

Energy-Efficient Path Planning: Designed Software Implementation

Vyacheslav Petrenko

Academic department of Organization
and Technology of Information
Protection
North-Caucasus Federal University
Stavropol, Russia
vip.petrenko@gmail.com

Fariza Tebueva

Academic department of Applied
Mathematics and Computer Security
North-Caucasus Federal University
Stavropol, Russia
fariza.teb@gmail.com

Vladimir Antonov

Academic department of Applied
Mathematics and Computer Security
North-Caucasus Federal University
Stavropol, Russia
ant.vl.02@gmail.com

Mikhail Gurchinsky

Academic department of Applied
Mathematics and Computer Security
North-Caucasus Federal University
Stavropol, Russia
gurcmikhail@yandex.ru

Nikolay Untewsky

Academic department of Applied
Mathematics and Computer Security
North-Caucasus Federal University
Stavropol, Russia
untewsky@yandex.ru

Abstract—The paper continues developing the method of a real-time quasi-optimal trajectory planning for anthropomorphic manipulator. Our goal includes software implementation of previously designed methods and algorithms of the quasi-optimal trajectory for anthropomorphic manipulator. The general structure and interrelation of the developed methods and algorithms is described, as well as their significance for obtaining the final result. Component diagrams and communication diagrams are presented, on the basis of which the interfaces of software components are built. Their structure and composition of each software component are given. The description of the interfaces of various components of the system allows the use of design components as part of other software systems, for example, designed to control the manipulators of an anthropomorphic robot or as part of other modeling systems. The illustration of the developed software implementation is given. For the software implementation performed by the author, with standard input data, the necessary variable memory capacity is about 212 Kbytes, which allows using the proposed software implementation on the computer systems of existing manipulators in real time.

Keywords—*anthropomorphic manipulator, path planning, energy efficient movement, software implementation, traffic planning, software components*

I. INTRODUCTION

The problem of energy consumption for robot arm movement is an important research issue around the world. This challenge is particularly acute for autonomous robotic systems feeding by non-stationary or portable power supply.

In previous papers [1-2], we considered energy-efficient path planning methods and algorithms in the presence of spherical, or parallelepiped obstacles.

We set the task involving an anthropomorphic manipulator with 7 degrees of mobility (Fig. 1). Note: Fig. 1 uses following notation: B_{1-4} are shoulder, elbow, wrist joints and end effector, respectively; A_{1-7} are degrees of the manipulator's mobility, A_8 is the manipulator's end effector.

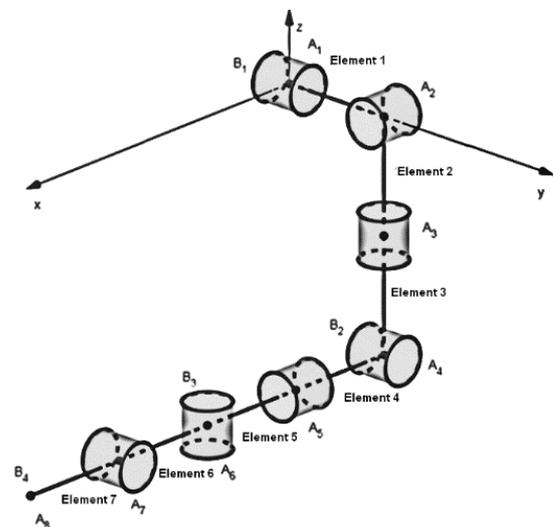


Fig. 1. Kinematic diagram for anthropomorphic manipulator with 7-degrees of mobility

The task of the path planning is to find a set of intermediate positions for the manipulator to pass through from the initial position to the target one, with obstacles to bypass.

The existing methods and algorithms of motion planning for AR manipulator in the workspace with a typical obstacle can be generalized to a method designed in [3]. This algorithm is presented in Fig. 2.

To calculate the manipulator's final position in the workspace with obstacle, we used the optimization method for the inverse kinematics problem based on the Denavit-Hartenberg representation and the generalized reduced gradient method where the condition for finding the optimal value of the function intends the condition for minimizing the energy consumed by manipulator engines. [4].

When adding some complications to the path we used iterative piecewise linear generation of the trajectory in the

workspace with a sphere and parallelepiped like obstacles [5].

After finding the way to move the manipulator around an obstacle, we solve the inverse problem of kinematics employing the method of forming a geometric solution to the inverse kinematics problem for chains with kinematic pairs of rotational type only [6-11].

Thus, the use of the developed mathematical methods in conjunction with algorithms allows energy efficient motion planning for the anthropomorphic robot manipulator in the presence of a typical obstacle.

This article is devoted to a software package that implements the above methods and algorithms.

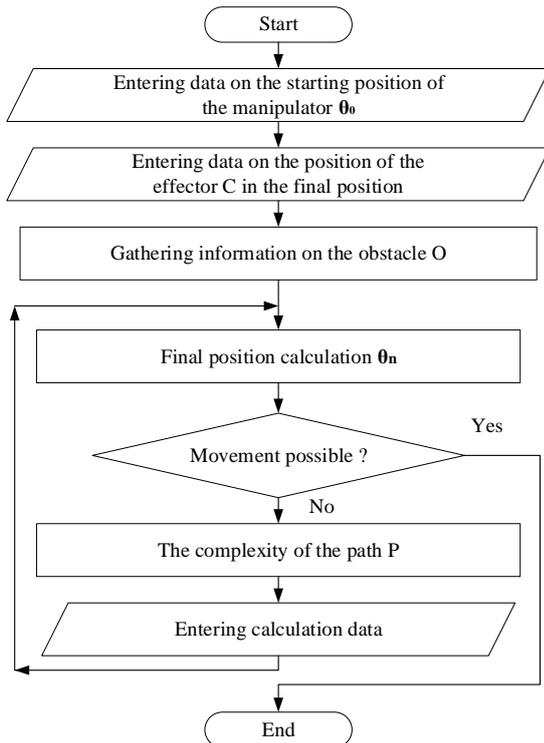


Fig. 2. Energy efficient trajectory planning algorithm performed by anthropomorphic robot arm

II. COMPONENT-BASED SOFTWARE ENGINEERING

We use systems engineering techniques to simulate the energy-efficient path for moving the manipulator in the presence of obstacles in the robot workspace.

For implementation of schemes describing various aspects of the complex being developed we employ analysis modeling tool UML 2.0 - universal modeling language (its formalism and notation). With respect to the systems engineering approach, a list of potential stakeholders - individuals who have some interest in the software package and its functionality is generated.

The diagram of the simulation system stakeholders use shown in Fig. 3.

Software architectures have been synthesized based on the usage of the software package. It is desirable to separate the design components implementing the algorithms and methods from the interface. This separation is founded on the

assumption of further use of the design components in robotic manipulator control system. In Fig. 4 a software system component diagram reflecting the software architecture.

The component “GUI” is a graphical user interface. The component of “ConsoleApp” is a program without a graphical shell, designed to automatically run the simulation system.

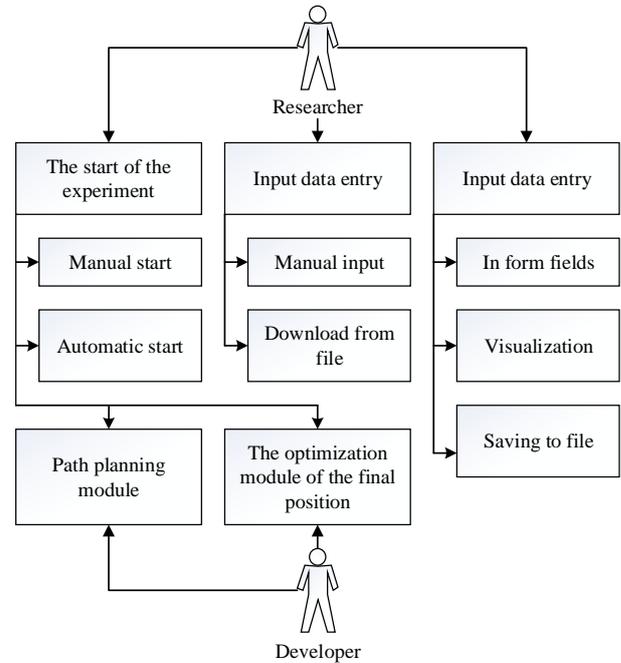


Fig. 3. Diagram of simulation system stakeholders use

The component “Planning” is a class library implementing the designed mathematical modeling for path planning, while the component “Optimization” is a class library for solving the optimization problem of finding generalized coordinates for the final position.

In Fig. 5 the sequence diagram of components interaction in the most difficult case - a numerical experiment with visual conclusion.

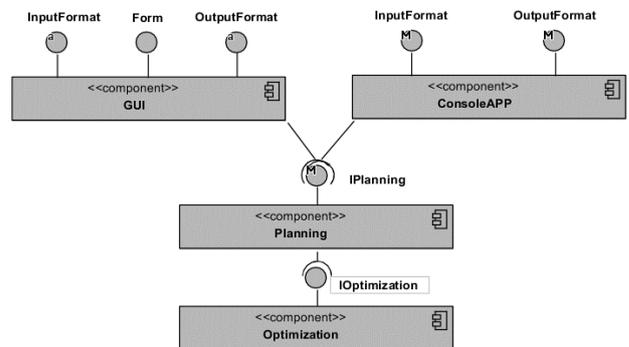


Fig. 4. Developed software system component diagram

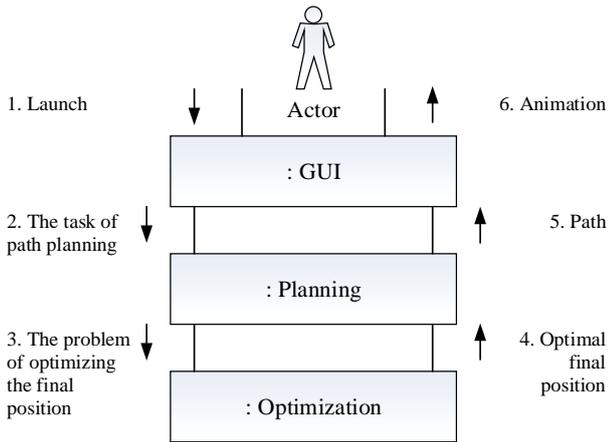


Fig. 5. Communication diagram of simulation system

Using the component and communication diagrams, we have designed component-based software interfaces.

A. Interface component “GUI” design

The component “GUI” must support the following functions: user input initial task; loading input data from a file; calculation start; exporting calculation results to form fields; visualizing of the results obtained; writing calculation results to a file.

A convenient software framework providing three-dimensional visualization is the key element when creating a software component.

For this purpose, Windows Presentation Foundation (WPF) providing rich graphical user interfaces and environments for applications have been selected as the most suitable. Since WPF uses DirectX technology the designed system is provided with hardware acceleration, thus improving the program performance based on WPF. In addition to rich opportunities that WPF offers in creating interfaces, it ensures high performance 3D graphics to visualize.

With regard to the software component purposes, it is important to adhere to the following logical line of work. Parameters for calculations can be set in the form fields or entered by loading a data file. In this case, click the Download button and select the corresponding file to download. The input data in the form fields can be saved if desired [12].

The algorithm for reading from the input file is in Fig. 6. Once the initial data have been entered, click Calculation button to start planning the path. Create and initialize a class object Planning, the getPath method is invoked (section 2.3). The result is displayed in the corresponding fields while the result visual interpretation in graphic area. To store the results to a file, click Save button. To visualize the results obtained earlier, click Download button. In this case, the results are exported to the corresponding fields, and visual interpretation is displayed in the graphic area. The algorithm of the program after pressing the “Calculation” button is shown in Fig. 7.

A. The software component design to automatically run a console application “ConsoleApp”

This component is a program designed for automatic path planning using a command line call. Read-write input-output correspondingly is performed to text files, the paths to files are indicated in the keys of the call program command. The format of the input file must follow a particular format in Fig. 8. The format of the output file must follow a particular format in Fig. 9.

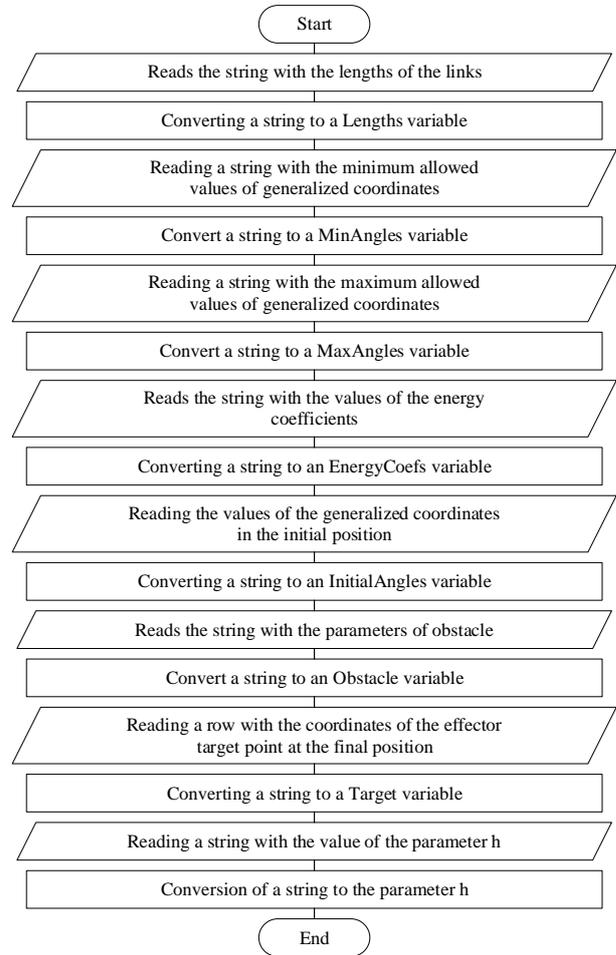


Fig. 6. Loading input data from a file

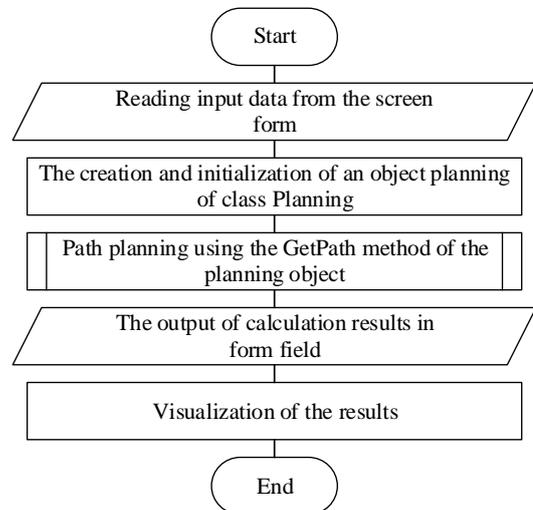


Fig. 7. The processing algorithm pressing the button “Calculate”

```

1 LinkLengths: 1 1 1
2 MinAngles: 1,57 -3,14 -3,14 -3,14 -3,14 -3,14 -1,57
3 MaxAngles: 4,71 0 0 0 0 0 1,57
4 EnergyCoefs: 6 5 4 3 2 1
5 InitialAngles: 0 0 0 0 0 0
6 Obstacle: 0,2 0,2 0,2 0,15
7 TargetCoords: 1 1 1
8 GapParameter: 0,01

```

Fig. 8. Input data file format

```

1 5
2 0 1,571 0 0 0 0
3 0,415880937106431 1,12329051567952 0 0 0 0
4 0,776300916017796 1,00536277797862 0 0 0 0
5 1,14933530512516 1,11917184095962 0 0 0 0
6 1,571 1,571 0 0 0 0

```

Fig. 9. Output data file format

In the input data, each line first indicates the data is decrypted with a single word, and then a colon is used followed by numerical values of the parameters, separated by whitespaces characters. A decimal point, is placed as separator.

Below data decryption:

- “LinkLengths” – link lengths of the manipulator.
- “MinAngles” – minimally permissible values of generalized coordinates.
- “MaxAngles” – maximally permissible values of generalized coordinates.
- “EnergyCoefs” – weighting coefficients of the target function.
- “InitialAngles” – generalized coordinates in the initial position.
- “Obstacle” – obstacle center point position and its characteristic (radius of a sphere and coordinates of the vertices for the parallelepiped).
- “TargetCoords” –coordinates of the end-effector final position.
- “GapParameter” – h parameter value.

In the output file, the first line indicates the number of positions the manipulator executed within the calculated path, then each line carries values of the generalized coordinates corresponding to each position vector in order, starting from the initial position to the final one. The numbers in the lines are separated using the tab character; a decimal point is placed as separator.

A Form-based interface (graphical application) is a kind of user interface. The form interface elements provide input of all required information for numerical experiment, display survey result in text form, as well as graphical visualization.

The scheme of the algorithm of this program is shown in Fig. 10.

The required input data include link lengths, limiting the angular rotation design, weighting coefficients of the objective function, obstacle data, generalized coordinates for the end effector initial position, Cartesian coordinates for the end effector final position, and the parameter h whose value is responsible for the distance from path to obstacle [5].

B. Component “Planning” design implementing path planning

The component “Planning” whose sole job is to create a Path object by the specified input basically employ the “GetPath” method. A generalized algorithm for energy-efficient trajectory planning designed by mathematical modeling is presented in [2]. In Fig. 11 the software implementation method [13].

The recursive path complicating algorithm is shown in Fig. 12 and in detail in [1-5].

C. Component “Optimization” design for solving inverse kinematics problem

The component “Optimization” is intended to provide optimal solution to inverse kinematics problem for determining the final position of the anthropomorphic robotic arm [2, 4].

In Fig. 13 the class interface implementing this function. Fig. 14 presents the “GetSolution” method.

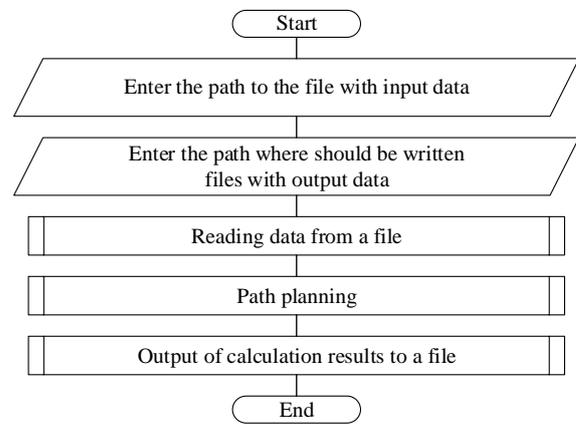


Fig. 10. The algorithm of the program “Console App”

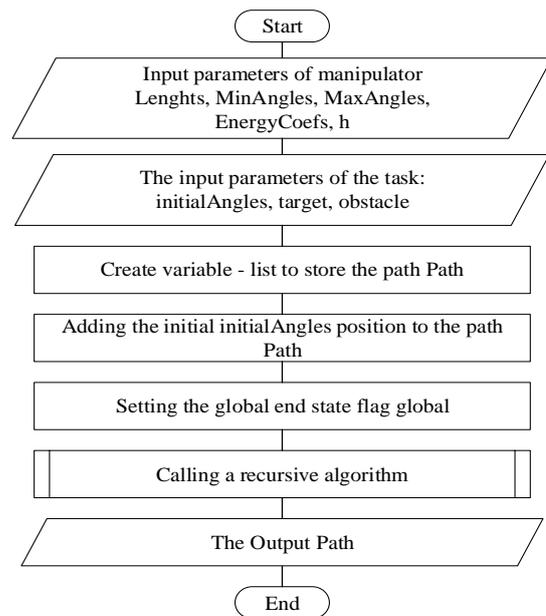


Fig. 11. Algorithm of the “GetPath” method of the “Planning” class

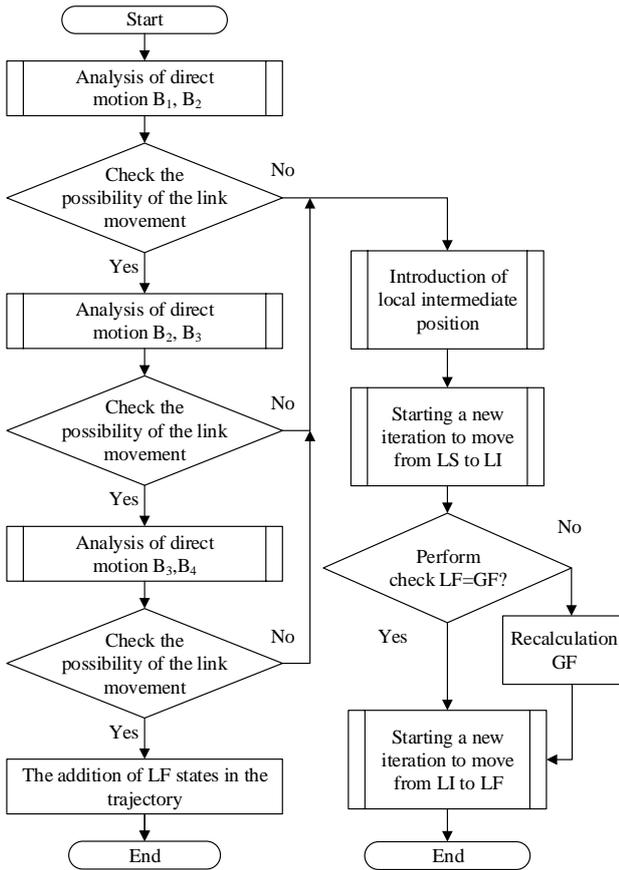


Fig. 12. The algorithm of the method of recursive path complication “Recursion”

The data that does not change during computer system operations are interface fields. An algorithm for solving inverse kinematics problem is allocated in a separate method, and the changing input data is its parameters.

In Fig. 13 the following notation applies:

- Field “Lengths” is an array of link lengths.
- Field “MaxAngles” is an array of maximum allowable value of generalized coordinates.
- Field “MinAngles” is an array of minimum allowable value of generalized coordinates.
- Field “EnergyCoefs” is an array of coefficients of the target function.
- Field “Obstacle” is an obstacle data.
- Method “GetSolution” is a solution method to optimization problem.
- Parameter “initialAngles” is initial (modeling) position values of generalized coordinates.
- Parameter “target” is Cartesian coordinates of the end effector final position.

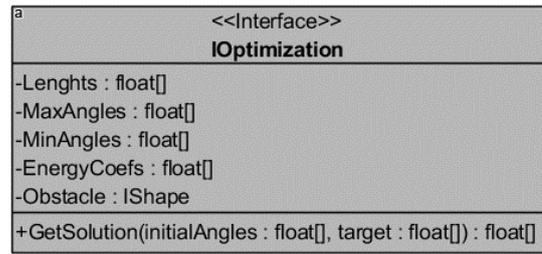


Fig. 13. Component interface “IOptimization”

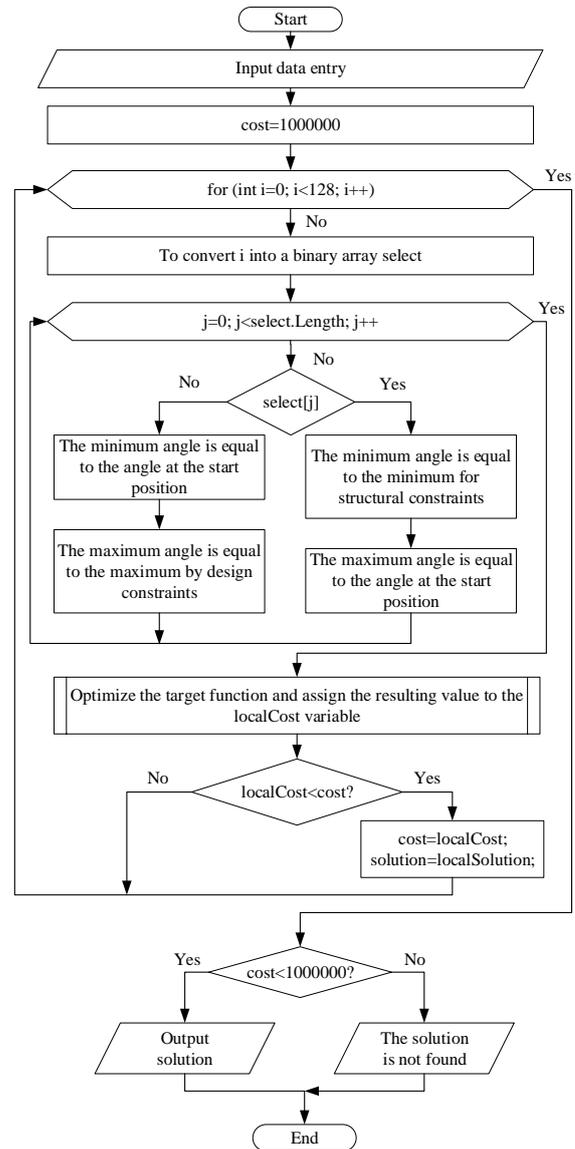


Fig. 14. Algorithm of the “GetSolution” method of the “Optimization” component

Once the initial data is entered, 128 subspaces are enumerated to solve the optimization problem. The numbers i are converted to binary code and then to Boolean array Select. The j -th element of the Select array determines where in the generalized coordinate the considered subspace is located: in the range from the minimum admissible value to the initial position value or from the initial position value to the maximum admissible one [14].

When the subspace boundaries are determined, the optimization problem is solved with Optimize function as a result of this function a solution localSolution with lowest energy localCost is found in this subspace. If the value of the target function is less than the value of the best global solution Cost, the local solution obtained becomes global.

Upon all search subspaces have been enumerated the values of objective function Cost are compared to the initial value. If the value has been changed, the found solution "Solution" is displayed. If there are no feasible solutions with minimum cost in any of subspaces No solution is displayed.

III. SIMULATION SYSTEMS

According to the designed model system, we present the results of the software operation and implementation of quasi-optimal energy path planning for anthropomorphic manipulator.

Fig. 15 shows the input and output data.

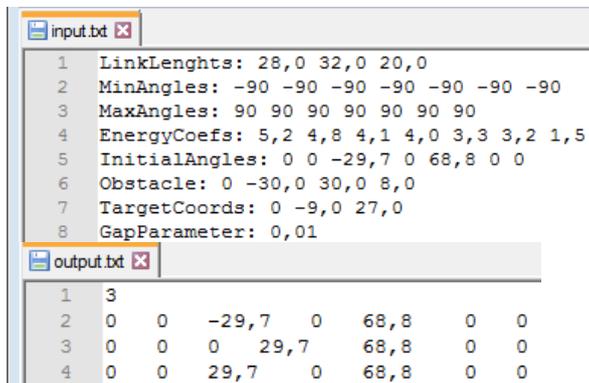


Fig. 15. The input and output data for modelling

Table 1 shows the module results after mathematical modeling have been applied. The results of a module that implements the mathematical modeling for planning an energy-efficient trajectory.

In Table 2 the program for inverse kinematics problem to calculate the final position of the anthropomorphic robot manipulator with respect to the input presented in Table 1.

In Fig. 16 presented implementation of "GUI" framework.

TABLE I. RESULTS OF ENERGY-EFFICIENT PATH PLANNING FOR ANTHROPOMORPHIC MANIPULATOR: PROGRAM IMPLEMENTATION

Parameter	Value						
LinkLengths	28	32			20		
MinAngles	-90	-90	-90	-90	-90	-90	-90
MaxAngles	90	90	90	90	90	90	90
EnergyCoefs	5.2	4.8	4.1	4.0	3.3	3.2	1.5
InitialAngles	0	0	-29.7	0	68.8	0	0
Obstacle	0	-30		30		8	
TargetCoords	0		-9			27	
GapParameter	0.01						

TABLE II. PROGRAM IMPLEMENTATION TO SOLVE INVERSE KINEMATICS PROBLEM

Parameter	Value						
LinkLengths	28	32			20		
MinAngles	-90	-90	-90	-90	-90	-90	-90
MaxAngles	90	90	90	90	90	90	90
EnergyCoefs	5.2	4.8	4.1	4.0	3.3	3.2	1.5
InitialAngles	0	0	-30	0	68.8	0	0
Obstacle	0	-30		30		8	
TargetCoords	0		-9			27	
GapParameter	0.01						
Best value of target function	691.3						
Value of target function after optimization	243.5						
Solution	0	0	29.7	0	68.8	0	0

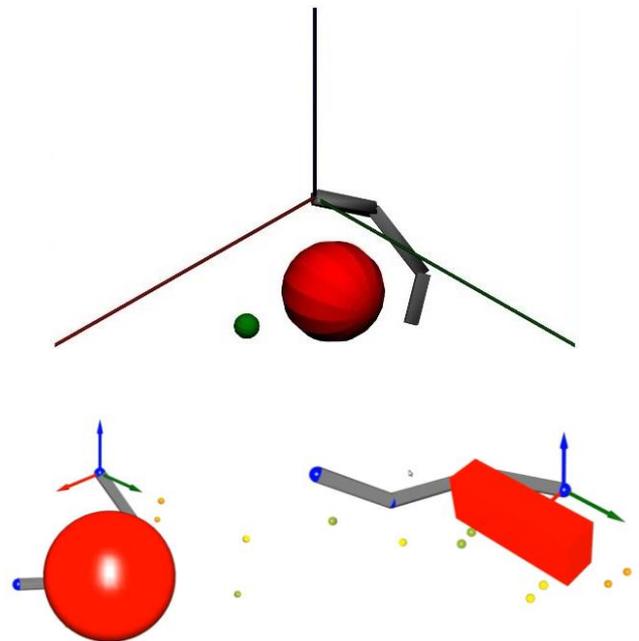


Fig. 16. Graphical user interface

IV. CONCLUSION

This paper presents a software for methods and algorithms of the quasi-optimal energy path planning in the presence of typical sphere or parallelepiped obstacles. The architecture of the developed simulation system includes two design and two interface components.

The component interface description allows using design components in other software packages, for example, software packages for robotic manipulator control or in other simulation systems.

The design component "Optimization" can be used for solving optimization problems of minimizing motion energy consumption between the initial and final nod positions of the robotic manipulator. The design component "Planning" is intended for the real time energy efficient trajectory planning for the robotic arm in the workspace with typical obstacles. This component is applicable both in combination with optimization and separately.

Employing the complex of problem-oriented programs, we have developed the interface components “ConsoleApp” and “GUI” for computational experiments. The first component is for text input and output, it can be launched automatically allowing batch data processing, and the second component is intended for motion planning visualization.

Thus, the developed software implementation of quasi-optimal energy path planning for anthropomorphic manipulator is flexible and expandable and can supply with a wide range of potential applications.

ACKNOWLEDGMENT

This research is financially supported by the Ministry of Science and Higher Education of the Russian Federation under the Grant agreement 14.575.21.0166 from 26 September 2017. The research topic: “Development of the software and hardware system of the control system based on the solution of the inverse problem of dynamics and kinematics” (Unique reference identifier of the agreement: RFMEFI57517X0166). Work on the project is carried out at the North-Caucasus Federal University (NCFU).

REFERENCES

- [1] Petrenko V.I., Tebueva F.B., Antonov V.O., Gurchinsky M.M., Untewsky N.Y. The method of the quasioptimal per energy efficiency design of the motion path for the anthropomorphic manipulator in a real time operation mode. – Proceedings of REMS 2018 – Russian Federation & Europe Multidisciplinary Symposium on Computer Science and ICT. – CEUR Workshop Proceedings. – 2018. – vol. 2254. – pp. 245-253.
- [2] Antonov V.O. A review of existing methods for planning the trajectory of anthropomorphic manipulators in a volumetric space with obstacles // The role and importance of science and technology for the development of modern society: materials of the International Scientific and Practical Conference (Volgograd, May 16, 2018). In 2 parts, Part 2. Ufa: Omega Science: 2018. P. 17-20 (in Russian).
- [3] Antonov V.O. Development of mathematical methods and algorithms for planning an energy-efficient path for moving an anthropomorphic robot manipulator in the presence of a typical obstacle. PhD thesis, North Caucasus Federal University, Stavropol, 2018 (in Russian).
- [4] Antonov V.O. Analysis of the computational complexity of the method of iterative piecewise linear generation of the trajectory of the three-link anthropomorphic manipulator in volumetric space with an obstacle / V.O. Antonov // Bulletin of the Southwestern State University. - 2018.- V.22. - No. 3 (78). - P. 13-29 (in Russian).
- [5] Petrenko, V.I. The method of planning the trajectory of a point in space with an obstacle based on an iterative piecewise linear approximation / V.I. Petrenko, F.B. Tebuyeva, V.O. Antonov, M.M. Gurchinsky // Control, communication and security systems. - 2018. - No. 1. - P. 168-182 (in Russian).
- [6] Petrenko, V.I., Tebueva, F.B., Gurchinsky, M.M., Antonov, V.O., Shutova, J.A. The method of forming a geometric solution of the inverse kinematics problem for chains with kinematic pairs of rotational type only // IOP Conference Series: Materials Science and Engineering. - 2018. - Volume 450. - Issue 4. - Номер статьи 042016.
- [7] Qi R., Zhou W., Wang T. An obstacle avoidance trajectory planning scheme for space manipulators based on genetic algorithm. *Jiqiren/Robot*, 2014, vol. 36, no. 3, pp. 263-270.
- [8] Xidias E. K. Time-optimal trajectory planning for hyper-redundant manipulators in 3D workspaces. *Robotics and computer-integrated manufacturing*, 2018, no. 50, pp. 286-298.
- [9] Liu W., Chen D., Zhang L. Trajectory generation and adjustment method for robot manipulators in human-robot collaboration. *Jiqiren Robot*, 2016, vol. 38, no. 4, pp. 504-512.
- [10] Ren Z. W., Zhu Q. G., Xiong, R. Trajectory planning of 7-DOF humanoid manipulator under rapid and continuous reaction and obstacle avoidance environment. *Zidonghua Xuebao/Acta Automatica Sinica*, 2015, vol. 41, no. 6, pp. 1131-1144.
- [11] Howard T., Pivtoraiko M., Knepper R. A., Kelly A. Model-predictive motion planning. *IEEE Robotics and Automation Magazine*, 2014, vol. 21, no. 1, pp. 64-73.
- [12] Certificate 2018617978 Russian Federation. Certificate of official registration of the computer program. Program planning the trajectory of a three-link anthropomorphic manipulator on the basis of the iterative piecewise-linear approximation / V. I. Petrenko, F. B. Tebueva, V. O. Antonov, M. M. Gurchinsky, applicant and owner Federal STATE Autonomous educational institution North Caucasus Federal University (RU). №. 2018615095; application. 21.05.2018; publ. 05.07.2018, Register of computer programs. 1p (in Russian).
- [13] Certificate 2018617979 Russian Federation. Certificate of official registration of the computer program. Software complex visualization of the trajectory of the three-link anthropomorphic manipulator in a bulk space with an obstacle / V. I. Petrenko, F. B. Tebueva, V. O. Antonov, M. M. Gurchinsky, E. N. Laskin, applicant and owner Federal STATE Autonomous educational institution North Caucasus Federal University (RU). №. 2018615094; application. 21.05.2018; publ. 05.07.2018, Register of computer programs. 1p (in Russian).
- [14] Certificate 2018617980 Russian Federation. Certificate of official registration of the computer program. The program of calculating the optimal values of generalized coordinates anthropomorphic manipulator according to the criterion of energy consumption, to move the effector to a given point / V. I. Petrenko, F. B. Tebueva, V. O. Antonov, M. M. Gurchinsky, D. E. Pijevskii, applicant and owner Federal STATE Autonomous educational institution North Caucasus Federal University (RU). №. 2018615092; application. 21.05.2018; publ. 05.07.2018, Register of computer programs. 1p (in Russian).
- [15] Intelligent tools to research plan development Popov, D., Rizvanov, D., Sjunturenko, O., Yusupova, N., Khisamutdinov, R. 2005 WMSCI 2005 - The 9th World Multi-Conference on Systemics, Cybernetics and Informatics, Proceedings 9, c. 83-88.