \Gamma(\lambda_j+2) & \frac{\Gamma'(\lambda_j+2)}{1!} \\ 0 & 0 & 0 & \cdots & 0 & \Gamma(\lambda_j+2) \end{pmatrix}$$

Using the properties of the derivative of the Gamma function and the properties of the Gamma function itself:

$$\begin{pmatrix} \Gamma(\lambda_j+2) & \frac{\Gamma'(\lambda_j+1)}{1!} & \frac{\Gamma''(\lambda_j+1)}{2!} & \dots & \frac{\Gamma^{(r_j-2)}(\lambda_j+1)}{(r_j-2)!} & \frac{\Gamma^{(r_j-1)}(\lambda_j+2)}{(r_j-1)!} \\ 0 & \Gamma(\lambda_j+2) & \frac{\Gamma'(\lambda_j+2)}{1!} & & \dots & \frac{\Gamma^{(r_j-2)}(\lambda_j+1)}{(r_j-2)!} \\ 0 & 0 & \Gamma(\lambda_j+2) & \ddots & \ddots & \vdots \\ \vdots & \vdots & \vdots & \ddots & \frac{\Gamma'(\lambda_j+2)}{1!} & \frac{\Gamma''(\lambda_j+2))}{2!} \\ 0 & 0 & 0 & \dots & \Gamma(\lambda_j+2) & \frac{\Gamma'(\lambda_j+2)}{1!} \\ 0 & 0 & 0 & \cdots & 0 & \Gamma(\lambda_j+2) \end{pmatrix} =$$

$$= \begin{pmatrix} (\lambda_i+1)\Gamma(\lambda_i+1) & \dfrac{(\lambda_i+1)\Gamma'(\lambda_i+1)+\Gamma(\lambda_i+1)}{1!} & \cdots \\ 0 & (\lambda_i+1)\Gamma(\lambda_i+1) & \cdots \\ \vdots & \vdots & \ddots \\ 0 & 0 & 0 \end{pmatrix}$$

$$\begin{pmatrix} \cdots & \dfrac{(\lambda_i+1)\Gamma^{(r_i-1)}(\lambda_i+1)+(r_i-1)\Gamma^{(r_i-2)}(\lambda_i+1)}{(r_i-1)!} \\ \cdots & \vdots \\ \ddots & \dfrac{(\lambda_i+1)\Gamma'(\lambda_i+1)+\Gamma(\lambda_i+1)}{1!} \\ 0 & (\lambda_i+1)\Gamma(\lambda_i+1) \end{pmatrix}$$

Now we split the obtained matrix into two separate matrices, grouping all terms with the coefficient (λ_i+1) into the first, all others into the second:

$$\begin{pmatrix} (\lambda_i+1)\Gamma(\lambda_i+1) & \dfrac{(\lambda_i+1)\Gamma'(\lambda_i+1)}{1!} & \cdots & \dfrac{(\lambda_i+1)\Gamma^{(r_i-1)}(\lambda_i+1)}{(r_i-1)!} \\ 0 & (\lambda_i+1)\Gamma(\lambda_i+1) & \cdots & \vdots \\ \vdots & \vdots & \ddots & \dfrac{(\lambda_i+1)\Gamma'(\lambda_i+1)}{1!} \\ 0 & 0 & 0 & (\lambda_i+1)\Gamma(\lambda_i+1) \end{pmatrix} +$$

$$+ \begin{pmatrix} 0 & \dfrac{\Gamma((\lambda_i+1))}{1!} & \cdots & \dfrac{(r_i-1)\Gamma^{(r_i-2)}(\lambda_i+1)}{(r_i-1)!} \\ 0 & 0 & \cdots & \vdots \\ \vdots & \vdots & \ddots & \dfrac{\Gamma((\lambda_i+1))}{1!} \\ 0 & 0 & 0 & 0 \end{pmatrix};$$

subtract (λ_i+1) from the first term as a matrix and reduce the factorial and coefficient in the second:

$$(\lambda_i+1)\begin{pmatrix} \Gamma(\lambda_i+1) & \dfrac{\Gamma'(\lambda_i+1)}{1!} & \cdots & \dfrac{\Gamma^{(r_i-1)}(\lambda_i+1)}{(r_i-1)!} \\ 0 & \Gamma(\lambda_i+1) & \cdots & \vdots \\ \vdots & \vdots & \ddots & \dfrac{\Gamma'(\lambda_i+1)}{1!} \\ 0 & 0 & 0 & \Gamma(\lambda_i+1) \end{pmatrix} +$$

$$+ \begin{pmatrix} 0 & \dfrac{\Gamma((\lambda_i+1))}{1!} & \dfrac{\Gamma''((\lambda_i+1))}{2!} & \cdots & \dfrac{\Gamma^{(r_i-2)}(\lambda_i+1)}{(r_i-2)!} \\ 0 & 0 & \ddots & \ddots & \vdots \\ \vdots & \vdots & \ddots & \dfrac{\Gamma((\lambda_i+1))}{1!} & \dfrac{\Gamma''((\lambda_i+1))}{2!} \\ \vdots & \vdots & & 0 & \dfrac{\Gamma((\lambda_i+1))}{1!} \\ 0 & 0 & \cdots & 0 & 0 \end{pmatrix} =$$

$$= (\lambda_i + 1)\Gamma(J_{r_i}(\lambda_i)) + \Gamma(J_{r_i}(\lambda_i))J_{r_i}(0).$$

That is, we get the following identity:

$$\Gamma(J_{r_i}(\lambda_i) + I) = (\lambda_i + 1)\Gamma(J_{r_i}(\lambda_i)) + \Gamma(J_{r_i}(\lambda_i))J_{r_i}(0).$$

Generalizing for the matrix $A = U J_{r_i}(\lambda_i)U^{-1}$:

$$\Gamma(A + I) = (\lambda_i + 1)\Gamma(A) + \Gamma(A)U J_{r_i}(0)U^{-1}. \qquad (2.5)$$

As we can see, it was possible to prove a property similar to (2.3), but to which the correcting term $\Gamma(A)U J_{r_i}(0)U^{-1}$ is added. Indeed, if we take the dimension of the matrix A as (1×1), i.e. return to a scalar variable, then the second term of the identity will be equal to 0 and we will return to the widely-known identity (2.3).

1.3 Euler's reflection formula

Before moving on to the generalization of the reflection formula, we give an additional auxiliary property:

$$\Gamma(mJ_r(\lambda_i)) = \int_0^\infty e^{mJ_r(\lambda_i)\ln x}e^{-x}dx = \int_0^\infty \begin{pmatrix} e^{m\lambda_i \ln x} & \cdots & \dfrac{(m\ln x)^{r_i-1}e^{m\lambda_i \ln x}}{(r_i-1)!} \\ \vdots & \ddots & \vdots \\ 0 & \cdots & e^{m\lambda_i \ln x} \end{pmatrix} e^{-x}dx =$$

$$= \begin{pmatrix} \Gamma(m\lambda_i+1) & \dfrac{\Gamma'(m\lambda_i+1)}{1!} & \cdots & \dfrac{\Gamma^{(r_i-1)}(m\lambda_i+1)}{(r_i-1)!} \\ 0 & \Gamma(\lambda_i+1) & \cdots & \vdots \\ \vdots & \vdots & \ddots & \dfrac{\Gamma'(m\lambda_i+1)}{1!} \\ 0 & 0 & 0 & \Gamma(m\lambda_i+1) \end{pmatrix}. \qquad (3.9)$$

Now let's return to Euler's reflection formula:

$$\Gamma(J_r(\lambda_i))\Gamma(I_{r_i} - J_r(\lambda_i)) =$$

$$= \left| I_{r_i} - J_r(\lambda_i) = \begin{pmatrix} 1-\lambda_i & -1 & 0 & \cdots & 0 \\ 0 & 1-\lambda_i & -1 & \ddots & \vdots \\ \vdots & 0 & \ddots & \ddots & 0 \\ \vdots & \vdots & \ddots & 1-\lambda_i & -1 \\ 0 & \cdots & \cdots & 0 & 1-\lambda_i \end{pmatrix} = -J_{r_i}(\lambda_i - 1) \right| =$$

$$= \Gamma(J_{r_i}(\lambda_i))\Gamma(-J_{r_i}(\lambda_i+1)) \underset{(3.9)}{\overset{вл.}{=}} \begin{pmatrix} \Gamma(\lambda_i+1) & \dfrac{\Gamma'(\lambda_i+1)}{1!} & \cdots & \dfrac{\Gamma^{(r_i-1)}(\lambda_i+1)}{(r_i-1)!} \\ 0 & \Gamma(\lambda_i+1) & \cdots & \vdots \\ \vdots & \vdots & \ddots & \dfrac{\Gamma'(\lambda_i+1)}{1!} \\ 0 & 0 & 0 & \Gamma(\lambda_i+1) \end{pmatrix} \times$$

$$\times\begin{pmatrix} \Gamma(-\lambda_i+2) & \frac{\Gamma'(-\lambda_i+2)}{1!} & \dots & \frac{\Gamma^{(r_i-1)}(-\lambda_i+2)}{(r_i-1)!} \\ 0 & \Gamma(-\lambda_i+2) & \dots & \vdots \\ \vdots & \vdots & \ddots & \frac{\Gamma'(-\lambda_i+2)}{1!} \\ 0 & 0 & 0 & \Gamma(-\lambda_i+2) \end{pmatrix} =$$

$$= \Gamma(J_{r_i}(\lambda_i))\Gamma(J_{r_i}(1-\lambda_i))$$

Derivation of Euler's reflection formula:
Consider the following product:

$$\Gamma\left(\frac{1}{2}I_{r_i}+J_{r_i}(\lambda_i)\right)\Gamma\left(\frac{1}{2}I_{r_i}-J_{r_i}(\lambda_i)\right);$$

$$\frac{1}{2}I_{r_i}+J_r(\lambda_i)=\begin{pmatrix} \frac{1}{2}+\lambda_i & 1 & 0 & \dots & 0 \\ 0 & \frac{1}{2}+\lambda_i & 1 & \ddots & \vdots \\ \vdots & 0 & \ddots & \ddots & 0 \\ \vdots & \vdots & \ddots & \frac{1}{2}+\lambda_i & 1 \\ 0 & \dots & \dots & 0 & \frac{1}{2}+\lambda_i \end{pmatrix}=J_r\left(\lambda_i+\frac{1}{2}\right);$$

$$\frac{1}{2}I_{r_i}-J_r(\lambda_i)=\begin{pmatrix} \frac{1}{2}-\lambda_i & -1 & 0 & \dots & 0 \\ 0 & \frac{1}{2}-\lambda_i & -1 & \ddots & \vdots \\ \vdots & 0 & \ddots & \ddots & 0 \\ \vdots & \vdots & \ddots & \frac{1}{2}-\lambda_i & -1 \\ 0 & \dots & \dots & 0 & \frac{1}{2}-\lambda_i \end{pmatrix}=-J_r\left(\lambda_i-\frac{1}{2}\right).$$

$$\Gamma\left(\frac{1}{2}I_{r_i}+J_{r_i}(\lambda_i)\right)\Gamma\left(\frac{1}{2}I_{r_i}-J_{r_i}(\lambda_i)\right)=\Gamma\left(J_r\left(\lambda_i+\frac{1}{2}\right)\right)\Gamma\left(-J_r\left(\lambda_i-\frac{1}{2}\right)\right)=$$

$$\begin{pmatrix} \Gamma\left(\lambda_i+\frac{3}{2}\right) & \frac{\Gamma'\left(\lambda_i+\frac{3}{2}\right)}{1!} & \dots & \frac{\Gamma^{(r_i-1)}\left(\lambda_i+\frac{3}{2}\right)}{(r_i-1)!} \\ 0 & \Gamma\left(\lambda_i+\frac{3}{2}\right) & \dots & \vdots \\ \vdots & \vdots & \ddots & \frac{\Gamma'\left(\lambda_i+\frac{3}{2}\right)}{1!} \\ 0 & 0 & 0 & \Gamma\left(\lambda_i+\frac{3}{2}\right) \end{pmatrix}\times$$

$$\times\begin{pmatrix} \Gamma\left(-\lambda_i+\frac{1}{2}\right) & \frac{\Gamma'\left(-\lambda_i+\frac{1}{2}\right)}{1!} & \cdots & \frac{\Gamma^{(r_i-1)}\left(-\lambda_i+\frac{1}{2}\right)}{(r_i-1)!} \\ 0 & \Gamma\left(-\lambda_i+\frac{1}{2}\right) & \cdots & \vdots \\ \vdots & \vdots & \ddots & \frac{\Gamma'\left(-\lambda_i+\frac{1}{2}\right)}{1!} \\ 0 & 0 & 0 & \Gamma\left(-\lambda_i+\frac{1}{2}\right) \end{pmatrix} = \Gamma(J_{r_i}(\lambda_i))\Gamma(J_{r_i}(1-\lambda_i)) =$$

$$= \begin{pmatrix} \Gamma\left(\lambda_i+\frac{3}{2}\right)\times & & & \swarrow\begin{pmatrix} i=0 \\ j=r-1 \end{pmatrix} \\ \Gamma\left(-\lambda_i+\frac{1}{2}\right) & \cdots & \cdots & \\ 0 & \Gamma\left(\lambda_i+\frac{3}{2}\right)\times & \sum_{k=0}^{j-i}\frac{\Gamma^{(k)}\left(\lambda_i+\frac{3}{2}\right)\Gamma^{(r-1-i-k)}\left(-\lambda_i+\frac{1}{2}\right)}{k!(r-1-i-k)!} & \vdots \\ & \Gamma\left(-\lambda_i+\frac{1}{2}\right) & & \\ \vdots & \ddots & \Gamma\left(\lambda_i+\frac{3}{2}\right)\times & \vdots \\ & & \Gamma\left(-\lambda_i+\frac{1}{2}\right) & \\ & & & \Gamma\left(\lambda_i+\frac{3}{2}\right)\times \\ 0 & \cdots & 0 & \Gamma\left(-\lambda_i+\frac{1}{2}\right) \end{pmatrix}$$

2 BETA FUNCTION

2.1 Definition of the matrix-variate beta function

Similarly to the previous subsection, let's start with the definition of the matrix-variate Beta function, using the integral definition of the Beta function:

$$B(x,y)=\int_0^1 t^{x-1}(1-t)^{y-1}dt.$$

For two matrices $X,Y\in\mathbb{R}^{(k\times k)}$ with Jordan canonical forms $X=U_1J_1U_1^{-1}$; $Y=U_2J_2U_2^{-1}$ we consider the function of two matrix variables. We will first perform the following calculations for their Jordan matrices, respectively:

$$B(J_r(\lambda_1),J_r(\lambda_2))=\int_0^1 e^{\ln(t)J_r(\lambda_1)}e^{\ln(1-t)J_r(\lambda_2)}dt$$

Now we present and analyze the integral product separately:

$$e^{\ln(t)J_r(\lambda_1)}e^{\ln(1-t)J_r(\lambda_2)} = \begin{pmatrix} e^{\lambda_1\ln(t)}e^{\lambda_2\ln(1-t)} & \dots & M_{(l,m)} \\ \vdots & \ddots & \vdots \\ 0 & \dots & e^{\lambda_1\ln(t)}e^{\lambda_2\ln(1-t)} \end{pmatrix},$$

where $M_{(l,m)}$ is an arbitrary product of the l-th row and m-th column of the initial matrices, and $m \geq l$:

$$M_{(l,m)} = \sum\left(0 \quad \dots \quad 0 \quad e^{\lambda_1\ln(t)}e^{\lambda_2\ln(1-t)}\frac{(\ln(1-t))^{m-l}}{((m-l)!} \quad \dots \quad 0 \quad \dots \quad 0\right).$$

The number of zeros at the beginning and at the end of the product is l and $k-m-l$, respectively, so the resulting sum will consist of the middle part of the vector:

$$M_{(l,m)} = \sum_{j=0}^{m-l}\frac{(\ln(t))^j(\ln(1-t))^{(m-l)-j}}{j!((m-l)-j)!}e^{\lambda_1\ln(t)}e^{\lambda_2\ln(1-t)}.$$

Then, returning to the integral, we get the following:

$$\int_0^1 e^{\lambda_1\ln(t)}e^{\lambda_2\ln(1-t)}dt = B(\lambda_1+1,\lambda_2+1) \text{ — elements on the main diagonal;}$$

$$\int_0^1\sum_{j=0}^{m-l}\frac{(\ln(t))^j(\ln(1-t))^{(m-l)-j}}{j!((m-l)-j)!}e^{\lambda_1\ln(t)}e^{\lambda_2\ln(1-t)} =$$

$$= \begin{pmatrix} B(\lambda_1+1,\lambda_2+1) & \dots & \sum_{j=0}^{m-l}\frac{1}{j!((m-l)-j)!}\frac{\partial^{(m-l)}B(x,y)}{\partial x^j\partial y^{(m-l)-j}}\Big|_{\substack{x=\lambda_1+1\\ y=\lambda_2+1}} \\ \vdots & \ddots & \vdots \\ 0 & \dots & B((m-l)-j) \end{pmatrix}.$$

It should be noted that the resulting matrix, namely an arbitrary element in the form of the sum $M_{(l,m)}$ and the corresponding resulting sums of derivatives depend only on the difference of indices (l,m), and not on each of them separately. This, in turn, means that these elements are equal to each other for m and l on the corresponding diagonals, which significantly reduces the number of necessary calculations for finding the explicit form of the matrix $B(,)$ for specific values.

Now let us return to the initial general definition for arbitrary matrices X, Y:

$$B(X,Y) = \int_0^1 e^{\ln(t)X}e^{\ln(1-t)Y}dt = \begin{matrix}\text{using the Jordan} \\ \text{decomposition}\end{matrix} =$$

$$= \int_0^1 U_1e^{\ln(t)J_1}U_1^{-1}U_2e^{\ln(1-t)J_2}U_2^{-1}dt;$$

Compared to the situation with functions of one variable (e.g. Gamma function), when we start working with functions of several matrix variables, we have two different Jordan transformations, and that is, two different matrices U_1, U_2, which greatly complicates the task and makes it impossible to establish a direct relationship between $B(X,Y)$ and $B(J_1,J_2)$ to obtain a clear analytical view of the resulting matrix. In this regard, it is advisable to further consider the matrices X, Y as a pair of commuting matrices, which will give us the opportunity to find a common Jordan basis for them, i.e. $U_1 = U_2$. Also, in several points, it will allow the use of properties of the matrix exponent only for commuting matrices. Therefore, taking this into account, we get the following result:

$$B(X,Y) = \int_0^1 U e^{\ln(t)J_1} U^{-1} U e^{\ln(1-t)J_2} U^{-1} dt == U \int_0^1 e^{\ln(t)J_1} e^{\ln(1-t)J_2} dt\, U^{-1} =$$

$$= UB(J_1,J_2)U^{-1}.$$

2.2 Certain properties of the Beta function

1) Symmetry: $B(x,y) = B(y,x)$

Since commuting matrices were chosen for research from the previous point, symmetry for matrix arguments is also preserved.

2) Partial case of the function $B(1,x) = 1/x$.

For the matrix-variate function, let's start with $J_r(\lambda_1)$:

$$B(I,J_r) = \int_0^1 e^{\ln(1-t)J_r(\lambda_1)} dt$$

In this case, we have a single matrix of the form:

$$e^{\ln(1-t)J_r(\lambda_1)} = \begin{pmatrix} e^{\lambda_1 \ln(1-t)} & \dots & \dfrac{(\ln(1-t))^{k-1} e^{\lambda_1 \ln(1-t)}}{(k-1)!} \\ \vdots & \ddots & \vdots \\ 0 & \dots & e^{\lambda_1 \ln(1-t)} \end{pmatrix};$$

Then, with the absence of a product, we go directly to the integral:

$$B\big(I,J_r(\lambda_1)\big) = \begin{pmatrix} B(1,\lambda_1+1) & \dots & \dfrac{1}{(k-1)!} \dfrac{\partial^{(k-1)} B(x,y)}{\partial y^{k-1}}\Big|_{\substack{x=1 \\ y=\lambda_1+1}} \\ \vdots & \ddots & \vdots \\ 0 & \dots & B(1,\lambda_1+1) \end{pmatrix} =$$

$$= \begin{pmatrix} B(1,\lambda_1+1) & \dots & \dfrac{1}{(k-1)!} \dfrac{\partial^{(k-1)} B(1,y)}{\partial y^{k-1}}\Big|_{y=\lambda_1+1} \\ \vdots & \ddots & \vdots \\ 0 & \dots & B(1,\lambda_1+1) \end{pmatrix} =$$

$$= \begin{pmatrix} \frac{1}{\lambda_1+1} & \cdots & \frac{(-1)^{k-1}}{(k-1)!(\lambda_1+1)^{k-1}} \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \frac{1}{\lambda_1+1} \end{pmatrix} = \begin{matrix} \text{property of the} \\ \text{jordan matrix} \end{matrix} = f(J(\lambda_1+1)),$$

where $f(x) = \frac{1}{x}$.

So, we see a complete analogy with the property of the scalar Beta function.

Summarizing:

$$B(I,X) = UB(I,J)U^{-1}.$$

2.3 Pascal's rule or the Beta function recurrence relation

Pascal's rule is one of the key identities in combinatorics and given the relationship between the Beta function and binomial coefficients, as well as its use for the recurrent computation of the Beta function, it will be appropriate to try to generalize it for two arbitrary commuting matrices.

$$B(X+I,Y) + B(X,Y+I) = \int_0^1 e^{\ln(t)(X+I)} e^{\ln(1-t)Y} dt +$$

$$+ \int_0^1 e^{\ln(t)X} e^{\ln(1-t)(Y+I)} dt = \int_0^1 (e^{\ln(t)c} e^{\ln(1-t)Y} + e^{\ln(t)X} e^{\ln(1-t)(Y+I)}) dt =$$

$$\int_0^1 (Ue^{\ln(t)J_1} U^{-1} e^{\ln(t)I} Ue^{\ln(1-t)J_2} U^{-1} + Ue^{\ln(t)J_1} U^{-1} Ue^{\ln(1-t)J_2} U^{-1} e^{\ln(1-t)I}) dt =$$

$$= \text{according to properties of commuting matrices and exponent} =$$

$$\int_0^1 (Ue^{\ln(t)J_1} e^{\ln(1-t)J_2} U^{-1} e^{\ln(t)I} + Ue^{\ln(t)J_1} e^{\ln(1-t)J_2} U^{-1} e^{\ln(1-t)I}) dt =$$

$$\int_0^1 (Ue^{\ln(t)J_1} e^{\ln(1-t)J_2} U^{-1} (e^{\ln(t)I} + e^{\ln(1-t)I})) dt .$$

According to the property of the matrix exponent, the two terms obtained are found as exponents of the diagonal matrix:

$$e^{\ln(t)I} + e^{\ln(1-t)I} = \begin{pmatrix} e^{\ln(t)} & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & e^{\ln(t)} \end{pmatrix} + \begin{pmatrix} e^{\ln(1-t)} & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & e^{\ln(1-t)} \end{pmatrix} = I.$$

So, the end result is as follows:

$$B(X+I,Y) + B(X,Y+I) = \int_0^1 (Ue^{\ln(t)J_1} e^{\ln(1-t)J_2} U^{-1} I) dt =$$

$$= U\int_0^1 e^{\ln(t)J_1} e^{\ln(1-t)J_2}\, dt U^{-1} = UB(J_1, J_2)U^{-1} = B(X,Y).$$

Similarly to the scalar Beta function, Pascal's rule holds and has the same form, unlike many other properties that have additional constructions when working with matrix variables.

3 JACOBI THETA FUNCTION

3.1 Definition of the main Jacobi Theta function in the form of an infinite sum

$$\vartheta(J_r(\lambda_1), J_r(\lambda_2)) = \sum_{n=-\infty}^{\infty} Q^{n^2} \mathrm{H}^n,$$

where $Q = e^{\pi i J_r(\lambda_2)}, \mathrm{H} = e^{2\pi i J_r(\lambda_1)}$.

As in the previous section, let's start with each of the factors separately and then move on to the general form of the product:

$$Q^{n^2} = e^{\pi i n^2 J_r(\lambda_2)} =$$

$$= \begin{pmatrix} e^{\pi i n^2 \lambda_2} & \frac{\pi i n^2 e^{\pi i n^2 \lambda_2}}{1!} & \cdots & \frac{(\pi i n^2)^{k-2} e^{\pi i n^2 \lambda_2}}{(k-2)!} & \frac{(\pi i n^2)^{k-1} e^{\pi i n^2 \lambda_2}}{(k-1)!} \\ 0 & e^{\pi i n^2 \lambda_2} & \cdots & \cdots & \frac{(\pi i n^2)^{k-2} e^{\pi i n^2 \lambda_2}}{(k-2)!} \\ 0 & 0 & \ddots & \ddots & \vdots \\ \vdots & \vdots & & e^{\pi i n^2 \lambda_2} & \frac{\pi i n^2 e^{\pi i n^2 \lambda_2}}{1!} \\ 0 & 0 & \cdots & 0 & e^{\pi i n^2 \lambda_2} \end{pmatrix}$$

$$\mathrm{H}^n = e^{2\pi i n J_r(\lambda_1)} =$$

$$= \begin{pmatrix} e^{\pi i n \lambda_1} & \frac{\pi i n e^{\pi i n \lambda_1}}{1!} & \cdots & \frac{(\pi i n)^{k-2} e^{\pi i n \lambda_1}}{(k-2)!} & \frac{(\pi i n)^{k-1} e^{\pi i n \lambda_1}}{(k-1)!} \\ 0 & e^{\pi i n^2 \lambda_2} & \cdots & \cdots & \frac{(\pi i n)^{k-2} e^{\pi i n \lambda_1}}{(k-2)!} \\ 0 & 0 & \ddots & \ddots & \vdots \\ \vdots & \vdots & & e^{\pi i n^2 \lambda_2} & \frac{\pi i n e^{\pi i n \lambda_1}}{1!} \\ 0 & 0 & \cdots & 0 & e^{\pi i n^2 \lambda_2} \end{pmatrix}.$$

Then the product of these matrices will be:

$$\left(Q^{n^2}\ \mathrm{H}^n\right) = \begin{pmatrix} e^{2\pi i n \lambda_1} e^{\pi i n^2 \lambda_2} & \cdots & M_{(l,m)} \\ \vdots & \ddots & \vdots \\ 0 & \cdots & e^{2\pi i n \lambda_1} e^{\pi i n^2 \lambda_2} \end{pmatrix},$$

where $M_{(l,m)}$ is the product of the l-th row and the m-th column of the initial matrices for $m \geq l$.

$$M_{(l,m)} = \sum\left(0\ldots 0 e^{2\pi i n\lambda_1} e^{\pi i n^2 \lambda_2} \frac{(\pi i n^2)^0}{0!} \frac{(2\pi i n)^{m-l}}{(m-l)!} \ldots \frac{(\pi i n^2)^{m-l}}{(m-l)!} \frac{(2\pi i n)^0}{0!} 0\ldots 0\right),$$

while the number of zeros at the beginning and end will be equal to l and $k-m-1$ respectively. Then as a result we get the sum:

$$M_{(l,m)} = \sum_{j=0}^{m-l} \frac{(\pi i n^2)^j (2\pi i n)^{m-l-j}}{j!(m-l-j)!} e^{2\pi i n\lambda_1} e^{\pi i n^2 \lambda_2}.$$

Thus, similar to the product for the Beta function, we get a matrix element that depends only on the difference in the indices of the initial row and column, i.e. all the elements of the resulting matrix will be equal on the corresponding diagonals.

The next step is to return to the initial form of the function, namely to the sum:

$$\vartheta(J_r(\lambda_1), J_r(\lambda_2)) = \begin{pmatrix} \vartheta(\lambda_1,\lambda_2) & \ldots & \sum_{j=0}^{m-l} \frac{1}{j!(m-l-j)!} \frac{\delta^{m-l}\vartheta(z,\tau)}{\delta z^{m-l-j}\delta\tau^j}\Big|_{\substack{z=\lambda_1 \\ \tau=\lambda_2}} \\ \vdots & \ddots & \vdots \\ 0 & \ldots & \vartheta(\lambda_1,\lambda_2) \end{pmatrix};$$

$$\vartheta(Z,T) = \underset{\text{commuting}}{\overset{\text{for } Z,T}{=}} U\vartheta(J_1,J_2)U^{-1}$$

3.2 The period of the Jacobi theta function

The scalar Jacobi theta function is periodic with a period of 1 in z: $\vartheta(z\pm 1,\tau) = \vartheta(z,\tau)$, and by completing the square, τ — quasiperiodic in z:

$$\vartheta(z\pm\tau,\tau) = e^{-\pi i\tau \mp 2\pi i z}\vartheta(z,\tau).$$

In the case of matrix variables, the 1-periodicity of the first variable is transformed into the periodicity of the unit matrix I:

$$\vartheta(J_r(\lambda_1)+I, J_r(\lambda_2)) = J_{r_1}(\lambda_1) + I = J_{r_1}(\lambda_1+1) = \vartheta(J_r(\lambda_1+1), J_r(\lambda_2)) =$$

$$= \begin{pmatrix} \vartheta(\lambda_1+1,\lambda_2) & \ldots & \sum_{j=0}^{m-l} \frac{1}{j!(m-l-j)!} \frac{\delta^{m-l}\vartheta(z,\tau)}{\delta z^{m-l-j}\delta\tau^j}\Big|_{\substack{z=\lambda_1+1 \\ \tau=\lambda_2}} \\ \vdots & \ddots & \vdots \\ 0 & \ldots & \vartheta(\lambda_1+1,\lambda_2) \end{pmatrix} =$$

$$= \begin{pmatrix} \vartheta(\lambda_1,\lambda_2) & \dots & \sum_{j=0}^{m-l} \frac{1}{j!(m-l-j)!} \frac{\delta^{m-l}\vartheta(z,\tau)}{\delta z^{m-l-j}\delta\tau^{j}}\Big|_{\substack{z=\lambda_1\\ \tau=\lambda_2}} \\ \vdots & \ddots & \vdots \\ 0 & \dots & \vartheta(\lambda_1,\lambda_2) \end{pmatrix} = \vartheta(J_r(\lambda_1),J_r(\lambda_2)).$$

Then for two arbitrary commuting matrices Z, T:

$$\vartheta(Z+I,T) = \sum_{n=-\infty}^{\infty} e^{2\pi i n(Z+1)} e^{2\pi i n^2 T} = \sum_{n=-\infty}^{\infty} e^{2\pi i n Z} e^{2\pi i n I} e^{2\pi i n^2 T} =$$

$$= \sum_{n=-\infty}^{\infty} U e^{2\pi i n J_1} U^{-1} e^{2\pi i n I} U e^{2\pi i n^2 J_2} \mathrm{U}^{-1} = \sum_{n=-\infty}^{\infty} U e^{2\pi i n J_1} e^{2\pi i n I} e^{2\pi i n^2 J_2} \mathrm{U}^{-1} =$$

$$= \sum_{n=-\infty}^{\infty} U e^{2\pi i n (J_1+I)} e^{2\pi i n^2 J_2} \mathrm{U}^{-1} = U\vartheta(J_r(\lambda_1)+I, J_r(\lambda_2))U^{-1} =$$

$$= U\vartheta(J_1,J_2)U^{-1} = \vartheta(Z,T).$$

4 COMPARISON OF THE OBTAINED RESULTS WITH EXISTING METHODS OF WORKING WITH MATRIX-VARIATE FUNCTIONS

4.1 Comparison of obtaining the matrix Gamma function using the Lanczos approximation method and the obtained method

Computing the matrix Gamma function by the Lanczos method [3] is performed on the basis of the following formula:

$$\Gamma(A) = \sqrt{2\pi}\left(A+\left(\alpha+\frac{1}{2}\right)I\right)^{A+\frac{1}{2}I} e^{-\left(A+\left(\alpha+\frac{1}{2}\right)I\right)} \times$$

$$\times\left(c_0(\alpha)I + \sum_{k=1}^{m} c_k(\alpha)(A+(k-1)I)^{-1} + e_{\alpha,m}(A)\right),$$

where $c_k(\alpha)$ are the Lanczos coefficients that depend on the parameter α.

Typically, pre-logarithmization is used to optimize calculations and avoid overflow problems during calculations:

$$\ln(\Gamma(A)) = \frac{1}{2}\ln(2\pi) + \left(A+\frac{1}{2}I\right)\ln\left(A+\left(\alpha+\frac{1}{2}\right)I\right) - \left(A+\left(\alpha+\frac{1}{2}\right)I\right) +$$

$$+ \ln\left(c_0 cI + \sum_{k=1}^{m} c_k \alpha(A+(k-1)I)^{-1} + e_{\alpha,m}(A)\right).$$

It is also important to note that the set of coefficients $c_k(\alpha)$ is found empirically [3] and for the example for the pair $\alpha = 9,\ m = 10$ the following values are used:

k	$c_k(9)$
0	1.000000000000000174663
1	5716.400188274341379136
2	− 14815.30426768413909044
3	14291.49277657478554025
4	− 6348.160217641458813289
5	1301.608286058321874105
6	− 108.1767053514369634679
7	2.605696505611755827729
8	$-0.742345251020141615\times10^{-2}$
9	$0.538413643250956406\times10^{-2}$
10	$-0.402353314126823637\times10^{-2}$

Now let's compare the actual algorithms for finding matrices by these two methods:

Algorithm for finding the function using the Lanczos method	**Algorithm for finding the function using the Jordan form**
1. Set $\alpha = 9$; $m = 10$; $S = c_0 I + c_1 A^{-1}$;	1. Eigenvalues λ_i of matrix A;
2. for $k = 2:10$	2. Eigen and adjoint vectors x_i;
3. $S = S + c_k (A + (k-1)I)^{-1}$;	3. Jordan form J;
4. end	4. $\Gamma(J)$ and transitional matrix U based on x_i;
5. $L = \frac{1}{2}\ln(2\pi)I + \left(A - \frac{1}{2}I\right)\ln\left(A + \frac{17}{2}I\right) - \left(A + \frac{17}{2}I\right) + \ln(S)$;	5. $\Gamma(A) = U\Gamma(j)U^{-1}$.
6. $\Gamma(A) \approx e^L$.	–

So, as we can see, the proposed algorithm is much more convenient for actually finding the values of the matrix Gamma function $\Gamma(A)$ in comparison with some existing numerical methods. It is also in addition to the above that our method has the advantage of being able to use the obtained function and its properties in further research. Similar results were obtained for Spouge's approximation method [3], since both of them have similar algorithms.

4.2 Research using the Schur decomposition

The Schur decomposition method [4] is based on the decomposition of the input matrix and its representation through unitary and upper triangular:

$$\forall A, B \in \mathbb{C}(n \times n): A * B = B * A,\ \exists U,\ R_1, R_2 : A = QR_1Q^*;\ B = QR_2Q^*.$$

Thanks to this, in their work L. Jódar, J. CCortés [5] for two commuting matrices proved several properties of the matrix-variate Beta function, namely the symmetry of the variables and the connection with the matrix Gamma function:

$$B(P,Q) = \Gamma(P)\Gamma(Q)\Gamma - 1(P+Q).$$

It should be noted that the last property was proved only for diagonalizable, commuting matrces P and Q. Compared to the obtained results, we coan sii that the main advantage of using the Jordan canonical form is the presence of an explicit form of the resulting matrix. This, in turn, gives us the following advantages compared to the Shur Schedule:

- Ability to derive properties associated with certain partial cases and specific function values;
- From the point of view of computational complexity, although historically the calculation of the Jordan canonical form was usually considered a very difficult task, the properties of the matrix function from the Jordan matrix allow us to bypass this step, and so the need remains only to find the eigenvalues and the corresponding vectors to form a basis. Then, comparing to the Schur decomposition, which has a computational complexity of $O(n^3)$, our method will have an approximate complexity of $O(n^{\omega})$, $2 < \omega < 2.376$.

4.3 The zonal polynomials method

The method of zonal polynomials [6] is one of the methods for studying such functions using integrals and the difference in approach will be illustrated on its example.

In this method, the studied function differs from others, namely, it has the following form:

$$B_m(a,b) = \int_0^{I_m} \det(X)^{a-\frac{(m+1)}{2}} \det(I_m - X)^{b-\frac{(m+1)}{2}} dX.$$

Additional results and generalizations of this function were found using zonal polynomials and evaluating the resulting integral for them. The main use case of it and its generalized forms is the matrix beta distribution:

For $U \sim B_p^I(a,b)$, the distribution density of the positive definite square matrix U:

$$f(U) = \frac{1}{B_p(a,b)} \det(U)^{a-\frac{p+1}{2}} \det(I_p - U)^{b-\frac{p+1}{2}}.$$

As we can see, this function and similar functions of this type contain only matrix determinants and, in some cases, trace. This means that these functions are limited to uses only in problems in which the input signal has a matrix form, and the output signal is already scalar. This has a number of disadvantages in solving some statistical problems in which it is important to leave connections between certain vectors or blocks of vectors, like the problems that were mentioned in the introductory section.

REFERENCES

1. Sujit Kumar Mitra, "A Density-Free Approach to the Matrix Variate Beta Distribution," *Sankhyā: The Indian Journal of Statistics, Series A* (*1961-2002*), vol. 32, no. 1, pp. 81–88, 1970. Available: http://www.jstor.org/stable/25049638
2. Amr Elrawy, Mohammed Saleem, and Z. Kishka, "The Matrix of Matrices Exponential and Application," *International Journal of Mathematical Analysis*, 13, pp. 81–97, 2019. doi: 10.12988/ijma.2019.9210
3. João Cardoso, Amir Sadeghi, "Computation of matrix gamma function," *BIT Numerical Mathematics*, 59, 2019. doi: 10.1007/s10543-018-00744-1
4. Maya Neytcheva, "On element-by-element Schur complement approximations," *Linear Algebra and its Applications*, vol. 434, issue 11, pp. 2308–2324, 2011. Available: https://doi.org/10.1016/j.laa.2010.03.031
5. L Jódar, J.C Cortés, "Some properties of Gamma and Beta matrix functions," *Applied Mathematics Letters*, vol.11, issue 1, pp. 89–93, 1998. Available: https://doi.org/10.1016/S0893-9659(97)00139-0
6. Daya Nagar, Sergio Gómez-Noguera, and Arjun Gupta, "Generalized Extended Matrix Variate Beta and Gamma Functions and Their Applications," *Ingeniería y Ciencia*, 12, pp. 51–82, 2016. doi: 10.17230/ingciencia.12.24.3

Received 02.11.2023

INFORMATION ON THE ARTICLE

Dmytro O. Shutiak, ORCID: 0009-0008-6480-3706, World Data Center for Geoinformatics and Sustainable Development of the National Technical University of Ukraine "Igor Sikorsky Kyiv Polytechnic Institute", Ukraine, e-mail: dima.shutyak@gmail.com

Gleb B. Podkolzin, ORCID: 0000-0002-7120-2772, Educational and Research Institute for Applied System Analysis of the National Technical University of Ukraine "Igor Sikorsky Kyiv Polytechnic Institute", Ukraine, e-mail: podkolzin.gleb@lll.kpi.ua

Victor G. Bondarenko, ORCID: 0000-0003-1663-4799, Educational and Research Institute for Applied System Analysis of the National Technical University of Ukraine "Igor Sikorsky Kyiv Polytechnic Institute", Ukraine, e-mail: bondarenvg@gmail.com

Yury A. Chapovsky, ORCID: 0009-0001-8981-4742, Educational and Research Institute for Applied System Analysis of the National Technical University of Ukraine "Igor Sikorsky Kyiv Polytechnic Institute", Ukraine

КЛАСИЧНІ СПЕЦІАЛЬНІ ФУНКЦІЇ З МАТРИЧНИМИ ЗМІННИМИ / Д.О. Шутяк, Г.Б. Подколзін, В.Г. Бондаренко, Ю.А. Чаповський

Анотація. Розглянуто декілька найбільш часто використовуваних спеціальних функцій та їх ключові властивості, а також запропоновано аналітичний підхід до побудови їх аналогів із метричними змінними. Щоб досягти цього, ми уникали використання будь-яких алгоритмів чисельного наближення та натомість покладались на властивості матриць, матричної експоненти та Жорданову нормальну форму для представлення матриць. Ми зосередились на таких функціях: гамма-функція як приклад функції однієї змінної з великою кількістю властивостей і застосувань; бета-функція, щоб підкреслити подібності та відмінності від додавання другої змінної до функції матричної змінної; тета-функція Якобі. Побудовано явні представлення функцій і доведено декілька ключових властивостей для цих функцій; висвітлено та порівняно інші підходи, які використовувалися в минулому для вирішення цих задач.

Ключові слова: матриця, спеціальна функція, гамма-функція, бета-функція, тета-функції Якобі, Жорданова нормальна форма.

UDC 004.942:537.525
DOI: 10.20535/SRIT.2308-8893.2024.4.11

AN ADVANCED METHOD OF INTERPOLATION OF SHORT-FOCUS ELECTRON BEAMS BOUNDARY TRAJECTORIES USING HIGHER-ORDER ROOT-POLYNOMIAL FUNCTIONS AND ITS COMPARATIVE STUDY

I. MELNYK, A. POCHYNOK, M. SKRYPKA

Abstract. The comparison of three advanced novel methods for estimating the boundary trajectory of electron beams propagated in ionized gas, including lower-order interpolation, self-connected interpolation, and extrapolation, as well as higher-order interpolation, is considered and discussed in the article. All estimations of the corresponding errors have been provided relative to numerically solving the set of algebra-differential equations that describe the boundary trajectory of the electron beam. By providing analysis, it is shown and proven that lower-order interpolation usually gives the minimal value of average error, using the method of self-connected interpolation and extrapolation gives the minimal error for estimation of focal beam parameters, and higher-order interpolation is suitable to obtain a uniform error value over the entire interpolation interval. All results of error estimation were obtained using original computer software written in Python.

Keywords: interpolation, extrapolation, lower-order interpolation, higher-order interpolation, root-polynomial function, ravine function, average error, electron beam, boundary trajectory, high voltage glow discharge, electron beam technologies.

INTRODUCTION

Interpolation of boundary trajectories of electron beams is very important task today, taking into account the high level of development electron beam technologies and its applying in modern industry [1–9].

Really, industrial electron-beam technologies have been developed and widely applied in industry since 60–70-th years of XX century [6–9], but its application in modern industry is also continued. Therefore, adaptation of traditional electron-beam technologies to corresponded advanced technological processes is successfully provided today [1–5; 10–13].

Today main branches of industry, where electron beam technologies find high level of application, are follows: metallurgy, mechanical engineering, electrical engineering, instrument making, microelectronic production, automotive, aircraft, and space industries [1–5].

For example, in microelectronic production point-focus electron beams with focal beam radius can be successfully applied for making contacts in precision cryogenic devices [14; 15]. Corresponded estimation of diameter of welding seam in electron-beam technologies have been provided recently in the paper [16; 17]. Other advanced application of electron-beam technologies in microelectronic production is refining of silicon [10–13] and obtaining of chemically-complex

ceramic films for high quality capacitors and for microwave transmitters and receivers in advanced communication systems [18–20]. Generally, advanced possibility of using electron beam technologies in modern microelectronic production are described in manual book [5].

In metallurgy advanced electron-beam technologies are widely used, since 60-years of XX century, for refining of refractory metals [21–23] and other refractory materials, in particular silicon for the microelectronics industry [11–13]. Among other, advanced application of electron beam heating in metallurgy, which have been developed in last few years, three-dimensional printing by the metals, including forming of details with complex spatial shape for aircraft and space industry, have to be mentioned and considered [24–26].

In energetic industry and electric vehicle production electron beam technologies are widely used for deposition high-quality ceramic insulator films [27–30]. Other advanced application of high-quality chemically-complex ceramic films in automotive, aircraft and space industry is obtaining heat-resistant and heat-protective thick films for details of engines, which operated under conditions of high temperature. For deposition such kind of films advanced method of physical vapor deposition by electron-beam heating is usually applied [27–30].

Special issue is applying of high-energy intensive electron beams, obtained in the accelerators, for changing the properties of treated materials. Corresponded technologies are described in the works [4; 31–33].

The advantages of electron beams, which caused its wide application in modern industrial technologies, are as follows [1–5].

1. High total power and power density in beam focus.

2. High energetic efficiency of electron beam sources.

3. Simplicity of fast control of beam power and spatial position of beam focus using electric and magnetic fields.

4. Wide range of different technological operations, which can be realized by electron beam heating and chemical treatment.

Taking into account pointed out advantages of electron beam technologies, elaboration of advanced improving industrial constructions of electron beam sources, which are called electron guns, is important scientific and engineering task today. Usually, this task solving using two different ways, which are, generally, follows.

1. Improving the constructions of electron guns with heated cathode, operated in conditions of high vacuum. Such kind of electron guns are traditional and widely use since 60–70-th years of XX century [6–9].

2. Elaboration of novel types of electron guns, based on auto-electronic emission of electrons in the string electric fields, photoemission, as well as on emission in gas discharges. Among this types of guns special place occupied high-voltage glow discharge electron guns, which particularities of operation will be considered in next part of this paper.

HIGH VOLTAGE GLOW DISCHARGE ELECTRON GUNS AND ADVANTAGES ITS APPLYING IN ELECTRON BEAM TECHNOLOGIES

In the last few decades, in some technological processes that are implemented in a soft vacuum in an environment of air or active gases, instead of the traditionally used guns with a heated cathode, alternative guns have been successfully applied, the operation of which is based on the use of a high-voltage glow discharge (HVGD). From a physical point of view, HVGD is considered a kind of dis-

charge, taking place under voltage between electrodes 5–40 kV and pressure in the discharge volume range of 0.1–10 Pa [34–36]. In the work [34–36] the basic principles of simulation HVGD electron guns have been considered, and corresponding mathematical relations were also given and analyzed. In the papers [37; 38] main advantages of applying HVGD electron guns for welding, melting processes, as well as for deposition of ceramic films have been pointed out. These advantages are as follows [37; 38].

1. High stability of operation in conditions of soft vacuum.
2. Relative simplicity of gun construction.
3. Relative simplicity of evacuation equipment for obtaining soft vacuum.
4. Since the current density from the cold cathode in HVGD conditions is not so large, range of 0.01 A/cm^2, using of enlarges cathode surface and suitable self-maintained electron-ion optics allows forming profile electron beams with linear and ring-like focus [37; 38].
5. Simplicity of control of gun current, both by relatively slow aerodynamic method using electromagnetic valve [39], and by fast electric method with lighting of low-voltage additional discharge in anode plasma region [40].
6. Possibility of providing operation of HVGD electron guns in impulse regime with obtaining advanced technological possibilities of pulsed electron beams [41–43].

The regulation time for slow electrodynamic control systems was estimated in the paper [39], and for fast electric control systems correspondently, in papers [41; 43].

However, simulation of HVGD electron guns is realized today mostly by solving of complex algebra-differential equations, described forming and interaction of charged particles flows in the soft vacuum conditions. The main problem in this task is defining of anode plasma boundary form and position. It caused by the fact, that in HVGD anode plasma is considered as the source of ions and as electrode with fixed potential, which is transparent to beam electrons [34–36]. Simplified analytical models for defining of focal beam parameters in HVGD aren't existed [34]. But namely such approximative estimations are very important for defining the technological possibilities of electron beams, especially on the first stage of gun designing [16; 17]. Absence of such simple approach of analytical calculations of focal beam parameters significantly hinders development and implementation in industry of HVGD electron guns, which advantages have been described above. Also using of sophisticated numerical calculation methods is lead to increasing the complexity of solving simulation problems in case of implementing cloud computing. Corresponded estimations have been given in works [44–46]. Therefore, finding the corresponding analytical relations for estimation focal parameters of electron beams, formed by HVGD electron guns, is very important scientific and engineering task today. This task will be considered in the next section of the article.

GENERAL STATEMENT OF PROBLEM OF INTERPOLATION BOUNDARY TRAJECTORY OF ELECTRON BEAM, PROPAGATED IN IONIZED GAS, AND ESTIMATION OF ERRORS

Firstly, the basic approach to interpolation the boundary trajectories of electron beams have been proposed in the years 2019–2020 in the papers [17; 47–49]. Generally, this approach is based on the following presumptions.

1. Numerical solving of basic set of algebra-differential equations for the boundary trajectory of short-focus electron beam, propagated in the soft vacuum conditions in the medium of ionized gas, which is, in general form, written as follows [1–6; 34; 49]:

$$f = \frac{n_e}{n_{i0} - n_e};\ C = \frac{I_b(1 - f - \beta^2)}{4\pi\varepsilon_0\sqrt{\frac{2e}{m_e}}U_{ac}^{1.5}};\ \frac{d^2 r_b}{dz^2} = \frac{C}{r_b};\ \theta = \frac{dr_b}{dz} + \theta_s;$$

$$n_e = \frac{I_b}{\pi r_b^2};\ v_e = \sqrt{\frac{2eU_{ac}}{m_e}};\ n_{i0} = r_b^2 B_i p n_e \sqrt{\frac{\pi M_i \varepsilon_0 n_e}{m_e U_{ac}}} \exp\left(-\frac{U_{ac}}{\varepsilon_0 n_e r_b^2}\right); \quad (1)$$

$$\gamma = \sqrt{1-\beta^2};\ \tan\left(\frac{\theta_{\min}}{2}\right) = \frac{10^{-4} Z_a^{4/3}}{2\gamma\beta^2};\ \tan\left(\frac{\theta_{\max}}{2}\right) = \frac{Z_a^{3/2}}{2\gamma\beta^2};$$

$$\overline{\theta}^2 = \frac{8\pi(r_b Z_a)^2 dz}{n_e} \ln\left(\frac{\theta_{\min}}{\theta_{\max}}\right),\ \beta = \frac{v_e}{c},$$

where U_{ac} is the voltage of HVGD lighting; I_b is the current of electron beam; p is the residual gas pressure in the volume of HVGD lighting; z is the longitudinal coordinate; r_b is the radius of the boundary trajectory of the electron beam; dz is the length of the electron path in the longitudinal direction at the current iteration; n_{i0} is the concentration of residual gas ions on the beam symmetry axis; n_e is the concentration of beam electrons'; v_e is the average velocity of the beam electrons; $\theta_{\min}$ and $\theta_{\max}$ are the minimum and maximum scattering angles of the beam electrons, corresponding to Rutherford model [1–6]; $\overline{\theta}$ $\theta_{\min}$ $\theta_{\min}$ is the average scattering angle of the beam electrons; f is the residual level of gas ionization; B_i is the gas ionization level; m_e is the electron mass; ε_0 is the dielectric constant; c is the light velocity; γ is the relativistic factor; M_i is the molecular mass of residual gas atoms, and Z_a is its' nuclear charge.

2. Choosing of k basic points (r_b, z) on the calculated boundary trajectory.

3. Interpolation of defined function $r_b(z)$ using ravine root-polynomial function [47–49]:

$$r_b(z) = \sqrt[n]{C_n z^n + C_{n-1} z^{n-1} + \ldots + C_1 z + C_0}, \quad (2)$$

where $n = k - 1$ is the degree of the polynomial and the order of the root-polynomial function, and $C_0, \ldots, C_n$ are the polynomial coefficients.

4. Defining of interpolation error using relation [47–49]:

$$\varepsilon(\mathrm{z}) = \frac{r_{b_{num}}(\mathrm{z}) - r_{b_{int}}(\mathrm{z})}{r_{b_{num}}(\mathrm{z})} \cdot 100\%, \quad (3)$$

where $r_{b_{num}}(\mathrm{z})$ is numerical and $r_{b_{int}}(\mathrm{z})$ is interpolated values of beam radius r_b.

Generally, described above method of interpolation of electron beam boundary trajectory is based on the presumption, that dependence $r_{b_{num}}(z)$ is considered as ravine one with one global minimum and quasilinear dependence outside the region of minimum. This presumption is fully corresponded to the conception of physics of the flows of charged particles, have been described in [1–6]. Provided theoretical researches shown, that main particularities of root-polynomial function (2) are usually generally suitable to this presumption. Therefore, the interpolation error, defined by relation (3), is always very small, range of few percent, and in some cases is even smaller [47–49]. Typical dependence of obtained relative error of interpolation on z coordinate for different order of polynomial function (2) is presented in Fig. 1 [49]. In this figure the solid line corresponded to a second-order function, the dotted line – to a third-order function, the dashed line – to a fourth-order function, and the dash-dotted line – to a fifth-order function. Model parameters for the numerical data, presented at Fig. 1, are follows: $U_{ac} = 10$ kV, $I_b = 0.5$ A, $p = 0.1$ Pa.

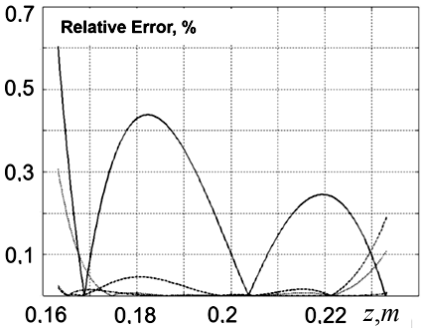


Fig. 1. Errors of interpolation of the boundary trajectory of the electron beam depend on *z* coordinate [49]

Also provided researches shown, that the interpolation error is strongly depend on position of interpolation points $P_i(r_{b_i}, z_i)$ on the interpolated interval. It has been proven, that the minimum error of interpolation is provided to symmetric interval of ravine data with location minimum in the medium point. And if the ravine data is asymmetric, the value of error is significantly increased. Namely this established rule constituted a theoretical basis for further research, which will be described later in this article.

By the reason of results of this researches the method of interpolation by higher-order root-polynomial functions is proposed, which will be considered in next section of the article.

In the paper [50] was described the method of approximation of the trajectories of electron beams, propagated in ionized gas, using third-order root-polynomial function (2). Another approach to simulation of focal beam parameters of HVGD electron guns was given in the paper [51].

ASYMMETRIC RAVINE NUMERICAL DATA AND STATEMENT THE PROBLEM OF ITS INTERPOLATION AND EXTRAPOLATION

Let's considering left-hand and right-hand asymmetric ravine root-polynomial functions, which in general form presented in Fig. 2. Here given the basic parameters of these functions, such as radiuses of electron beam in the Start Point SP r_{start} and End Point EP r_{end}, location of this points z_{start} and z_{end}, and position of beam focus z_f.

It is clear from Fig. 2, that, corresponding to the theory of interpolation, basic principles of numerical methods, and probability theory [52–58], for interpola-

tion ravine dependences using root-polynomial functions (2) the additional reference point with coordinate z_{bp} is considered, and its location is defined by following arithmetic-logic relation:

$$z_{bp}(i)=(r_{end}>r_{start})\cdot\left(\begin{pmatrix} r(i)<r_{start} \\ i_0=1; i=i+1 \end{pmatrix}\cdot z(i)+\begin{pmatrix} r(i)\geq r_{start} \\ i=i+1 \end{pmatrix}\cdot z_{bp}(i-1)\right)+$$

$$+(r_{start}\geq r_{end})\cdot\left(\begin{pmatrix} r(i)<r_{end} \\ i_0=N_P; i=i-1 \end{pmatrix}\cdot z(i)+\begin{pmatrix} r(i)\geq r_{end} \\ i=i-1 \end{pmatrix}\cdot z_{bp}(i+1)\right), \qquad (4)$$

where N_p is the whole number of points in numerical calculation of beam trajectories, which is usually in range $N_P>10^4$. On the contrary, the value of the basic points N_{BP} for solving interpolation tasks is significantly smaller: $N_{BP}=n+1$.

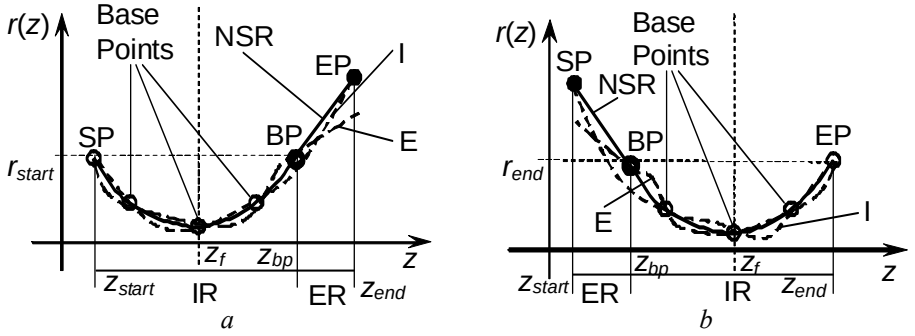


Fig. 2. Right-hand (*a*) and left-hand (*b*) asymmetric ravine functions: IR — Interpolation Region; ER — Extrapolation Region; SP — Start Point; EP — End Point; BP — Boundary Point; NSR — numerical simulation result; I — interpolation result; E — extrapolation result

The basic principles of forming arithmetic-logic relations have been considered in the book chapter [45].

After using iterative relation (4), finding of location of basic points BP on interval $[z_{start}, z_{bp}]$ for right-hand asymmetric function or in interval $[z_{bp}, z_{end}]$ for left-hand one is defined by following relations:

$$N_{BP}(j)=(z_j\geq z_f)\,N_f(1+k_f(j))+(z_j<z_f)N_f(1-k_f(j)), \qquad (5)$$

where $k_f(j)$ is the coefficient, which depend on the order of root-polynomial function and provided the minimal error of interpolation for symmetric ravine numerical data. For functions of even order usually one basis point is located at the focus position z_f and other symmetrically on the interval of interpolation $[z_{start}, z_{bp}]$ or $[z_{bp}, z_{end}]$. For example, for fifth-order root polynomial function: $k_f(2)=\frac{1}{3}$ and $k_f(3)=\frac{1}{20}$. For six-order function: $k_f(2)=\frac{1}{3}$ and $k_f(3)=\frac{1}{3}$. Correspondent approach to calculation the coefficients $k_f(j)$ is related with the theory of numerical methods [52; 53; 55; 56] and was considered in the paper [49]. Really, for symmetric ravine function:

$$\left(\left\lceil \frac{N_p}{2} \right\rceil - lk_f(l)N_p\right) = \left(\left\lceil \frac{N_p}{2} \right\rceil + lk_f(l)N_p\right), \quad l = \begin{cases} \left[1; \left\lfloor \frac{n+1}{2} \right\rfloor\right], \text{ for odd } n; \\ \left[1; \left\lfloor \frac{n}{2} \right\rfloor\right], \text{ for even } n. \end{cases} \quad (6)$$

For asymmetric ravine function problem of minimizing error of interpolation can be solved by to following ways.

1. Using interpolation method with defining position of basic points by relations (4)–(6). In this case the points are located evenly on interval $[z_{start}, z_{end}]$, and the focus position z_p can't be considered as a region of minimal error, because the coefficients k_f are calculated for ravine function.

2. Using interpolation method with defining position of basic points by relations (4)–(6) only for the symmetric region IR (see Fig. 2), and for region ER solving extrapolation task [53; 54]. In this case the additional basic boundary point BP is defined by relation (4) and used. Therefore, corresponding to Fig. 2, for right-hand asymmetric ravine data interpolation provided on the interval $[z_{start}, z_{bp}]$, and extrapolation on the interval $[z_{bp}, z_{end}]$. In contrary, for left-hand asymmetric ravine data interval of extrapolation is $[z_{start}, z_{bp}]$ and interval of interpolation is $[z_{bp}, z_{end}]$. For such self-connected interpolation-extrapolation task maximal error is always observed on the region ER, but in the region of focus position it is minimized.

Other method of interpolation, which give the average value of error on the whole interpolation interval, will be considered in the next section of the article.

INTERPOLATION OF ASYMMETRIC RAVINE NUMERICAL DATA USING ROOT-POLYNOMIAL FUNCTION OF HIGHER ORDER

The main distinguishing feature of proposed method of interpolation is solving of interpolation task on the whole interval of asymmetric ravine function $[z_{start}, z_{end}]$, but with including into consideration the boundary point BP, which coordinate, corresponding to Fig. 2, is z_{bp}. In such conditions other basic point are located in the IR region, but in ER region used interpolation by root-polynomial function (2) with the same polynomial coefficients. The order of this function is N_{BP}, where N_{BP} is number of basic points, located in the IR region. Therefore, all basic points are located evenly in interpolation interval IE using relations (4)–(6). The arithmetic-logic relation for defining set of coefficients $\{C_n \ldots C_0\}$ of the root-polynomial function of higher order $f_{n+1}(z)$ by corresponding set of basic points $\Omega(P_{start}, P_{bp}, P_{end})$ is written as follows:

$$\Omega(P_{start}, P_{bp}, P_{end}) =$$

$$= (r_{start} < r_{start}) \cdot (\{z_{start}, r_{start}\}, \{z_{BP_j}, r_{BP_j}\}\Big|_{j=1\ldots n-2}, \{z_{bp}, r_{bp}\}, \{z_{end}, r_{end}\}) +$$

$$+ (r_{start} \geq r_{end}) \cdot (\{z_{start}, r_{start}\}, \{z_{bp}, r_{bp}\}, \{z_{BP_j}, r_{BP_j}\}\Big|_{j=1\ldots n-2}, \{z_{end}, r_{end}\}), \quad (7)$$

$$\Omega(C_j\Big|_{j=0}^{j=n+1}) = \mathbf{F}(z_j\Big|_{j=0}^{j=n+1}, r_j\Big|_{j=0}^{j=n+1}).$$

Analytical relations for defining the set of coefficients of the root-polynomial function $\Omega(C_j\Big|_{j=0}^{j=n+1})$ through vector-function $\mathbf{F}(z_j\Big|_{j=0}^{j=n+1}, r_j\Big|_{j=0}^{j=n+1})$ for function from second to sixth order have been given and analyzed in the paper [49].

Some examples of using relations (4)–(7) for defining the coefficient of root-polynomial function (2), as well as comparing the error of interpolation using high-order, low-order functions and combined interpolation-extrapolation method, will be presented in the next part of the article.

OBTAINED RESULTS OF INTERPOLATION AND EXTRAPOLATION OF ASYMMETRIC RAVINE NUMERICAL DATA USING ROOT-POLYNOMIAL FUNCTION OF LOW AND HIGHER ORDER

Comparing study of applying interpolation and combined interpolation-extrapolation methods, described above, has been provided by comparing such types of errors: maximal error $\varepsilon_{\max}$, average error ε_{av}, error of estimation the focus position ε_F, and error of estimation focal beam radius ε_{rf}.

Average error is defined by the well-known method of optimization technique [53; 54] and of mathematical statistics [57; 58] as follows:

$$\varepsilon_{av} = \frac{\sum_{i=1}^{N_p} |r_{est} - r_{sim}|}{N_p}, \tag{8}$$

where r_{sim} is the radius of the electron beam, calculated numerically by the set of equations (1) using the fourth-order Runge-Kutt method [55; 56], and r_{est} is the value of the beam radius, estimated using relation (2). Local error of interpolation and extrapolation at considered point z has been defined, using relation (3).

All errors have been estimated for different order of root-polynomial functions n and length of extrapolation region L_{add}. Task parameter L_{add} is given in the tables of obtained testing results in absolute value, in meters, and relatively to the length of the interpolation region IR, in percents.

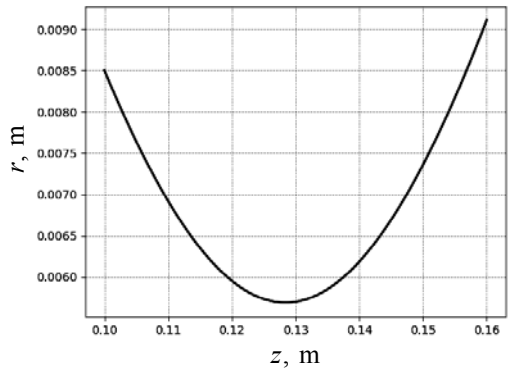


Fig. 3. Dependence $r(z)$ for $U_{ac} = 15\,\text{kV}$, $I_b = 5.5\,\text{A}$, $P = 4.5\,\text{Pa}$, end point $z_{end} = 0.16\,\text{m}$ (screen copy)

Task 1. $U_{ac} = 15\,\text{kV}$, $I_b = 5.5\,\text{A}$, $p = 4.5\,\text{Pa}$, $r_{start} = 8.5\,\text{mm}$, $\theta = 10.5^0$, $z_{start} = 0.1\,\text{m}$. End points: 1. $z_{end} = 0.16\,\text{m}$; 2. $z_{end} = 0.165\,m$; 3. $z_{end} = 0.17\,\text{m}$; 4. $z_{end} = 0.175\,\text{m}$; 5. $z_{end} = 0.18$. Additional boundary basic point: $z_{bp} = 0.156822\,\text{m}$.

For this example, the dependence $r(z)$, defined by numerical solving the set of equations (1), is presented in Fig. 3.

It is clear that the dependence presented in Fig. 3, corresponding to classification, given at Fig. 2, is a right-hand asymmetric ravine function. Errors in solving interpolation and self-connected interpolation-extrapolation tasks for this example are presented in Table 1. Corresponded polynomial coefficients are given in Table 2.

It is clear that the dependence presented in Fig. 3, corresponding to classification, given at Fig. 2, is a right-hand asymmetric ravine function. Errors in solving interpolation and self-connected interpolation-extrapolation tasks for this example are presented in Table 1. Corresponded polynomial coefficients are given in Table 2.

T a b l e 1. Errors of estimation for Task 1

Type of Error	Estimation methods								L_{addb}, m /%
	Interpolation Function Order, n			Extrapolation Function Order, n			Orders n_h of Higher-Order Method		
	4	5	6	4	5	6	5	6	
ε_{max}, %	0.51	0.865	0.118	1.355	2.26	0.76	3.2	0.26	$3.17\cdot10^{-3}$ / 5.6
ε_{av}, %	0.1586	0.377	$3\cdot10^{-2}$	0.19	0.338	$6\cdot10^{-2}$	0.926	0.11	
ε_F, %	$3.73\cdot10^{-2}$	$9.3\cdot10^{-3}$	$1.87\cdot10^{-2}$	0	0	0	0.129	0	
ε_{rf}, %	$1.45\cdot10^{-4}$	0.1536	$4.07\cdot10^{-5}$	$4.33\cdot10^{-12}$	$6.8\cdot10^{-12}$	$7.12\cdot10^{-10}$	0.014	$5\cdot10^{-2}$	
ε_{max}, %	0.98	1.434	0.281	1.355	2.26	0.76	3.35	0.3	$3.82\cdot10^{-3}$ / 14.388
ε_{av}, %	0.275	0.595	$6.6\cdot10^{-2}$	0.19	0.338	0.059	0.96	0.16	
ε_F, %	0.1367	$2\cdot10^{-2}$	$7.1\cdot10^{-2}$	0	0	0	0.82	0	
ε_{rf}, %	$2\cdot10^{-3}$	0.25	$5.3\cdot10^{-4}$	$4.33\cdot10^{-12}$	0.068	$7.12\cdot10^{-10}$	0.069	0.0533	
ε_{max}, %	1.615	2.263	0.537	2.65	4.45	1.72	4.616	0.843	$1.3175\cdot10^{-2}$ / 23.187
ε_{av}, %	0.47	0.9	0.1376	0.323	0.55	0.147	1.4	0.31	
ε_F, %	0.294	$6\cdot10^{-2}$	0.17445	0	0	0	0.36	0	
ε_{rf}, %	$8.8\cdot10^{-3}$	0.394	$3.4\cdot10^{-3}$	$1.32\cdot10^{-11}$	0.0544	$1.25\cdot10^{-9}$	0.0136	0.2478	
ε_{max}, %	2.4	3.447	0.89	4.13	6.92	2.9	2.7845	0.563	$1.8175\cdot10^{-2}$ / 32
ε_{av}, %	0.7465	1.33	0.256	0.53	0.9	0.2925	0.908	0.24	
ε_F, %	0.5257	0.1577	0.345	$5.84\cdot10^{-3}$	0	0	1.0	0	
ε_{rf}, %	$2.8\cdot10^{-2}$	0.6	$1.39\cdot10^{-2}$	$3.89\cdot10^{-7}$	0.0454	$3\cdot10^{-11}$	0.1147	0.0563	
ε_{max}, %	1.219	5.11	1.32	5.8	9.6	4.367	3.13	0.977	$2.1375\cdot10^{-2}$ / 40.8
ε_{av}, %	3.385	1.87	0.432	0.8166	1.379	0.51725	1.1168	0.37	
ε_F, %	3.078	0.6541	0.6168	0	0	0	1.05	0	
ε_{rf}, %	1.4	0.8752	$4.67\cdot10^{-2}$	$9.76\cdot10^{-12}$	0.0389	$4.28\cdot10^{-12}$	0.13	0.06	

T a b l e 2. Coefficients of root-polynomial function (2) for Task 1, z_{end} = 0.16 m

Estimation methods	n	Coefficients of root-polynomial function (2)						
		C_6	C_5	C_4	C_3	C_2	C_1	C_0
Lower-order interpolation	4	–	–	$3.14\cdot10^{3}$	$-1.6\cdot10^{3}$	$3.13\cdot10^{4}$	$-2.72\cdot10^{5}$	$8.96\cdot10^{7}$
	5	–	$4.944\cdot10^{-5}$	$9.97\cdot10^{6}$	$-1.33\cdot10^{5}$	$3.1\cdot10^{6}$	$-2.6\cdot10^{7}$	$9.9\cdot10^{9}$
	6	$2.2\cdot10^{4}$	$-1.7\cdot10^{-4}$	$5.487\cdot10^{-5}$	$9.45\cdot10^{6}$	$9.184\cdot10^{-7}$	$-4.778\cdot10^{8}$	$1.04\cdot10^{9}$
Interpolation and extrapolation	4	–	–	$3.06\cdot10^{3}$	$-1.57\cdot10^{3}$	$3.05\cdot10^{4}$	$-2.66\cdot10^{5}$	$8.774\cdot10^{7}$
	5	–	$-8.96\cdot10^{7}$	$3.98\cdot10^{5}$	$-2.03\cdot10^{5}$	$3.9\cdot10^{6}$	$-3.37\cdot10^{7}$	$1.1\cdot10^{8}$
	6	$2.11\cdot10^{4}$	$-1.626\cdot10^{-4}$	$5.239\cdot10^{-5}$	$-9.036\cdot10^{6}$	$8.798\cdot10^{-7}$	$-4.586\cdot10^{8}$	10^{9}
Higher-order interpolation	5	–	$4.84\cdot10^{-4}$	$2.744\cdot10^{-4}$	$6.074\cdot10^{-5}$	$-6.474\cdot10^{-6}$	$-3.245\cdot10^{-7}$	$-5.8\cdot10^{9}$
	6	$2.63\cdot10^{4}$	$-2.03\cdot10^{4}$	$6.53\cdot10^{5}$	$-1.222\cdot10^{-5}$	$1.1\cdot10^{-6}$	$-5.63\cdot10^{8}$	$1.22\cdot10^{9}$

Results of estimations in graphic form for the point $z_{end} = 0.16$ m are presented at Fig. 4.

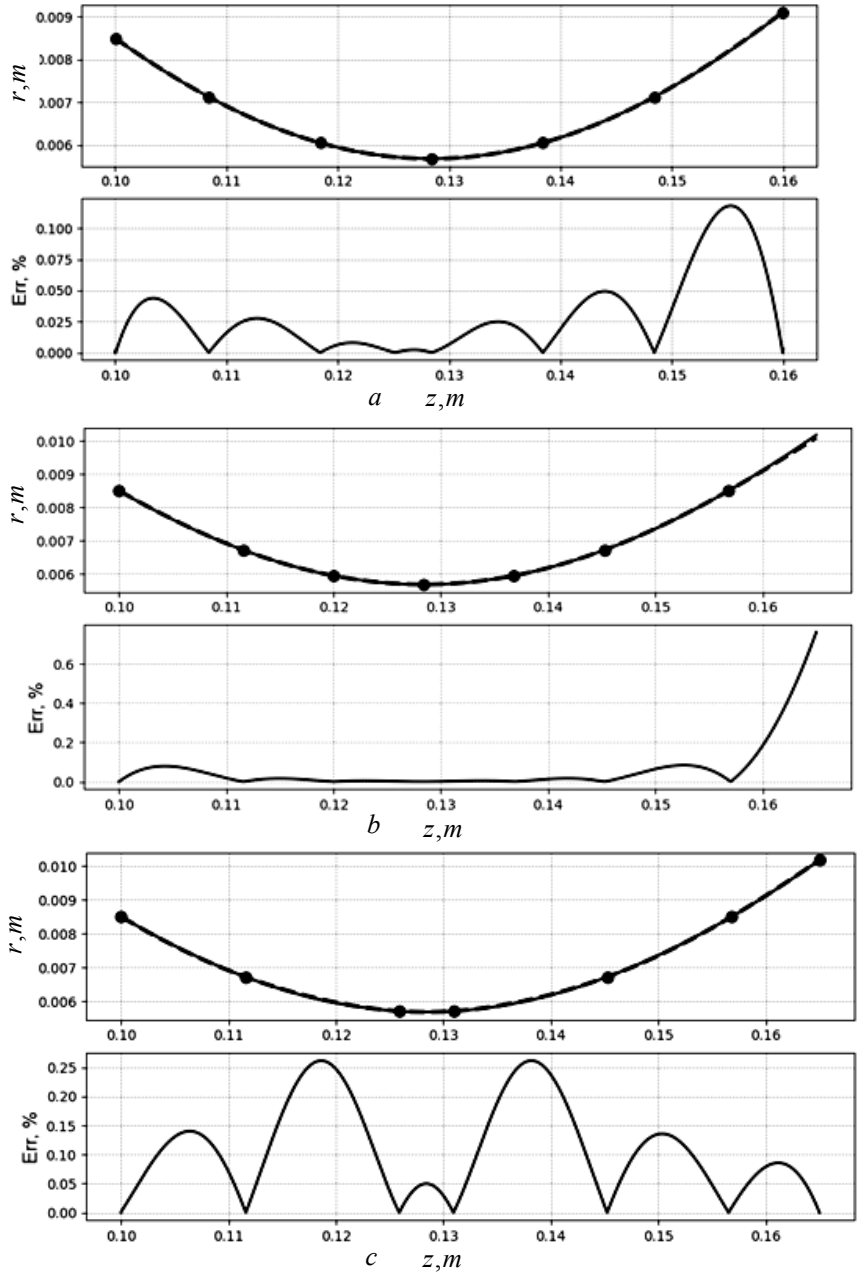


Fig. 4. Lower-order interpolation (*a*), extrapolation (*b*) and higher-order interpolation (*c*) for the Task 1, $n = 6$

In the upper graphs straight line correspond to numerical solving the set of equation (1) and dash line to estimation of numerical solution in dependences on length of propagation of electron beam. On the lower graphs shown the error of estimation in dependences on length of propagation of electron beam (screen copy).

Dependence of error of estimation for extrapolation and higher-order interpolation tasks on the length of extrapolation region L_{add} for different orders of root-polynomial function presented at Fig. 5.

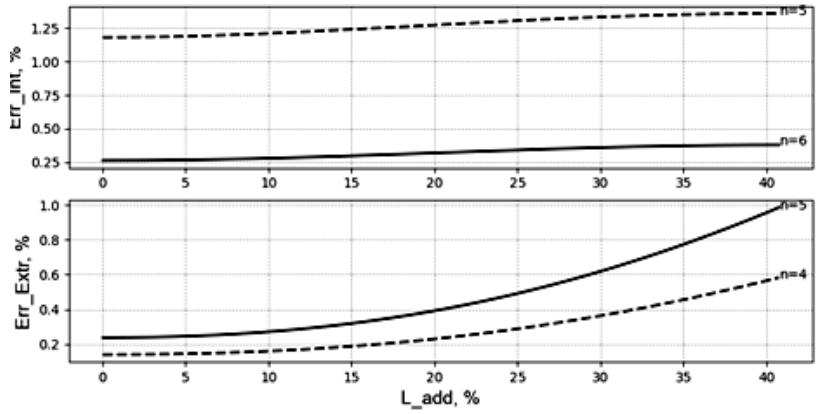


Fig. 5. Dependences of errors of higher-order interpolation (upper) and extrapolation (lower) tasks on the relative length of extrapolation region L_{add} and order of root-polynomial function n for Task 1 (screen copy)

Task 2. $U_{ac} = 15\,\text{kV}$; $I_b = 5.5\,\text{A}$, $p = 4.5\,\text{Pa}$, $r_{srart} = 8.5\,\text{mm}$, $\theta = 10.5^0$, $z_{start} = 0.1\,\text{m}$. End points: $z_{end} = 0.155\,\text{m}$, $z_{end} = 0.153\,\text{m}$, $z_{end} = 0.15\,\text{m}$, $z_{end} = 0.147\,\text{m}$, and $z_{end} = 0.145\,\text{m}$.

For this example, the dependence $r(z)$, defined by numerical solving the set of equations (1), is presented in Fig 6.

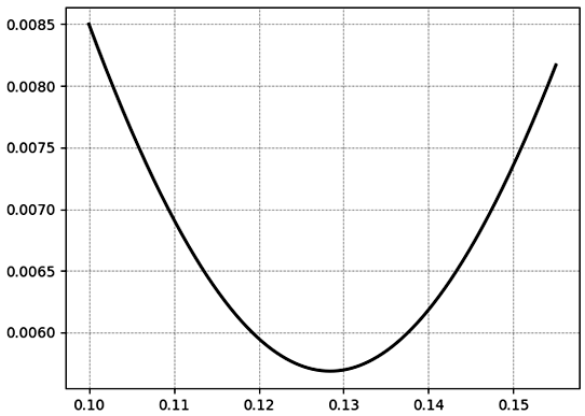


Fig. 6. Dependence $r(z)$ for $U_{ac} = 15\,\text{kV}$, $I_b = 5.5\,\text{A}$, $P = 4{,}5\,\text{Pa}$, end point $z_{end} = 0.155\,\text{m}$ (screen copy)

It is clear that the dependence presented in Fig. 6, corresponding to classification, given at Fig. 2, is a left-hand asymmetric ravine function. The position of boundary point for left-hand asymmetric are always different. Corresponded values for this task are presented at Table 3. Errors in solving interpolation and self-connected interpolation-extrapolation tasks for this example are presented in Table 4. Obtained polynomial coefficients for different estimation methods are given in Table 5.

Table 3. Position of boundary points z_{bp} in depend on position of end point z_{end} for left-hand asymmetric ravine function, given in Task 2

z_{end}, m	0.155	0.153	0.15	0.147	0.145
z_{bp}, m	0.102	0.1038	0.1044	0.1074	0.10942

Table 4. Errors of estimation for Task 2

Type of Error	Estimation methods								L_{add}, m / %
	Interpolation Function Order, n			Extrapolation Function Order, n			Orders n_h of Higher-Order Method		
	4	5	6	4	5	6	5	6	
ε_{max}, %	0.344	0.5224	$6.6\cdot10^{-2}$	0.2357	0.38	0.0735	1.347	0.145	$1.83\cdot10^{-3}$ / 3.44
ε_{av}, %	0.1	0.24	$1.7\cdot10^{-2}$	$9.147\cdot10^{-2}$	0.1887	0.01455	0.4	$5.65\cdot10^{-2}$	
ε_F, %	$3\cdot10^{-2}$	$4.3\cdot10^{-3}$	$4.3\cdot10^{-3}$	0	0	0	0.6425	0	
ε_{rf}, %	$8.62\cdot10^{5}$	0.09	$3.89\cdot10^{-6}$	$1.24\cdot10^{-11}$	0.0634	$1.91\cdot10^{-9}$	$4.426\cdot10^{-2}$	0.0297	
ε_{max}, %	0.344	0.44	$7.54\cdot10^{-2}$	0.38143	0.63	0.164	1.03	0.084	$3.8\cdot10^{-3}$ / 7.79
ε_{av}, %	$9.7\cdot10^{-2}$	0.202	$1.58\cdot10^{-2}$	0.0736	0.134	0.0137	0.3	0.0343	
ε_F, %	$3\cdot10^{-2}$	$4.1\cdot10^{-3}$	$8.26\cdot10^{-3}$	0	0	0	0.425	0	
ε_{rf}, %	$8.62\cdot10^{-2}$	$7.07\cdot10^{-2}$	$1.05\cdot10^{-5}$	$1.34\cdot10^{-11}$	0.0336	$1.14\cdot10^{-9}$	$1.94\cdot10^{-2}$	$1.5\cdot10^{-2}$	
ε_{max}, %	0.3725	0.3685	$8.97\cdot10^{-2}$	0.66	1.17	0.3	0.6386	0.034	$4.415\cdot10^{-3}$ / 9.685
ε_{av}, %	0.096	0.1628	$1.69\cdot10^{-2}$	0.069	0.12	0.019	0.172	0.0185	
ε_F, %	$4.3\cdot10^{-2}$	$3.893\cdot10^{-3}$	$1.16\cdot10^{-2}$	0	0	0	0.214	0	
ε_{rf}, %	$1.79\cdot10^{-4}$	$4.87\cdot10^{-2}$	$1.3\cdot10^{-2}$	$1.3\cdot10^{-11}$	0.012	$3.44\cdot10^{-10}$	$5\cdot10^{-3}$	$5.4\cdot10^{-3}$	
ε_{max}, %	0.3224	0.5224	$6.6\cdot10^{-2}$	0.2357	0.38	$7.35\cdot10^{-2}$	1.347	0.145	$7.422\cdot10^{-3}$ / 18.75
ε_{av}, %	0.1054	0.24	$1.7\cdot10^{-2}$	$9.15\cdot10^{-2}$	0.1887	$1.45\cdot10^{-2}$	0.4	$5.65\cdot10^{-2}$	
ε_F, %	0.017	$4.3\cdot10^{-3}$	$4.3\cdot10^{-3}$	0	0	0	0.6425	0	
ε_{rf}, %	$2.52\cdot10^{-5}$	$8.9\cdot10^{-2}$	$3.89\cdot10^{-2}$	$1.24\cdot10^{-11}$	$6.34\cdot10^{-2}$	$1.91\cdot10^{-9}$	$4.4\cdot10^{-2}$	$2.97\cdot10^{-2}$	
ε_{max}, %	0.4	0.398	0.112361	1.107	2.073	0.58	0.22	$7.66\cdot10^{-2}$	$9.42\cdot10^{-2}$ / 26.5
ε_{av}, %	0.11	0.129	$2.2\cdot10^{-2}$	0.11	0.2065	$4.81\cdot10^{-2}$	0.084	$1.66\cdot10^{-2}$	
ε_F, %	0.042	$3.5\cdot10^{-3}$	$1.05\cdot10^{-2}$	0	0	0	0.06	0	
ε_{rf}, %	$1.95\cdot10^{-4}$	0.024	$1.31\cdot10^{-5}$	$9.1\cdot10^{-12}$	$1.54\cdot10^{-3}$	$9.76\cdot10^{-10}$	$4.08\cdot10^{-4}$	$7.3\cdot10^{-4}$	

Table 5. Coefficients of root-polynomial function (2) for Task 2, $z_{end} = 0.15$ m

Estimation methods	n	Coefficients of root-polynomial function (2)						
		C_6	C_5	C_4	C_3	C_2	C_1	C_0
Lower-order interpolation	4	–	–	$2.95\cdot10^{3}$	$-1.512\cdot10^{-3}$	$2.956\cdot10^{-4}$	$-2.582\cdot10^{-5}$	$8.538\cdot10^{-7}$
	5	–	$-9.963\cdot10^{-5}$	$9.966\cdot10^{-5}$	$-3.4734\cdot10^{-5}$	$5.6475\cdot10^{-6}$	$-4.408\cdot10^{-7}$	$1.34\cdot10^{-8}$
	6	$2\cdot10^{-4}$	$-1.549\cdot10^{-4}$	$5\cdot10^{-5}$	$-8.64\cdot10^{-6}$	$8.43\cdot10^{-7}$	$-4.4046\cdot10^{-8}$	$9.628\cdot10^{-10}$
Interpolation and extrapolation	4	–	–	$2.8\cdot10^{-3}$	$-1.435\cdot10^{-3}$	$2.792\cdot10^{-4}$	$-2.43\cdot10^{-5}$	$8.06\cdot10^{-7}$
	5	–	$-8.5335\cdot10^{-7}$	$3.25\cdot10^{-5}$	$-1.655\cdot10^{-5}$	$3.198\cdot10^{-6}$	$-2.766\cdot10^{-7}$	$9.032\cdot10^{-9}$
	6	$1.84\cdot10^{-4}$	$-1.417\cdot10^{-4}$	$4.57\cdot10^{-5}$	$-7.9\cdot10^{-6}$	$7.7\cdot10^{-7}$	$-4.03\cdot10^{-8}$	$8.821\cdot10^{-10}$
Higher-order interpolation	5	–	$4.03\cdot10^{-4}$	$2.9\cdot10^{-4}$	$-8.2\cdot10^{-5}$	$1.15\cdot10^{-5}$	$-8.018\cdot10^{-7}$	$2.227\cdot10^{-8}$
	6	$2.22\cdot10^{-4}$	$-1.71\cdot10^{-4}$	$5.51\cdot10^{-5}$	$-9.495\cdot10^{-6}$	$9.235\cdot10^{-7}$	$-4.81\cdot10^{-8}$	$1.046\cdot10^{-9}$

Results of estimations in graphic form for the point $z_{end} = 0.15$ m are presented at Fig. 7. Dependence of error of estimation for extrapolation and higher-order interpolation tasks on the length of extrapolation region L_{add} for different orders of root-polynomial function presented at Fig. 8.

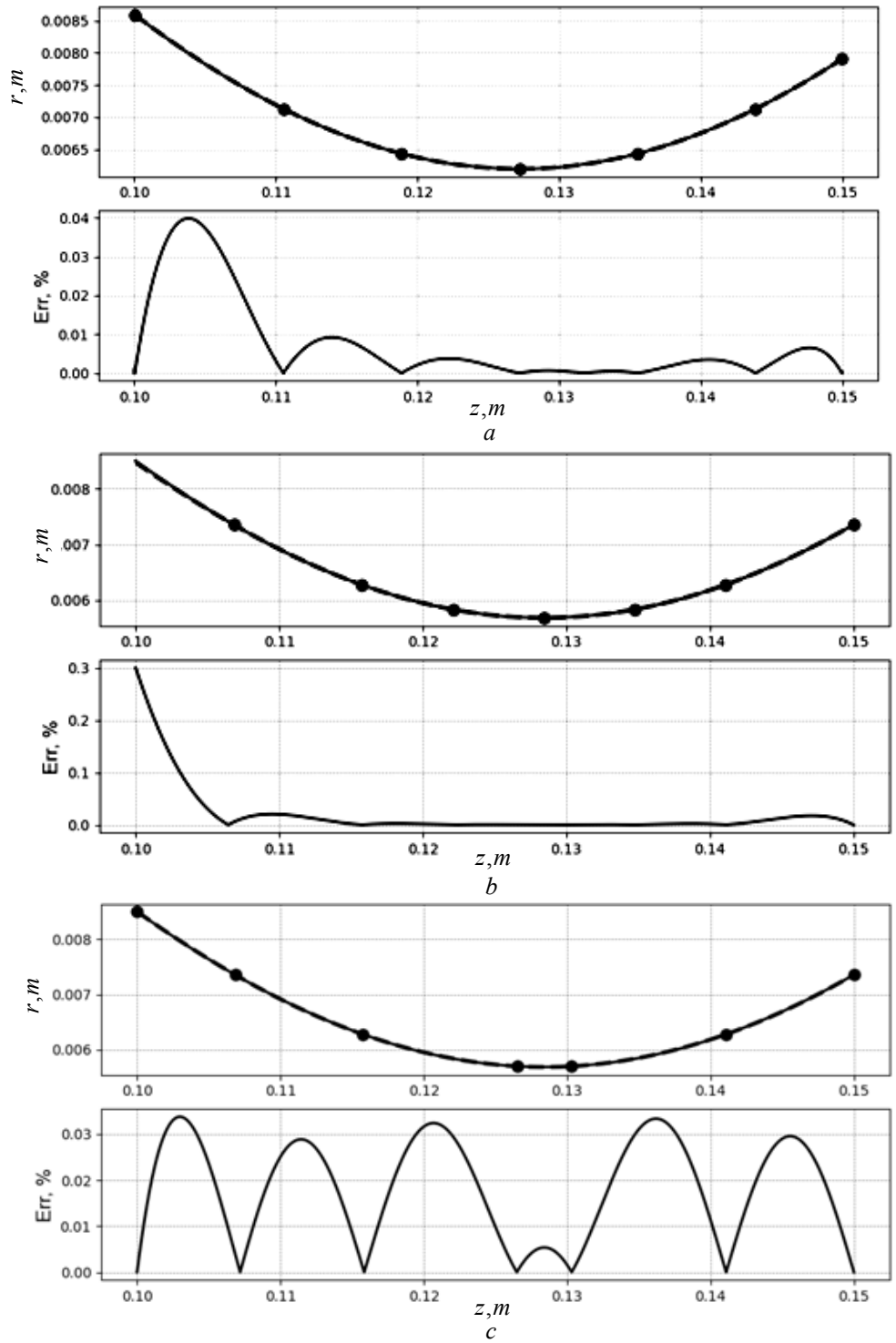


Fig. 7. Lower-order interpolation (*a*), extrapolation (*b*) and higher-order interpolation (*c*) for the Task 2, $n = 6$. In the upper graphs straight line correspond to numerical solving the set of equation (1) and dash line to estimation of numerical solution in dependences on length of propagation of electron beam. On the lower graphs shown the error of estimation in dependences on length of propagation of electron beam (screen copy)

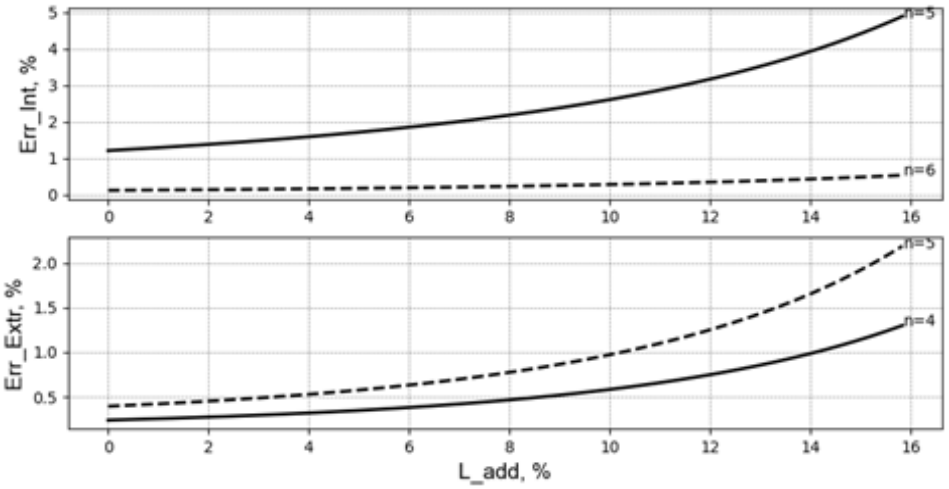


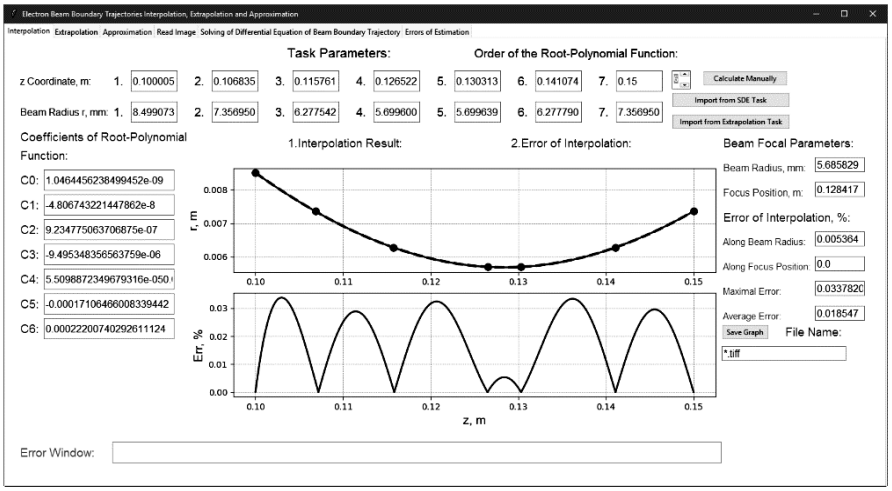
Fig. 8. Dependences of errors of higher-order interpolation (upper) and extrapolation (lower) tasks on the relative length of extrapolation region L_{add} and order of root-polynomial function n for Task 1 (screen copy)

PARTICULARITIES OF ELABORATED COMPUTER SOFTWARE

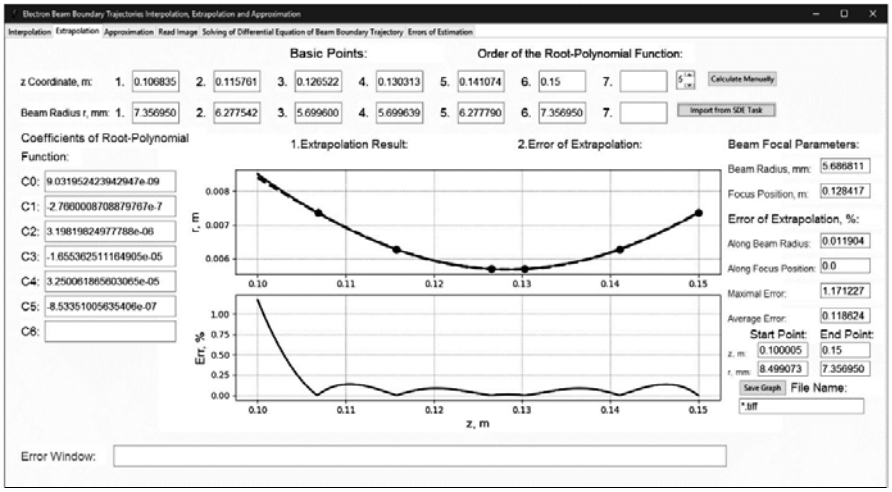
All simulation results presented in this paper have been obtained using original software, which has been elaborated for simulation and numerical estimation of the boundary trajectory of an electron beam propagated in ionized gas. The source program code has been written using the means of programing language Python, including advanced mathematic and graphic libraries such as tkinter, nympy, and matplotlib [59; 60]. The distinguishing feature of elaborated software from the point of view of the means of programming is including additional advanced libraries for creating scientific plots from module matplotlib into traditional elements of the interface window created using the function of module tkinter [59; 60]. For the correct solution of this sophisticated programming task, specific system tools have been used, including the definition of virtual variables and creating on its base the virtual environment for forming a virtual disk in the operative memory of a local computer [59; 60]. Corresponding graphic interface windows of elaborated software for the bookmarks "Interpolation" and "Extrapolation" are presented in Fig. 9. For saving and further analyzing the obtained graphic information, the bottom "Save Graph" has been provided in both interface windows.

For automatic creation of root-polynomial functions on both bookmarks, the bottoms "Import from SDE Task" have been provided. Using this program's functionality is possible only after solving the simulation task for the established electron beam parameters in the corresponded bookmark "Solving of Differential Equation of Beam Boundary Trajectory". But the manual creation of the root-polynomial function by the r and z coordinates, which have to be input in the corresponded textboxes, is also possible by pressing the bottom "Calculate Manually".

Errors of estimation, presented in Tables 1 and 4, as well as coefficients of root-polynomial functions, presented in Tables 2 and 5, are written out in the established output text windows on the corresponded bookmarks. All described elements of the graphic user interface are shown in the copy of these bookmarks, presented in Fig. 9.



a



b

Fig. 9. Interface windows for bookmarking “Interpolation” (*a*) and “Extrapolation” (*b*) in elaborated computer software (screen copy)

ANALYSIS OF OBTINED RESULTS AND DISCUSSION

The computer simulation results described in this paper showed that higher-order interpolation for asymmetric ravine functions gives an average error value. No minimum error value was detected for this novel estimation method. In general, from a theoretical point of view, this is due to the location of the reference points for root-polynomial functions of the appropriate order. Indeed, the k_f values determined by relations (5), (6) were chosen correctly only for the corresponded lower-order of the odd or even root polynomial function (2).

For example, for higher-order interpolation with order of function $n_h = 5$, the basic points are located as for forth order symmetric function, and additional point, located at the start of interpolated interval for left-hand asymmetric function or on the end of this interval for right-hand asymmetric function, is artificially added.

Generally, corresponding to Tables 1 and 4, minimal values of maximal and average interpolation error are corresponded to standard low-order interpolation,

but self-connected interpolation-extrapolation task usually given the minimal errors in estimation of focal parameters of electron beam. The same conclusion are follows from graphic dependences, presented at Fig. 4 and Fig. 7.

But, in any case, average integral error of estimation the beam trajectory by the higher-order root-polynomial function in the whole segment of interpolation isn't so large, therefore such estimation can be preferable in some solutions for practice application. For simplifying the further corresponded analysis in the digital presentation all estimation errors for the end point $z_{end} = 0.15$ are rewritten from extended Table 4 to smaller Table 6.

Table 6. Errors of estimation for Task 2 for end point $z_{end} = 0.15$ m

Methods and function order		**Standard Interpolation**			**Interpolation and Extrapolation**			**Higher-Order Interpolation**	
N		4	5	6	4	5	6	5	6
Errors	ε_{max}, %	0.3725	0.3685	$897 \cdot 10^{-2}$	0.66	1.17	0.3	0.6386	0.034
	ε_{av}, %	0.096	0.1628	$1.69 \cdot 10^{-2}$	0.069	0.12	0.019	0.172	0.0185
	ε_F, %	$4.3 \cdot 10^{-2}$	$3.893 \cdot 10^{-3}$	1.1610^{-2}	0	0	0	0.214	0
	ε_{rf}, %	$1.79 \cdot 10^{-4}$	$4.87 \cdot 10^{-2}$	$1.3 \cdot 10^{-5}$	$1.3 \cdot 10^{-11}$	0.012	$3.44 \cdot 10^{-10}$	$5 \cdot 10^{-3}$	$5.4 \cdot 10^{-3}$

From the calculation results, presented in Table 6, it is clear, that for higher-order interpolation the for $n = 6$ average error ($\varepsilon_{av} = 0.0185$ %) isn't so small, than for standard interpolation by the function of same order ($\varepsilon_{av} = 0.0169$ %), but the difference of these errors isn't so large. Also, and it is very significant and important that the estimation using higher-order interpolation for $n = 6$ gives the minimal value of the maximal error, $\varepsilon_{max} = 0.034$ %.

It is clear also from numerical data, presented in Table 6, that the best results for estimation of focal radius of electron beam giving the method of interpolation and extrapolation by forth and six order functions, the level of error ε_{rf} is range of from 10^{-11} % to 10^{-10} %. But such precision estimation of focal beam parameters usually isn't necessary for the practical applications. Estimation using higher-order interpolation method give the value of error $\varepsilon_{rf} = 5.4 \cdot 10^{-3}$ %, which, certainly, isn't so small, but usually is suitable for the most of practical applications [16]. It is also interesting and important, that for self-connected interpolation and extrapolation method the error of estimation focus position is $\varepsilon_F = 0$ %, but the same result is observed for higher-order interpolation function in the case of $n = 6$.

As it is clear from Tables 1 and 4, the particularities of the different methods of interpolation and extrapolation described above are similar for all positions of the end point, including left-hand and right-hand ravine functions. But, in any case, the error in the estimation of electron beam boundary trajectories by using the root-polynomial function (2) is very small, in the range of a fraction of a percent. This result is confirming the pervious preliminary theoretical estimations, have been provided in the works [47–49].

All research work described in this paper has been provided in the Scientific and Educational Laboratory of Electron Beam Technological Devices of the National Technical University of Ukraine "Igor Sikorsky Kyiv Polytechnical Institute".

CONCLUSION

Generally, provided research has shown that usually the minimal average error ε_{av} of estimation of the boundary electron beam trajectory using the root-polynomial

function (2) corresponds to the lower-order interpolation method. The best orders of these functions are even values, such as $n = 2$, $n = 4$, and $n = 6$. The best estimations of electron beam focal parameters have been obtained using the self-connected interpolation-extrapolation method. The level of error in the estimation of the focal beam radius ε_{rf} for this method has been significantly small, ranging from 10^{-11} % to 10^{-10} %, and the estimation by the focus position has been exactly precise without error. The best results for this method also give the even values of the order of the root-polynomial function, such as $n = 4$ and $n = 6$. It can be generally explained by the suitable choice of base points position for the symmetric part of the ravine function, which is evaluated. The proposed method of higher-order root-polynomial interpolation gives an average value of error both in the focal region and at the start and end basic points. The larger values of the average error in this case are explained by the location of the basic points. Unfortunately, solving the optimization task of defining the basic points position in this case is impossible.

All simulation results presented in this paper have been obtained using original computer software elaborated and developed by applying the advanced mathematical and graphic means of the Python programming language.

Obtained scientific results and practical recommendations can be interesting to a wide range of experts in the fields of the physics of electron beams and advanced electron beam technologies, as well as in the computational mathematics and methods of interpolation and extrapolation of ravine functions.

REFERENCES

1. M. Reiser, *Theory and Design of Charged Particle Beams*. John Wiley & Sons, 2008, 634 p. Available: https://www.wiley.com/en-us/Theory+and+ Design +of+Charged+ Particle+Beams-p-9783527617630
2. M. Szilagyi, *Electron and Ion Optics*. Springer Science & Business Media, 2012, 539 p. Available: https://www.amazon.com/Electron-Optics-Microdevices-Miklos-Szilagyi/dp/1461282470
3. S.J.R. Humphries, *Charged Particle Beams*. Courier Corporation, 2013, 834 p. Available: https://library.uoh.edu.iq/admin/ebooks/76728-charged-particle-beams---s.-humphries.pdf
4. R.C. Davidson, H. Qin, *Physics of Intense Charged Particle Beams in High Energy Accelerators*. World Scientific, Singapore, 2001, 604 p. Available: https://books.google.com.ua/books/about/Physics_Of_Intense_Charged_Particle_Beam.html?id=5M02DwAAQBAJ&redir_esc=y
5. G. Brewer, *Electron-Beam Technology in Microelectronic Fabrication*. Elsevier, 2012, 376 p. Available: https://books.google.com.ua/books?id= snU5sOQD6noC&hl=uk&source=gbs_similarbooks
6. J.D. Lawson, *The Physics of Charged-Particle Beams*. Oxford: Clarendon Press, 1977, 446 p. Available: https://www.semanticscholar.org/paper/The-Physics-of-Charged-Particle-Beams-Stringer/80b5ee5289d5efd8f480b516ec4bade0aa529ea6
7. S. Schiller, U. Heisig, and S. Panzer, *Electron Beam Technology*. John Wiley & Sons Inc, 1995, 508 p. Available: https://books.google.com.ua/books/about/ Electron_ Beam_Technology.html?id=QRJTAAAAMAAJ&redir_esc=y
8. H. Schultz, *Electron Beam Welding*. Woodhead Publishing, 1993, 240 p. Available: https://books.google.com.ua/books?id=I0xMo28DwcIC&hl=uk&source= gbs_book_similarbooks
9. R.A. Bakish, *Introduction to Electron Beam Technology*. Wiley, 1962, 452 p. Available: https://books.google.com.ua/books?id=GghTAAAAMAAJ&hl= uk&source= gbs_similarbooks

10. T. Kemmotsu, T. Nagai, and M. Maeda, "Removal Rate of Phosphorous form Melting Silicon," *High Temperature Materials and Processes*, vol. 30, issue 1–2, pp. 17–22, 2011. Available: https://www.degruyter.com/journal/key/ htmp/30/1-2/html
11. J.C.S. Pires, A.F.B. Barga, and P.R. May, "The purification of metallurgically grade silicon by electron beam melting," *Journal of Materials Processing Technology*, vol. 169, no. 1, pp. 347–355, 2005. Available: https://www.academia.edu/9442020/ The_purification_of_metallurgical_grade_silicon_by_electron_beam_melting
12. D. Luo, N. Liu, Y. Lu, G.Zhang, and T. Li, "Removal of impurities from metallurgically grade silicon by electron beam melting," *Journal of Semiconductors*, vol. 32, issue 3, article ID 033003, 2011. Available: http://www.jos.ac.cn/en/article/ doi/10.1088/1674-4926/32/3/033003
13. D. Jiang, Y. Tan, S. Shi, W. Dong, Z. Gu, and R. Zou, "Removal of phosphorous in molten silicon by electron beam candle melting," *Materials Letters*, vol. 78, pp. 4–7, 2012.
14. A.A. Druzhinin, I.P. Ostrovskii, Y.N. Khoverko, N.S.Liakh-Kaguy, and A.M. Vuytsyk, "Low temperature characteristics of germanium whiskers," *Functional materials 21*, no. 2, pp. 130–136, 2014. Available: http://dspace.nbuv.gov.ua/bitstream/ handle/123456789/120404/02-Druzhinin.pdf?sequence=1
15. A.A. Druzhinin, I.A. Bolshakova, I.P. Ostrovskii, Y.N. Khoverko, and N.S. Liakh-Kaguy, "Low temperature magnetoresistance of InSb whiskers," *Materials Science in Semiconductor Processing*, vol. 40, pp. 550–555, 2015. Available: https://academic-accelerator.com/search?Journal=Druzhinin
16. I. Melnyk, S. Tuhai, M. Surzhykov, I. Shved, V. Melnyk, and D. Kovalchuk, "Analytical Estimation of the Deep of Seam Penetration for the Electron-Beam Welding Technologies with Application of Glow Discharge Electron Guns," *2022 IEEE 41-st International Conference on Electronics and Nanotechnology (ELNANO), 2022*, pp. 1–5. doi: 10.1109/ELNANO54667_2022_9927071
17. I. Melnyk, S. Tuhai, and A. Pochynok, "Calculation of Focal Paramters of Electron Beam Formed in Soft Vacuum at the Plane which Sloped to Beam Axis," *The Forth IEEE International Conference on Information-Communication Technologies and Radioelectronics UkrMiCo'2019. Collections of Proceedings of the Scientific and Technical Conference, Odesa, Ukraine, September 9-13, 2019*. doi: 10.1109/UkrMiCo47782.2019.9165328
18. A. Zakharov, S. Rozenko, S. Litvintsev, and M. Ilchenko, "Trisection Bandpass Filter with Mixed Cross-Coupling and Different Paths for Signal Propagation," *IEEE Microwave Wireless Component Letters*, vol. 30, no. 1, pp. 12–15, Jan. 2020. doi: 10.1109/LMWC.2019.2957207
19. A. Zakharov, S. Litvintsev, and M. Ilchenko, "Trisection Bandpass Filters with All Mixed Couplings," *IEEE Microwave Wireless Components Letter*, vol. 29, no. 9, pp. 592–594, 2019. Available: https://ieeexplore.ieee.org/abstract/document/ 8782802
20. A. Zakharov, S. Rozenko, and M. Ilchenko, "Varactor-tuned microstrip bandpass filter with loop hairpin and combline resonators," *IEEE Transactions on Circuits Systems. II. Experimental Briefs*, vol. 66, no. 6, pp. 953–957, 2019. Available: https://ieeexplore.ieee.org/document/8477112
21. A. Mitchell, T. Wang, "Electron beam melting technology review," *Proceedings of the Conference "Electron Beam Melting and Refining State of the Art 2000, Reno, NV, USA, 2000*, ed. R. Bakish, pp. 2–13.
22. D.V. Kovalchuk, N.P. Kondraty, "Electron-beam remelting of titanium – problems and development prospects," *Titan 2009*, no. 1(23), pp. 29–38.
23. V.A. Savenko, N.I. Grechanyuk, and O.V. Churakov, "Electron beam refining in production of platinum and platinum-base alloys. Information 1. Electron beam refining of platinum," *Advances in Electrometallurgy*, no. 1, pp. 14–16, 2008.
24. J. Zhang et al., "Fine equiaxed β grains and superior tensile property in Ti–6Al–4V alloy deposited by coaxial electron beam wire feeding additive manufacturing," *Acta Metallurgica Sinica (English Letters)*, 33(10), pp. 1311–1320, 2020. doi: 10.1007/s40195-020-01073-5
25. D. Kovalchuk, O. Ivasishin, "Profile electron beam 3D metal printing," in *Additive Manufacturing for the Aerospace Industry*. Elsevier Inc., 2019, pp. 213–233.
26. M. Wang et al., "Microstructure and mechanical properties of Ti-6Al-4V cruciform structure fabricated by coaxial electron beam wire-feed additive manufacturing," *Journal of Alloys and Compounds*, vol. 960. article 170943. doi: https://doi.org/10.1016/ j.jallcom.2023.170943

27. T.O. Prikhna et al., "Electron-Beam and Plasma Oxidation-Resistant and Thermal-Barrier Coatings Deposited on Turbine Blades Using Cast and Powder Ni(Co)CrALY(Si) Alloys I. Fundamentals of the Production Technology, Structure, and Phase Composition of Cast NiCrAlY Alloys," *Powder Metallurgy and Metal Ceramics*, vol. 61, issue 1-2, pp. 70–76, 2022. doi: 10.1007/s11106-022-00320-x
28. T.O. Prikhna et al., "Electron-Beam and Plasma Oxidation-Resistant and Thermal-Barrier Coatings Deposited on Turbine Blades Using Cast and Powder Ni(Co)CrAlY(Si) Alloys Produced by Electron-Beam Melting II. Structure and Chemical and Phase Composition of Cast CoCrAlY Alloys," *Powder Metallurgy and Metal Ceramicsthis*, vol. 61, issue 3-4, pp. 230–237, 2022. doi: 10.1007/s11106-023-00333-0
29. I.M. Grechanyuk et al., "Electron-Beam and Plasma Oxidation-Resistant and Thermal-Barrier Coatings Deposited on Turbine Blades Using Cast and Powder Ni(Co)CrAlY(Si) Alloys Produced by Electron Beam Melting IV. Chemical and Phase Composition and Structure of Cocralysi Powder Alloys and Their Use," *Powder Metallurgy and Metal Ceramics*, vol. 61, issue 7-8, pp. 459–464, 2022. doi: 10.1007/s11106-022-00310-z
30. M.I. Grechanyuk et al., "Electron-Beam and Plasma Oxidation-Resistant and Thermal-Barrier Coatings Deposited on Turbine Blades Using Cast and Powder Ni (Co)CrAlY (Si) Alloys Produced by Electron Beam Melting III. Formation, Structure, and Chemical and Phase Composition of Thermal-Barrier Ni(Co)CrAlY/ZrO_2–Y_2O_3 Coatings Produced by Physical Vapor Deposition in One Process Cycle," *Powder Metallurgy and Metal Ceramics*, vol. 61, issue 5-6, pp. 328–336, 2022. doi: 10.1007/s11106-022-00320-x
31. A.F. Tseluyko, V.T. Lazurik, D.L. Ryabchikov, V.I. Maslov, and I.N. Sereda, "Experimental study of radiation in the wavelength range 12.2-15.8 nm from a pulsed high-current plasma diode," *Plasma Physics Reports*, 34(11), pp. 963–968, 2008. doi: 10.1134/S1063780X0811010X
32. V.G. Rudychev, V.T. Lazurik, and Y.V. Rudychev, "Influence of the electron beams incidence angles on the depth-dose distribution of the irradiated object," *Radiation Physics and Chemistry*, 186, 109527, 2021. doi: 10.1016/j.radphyschem.2021.109527
33. V.M. Lazurik, V.T. Lazurik, G. Popov, and Z. Zimek, "Two-parametric model of electron beam in computational dosimetry for radiation processing, *Radiation Physics and Chemistry*, 124, pp. 230–234, 2016. doi: 10.1016/j.radphyschem. 2015.12.00
34. I. Melnyk, S. Tuhai, and A. Pochynok, "Universal Complex Model for Estimation the Beam Current Density of High Voltage Glow Discharge Electron Guns," *Lecture Notes in Networks and Systems*; Eds: M. Ilchenko, L. Uryvsky, L. Globa, vol. 152, pp. 319–341, 2021. doi: 10.1007/978-3-030-58359-0_18
35. I.V. Melnyk, "Numerical simulation of distribution of electric field and particle trajectories in electron sources based on high-voltage glow discharge," *Radioelectronic and Communication Systems*, vol. 48, no. 6, pp. 61–71, 2005. doi: https://doi.org/10.3103/S0735272705060087
36. S.V. Denbnovetsky, J. Felba, V.I. Melnik, and I.V. Melnik, "Model of Beam Formation in a Glow Discharge Electron Gun with a Cold Cathode," *Applied Surface Science*, 111, pp. 288–294, 1997. doi: 10.1016/S0169-4332(96)00761-1
37. S.V. Denbnovetsky, V.G. Melnyk, and I.V. Melnyk, "High voltage glow discharge electron sources and possibilities of its application in industry for realizing different technological operations," *IEEE Transactions on Plasma Science*, vol. 31, issue 5, pp. 987–993, October, 2003. doi: 10.1109/TPS.2003.818444
38. S. Denbnovetskiy et al., "Principles of operation of high voltage glow discharge electron guns and particularities of its technological application," *Proceedings of SPIE, The International Society of Optical Engineering*, pp. 10445–10455, 2017. doi: 10.1117/12.2280736
39. S.V. Denbnovetsky, V.I. Melnyk, I.V. Melnyk, and B.A. Tugay, "Model of control of glow discharge electron gun current for microelectronics production applications," *Proceedings of SPIE. Sixth International Conference on "Material Science and Material Properties for Infrared Optoelectronics"*, vol. 5065, pp. 64–76, 2003. doi: https://doi.org/10.1117/12.502174
40. I.V. Melnyk, S.B. Tugay, "Analytical calculations of anode plasma position in high-voltage discharge range in case of auxiliary discharge firing," *Radioelectronic and Communication Systems*, vol. 55(11), pp. 50–59. 2012. doi: https://doi.org/10.3103/S0735272712110064
41. I.V. Melnyk, "Estimating of current rise time of glow discharge in triode electrode system in case of control pulsing," *Radioelectronic and Communication Systems*, vol. 56, no. 12, pp. 51–61, 2017. doi: 10.3103/S0735272713120066

42. S.V. Denbnovetskiy, I.V. Melnyk, V.G. Melnyk, B.A. Tugai, and S.B. Tuhay, "Investigation of Emission Properties of Cold Cathodes in Triode Impulse High Voltage Glow Discharge Electron Guns," *XXXV IEEE International Scientific Conference "Electronic And Nanotechnology (ELNANO)", Conference Proceedings, Kyiv, Ukraine, April 21-24, 2015*, pp. 450–453. doi: 10.1109/ELNANO.2015.7146931
43. I.V. Melnyk, "Improving Estimation of Rising Time of High Voltage Glow Discharge Current in Triode Electrodes Systems with Taking into Account Changing of Anode Plasma Parameters," *XXXV IEEE International Scientific Conference "Electronic And Nanotechnology (ELNANO)", Conference Proceedings, Kyiv, Ukraine, April 21-24, 2015,* pp. 461–464. doi: 10.1109/ELNANO.2015.7146930
44. R.W. Hockney, J.W. Eastwood, *Computer Simulation Using Particles*. CRC Press, 1988, 540 p.
45. A.O. Luntovskyy, I.V. Melnyk, "Simulation of Technological Electron Sources with Use of Parallel Computing Methods," *XXXV IEEE International Scientific Conference "Electronic And Nanotechnology (ELNANO)", Conference Proceedings, Kyiv, Ukraine, April 21-24, 2015*, pp. 454 – 460. doi: 10.1109/ELNANO.2015.7146929
46. I. Melnyk, A. Luntovskyy, "Estimation of Energy Efficiency and Quality of Service in Cloud Realizations of Parallel Computing Algorithms for IBN," in Klymash, M., Beshley, M., Luntovskyy, A. (eds) *Future Intent-Based Networking. Lecture Notes in Electrical Engineering*, vol. 831, Springer, Cham, pp. 339–379. doi: https://doi.org/10.1007/978-3-030-92435-5_20
47. I. Melnyk, S. Tuhai, and A. Pochynok, "Interpolation of the Boundary Trajectories of Electron Beams by the Roots from Polynomic Functions of Corresponded Order," *2020 IEEE 40th International Conference on Electronics and Nanotechnology (ELNANO)*, pp. 28–33. doi: 10.1109/ELNANO50318.2020.9088786
48. I. Melnik, S. Tugay, and A. Pochynok, "Interpolation Functions for Describing the Boundary Trajectories of Electron Beams Propagated in Ionised Gas," *15-th International Conference on Advanced Trends in Radioelectronics, Telecommunications and Computer Engineering (TCSET – 2020)*, pp. 79–83. doi: 10.1109/TCSET49122.2020.235395
49. I.V. Melnyk, A.V. Pochynok, "Study of a Class of Algebraic Functions for Interpolation of Boundary Trajectories of Short-Focus Electron Beams," *System Researches and Information Technologies*, no. 3, pp. 23–39, 2020. doi: https://doi.org/10.20535/SRIT.2308-8893.2020.3.02
50. I. Melnyk, S. Tuhai, M. Skrypka, A. Pochynok, and D. Kovalchuk, "Approximation of the Boundary Trajectory of a Short-Focus Electron Beam using Third-Order Root-Polynomial Functions and Recurrent Matrixes Approach," *2023 International Conference on Information and Digital Technologies (IDT)*, Zilina, Slovakia, 2023, pp. 133–138, doi: 10.1109/IDT59031.2023.10194399
51. J.I. Etcheverry, N. Mingolo, J.J. Rocca, and O.E. Martınez, "A Simple Model of a Glow Discharge Electron Beam for Materials Processing," *IEEE Transactions on Plasma Science*, vol. 25, no. 3, pp. 427–432, June, 1997. doi: 10.1109/27.597256
52. G.M. Phillips, *Interpolation and Approximation by Polynomials*. Springer, 2023, 312 p. Available: http://bayanbox.ir/view/2518803974255898294/George-M.-Phillips-Interpolation-and-Approximation-by-Polynomials-Springer-2003.pdf
53. N. Draper, H. Smith, *Applied Regression Analysis*; 3 Edition. Wiley Series, 1998, 706 p. Available: https://www.wiley.com/en-us/Applied+Regression+Analysis,+3rd+Edition-p-9780471170822
54. C. Mohan, K. Deep, *Optimization Techniques*. New Age Science, 2009, 628 p. Available: https://www.amazon.com/Optimization-Techniques-C-Mohan/dp/ 1906574219
55. M.K. Jain, S.R.K. Iengar, and R.K. Jain, *Numerical Methods for Scientific & Engineering Computation*. New Age International Pvt. Ltd., 2010, 733 p. Available: https://www.google.com.ua/url?sa=t&rct=j&q=&esrc=s&source=web&cd=&ved=2ahUKEwippcuT7rX8AhUhlYsKHRfBCG0QFnoECEsQAQ&url=https%3A%2F%2Fwww.researchgate.net%2Fprofile%2FAbiodun_Opanuga%2Fpost%2Fhow_can_solve_a_non_linear_PDE_using_numerical_method%2Fattachment%2F59d61f7279197b807797de30%2FAS%253A284742038638596%25401444899200343%2Fdownload%2FNumerical%2BMethods.pdf&usg=AOvVaw0MjNl3K877lVWUWw-FPwmV
56. S.C. Chapra, R.P. Canale, *Numerical Methods for Engineers*; 7th Edition. McGraw Hill, 2014, 992 p. Available: https://www.amazon.com/Numerical-Methods-Engineers-Steven-Chapra/dp/007339792X

57. E. Wentzel, L. Ovcharov, *Applied Problems of Probability Theory*. Mir, 1998, 432 p. Available: https://mirtitles.org/2022/06/03/applied-problems-in-probability-theory-wentzel-ovcharov/
58. J.A. Gubner, *Probability and Random Processes for Electrical and Computer Engineers*. UK, Cambridge: Cambridge University Press, 2006. Available: http:// www.amazon.com/Probability-Processes-Electrical-Computer-Engineers/dp/ 0521864704
59. M. Lutz, *Learning Python*; 5th Edition. O'Reilly, 2013, 1643 p.
60. W. McKinney, *Python for Data Analysis: Data Wrangling with Pandas, NumPy, and Jupyter*; 3rd Edition. O'Reilly Media, 2023, 579 p.

Received 11.11.2023

INFORMATION ON THE ARTICLE

Igor V. Melnyk, ORCID: 0000-0003-0220-0615, National Technical University of Ukraine "Igor Sikorsky Kyiv Polytechnic Institute", Ukraine, e-mail: imelnik@phbme.kpi.ua

Alina V. Pochynok, ORCID: 0000-0001-9531-7593, Research Institute of Electronics and Microsystem Technology of the National Technical University of Ukraine "Igor Sikorsky Kyiv Polytechnic Institute", Ukraine, e-mail: alina_pochynok@yahoo.com

Mykhailo Yu. Skrypka, ORCID: 0009-0006-7142-5569, National Technical University of Ukraine "Igor Sikorsky Kyiv Polytechnic Institute", Ukraine, e-mail: scientetik@gmail.com

УДОСКОНАЛЕНИЙ МЕТОД ІНТЕРПОЛЯЦІЇ ГРАНИЧНИХ ТРАЄКТОРІЙ КОРОТКОФОКУСНИХ ЕЛЕКТРОННИХ ПУЧКІВ ЗА ДОПОМОГОЮ КОРЕНЕВИХ ПОЛІНОМІАЛЬНИХ ФУНКЦІЙ ВИЩОГО ПОРЯДКУ ТА ЙОГО ПОРІВНЯЛЬНЕ ДОСЛІДЖЕННЯ / І.В. Мельник, А.В. Починок, М.Ю. Скрипка

Анотація. Розглянуто та обговорено узагальнене порівняння трьох сучасних, нових методів оцінювання граничної траєкторії електронних пучків, що поширюються в іонізованому газі, включаючи інтерполяцію нижчого порядку, самоузгоджену інтерполяцію та екстраполяцію, а також інтерполяцію вищого порядку. Усі оцінки відповідних похибок були проведені відносно числового розв'язування системи алгебра-диференціальних рівнянь, що описують граничну траєкторію електронного пучка. Через виконаний аналіз показано та доведено, що інтерполяція нижчого порядку зазвичай дає мінімальне значення середньої похибки, використання методу самоузгодженої інтерполяції та екстраполяції дає мінімальну похибку щодо оцінки фокальних параметрів електронного променя, а інтерполяція вищого порядку може бути використана для отримання рівномірного значення похибки на всьому інтервалі інтерполяції. Усі результати оцінювання похибок отримано з використанням оригінального комп'ютерного програмного забезпечення, створеного засобами мови програмування Python.

Ключові слова: інтерполяція, екстраполяція, інтерполяція нижчого порядку, інтерполяція вищого порядку, коренево-поліноміальна функція, яружна функція, середня похибка, електронний пучок, гранична траєкторія, високовольтний тліючий розряд, електронно-променеві технології.

ВІДОМОСТІ ПРО АВТОРІВ

Абрамов Геннадій Серафимович,
доцент, кандидат фізико-математичних наук, доцент кафедри судноводіння Херсонської державної морської академії, Україна, Херсон

Альохін Олексій Борисович,
професор, доктор економічних наук, професор кафедри системного аналізу та інформаційних технологій Маріупольского державного університету, Україна, Київ

Бідюк Петро Іванович,
професор, доктор технічних наук, професор кафедри математичних методів системного аналізу ННІПСА КПІ ім. Ігоря Сікорського, Україна, Київ

Бондаренко Віктор Григорович,
професор, доктор технічних наук, професор кафедри математичних методів системного аналізу ННІПСА КПІ ім. Ігоря Сікорського, Україна, Київ

Брутман Анна Богданівна,
доцент, кандидат економічних наук, завідувач кафедри іноземних мов професійного спілкування Національного університету "Запорізька політехніка", Україна, Запоріжжя

Городецький Віктор Георгійович,
доцент, кандидат фізико-математичних наук, доцент кафедри автоматизації електротехнічних та мехатронних комплексів Навчально-наукового інституту енергозбереження та енергоменеджменту КПІ ім. Ігоря Сікорського, Україна, Київ

Грабовий Олександр Миколайович,
професор, доктор медичних наук, професор та в.о. завідувача кафедри гістології та ембріології Національного медичного університету імені О.О. Богомольця, Україна, Київ

Гущин Іван Валерійович,
старший викладач кафедри штучного інтелекту та програмного забезпечення Харківського національного університету імені В.Н. Каразіна, Україна, Харків

Канцедал Георгій Олегович,
аспірант, асистент кафедри математичних методів системного аналізу ННІПСА КПІ ім. Ігоря Сікорського, Україна, Київ

Мацукі Йошіо,
професор, доктор технічних наук, консультант Навчально-наукового центру "Світовий центр даних з геоінформатики та сталого розвитку" КПІ ім. Ігоря Сікорського, Україна, Київ

Мельник Ігор Віталійович,
професор, доктор технічних наук, професор кафедри електронних пристроїв та систем факультету електроніки КПІ ім. Ігоря Сікорського, Україна, Київ

Панкратова Наталія Дмитрівна,
член-кореспондент НАН України, професор, доктор технічних наук, заступник директора з наукової роботи ННІПСА КПІ ім. Ігоря Сікорського, Україна, Київ

Паржин Юрій Володимирович,
професор, доктор технічних наук, професор кафедри системного аналізу та інформаційно-аналітичних технологій Навчально-наукового інституту комп'ютерних наук та інформаційних технологій Національного технічного університету "Харківский політехнічний інститут", Україна, Харків

Перевозник Кирило Максимович,
студент Навчально-наукового інституту комп'ютерних наук та інформаційних технологій Національного технічного університету "Харківский політехнічний інститут", Україна, Харків

Петренко Анатолій Іванович,
професор, доктор технічних наук, професор кафедри системного проектування ННІПСА КПІ ім. Ігоря Сікорського, Україна, Київ

Подколзін Гліб Борисович,
доцент, кандидат фізико-математичних наук, доцент кафедри математичних методів системного аналізу ННІПСА КПІ ім. Ігоря Сікорського, Україна, Київ

Починок Аліна Володимирівна,
кандидат технічних наук, старший науковий співробітник Науково-дослідного інституту електроніки та мікросистемної техніки КПІ ім. Ігоря Сікорського, Україна, Київ

Романенко Віктор Демидович,
професор, доктор технічних наук, заступник директора з науково-педагогічної роботи ННІПСА КПІ ім. Ігоря Сікорського, Україна, Київ

Сіренька Тетяна Олександрівна,
студентка магістратури Харківського національного університету імені В.Н. Каразіна, Україна, Харків

Скрипка Михайло Юрійович,
аспірант кафедри електронних пристроїв та систем факультету електроніки КПІ ім. Ігоря Сікорського, Україна, Київ

Тимощук Оксана Леонідівна,
доцент, кандидат технічних наук, завідувачка кафедри математичних методів системного аналізу ННІПСА КПІ ім. Ігоря Сікорського, Україна, Київ

Титаренко Андрій Миколайович,
аспірант ННІПСА КПІ ім. Ігоря Сікорського, Україна, Київ

Тішков Максим Олегович,
аспірант кафедри математичних методів системного аналізу ННІПСА КПІ ім. Ігоря Сікорського, Україна, Київ

Шабельник Тетяна Володимирівна,
професор, доктор економічних наук, завідувачка кафедри економічної кібернетики і системного аналізу Харківського національного економічного університету імені Семена Кузнеця, Україна, Харків

Чаповський Юрій Аркадійович,
доцент, кандидат фізико-математичних наук, доцент кафедри математичних методів системного аналізу ННІПСА КПІ ім. Ігоря Сікорського, Україна, Київ

Шутяк Дмитро Олегович,
молодший науковий співробітник Навчально-наукового центру “Світовий центр даних з геоінформатики та сталого розвитку” КПІ ім. Ігоря Сікорського, Україна, Київ

Зміст журналу
«Системні дослідження та інформаційні технології»
за 2024 р.

ЗМІСТ № 1

ЗМІСТ № 2

ЗМІСТ № 3

ЗМІСТ № 4

АВТОРИ СТАТЕЙ ЗА 2024 РІК