

Route planning of autonomous robots in three-dimensional logic space using mivar technologies

Anton Kotsenko¹, Aleksandr Andreev¹, Radmir Kim¹, Anastasiia Bazanova¹, Dmitry Aladin¹, David Todua¹, Aleksei Marushchenko¹, and Oleg Varlamov^{1}*

¹Bauman Moscow State Technical University, 2-ya Baumanskaya Street, 5/1, Moscow, 105005, Russian Federation

Abstract. The paper describes an approach to using mivar technologies for planning three-dimensional robot routes, taking into account the given obstacles. This system is of great importance in increasing the autonomy of robots, as it will ultimately help us get closer to creating artificial intelligence in mechanical engineering. At present, there are many tasks related to automation and robotics that require non-trivial solutions. The article describes three models of three-dimensional logical space and presents a visualization of route construction in these spaces for robotic complexes. Obstacles in three-dimensional logical space are understood as the absence of graph vertices and transitions between them in a certain specific area. The use of mivar technologies of logical artificial intelligence in the field of transport systems allows to significantly speed up route planning thanks to a unique mivar algorithm for processing information, which allows creating systems capable of making decisions in real time. This work is intended for researchers dealing with the problem of three-dimensional route planning using mivar technologies.

1 Introduction

In recent years, research in the field of control of robotic complexes (RC), for example, for unmanned transport, is actively developing. Vision systems are implemented primarily using neural network technologies. However, logical artificial intelligence should be used to make decisions in complex and non-standard situations.

Mivar technologies of logical artificial intelligence have been developed for quite some time [1]. For tasks in the format of production networks, they allow finding a solution with linear computational complexity. Recently, quite a few mivar expert systems have been created in various fields: systems for plant care [2], for managing educational programs [3] at the university [4], for intelligent management systems [5] of vehicles [6] and robots [7], system of making decisions about the safety of thermolabile blood components [8], for planning of robot actions [9] and for many other areas. Mivar technologies are also used in hybrid intelligent information systems (HIIS). HIIS can be used, for example, for

* Corresponding author: ovarlamov@gmail.com

generating advertising videos [10] or for analysis [11] and creating a brief overview [12] of judicial practice. In addition to the Mivar approach, metagraph representation [13] of data sets [14] can be used in HIIS to overcome the limitations [15] of existing knowledge bases [16].

The HIIS approach also includes neural network methods. Neural networks can be used for generating questions in Russian [17], for sentimental analysis of multilingual texts [18], for creating a brief summary of judicial acts [19] or for bypassing text CAPTCHAs [20] using convolutional neural networks [21]. The use of LiDAR has allowed solving a list of new tasks: evaluation of the quality of segmentation of an individual tree [22], determination of the location of trees and estimation of their diameters [23], inventory of trees [24] and classification of tree species [25] based on LiDAR data [26]. With the help of satellite imagery analysis [27] has been possible to solve tasks of segmenting trees damage [28] and identifying deforested areas [29] using neural networks [30]. In addition to the tasks described above, neural network technologies in HIIS allow solving a huge number of completely different tasks: management of data lakes [31], processing and visualization of tomography signals for making decisions on COVID [32], detection of energy theft in intelligent networks using explainable attention maps [33], for removing noise in audio and video [34], as well as for indoor navigation [35].

Due to the low requirements for computing power, it was decided to create autonomous decision support systems (DSS) for robots. Since in such robots a person does not make decisions, these systems were not called “decision support systems”, but “decision-making systems” (DMS). Subsequent research confirmed the correctness of the allocation at the management level of new systems, and were carried out based on Mivar expert systems (MES). Over the past few years, several working prototypes and layouts have been created based on MES, which together include: control systems (CS) for the movement of various robots and their groups; technical vision systems (TVS); DMS for robots, for example, the “RoboRazum” system. This DMS is created based on the server version of “KESMI (Wi!Mi)”, to which two blocks have been added to receive source data from TVS and to transmit control effects to CS for the movement of various robots and their groups. At present, the problem of creating Mivar models for DMS in robotic complexes is being solved.

Previously, a method for finding several trajectories of a robot’s movement in two-dimensional space was proposed, as well as an automated method for sequentially removing rules from the Mivar network and searching for new trajectories of the robot’s movement for their further comparison by the number of activated transition rules on the way from the initial location to the target. These techniques can be used to solve resource allocation optimization problems in mechanical artificial intelligence. Thus, at the moment, all the prerequisites have been created for the transition to the use of Mivar expert systems for planning RC routes in three-dimensional logical space. So, the creation of decision-making systems based on logic is a relevant and important task of robotics.

2 Application of Mivar expert systems for planning three-dimensional routes

First of all, it is necessary to generate an XML model that includes relationships, parameters and rules. There are three types of relationships depending on the weight with which the incoming vector arrives. The first type is unit, the second type is the diagonal of a unit square-the root of two, the third type is the diagonal of a unit cube-the root of three. The parameters are the vertices of the final graph. They reflect all possible states in space, that is, at what point an object can be located. Rules are transitions between states. Each

rule relates to one of the relations; accordingly, the weight of each rule coincides with the value specified in the relation formula.

Secondly, it is required to define obstacles and remove them from the model XML file. Obstacles can be of three types. The first type is obstacles similar in shape to a rectangular parallelepiped. To define them, you need to designate two opposite corners of the parallelepiped. The remaining points of the figure will be found automatically and removed from the XML file. The second type of obstacle is straight lines. They have a starting point and an ending point. For greater visualization functionality, these lines are displayed as a vector from the starting point to the ending point. The third type of obstacle is individual vertices of the graph, which for some reason cannot be reached. Next, obstacles of the second and third types are also removed from the model XML file. When deleting vertices from the model, the parameter and relations associated with that vertex are deleted, as well as all rules for which this point is an input or output.

After this, the resulting XML model is uploaded into the server version of the “KESMI Wi!Mi Razumator” using a POST request. If the upload was successful, then you can proceed to finding the route. To do this, you need to use a POST request to transfer to the server the ID of the uploaded model, the starting and the ending points of the route. Next, the response received from the server needs to be converted into the final route of the robot. If you need to find a route for several robots, then you need to perform several POST requests with starting and ending points. If the environment has changed, then the model needs to be generated again, then uploaded to the server, and then a route must be constructed taking into account the changed environment. After receiving the routes for the robots, you can visualize the results. For this purpose, data on all three types of obstacles is used, as well as on all constructed routes and their lengths. Visualization is implemented in the Python programming language using the “Matplotlib” library. An example of visualizing the route of one robot without taking into account obstacles is shown in Fig. 1. Examples of visualizing the routes of one robot and three robots also will also be demonstrated (Fig. 1), taking into account the three types of obstacles described above.

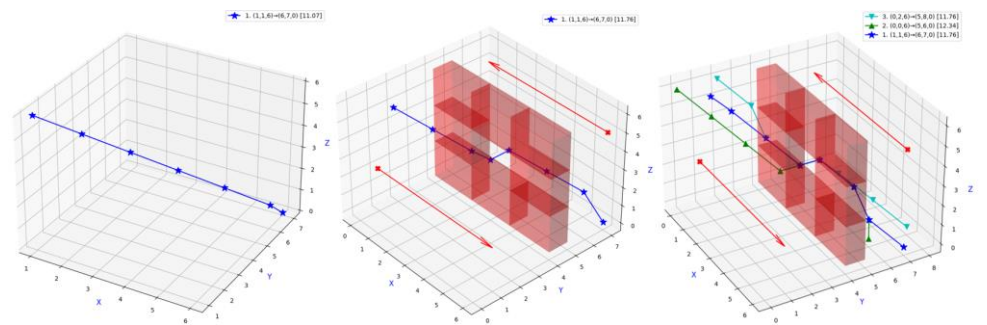


Fig. 1. An example of visualizing the routes of one and three robots taking obstacles into account.

3 Mivar knowledge bases of three-dimensional logical space

This work proposes three types of Mivar knowledge bases (MKB) of three-dimensional logical space. Visualization is done on a 5x5x5 model, as with a larger number of rules, clarity decreases (Fig. 2). Potentially, Mivar technologies can work with 5 million rules. Transitions in it go along the edges of the cubes. The model in the image contains 125 parameters and 600 rules (Fig. 2). This is an improved version, as transitions along the diagonals and diagonals of the cube faces have been added to it. This improvement allowed to build shorter routes in terms of the number of transitions, but here a problem arose that

for three-dimensional logical space all transitions must be unitary, and the side diagonal of the cube is equal to the square root of 2, and the diagonal of the cube is equal to the square root of 3. Also, there was a significant increase in the number of rules, which greatly affected performance. The model consists of 125 parameters and 2072 rules (Fig. 2). Transitions in this space go along the edges of tetrahedra and octahedra. In this type, it was possible to correct the shortcomings of the previous version, since it has almost 2 times fewer rules, unit edges, and a sufficient number of transitions. The model in the image contains 126 parameters and 1080 rules. Depending on the relation, the transition will have a certain weight. Constraints and classes are not used in this work, so they are left empty in the final XML file.

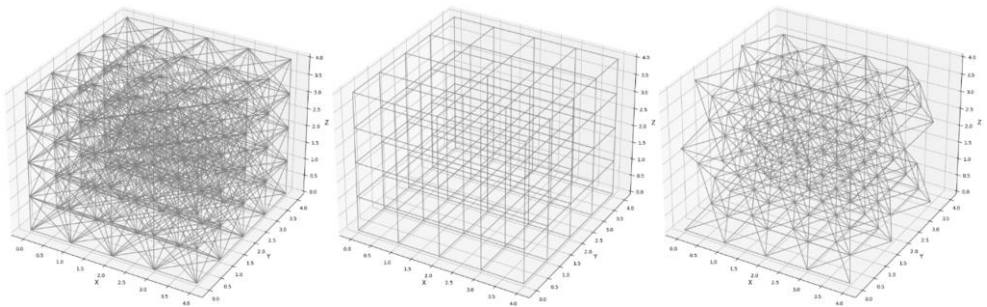


Fig. 2. Visualization of transition rules in three models of three-dimensional logic space.

4 Adding obstacles to Mivar knowledge bases

After generating an XML model of a Mivar knowledge base, which includes parameters and rules, it is necessary to set obstacles. An obstacle is an area that “KESMI (Wi!Mi)” cannot use for logical inference, as it is removed from the XML file of the Mivar model. In this work, only a spherical obstacle is taken as an example. For this obstacle, it is necessary to set the coordinates of the sphere and its radius. After that, all parameters and rules need to be checked. If the distance from the center of the sphere to a certain parameter is less than the radius of the sphere, then it is removed from the model. The rule has initial and final points. If one of these points is inside the sphere, then this rule is removed from the model. The visualization of the model in Fig. 3 is divided into different stages. At the first stage, the visualization of the Mivar knowledge base of three-dimensional logical space takes place, that is, those parameters and rules that did not fall into the spherical obstacle. Then, parameters and rules that fell into the sphere are added to the image. The sphere is drawn separately in Matplotlib using the plot_surface function. To keep the image of the sphere within the specified axes, a mask was added that nullifies values if there is a transition of zero axis values, and makes values maximum if there is a transition from the right.

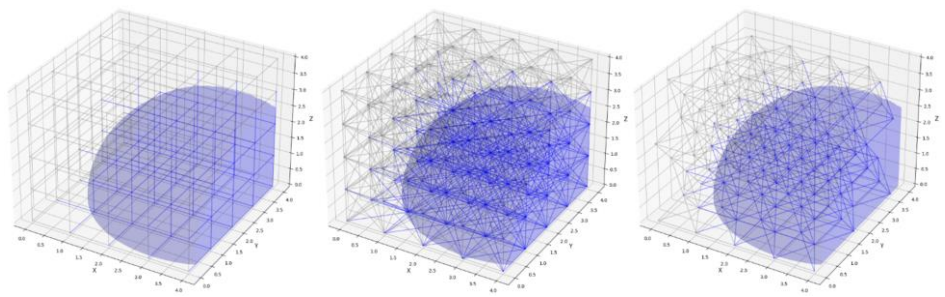


Fig. 3. Visualization of spherical obstacle in three models of three-dimensional logical space.

5 Application of Mivar technologies for planning RC routes

For route planning, the server version of “KESMI (Wi!Mi)” is used. At this stage, it is necessary to take a model without taking into account obstacles. The received XML file of the Mivar model is uploaded to KESMI using a POST request. If the upload was successful, then you can proceed to finding the route. For this, it is necessary to send the ID of the uploaded model to the server using a POST request, as well as the starting and ending points of the route. Then the response received from the server needs to be converted into the final RC route. If you need to find a route for several robots, then you should make several POST requests with starting and ending points. When the surrounding situation changes, the model needs to be regenerated, then uploaded to the server, and then a route is built taking into account the changed situation. After receiving the routes of the robots, the results can be visualized. The Python programming language and the “Matplotlib” library are used for visualization. Fig. 4 show the result of planning routes for three robots in spaces with transitions along the edges of cubes, with transitions along the edges, diagonals, and diagonals of cube faces, and with transitions along the edges of tetrahedra and octahedra, respectively.

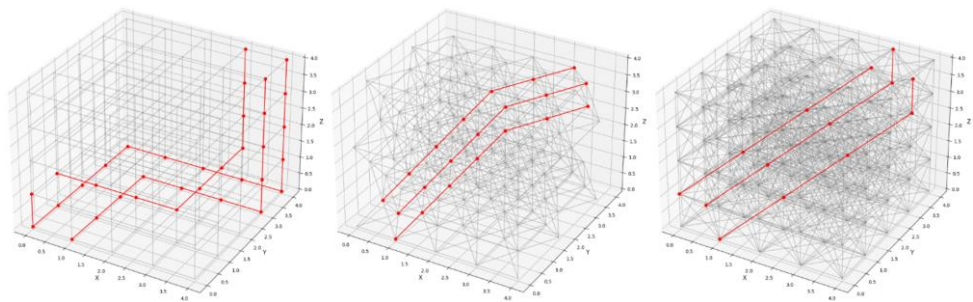


Fig. 4. Visualization of route planning in three models of three-dimensional logical space.

6 Planning RC routes taking into account obstacles

This section discusses group robot movement with a detour of a spherical obstacle. Initially, it is necessary to set the size of the three-dimensional logical space and generate an XML file of the Mivar model. After that, all rules and parameters that were inside the spherical obstacle are removed. Then the received model file is uploaded to “KESMI (Wi!Mi)”, and for three routes, three separate requests are made with initial and final coordinates. Since the parameters and rules related to the obstacle were removed from the model, Razumator

cannot use them to build a route. Accordingly, they will not be activated when the algorithm for constructing Mivar logical inference works. On the visualization, it can be observed that none of the routes were built through the obstacle. Fig. 5 shows route-planning taking into account a spherical obstacle in three-dimensional logical space with transitions along the edges of cubes, demonstrates the construction of routes in space with transitions along the edges, diagonals, and diagonals of cube faces and depicts route planning with transitions along the edges of tetrahedra and octahedra.

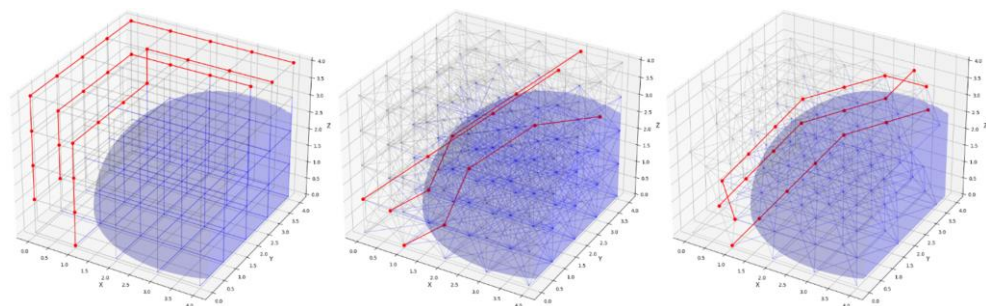


Fig. 5. Visualization of obstacle-aware route planning in three models of three-dimensional logical space.

7 Conclusion

The paper describes an approach to using mivar expert systems for planning three-dimensional robot routes taking into account obstacles. A method for generating an XML model is shown, with an example of the resulting file, and then with a description of how to define obstacles and remove them from the XML file. The program interacted with the server version of “KESMI Wi!Mi Razumator”. Examples of practical work of the decision-making system with visualization are given: the route of one robot without taking into account obstacles; routes for one and three robots, taking into account predefined obstacles. Further development of the idea of dynamic planning of robot and robotic systems’ trajectories in three-dimensional space using mivar technologies of logical artificial intelligence is proposed, which makes it possible to increase the intelligence and autonomy of robotic equipment.

Three types of Mivar knowledge bases of three-dimensional logical space are proposed to ensure the movement of robotic complexes. The work shows the possibility of software implementation of automatic generation of XML models. After the completion of model creation, they were tested using the server version of “KESMI (Wi!Mi)”. All parameters and rules are displayed correctly, therefore, the models are generated correctly.

The work also describes an approach to the application of Mivar technologies for planning RC routes taking into account obstacles. A method of setting obstacles and their removal from the XML file is described. The interaction of the program with KESMI is implemented. Examples of practical work of the system with the construction and visualization of routes for three robots in three types of Mivar knowledge bases of three-dimensional logical space, taking into account pre-set obstacles, are given.

The proposed development of the idea of dynamic planning of robot routes in three-dimensional space using Mivar technologies makes it possible to increase the independence and autonomy of robotic equipment, which will significantly help the development of autonomous robots and radio-controlled devices with logical artificial intelligence.

References

1. O. O. Varlamov, Evolutionary databases and knowledge for adaptive synthesis of intelligent systems. The Mivar Information Space (Radio and Communication) (2002)
2. D. V. Aladin, E. V. Aladina, D. A. Chuvikov, et al., IOP Conference Series: Earth and Environmental Science **954**, 012004 (2022) <https://doi.org/10.1088/1755-1315/954/1/012004>.
3. T. Guzeva, S. Egorov, K. Smetankin, et al., Lecture Notes in Networks and Systems **509** (2022) https://doi.org/10.1007/978-3-031-11058-0_64.
4. T. Guzeva, A. Parsheva, V. Babin, et al., Lecture Notes in Networks and Systems **509** (2022) https://doi.org/10.1007/978-3-031-11058-0_65.
5. O. O. Varlamov, Big Data Research **25**, 100241 (2021) <https://doi.org/10.1016/j.bdr.2021.100241>. (Q2).
6. D. A. Chuvikov, D. V. Aladin, L. E. Adamova, et al., Journal of Physics: Conference Series **2061(1)**, 012123 (2021) <https://doi.org/10.1088/1742-6596/2061/1/012123>.
7. D. V. Aladin, O. O. Varlamov, L. E. Adamova, et al., IOP Conference Series: Materials Science and Engineering **747(1)**, 012099. (2020) <https://doi.org/10.1088/1757-899X/747/1/012099>.
8. O. O. Varlamov, D. A. Chuvikov, V. N. Lemondzhava, et al., Biomedical Engineering **55**, 355–359 (2022) <https://doi.org/10.1007/s10527-022-10135-0>.
9. O. Varlamov, D. Aladin, Mach. Intell. Res. (2024) <https://doi.org/10.1007/s11633-023-1473-1>.
10. S. Chernobrovkin, I. Latkin, M. Belyanova, Y. Gapanyuk, *Using a hybrid intelligent information systems approach for advertising video generation*, in Proceedings of the Conference of Open Innovation Association, FRUCT 2021, 9347594 (2021)
11. M. O. Taran, G. I. Revunkov, Y. E. Gapanyuk, Studies in Computational Intelligence **1120**, 435–444 (2023)
12. M. O. Taran, G. I. Revunkov, Y. E. Gapanyuk, Studies in Computational Intelligence **1064**, 466–475 (2023)
13. Y. E. Gapanyuk, Pattern Recognition and Image Analysis **33(3)**, 300–305 (2023)
14. D. Fedyukin, D. Gromozdov, Y. Gapanyuk, Studies in Systems, Decision and Control **457**, 207–218 (2023)
15. V. Terekhov, Y. Gapanyuk, A. Kanev, *Metagraph representation for overcoming limitations of existing knowledge bases*, in Proceedings of the Conference of Open Innovation Association, FRUCT 2021, 9347601 (2021)
16. V. Tarassov, Y. Kaganov, Y. Gapanyuk, Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics) **12948**, 135–151 (2021)
17. M. A. Belyanova, A. M. Andreev, Y. E. Gapanyuk, Studies in Computational Intelligence **1008**, 217–223 (2022)
18. A. I. Kanev, G. A. Savchenko, I. A. Grishin, et al., *Sentiment analysis of multilingual texts using machine learning methods*, in Proceedings of the 2022 Conference of Russian Young Researchers in Electrical and Electronic Engineering, ElConRus 2022, 2022, pp. 326–331 (2022)
19. M. O. Taran, G. I. Revunkov, Y. E. Gapanyuk, Studies in Computational Intelligence **1008**, 292–298 (2022)

20. V. Terekhov, V. Chernenky, D. Ishkov, Communications in Computer and Information Science **1573**, 111–122 (2022)
21. D. O. Ishkov, V. I. Terekhov, *Text CAPTCHA Traversal with ConvNets: Impact of Color Channels*, in Proceedings of the 2022 4th International Youth Conference on Radio Electronics, Electrical and Power Engineering, REEPE 2022 (2022)
22. I. A. Grishin, T. Y. Krutov, A. I. Kanev, V. I. Terekhov, Optical Memory and Neural Networks (Information Optics) **32**, 270–276 (2023)
23. I. A. Grishin, V. I. Terekhov, *Procedure for Locating Trees and Estimating Diameters Using LiDAR Data*, in Proceedings of the 2023 5th International Youth Conference on Radio Electronics, Electrical and Power Engineering, REEPE 2023 (2023)
24. I. A. Grishin, E. K. Sakharova, S. M. Ustinov, et al., Studies in Computational Intelligence **1064**, 3–11 (2023)
25. E. K. Sakharova, D. D. Nurlyeva, A. A. Fedorova, et al., Studies in Computational Intelligence **1008**, 319–324 (2022)
26. C. Zhang, Y. Huang, E. K. Sakharova, et al., Studies in Computational Intelligence **1120**, 32–38 (2023)
27. D. A. Gurianov, K. S. Myshenkov, V. I. Terekhov, *Software Development Methodologies: Analysis and Classification*, in Proceedings of the 2023 5th International Youth Conference on Radio Electronics, Electrical and Power Engineering, REEPE 2023 (2023)
28. N. S. Podoprigrorova, G. A. Savchenko, K. R. Rabcevich, et al., Studies in Computational Intelligence **1120**, 380–388 (2023)
29. A. I. Kanev, A. V. Tarasov, A. N. Shikhov, et al., Sovremennye Problemy Distsionnogo Zondirovaniya Zemli iz Kosmosa **20(3)**, 136–151 (2023)
30. A. Volkov, O. Varlamov, Journal of Physics: Conference Series **2131**, 032003 (2021) <https://doi.org/10.1088/1742-6596/2131/3/032003>.
31. A. A. Sukhobokov, Y. E. Gapanyuk, A. S. Zenger, A. K. Tsvetkova, Procedia Computer Science **213(C)**, 407–414 (2022)
32. D. O. Ishkov, B. S. Goryachkin, V. I. Terekhov, *Processing and Visualization of Tomography Signal with AI: Decision Making on COVID-19*, in Proceedings of the 2022 4th International Youth Conference on Radio Electronics, Electrical and Power Engineering, REEPE 2022 (2022)
33. D. O. Ishkov, V. I. Terekhov, K. S. Myshenkov, *Energy Theft Detection in Smart Grids via Explainable Attention Maps*, in Proceedings of the 2023 5th International Youth Conference on Radio Electronics, Electrical and Power Engineering, REEPE 2023, 2023.
34. A. Kanev, M. Nazarov, D. Uskov, et al., *Research of Different Neural Network Architectures for Audio and Video Denoising*, in Proceedings of the 2023 5th International Youth Conference on Radio Electronics, Electrical and Power Engineering, REEPE 2023 (2023)
35. A. D. Volgina, D. S. Kirillov, A. N. Kravtsov, et al., *The Robot-Guide for Indoor Navigation*, in Proceedings of the 2023 5th International Youth Conference on Radio Electronics, Electrical and Power Engineering, REEPE 2023 (2023)