UDC 621.004.8(045)
DOI:10.18372/1990-5548.81.18991

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

COMPARATIVE ANALYSIS OF THE METHODS OF PLANNING AND COORDINATING OF MANIPULATOR ROBOT MOVEMENT

[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*—*This paper presents a comparative analysis of two methods for planning and coordinating the movement of robot manipulators in dynamic environments: a neural network-based approach for solving dynamic production scenarios and the rapidly exploring random trees algorithm. The study aims to enhance the trajectory planning of robot manipulators by leveraging the strengths of intelligent systems. The neural network method is designed to perceive the environment, generate accurate control commands, and adapt to changing conditions in real-time. The paper the processes involved in environmental analysis, collision avoidance, and control signal generation for actuators, with an emphasis on the neural network architecture tailored for these tasks. The results demonstrate that the neural network approach offers significant improvements in adaptability and efficiency, providing a robust solution for optimizing automated processes in dynamic production environments.*

Index Terms—Robot manipulators; trajectory planning; neural networks; dynamic environments; collision avoidance; intelligent control systems; automated processes; real-time adaptation; production scenarios.

I. INTRODUCTION

At the initial stages of designing a robot manipulator, it is essential to precisely define the dynamic model and the related system parameters for effective controller design [1]. Traditional control design methods, such as computed torque control and inverse dynamics control, have proven effective by calculating the manipulator's torque and establishing a dynamic equation to achieve satisfactory control performance [2], [3]. However, these methods rely on the assumption of an accurate data model, which can be challenging to obtain during real-world operation [4].

In many scenarios, robots must adapt to new conditions or even learn entirely new behaviors. For instance, a robot involved in car manufacturing may occasionally need to adapt to new car models. While it may be feasible to manually program the required behaviors in some real-world applications, this approach often falls short when the environment changes frequently or is unknown in advance to the engineers.

Modern demands for automated systems necessitate the development of innovative motion planning methods to ensure precise and optimal robot actions in dynamic production environments. Existing approaches often have limitations, lacking the flexibility needed to address dynamic production scenarios. This requirement stems from the dynamic nature of production environments where robots must operate.

Incorporating dynamic changes into motion planning can enhance the accuracy and efficiency of robots, leading to better task performance, resource savings, and increased productivity. Additionally, this approach reduces the risk of collisions, as improved motion planning significantly lowers the likelihood of emergencies. Consequently, this will broaden the scope of robot applications, as the ability to adapt to dynamic changes will render robots more versatile, enabling their deployment in a wider range of tasks.

Therefore, there is a pressing need to develop a motion planning method that accounts for dynamic production scenarios. The practical significance of this research lies in its potential to ensure the safe and effective use of robots in environments where they must interact with dynamic surrounding objects, such as other robots. In recent years, machine learning has revolutionized robotics and automation. With the help of algorithms, robots can now be trained to perform various tasks and even independently navigate complex environments, interact more naturally with humans, and carry out production tasks more efficiently.

Machine learning empowers robots to process vast amounts of data in real-time, enabling faster and more accurate decision-making. These robots gain a deeper understanding of their environment and the objects within it. For example, they can be programmed to identify objects using a combination of visual, tactile, and auditory sensors, allowing them to recognize different objects and respond accordingly.

II. ANALYSIS OF THE ROBOT MOVEMENT PLANNING AND COORDINATION METHOD USING NEURAL NETWORKS FOR SOLVING DYNAMIC SCENARIOS AND THE RAPIDLY EXPLORING RANDOM TREES METHOD

In modern conditions of a rapidly changing production environment, it is urgent to create methods of planning and coordinating the movement of robots that ensure accuracy and adaptability in real time. This section discusses two approaches to solving this problem: the use of neural networks for manipulator motion planning and the Rapidly Exploring Random Trees (RRT) algorithm.

A) The method of motion planning using neural networks

The method of robot movement planning and coordination using neural networks is based on the ability of artificial neural networks to learn complex patterns of behavior based on large volumes of data. Neural networks, such as convolutional neural networks (CNN), recurrent neural networks (RNN), as well as transformers, provide high flexibility and adaptability to changing environmental conditions [5].

This method allows you to create models that can not only recognize objects in space, but also track their movements and changes over time. For example, the use of CNN provides effective real-time object recognition, which is key to safe manipulator motion planning. At the same time, RNN, LSTM or transformers allow modeling time dependencies and predicting future states of the system, which is important for avoiding collisions and optimal planning of the movement trajectory [6].

Hybrid architectures that combine CNNs with RNNs, LSTMs, or transforms provide an even deeper understanding of dynamic scenes, allowing the system not only to recognize objects, but also to track their movements and changes over time. This significantly expands the capabilities of automated systems and provides flexibility in solving dynamic production scenarios.

B) Rapidly RRT method

The rapidly exploring random trees algorithm is one of the most common methods of trajectory planning in the space with obstacles. It is based on the idea of building a tree by randomly exploring the state space, gradually finding a path from the starting point to the final point. The main advantage of the RRT is its ability to efficiently explore large spaces with complex obstacles [7].

Rapidly exploring random trees is a deterministic algorithm that allows you to quickly find the motion trajectory, but it has certain limitations in dynamic environments. In particular, the algorithm does not take into account possible changes in the environment during the movement of the robot, which can lead to collisions or incorrect trajectory planning. This limitation makes RRT less suitable for scenarios where it is important to adapt quickly to changing conditions [8].

Comparing the two approaches, it can be noted that the neural network method provides greater flexibility and adaptability compared to RRT, especially in dynamic production scenarios. Neural networks are able not only to plan the trajectory, but also to quickly adapt it to changes in the environment, which reduces the risk of collisions and provides optimal conditions for the operation of manipulator robots in real time.

On the other hand, RRT remains an effective tool for rapid trajectory planning in static conditions where no significant changes in the environment are expected. This makes it suitable for tasks that do not require high adaptability, but require a quick and reliable solution [9].

That is, for dynamic scenarios of the production environment, neural networks offer a more modern and flexible approach, while RRT is an effective method for static or slightly changing conditions.

III. PROBLEM SOLUTION

Training a robot motion planning and coordination system requires large training samples and realistic trajectories performed by the robot control system. This is an expensive process both in terms of time and resources. Therefore, it is important to create a realistic simulation environment that allows you to effectively simulate the work of robots. Unity3D, developed by Unity Software, based in the United States of America, is used to implement such an environment (Fig. 1).

Fig. 1. Examples of using the Unity3D environment for modeling robot movements using the example of a manipulator robot

Unity is used to create realistic datasets and validate planning results. In particular, the Unity Robotics Hub contains a standalone programming system that can integrate with the ROS or MoveIt module. The Unity Robotics Hub module supports integration with original robot controllers and provides simulation accuracy to 0.00005 radians and 1% cycle time.

The geometry of the workspace is represented through 3D polygon mesh models in the modeling software. Polygon mesh models are exported and collected as raw data to train a neural network that represents the dynamic environment. The transformation of the 3D obstacle mesh into voxel models is chosen because of its convenient data format, which is well suited for analysis and presentation and can be easily adapted to different requirements for different robot tasks.

For example, in some high-speed tasks, the robot must keep a safe distance from obstacles in the environment. In this case, the size of the voxels responsible for the obstacles should be increased to leave enough space between the robot and the obstacles. At the same time, for tasks that require delicate operations, such as spot welding, where the robot must pass through narrow areas, the resolution of the voxel models must be increased to represent more detailed elements.

To validate the proposed approach, both methods, namely the neural network method and the RRT method, will be tested in the Unity Robotics Hub virtual environment. The robots will perform the task of selecting and placing objects in various environments that contain both static and dynamic obstacles. It is important to note that although the environment contains two SCARA robots, motion planning is only necessary for one robot, while the other robot is treated as a static or dynamic obstacle.

Unity Robotics Hub motion planners were used to control the robot. Some robot movements were generated using high-level motion commands that were programmed manually. To evaluate the proposed approach, two robots will perform the task of picking up and placing objects in different environments that include static and dynamic obstacles (Table I). It is important to note that, despite the presence of two SCARA robots, motion planning is performed for only one robot, while the other robot is treated as a static or dynamic obstacle.

Table I shows the static and dynamic interference in the four different categories of environments. To evaluate the proposed approach, 100 environments were created that were not used during training. In each environment, 20 pairs of start and target coordinates were randomly generated. The performance of the proposed approach was evaluated in terms of validity, trajectory execution time, and computation time. In an application where the movements of SCARA robots are planned by different planners in an offline mode, the validation was performed only in visual aspects.

TABLE I. STATIC AND DYNAMIC INTERFERENCE IN FOUR ENVIRONMENTS

Environment type	Static interference	Dynamic interference
Simple static environments	Robot and a cube	None
Complex static environments	Robot and 3 cubes	
Simple dynamic environments	Cube	Movable robot
Complex dynamic environments	3 cubes	

The robot can move along an incorrectly generated trajectory, or it can follow the planned trajectory exactly (Fig. 2).

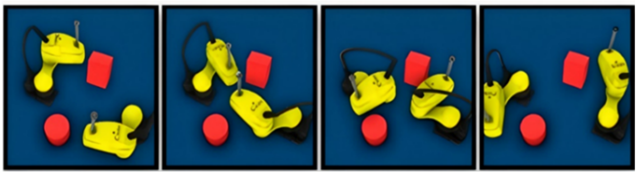

Fig. 2. An example of the correct trajectory of the SCARA robot

The Figure 2 shows that when the robot precisely moves along the given trajectory, the other robot, moving from the right side, crosses the common area earlier than the one moving from the left side.

After testing in 100 created environments, only 5 trajectories generated by the proposed approach contain errors (Table II).

Thus, in all scenarios of the experiment, the average discrepancy between the actual and predicted execution time of high-level movement commands is approximately 5%. It is also worth comparing the interpolation algorithm used to convert the planned robot movement into high-level commands, including RRT, with the trajectory generated by the developed system (Figs 3 – 5). It is important to note that the runtime of the robot's trajectory varies significantly depending on the distance between the start and end points of the movement. Thus, for the test cases, it is necessary to classify distances into three categories: small distance (less than 30% of the robot's manipulator range); medium distance (more than 30% but less than 60%); and large distance (more than 60%).

TABLE II. RELATIVE ERRORS OF THE TRAINED MODEL WHEN PREDICTING THE MOVEMENT AND THE EXECUTION TIME OF THE ACTUAL MOVEMENT

The values of the minimum and maximum possible robot speed, %	**Average time error when performing a movement**		**Average trajectory prediction error**	
	Point-to-point movement, %	***Linear movement, %***	***Point-to-point movement, %***	***Linear movement, %***
0–25	2.71	4.4	0.232	0.574
25–50	2.89	5.78	0.473	0.789
50–75	4.38	6.65	0.481	0.862
75–100	6.11	7.17	0.653	0.912

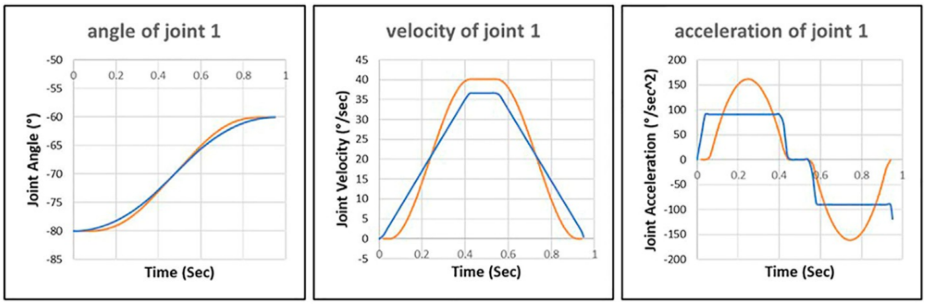

Fig. 3. “angle of joint 1” – change the angle of rotation of the joint; “velocity of joint 1” – change the velocity of the joint; “acceleration of joint 1” – change the acceleration of the joint

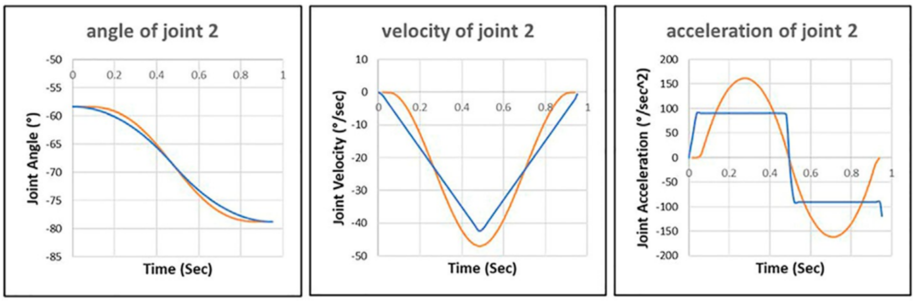

Fig. 4. “angle of joint 2” – change the angle of rotation of the joint; “velocity of joint 2” – change the velocity of the joint; “acceleration of joint 2” – change the acceleration of the joint.

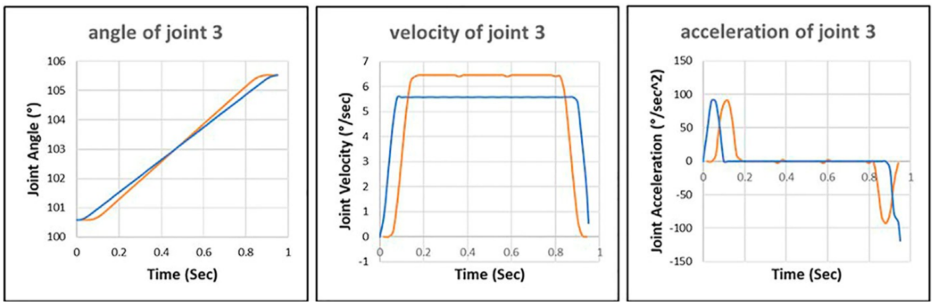

Fig. 5. “angle of joint 3” – change the angle of rotation of the joint; “velocity of joint 3” – change the velocity of the joint; “acceleration of joint 3” – change the acceleration of the joint

Thus, it is possible to calculate the execution time of the trajectories generated by the approach when using robot movement planning and coordination method using neural networks for solving dynamic scenarios and RRT (Table III).

Therefore, the movement of the robot planned by the existing system is significantly different from the movement proposed by the trajectory planning system. This is explained by the fact that the RRT control algorithm used at the planning stage differs from the control algorithm described in this paper. In the planning stage, RRT assumes that the joints can reach their maximum acceleration, while the actual robot control system uses only 60% and 45% of the maximum acceleration for the respective robot axes.

TABLE III. COMPARATIVE TABLE OF THE EXECUTION TIME OF TRAJECTORIES GENERATED BY THE APPROACH WHEN USING ROBOT MOVEMENT PLANNING AND COORDINATION METHOD USING NEURAL NETWORKS FOR SOLVING DYNAMIC SCENARIOS AND RRT

Environment	Distance between start-point and end-point	Average runtime, ms		
		Proposed approach	***RRT***	***Improved RRT***
Simple static environments	Low	221	212	213
	Average	422	543	515
	High	659	836	694
Complex static environments	Low	291	372	304
	Average	603	797	663
	High	732	904	756
Simple dynamic environments	Low	244	272	277
	Average	496	581	558
	High	734	958	826
Complex dynamic environments	Low	419	462	465
	Average	765	975	829
	High	1071	1294	1113

IV. RESULTS

In general, developed methods for planning the movement of manipulators already demonstrate significant advantages in solving the tasks of avoiding obstacles and optimizing trajectories. However, to achieve an even greater level of efficiency and flexibility, it is worth suggesting several areas of optimization. For example, tuning hyperparameters of neural networks. Further research and tuning of model parameters will allow to achieve an optimal balance between speed and accuracy.

Optimizing the weights and architecture of the networks can contribute to improving the training results and prediction accuracy. The use of deep reinforcement learning, i.e. DRL, is important. The application of DRL will allow the operation of the manipulator to learn in real time, adapting its strategies to new conditions. This can improve the robot's ability to quickly adapt to unpredictable scenarios in a manufacturing environment. There is also the integration of additional sensors and real data.

Including additional sources of information, such as additional cameras or sensors, can improve the perception of the robot's environment and provide a more accurate model of the working environment. It is worth paying attention to the improvement of genetic algorithms. Researching various variants of genetic algorithms and their adaptation to the specific requirements of the production process will allow achieving a greater balance between speed and the ability to optimize trajectories. In addition, defining and using clear metrics to measure system performance will help pinpoint improvements made.

Metrics can include robot speed, accuracy of predictions, and response time to changes in the environment. And the definition of opportunities is the use of quantum computing to optimize large volumes of calculations related to learning deep networks and optimizing trajectories. These areas of improvement are aimed at expanding the capabilities and improving the motion planning system of manipulators, providing them with the ability to effectively adapt to various conditions of the production process and constantly increase their productivity.

V. CONCLUSIONS

Based on the results of this study, there are several recommendations for practical application in the field of motion planning of manipulative robots. First of all, it is recommended to implement the developed approach in modern production processes that require autonomy and adaptability in the work of manipulators. It is important to focus on training staff to use this technology to optimize work flows. In addition, it is recommended to conduct other experiments and research aimed at improving the efficiency and speed of movement of manipulative robots in real production conditions. This will allow to expand the fields of application of the technology and increase its competitiveness. For practical implementation, cooperation with manufacturers of robotic equipment is recommended to integrate the proposed approach into new and existing manipulators. This will contribute to increasing the availability and speed of implementation of this technology on production lines.

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 and 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] V. M. Sineglazov and V. P. Khotsyanovsky, "Adaptive Control of Manipulator Robots in a Dynamic Environment Using Neural Networks," *Electronics and Control Systems*, no 2(80), pp. 36–42, 2024. https://doi.org/10.18372/1990-5548.79.18441

[6] V. M. Sineglazov and V. P. Khotsyanovsky, "Method of Planning and Coordination of Robot Movement Using Neural Networks for Solution of Dynamic Production Scenarios," *Electronics and Control Systems*, no. 1(79), pp. 48–53, 2024. https://doi.org/10.18372/1990-5548.80.18682

[7] C. C. Wong, C. J. Chen, K. Y. Wong, and H. M. Feng, "Implementation of a Real-Time Object Pick-and-Place System Based on a Changing Strategy for Rapidly-Exploring Random Tree," *Sensors*, 23(10), 4814, 2023. https://doi.org/10.3390/s23104814

[8] J. G. Kang, & J. W. Jung, *Post triangular rewiring method for shorter RRT robot path planning*, 2021. arXiv preprint arXiv:2107.05344. https://doi.org/10.5391/IJFIS.2021.21.3.213

[9] J. Yu, M. Wu, J. Ji, & W. Yang, "Neural network-based region tracking control for a flexible-joint robot manipulator," *Journal of Computational and Nonlinear Dynamics*, 19(2), article number 021003, 2023. https://doi.org/10.1115/1.4064201.

Received Yuly 09, 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. Порівняльний аналіз методів планування та координації руху робота-маніпулятора

У статті представлено порівняльний аналіз двох методів планування та координації руху роботів-маніпуляторів у динамічних середовищах: підхід на основі нейронної мережі для вирішення динамічних сценаріїв виробництва та алгоритм швидкого дослідження випадкових дерев. Дослідження спрямоване на покращення планування траєкторії роботів-маніпуляторів шляхом використання сильних сторін інтелектуальних систем. Метод нейронної мережі призначений для сприйняття навколишнього середовища, створення точних команд управління та адаптації до мінливих умов у режимі реального часу. Стаття описує процеси, пов'язані з аналізом навколишнього середовища, уникненням зіткнень і генерацією керуючого сигналу для приводів, з наголосом на архітектурі нейронної мережі, розробленій для цих завдань. Результати демонструють, що нейромережевий підхід пропонує значні покращення в адаптованості та ефективності, забезпечуючи надійне рішення для оптимізації автоматизованих процесів у динамічних виробничих середовищах.

Ключові слова: роботи-маніпулятори; планування траєкторії; нейронні мережі; динамічне середовище; уникнення зіткнення; інтелектуальні системи управління; автоматизовані процеси; адаптація в режимі реального часу; сценарії виробництва.

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