UDC 621.004.8(045)
DOI:10.18372/1990-5548.80.18682

[1]V. M. Sineglazov,
[2]V. P. Khotsyanovsky

ADAPTIVE CONTROL OF MANIPULATOR ROBOTS IN A DYNAMIC ENVIRONMENT USING NEURAL NETWORKS

[1,2]Faculty of Air Navigation, Electronics and Telecommunications, National Aviation University, Kyiv, Ukraine
E-mails: [1]svm@nau.edu.ua ORCID 0000-0002-3297-9060,
[2]sttt912@yahoo.com ORCID 0000-0003-0415-777X

***Abstract*—*The purpose of the study is to develop an approach to planning the trajectory of the manipulator robot using an intelligent system based on neural networks. For this purpose, the work considered the processes of planning and deploying the movement of the robot. The analysis of existing methods of planning the movement of manipulator robots and the review of intelligent control systems made it possible to obtain a complete picture of the current state of this issue. A system is proposed that can perceive the environment and control the movement of the robot by generating the correct control commands. For this, 3 tasks were solved, namely: analysis of the environment in order to determine its features, determination of the trajectory in order to neutralize the collision and determination of controlled influences for the executive authorities in order to implement the movement. The functionality and structure of the neural network for solving each of the tasks are proposed.*

Index Terms—Machine learning; neural networks; motion planning system; intelligent system; robotic manipulators; dynamic obstacles; environment analysis; automated systems.

I. INTRODUCTION

At the early stage of the design of the robot manipulator, the dynamic model system and the system parameters associated with it must be accurately described when designing the controller [1]. In traditional control design methods, such as computational torque control and inverse dynamics control, which works well [2], by calculating the torque of the robot manipulator and building dynamic equation, you can get a good control effect [3]. However, this suggests the possibility of obtaining an accurate data model. However, obtaining an accurate mathematical model of the robot during its real operation is difficult [4].

Because many scenarios require robots to adapt to new conditions or even learn completely new behavior. For example, a robot that manufactures cars will occasionally have to adapt to new car models.

For many real-world applications, it is sufficient to program the required behavior by hand, but often this is not possible because the environment may simply change too often or even be unknown in advance to the engineers programming the system.

However, modern requirements for automated systems require the development of new motion planning methods to ensure the accuracy and optimality of robot actions in dynamic production conditions, as existing approaches often have limitations and are unable to provide flexibility in solving dynamic production scenarios This need is caused by the dynamism of the production environment where robots have to function.

Motion planning taking into account dynamic changes opens the way to increasing the accuracy and efficiency of robots. This will lead to better performance of tasks, saving resources and increasing productivity. Also, this method will reduce the risk of collisions, as the probability of emergency situations will be significantly reduced due to better traffic planning.

As a result, this will lead to the expansion of the spheres of use of robots. Because the ability to adapt to dynamic changes will make robots more versatile and allow them to be used in a wider range of tasks. Therefore, there is a problem of developing a method of planning robot movement with the possibility of taking into account changes in dynamic production scenarios. The results of this research are needed in practice, because they determine the possibility of safe and effective use of robots in conditions where they must interact with dynamic surrounding objects, for example, other robots. In recent years, machine learning has revolutionized the field of robotics and automation. Using algorithms, robots can be taught to perform various tasks and even This has made it possible to create more perfect works that can independently navigate complex environments, interact with people in a more natural way, and perform production tasks more efficiently.

Machine learning enables robots to process vast amounts of data in real time, allowing them to make faster and more accurate decisions. These robots have a better understanding of the environment and the objects around them. For example, they can be programmed to identify objects using a combination of visual, tactile and sound sensors. This allows it to recognize different objects in the environment and react accordingly.

II. ANALYSIS OF EXISTING AND CUSTOM ROBOTIC MANIPULATOR MOTION PLANNING METHODS

Robotic arm positioning is the process of precisely controlling the position and orientation of a robotic arm limb (tool or gripper) to perform certain tasks. This position is crucial for the interaction of the hand with objects, manipulating them and accurately performing various operations.

A robotic arm usually consists of several joints that provide degrees of freedom (DOF) to the arm. DOF represents the number of independent parameters needed to describe the arm configuration. Each joint allows the arm to rotate or move along a specific axis, allowing the arm to move in multiple directions (Fig. 1).

Positioning of a robotic arm is based on the use of coordinate systems to determine the position and orientation of the arm. The most common coordinate system is the Cartesian system coordinates, where positions are defined by X, Y, and Z coordinates. Hand orientation can be represented by Euler angles, quaternions, or rotation matrices. Direct kinematics involves determining the position and orientation of the end effector based on the angles or lengths of the joints. On the other hand, inverse kinematics involves finding the angles or lengths of the joints necessary to achieve the desired position and orientation of limbs.

Trajectory planning involves creating a smooth and optimal trajectory along which the robot arm will move when moving from one position to another. It takes into account factors such as obstacle avoidance, joint constraints and path optimization based on criteria such as time, energy consumption or traffic. Trajectory planning ensures efficient and safe movement of the hand to the desired position.

In the analysis of traditional robot motion planning methods, key techniques such as geometric trajectory planning, inverse kinematics method, dynamic programming, and random positions and optimization methods should be addressed.

Geometric trajectory planning determines the movement of the robot based on the geometric characteristics of the workspace. This method allows you to specify the exact position and orientation of the manipulator, but may be limited by the complexity of solving problems for complex user convenience, it has its limitations, particularly in the area of adaptation to changing conditions [5].

The inverse kinematics method is used to determine the input angles or positions of the manipulator to achieve a specific position or trajectory. This method is effective in solving problems for specific points in space, but may lose accuracy in complex problems due to a large number of possible solutions. Also, it is used in most industrial robot control systems [6].

The inverse kinematics method allows you to determine the robot's kinematic parameters based on its position and orientation.

Dynamic programming considers the movement of the manipulator as a sequence of actions with criteria minimization. This method is effective for optimization problems and for planning trajectories, in particular in cases where the dynamic constraints of the robot are important. However, it can be computationally expensive for real time in complex environments, especially with a large number of dimensions of the decision space and complex tasks [7].

Regarding random positions and optimization methods, these approaches often use random points to reduce the number of intermediate points in the trajectories or use optimization methods to reach optimal solutions.

In order to perform a comparative analysis of the above-mentioned approaches, a table should be drawn up in which their main characteristics and differences will be displayed (Table I).

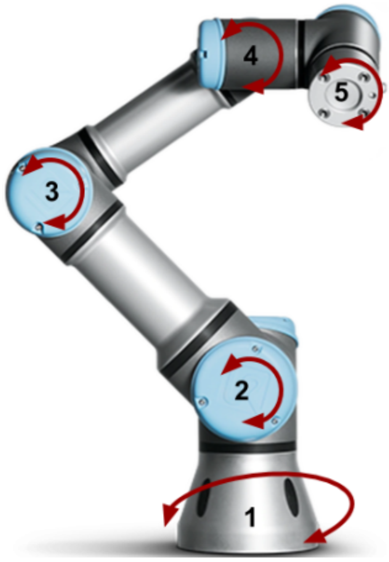

Fig. 1. Robotic Arm with 5 DOF

TABLE I. COMPARATIVE ANALYSIS OF EXISTING APPROACHES

Approach	Peculiarities	Advantages	Limitations:
Geometric trajectory planning	Determines the movement of the robot according to the geometric characteristics of the workspace	Ensures the exact position and orientation of the robot	Difficulty solving problems for complex configurations
		Convenient management for the user	Limited adaptation to changing conditions
The inverse kinematics method	Defines the input angles or positions of the manipulator to reach a certain point	Effective in solving problems for specific points	Loss of accuracy in complex tasks
		Used in most control systems	A large number of possible solutions in complex problems
Dynamic programming	Considers the movement of the manipulator as a sequence of actions with criteria minimization	Effective for optimization and planning of trajectories	Computationally expensive for real time
		Allows to take into account the dynamic limitations of the robot	Difficulty in use in real time conditions
Random positions and optimization methods	Using random points	Reducing the number of intermediate points in trajectories	Dependence on the initial selection of random points
	Use of optimization methods	Achieving optimal solutions	The need for computing resources, especially for complex tasks

III. PROBLEM STATEMENT

Modern requirements for automated systems necessitate the development of new motion planning methods to ensure the accuracy and optimality of robotic actions in dynamic production environments. Let the dynamics of work in discrete time occur as f_χ:

$$x_{k+1} = f_\chi\left(x_k, u_k\right),$$

where $x_k \in \chi$ and $u_k \in U$ denote the state and control input of the system at the kth search step.

The work considers static obstacles and dynamic obstacles, the movement of which is known. For example, for a multi-robot system, the movements of all robots are usually planned one at a time. When planning the movement of a given robot, the movements of other robots are known.

In this paper, the trajectory π is defined as a series of states and level control commands:

$$\pi = \left(x_0, u_0, x_{t_0}, x_1, u_1, x_{t_1}, ..., x_k, u_k, x_{t_k}\right),$$

where t_k is the time step of the kth intermediate point on the trajectory.

Existing approaches frequently exhibit limitations and fail to provide the required flexibility for addressing dynamic production scenarios. This need arises from the inherent dynamism of the production environment in which robots operate [9].

In this paper, the proposed approach is focused on the study of the possible solution space of traffic planning problems from previous experience to improve search efficiency. In other words, the proposed approach first learns to perceive the robot's environment. The robot dynamics then learns to accurately mimic realistic robot motion. Finally, the proposed approach learns which optimal high-level commands can move the robot to the target area with realistic dynamics in the perceived environment model at each search step.

For successful manipulation of the object, the task will be divided he task:

A. Determining the trajectory in order to neutralize the collision

Since studying the full solution space is very difficult and does not scale well for other problems, our approach starts with studying the local space of possible solutions f_{local}:

$$f_{\text{local}}\left(x_k, \varphi_k \middle| x_{\text{goal}} \to u_k\right).$$

The locally feasible solution space consists of all possible control policies that consider only the state of the local system (for example, the state of the environment φ_k, the current state of the robot x_k, and the goal state χ_{goal}) and direct the robot from the current state to the goal area with the kth control command u_k search steps.

B. Analyzing the environment in order to determine its features

Since this work considers environments with static and dynamic obstacles, the geometric and temporal information of the environment should be represented as the state of the environment φ_k at the *k*th step of the search and used in the subtask the task of determining the trajectory in order to neutralize the collision.

C. Determining controlled influences for executive bodies in order to implement the movement

It is necessary to calculate the execution time and interpolation of the robot's motion between two states to check for collisions between the robot and obstacles during the transition from one state to the next at each search step. Thus, the proposed approach studies realistic high-level motion-driven robot dynamics.

IV. PROBLEM SOLUTION

To address this issue, the implementation of intelligent control systems, specifically neural networks, is proposed. Neural networks are instrumental in enhancing and optimizing the movement of robotic manipulators due to their flexibility and adaptability to changing conditions. They also facilitate the automation and simplification of the calibration processes for moving robot components, thereby ensuring maximum accuracy and operational speed.

For precise calculation of the robot's actual motions and trajectory, the neural network must incorporate the robot's real motion dynamics and motion trajectory based on actual motion execution. Notably, the planned movement of the robot may differ slightly from its actual movement.

For training tasks, we consider a model of a robotic arm represented as an intelligent agent in a constrained and simplified environment. An agent is an entity that is capable of perceiving its environment through sensors, as well as influencing its environment through actuators. Agents can traditionally perceive their own actions, but not necessarily the effects of those actions on the environment.

An agent can be mathematically described by an agent function that affects the agent's action based on its entire perceptual position. Thus, the agent's behavior can be fully described by defining actions for each possible perception situation [10]. The scheme of interaction between the agent and the environment is presented in Fig. 2.

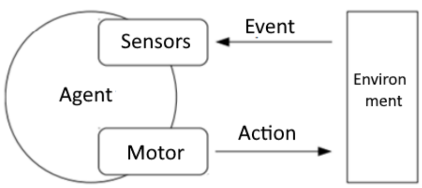

Fig. 2. Interaction of components of the educational environment

Neural networks such as convolutional neural networks, recurrent neural networks, and deep neural networks are best suited for manipulator robot motion planning systems (Fig. 3).

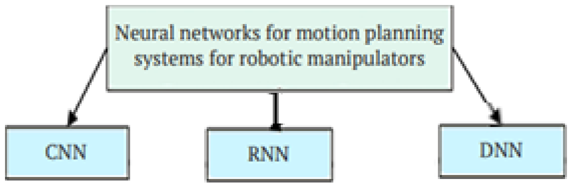

Fig. 3. Neural networks for robotic manipulators

When considering the use of neural networks to solve the described problem, it is important to focus on the analysis and description of architectures that optimally take into account the features of these systems. Neural networks such as convolutional neural networks, recurrent neural networks, and deep neural networks are best suited for motion planning systems of robot manipulators.

Consider a dynamic scene that includes moving objects or a change in the state of objects over time. The architecture of convolutional neural networks is suitable for object recognition. The use of CNN allows effective recognition of objects in real time, which is key to planning the safe movement of the manipulator around objects in the workspace [11].

However, convolutional neural networks (CNNs) alone may not be sufficient for effective analysis of dynamic scenes, and they do not account for the temporal sequence of events. In such cases, it makes sense to use recurrent neural networks (RNNs) or their enhancements, such as long-term variants of short-term memory (LSTM) and networks based on the attention mechanism (Transformers). These architectures can learn to recognize and retrieve temporal relationships, which are critical to properly understanding a dynamic scene.

Long short-term memory networks variants cope well with the problem of gradient vanishing, thanks to which they can remember information over long time intervals [12]. This is especially useful for tracking the trajectories of objects in time and predicting their future position, which is important for planning the motion of the manipulator (Fig. 4).

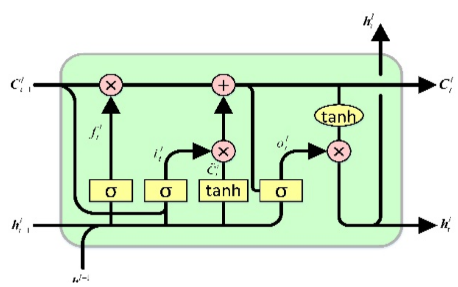

Fig. 4. LSTM architecture

Another promising architecture is Transformer (Fig. 5), which uses an attention mechanism to efficiently process a list of data. Transformers have shown impressive results in natural language processing tasks, but they have also been successfully applied to video analysis and other types of subsequent data [13]. The attention mechanism allows you to simulate focusing on important parts of the input situation, which contributes to a better understanding of the context and ensures high performance.

To solve the task of determining the trajectory in order to neutralize collisions for the manipulator robot, it is proposed to use a recurrent neural network (RNN). Recurrent neural networks are used to model motion dynamics (Fig. 6). Recurrent neural networks are suitable for taking into account time dependencies and modeling the dynamics of manipulator movement, which allows predicting future states of the system [14].

To improve work with dynamic scenes, you can combine CNN with RNN, LSTM or Transformer, creating hybrid models. For example, a CNN can be used to extract spatial features from live video, after which these features are fed to an LSTM or Transformer to process the temporal information.

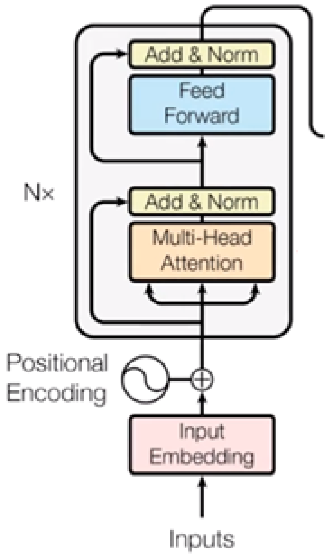

Fig. 5. Transformer architecture

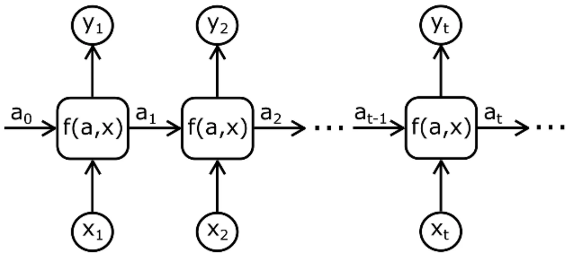

Fig. 6. RNN architecture

Such hybrid architectures provide a deeper understanding of dynamic scenes, allowing the model not only to recognize objects, but also to track their movements and changes over time. This creates opportunities for automated systems, such as new robotics, where it is important not only to recognize objects, but also to interact with them in real time.

V. Results

The article resulted in recommendations for an approach to solving the problem of increasing positioning accuracy and response to dynamic obstacles of manipulator robots using neural networks. Key recommendations include the following:

1) Combining convolutional neural networks with recurrent neural networks, long-short-term memories, or networks based on the attention mechanism (Transformers) to account for temporal dependencies in dynamic scenes. This provides a deeper understanding of the movements of objects and increases the accuracy of their recognition.

2) The use of LSTM and Transformer to predict the future positions of objects allows you to effectively plan manipulator movement trajectories, taking into account possible obstacles. This reduces planning time and increases overall system efficiency.

3) Creation of a network that transforms high-level commands into specific controlled signals for executive bodies of the manipulator. The input layer receives the commands, the hidden layers process the information, and the output layer generates specific controlled effects.

4) Using feedback to adapt network parameters based on the output signal and the results of the performed movements. This allows the system to learn from real data and adjust its actions to achieve greater accuracy and efficiency.

The implementation of these recommendations allows to significantly increase the positioning accuracy of manipulator robots and their ability to respond to dynamic obstacles in real time.

VI. Conclusions

As a result of the conducted research, this article proposes and analyzes modern approaches to solving the problem of increasing positioning accuracy and response to dynamic obstacles of manipulators using neural networks. Proposed combining CNN with RNN, LSTM, or Transformer significantly increases the system's ability to recognize objects and analyze their movements in dynamic scenes. This ensures high accuracy and speed of information processing, which is critically important for robotic systems.

Using LSTM and Transformer to predict the future positions of objects allows you to effectively plan manipulator trajectories, taking into account the possibility of obstacles. This reduces planning time and efficiency of tasks.

The proposed network for determining controlled influences allows transforming high-level commands into specific controlled signals for executive bodies of the manipulator. This ensures accurate execution of movements and adaptation of the system in real time. Use of feedback and parameterization of output signals in accordance with the requirements of executive bodies allows the system to learn on the basis of real data, adjust its actions and ensure high accuracy and reliability.

The proposed recommendations and architectural solutions open up new opportunities for the development of efficient and reliable robotic systems. The implementation of these approaches in practice allows to significantly increase the positioning accuracy of manipulator robots, their ability to respond to dynamic obstacles, as well as to provide more intelligent and adaptive control in various fields of application.

REFERENCES

[1] A. V. Duka, “Neural Network based Inverse Kinematics Solution for Trajectory Tracking of a Robotic Arm,” *Procedia Technol*, 12, 20–27, 2014. https://doi.org/10.1016/j.protcy.2013.12.451

[2] D. G. Arseniev, L. Overmeyer, H. Kälviäinen, and B. Katalinić, *Cyber-Physical Systems and Control*, Springer: Berlin/Heidelberg, Germany, 2019. https://doi.org/10.1007/978-3-030-34983-7

[3] S. Islam, and X. P. Liu, “Robust Sliding Mode Control for Robot Manipulators,” *IEEE Trans. Ind. Electron*, 58, 2011, 2444–2453. https://doi.org/10.1109/TIE.2010.2062472

[4] M. J. Yazdanpanah, G. Karimian Khosrowshahi, Robust Control of Mobile Robots Using the Computed Torque Plus H∞ Compensation Method. Available online: https://www.sciencegate.app/document/10.1109/cdc.2003.1273069

[5] S. B. Niku, “Industrial Robotics: Programming, Simulation and Applications,” *John Wiley & Sons*., 2010. https://doi.org/10.5772/40

[6] Ryo Kikuuwe, & Bernard Brogliato, “A New Representation of Systems with Frictional Unilateral Constraints and Its Baumgarte-Like Relaxation,” *Multibody System Dynamics*, pp. 267–290, 2017. https://doi.org/10.1007/s11044-015-9491-6

[7] X. Wang, Q. Wu, T. Wang, and Y. Cui, “A Path-Planning Method to Significantly Reduce Local Oscillation of Manipulators Based on Velocity Potential Field,” Sensors. 2023. URL: https://www.mdpi.com/1424-8220/23/23/9617.

[8] H. Liu and L. Wang, “Collision-Free Human-Robot Collaboration Based on Context Awareness,” *Robot. Comput.-Integr. Manuf*., 67, 101997, 2021. https://doi.org/10.1016/j.rcim.2020.101997

[9] V. Khotsyanivskyi, & V. Sineglazov, “Machine learning in the task of auto-calibration of moving elements of robotic systems on the example of stepper motor control,” *In International scientific and technical conference “AVIA,”* Kyiv: National Aviation University, 2023, pp. 9.37–9.41. [in Ukraine]

[10] Iaroslav Omelianenko, *Hands-On Neuroevolution with Python: Build high-performing artificial neural network architectures using neuroevolution-based algorithms*, Packt Publishing, 2019.

[11] R. L. Galvez, A. A. Bandala, E. P. Dadios, R. R. P. Vicerra and J. M. Z. Maningo, "Object Detection Using Convolutional Neural Networks," *TENCON 2018– 2018 IEEE Region 10 Conference*, Jeju, Korea (South), 2018, pp. 2023–2027, https://doi.org/10.1109/TENCON.2018.8650517.

[12] K. Greff, R. K. Srivastava, J. Koutník, B. R. Steunebrink, & J. Schmidhuber, "LSTM: A Search Space Odyssey". *IEEE Transactions on Neural Networks and Learning Systems*, 28(10), (2017), 2222–2232. https://doi.org/10.1109/TNNLS.2016.2582924.

[13] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, & N. Houlsby, "An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale," *International Conference on Learning Representations*, 2021.

[14] S. Hochreiter, & J. Schmidhuber, "Long Short-Term Memory," *Neural Computation*, 9(8), 1735–1780, 1997. https://doi.org/10.1162/neco.1997.9.8.1735.

Received February 26, 2024

Sineglazov Victor. ORCID 0000-0002-3297-9060. Doctor of Engineering Science. Professor. Head of the Department Aviation Computer-Integrated Complexes.
Faculty of Air Navigation Electronics and Telecommunications, National Aviation University, Kyiv, Ukraine.
Education: Kyiv Polytechnic Institute, Kyiv, Ukraine, (1973).
Research area: Air Navigation, Air Traffic Control, Identification of Complex Systems, Wind/Solar power plant, artificial intelligence.

Publications: more than 700 papers.
E-mail: svm@nau.edu.ua

Volodymyr Khotsyanovsky. ORCID 0000-0003-0415-777X. Post-graduate Student.
Faculty of Air Navigation, Electronics and Telecommunications, National Aviation University, Kyiv, Ukraine.
Education: National Aviation University, Kyiv (2020).
Research area: artificial intelligence.
Publications: more than 10.
E-mail: sttt912@yahoo.com

В. М. Синєглазов, В. П. Хоцянівський. Адаптивне керування роботами-маніпуляторами у динамічному середовищі за допомогою нейронних мереж

Метою дослідження є розробка підходу до планування траєкторії руху робота-маніпулятора за допомогою інтелектуальної системи на основі нейронних мереж. Для цього в роботі розглядалися процеси планування та розгортання руху робота. Аналіз існуючих методів планування руху роботів-маніпуляторів та огляд інтелектуальних систем управління дозволив отримати повну картину сучасного стану цього питання. Пропонується система, яка може сприймати навколишнє середовище та керувати рухом робота, генеруючи правильні команди керування. Для цього було вирішено три завдання, а саме: аналіз середовища з метою визначення його особливостей, визначення траєкторії з метою нейтралізації зіткнення та визначення контрольованих впливів для органів виконавчої влади з метою здійснення руху.

Ключові слова: машинне навчання; нейронні мережі; система планування руху; інтелектуальна система; роботи маніпулятори; динамічні перешкоди; аналіз середовища; автоматизовані системи.

Синєглазов Віктор Михайлович. ORCID 0000-0002-3297-9060.
Доктор технічних наук. Професор. Завідувач кафедрою авіаційних комп'ютерно-інтегрованих комплексів. Факультет аеронавігації, електроніки і телекомунікацій, Національний авіаційний університет, Київ, Україна.
Освіта: Київський політехнічний інститут, Київ, Україна, (1973).
Напрям наукової діяльності: аеронавігація, управління повітряним рухом, ідентифікація складних систем, вітроенергетичні установки, штучний інтелект.
Кількість публікацій: більше 700 наукових робіт.
E-mail: svm@nau.edu.ua

Хоцянівський Володимир Петрович. ORCID 0000-0003-0415-777X. Аспірант.
Факультет аеронавігації, електроніки та телекомунікацій, Національний авіаційний університет, Київ, Україна.
Освіта: Національний авіаційний університет, Київ (2020).
Напрям наукової діяльності: штучний інтелект.
Кількість публікацій: більше 10 наукових робіт.
E-mail: sttt912@yahoo.com