DOI 10.15622/ia.21.3.1

V. PSHIKHOPOV, M. MEDVEDEV, V. KOSTJUKOV, F. HOUSSEIN, A. KADHIM

TRAJECTORY PLANNING ALGORITHMS IN TWO-DIMENSIONAL ENVIRONMENT WITH OBSTACLES

Pshikhopov V., Medvedev M., Kostjukov V., Houssein F., Kadhim A. **Trajectory Planning Algorithms in Two-Dimensional Environment with Obstacles.**

Abstract. This article proposes algorithms for planning and controlling the movement of a mobile robot in a two-dimensional stationary environment with obstacles. The task is to reduce the length of the planned path, take into account the dynamic constraints of the robot and obtain a smooth trajectory. To take into account the dynamic constraints of the mobile robot, virtual obstacles are added to the map to cover the unfeasible sectors of the movement. This way of accounting for dynamic constraints allows the use of map-oriented methods without increasing their complexity. An improved version of the rapidly exploring random tree algorithm (multi-parent nodes RRT – MPN-RRT) is proposed as a global planning algorithm. Several parent nodes decrease the length of the planned path in comprise with the original one-node version of RRT. The shortest path on the constructed graph is found using the ant colony optimization algorithm. It is shown that the use of two-parent nodes can reduce the average path length for an urban environment with a low building density. To solve the problem of slow convergence of algorithms based on random search and path smoothing, the RRT algorithm is supplemented with a local optimization algorithm. The RRT algorithm searches for a global path, which is smoothed and optimized by an iterative local algorithm. The lower-level control algorithms developed in this article automatically decrease the robot's velocity when approaching obstacles or turning. The overall efficiency of the developed algorithms is demonstrated by numerical simulation methods using a large number of experiments.

Keywords: mobile robots, motion planning, path planning, motion control, robot motion.

1. Introduction. Path planning is a key problem in designing motion planning systems. The problem is to determine the trajectory that connects the initial and final position of the robot and ensures that there are no collisions with obstacles. In this case, the problem can be solved by taking into account the dynamics of the robot, the uncertainty and non-stationarity of the environment, the time for calculating the path, and the physical feasibility of the trajectory. The planning problem is often formulated as the problem of optimizing the state of the robot's current position relative to the target position. Most often, this problem is solved in the configuration space [1 – 3], which consists of a set of obstacles, kinematic and dynamic constraints, and a set of robot points. In this case, a configuration is understood as a set of variables that uniquely determine the robot's position in space. Planning methods are divided into global and local. Global methods build a route based on a known map, while local methods correct the path when unmapped obstacles are detected. Planning can be carried out in discrete and continuous spaces.

Cell decomposition methods [2] are the basis of a significant number of path-planning algorithms, which implies decomposing the original space into discrete cells, for example, according to the procedure [4]. After decomposition, the search for a path is already carried out in a discrete space that describes the original space with certain accuracy. In general, the division of space into cells that is optimal in terms of accuracy requires different resolutions for different regions of this space. In this regard, the methods of adaptive cell decomposition are widely used today [5]. The main problem of this method is the increase in computational complexity with the growth of requirements for the accuracy of planning the robot's path, which makes this problem remaining relevant [6].

Cell decomposition allows applying discrete search methods based on graph theory. The most popular methods are the A* and D* algorithms [7, 8].

A* algorithm is a development of Dijkstra's algorithm, in which the computational complexity is reduced due to the heuristic function for estimating the path cost. At the same time, the algorithm demands on memory, and its computational complexity increases significantly with the number of cells on the map. In this regard, there are modifications of A* aimed at eliminating these disadvantages. Such modifications include A* algorithms with iterative deepening [9], A* with memory constraints [10], hierarchical A* [11], and A* with dynamic change in edge weights [12, 13].

D* algorithm [8] provides for path planning under conditions when information about the environment can be updated while the robot is moving, for example, when new obstacles are detected.

It should be noted that the considered path planning algorithms give non-smooth paths. In this regard, these methods are supplemented by additional methods that are used to smooth the path [14]. Taking into account the dynamic constraints imposed by the inertia of the robot is also a separate laborious task.

Smooth paths can be obtained using the potential field method [15], which has been studied in detail in the works of various authors [16 - 19]. The potential field method is computationally efficient and allows path planning in uncertain environments. However, taking into account the dynamic properties of the target and obstacles is a separate problem since it is required to form the functions of attraction and repulsion depending on the positions and velocities. This problem is considered in [20].

The main disadvantage that limits the application of the potential field method in practice is the possible presence of local minima. The proposed approaches to solving this drawback solve the problem in particular cases. There is currently no good solution to this problem. In addition, the coefficients of attractive and repulsive forces are usually chosen heuristically, and the potential field method experiences difficulties when planning in a complex environment with narrow corridors, passages, and non-convex obstacles. Despite the partial solution of problems, for example, using vector histograms [19], it is required to involve other planning algorithms.

One way to solve the problem of local minima is to use random forces in them that push the robot out of the indicated minima. This idea has been developed in sampling methods, which include probabilistic planning methods.

At present, probabilistic path-planning algorithms in discrete space are widely used [21]. Algorithms of this type include Probabilistic Roadmap (PRM) [3, 22, 23] and rapidly-exploring random trees (RRT) [2, 24 - 26]. The PRM algorithm usually does not take into account the dynamics of the mobile robot. To consider it, significant computational costs are required [22, 27]. The RRT algorithm can take into account the dynamics of a mobile robot [24, 25] since it builds a feasible path in free space.

The main limitations of these methods are slow convergence to the optimal solution, high requirements for the memory used [22, 27], and limitations in planning under conditions of incomplete certainty of the environment and robot model [22]. In solving these problems, algorithms based on a probabilistic approach can be effectively used as global path planners.

In this regard, let us consider in more detail the work aimed at improving the efficiency of algorithms based on a probabilistic approach.

1.1. Related work. The problem of using the RRT and RPM algorithms under conditions of partial uncertainty is considered in [22], in which it is proposed to divide the entire area of operation into separate regions. Links between regions are searched using the RRT or RPM methods. Planning a local path within a particular region is carried out using a simple straight-line planning algorithm. In this case, the uncertainty of the robot model is taken into account by introducing an additive disturbing input such as white noise. The uncertainty of the map is taken into account by the probability that the cell is occupied by an obstacle. The main result of the method is a path that guarantees the successful achievement of the target point with a probability not less than a given p_{min}. The work [22] presents the results of analyzing and modeling the motion of the robot that is represented by the kinematic equations of a material point in a static environment. The main disadvantage of the method is a significant increase in computational costs for large values of the probability p_{min}. The problem of reducing computational costs and memory occupied is considered in [26, 27], which combines the RRT and potential field (P-RRT*) methods. In this method, a random tree is built in the direction of decreasing the potential field value, which makes it possible to increase the rate of convergence to the optimal trajectory. In [27], an improved version of the algorithm RRT* [28] is used, in which the initial path is calculated similarly to the basic RRT algorithm. Then the number of nodes in the configuration space is increased to optimize the initial path. In [28], the analysis of the convergence of the proposed algorithm is carried out and the results of numerical simulation are presented. The disadvantages of the method presented in [27, 28] are the non-smoothness of the trajectories and the still high computational complexity.

Note that the idea of using potential fields to improve the convergence and computational efficiency is not limited only to the RRT algorithms family, but it is also used with the PRM algorithms. Thus, in [23], a hybrid method is used that includes the PRM algorithm and artificial potential fields. The node distribution uses a segmented map to create areas of low and high potentials. To eliminate the problem of local minima, a reactive approach to planning local paths in the presence of static and dynamic obstacles is used. The proposed algorithms have been tested using numerical and field experiments.

In [26], the P-RRT* method is developed for the case of bidirectional search. In this method, nodes are generated from both the starting and the target points. Both trees create attractive fields for each other, which improves the convergence of the P-RRT* method, especially for finding narrow passages. The work [29] also investigates the problem of increasing the rate of convergence of the path planning process using the bidirectional search RRT algorithm. The outcomes of [26, 29] indicate that the bidirectional search in a combination with potential fields makes it possible to reduce the number of iterations for finding the shortest path. Also, such algorithms allow moving in narrow corridors without hitting local minima. To reduce the memory used, the calculation of two random trees is carried out sequentially in time, which can increase the calculation time.

The paper [30] also uses a two-step planning procedure for a dynamic environment. The global planner plans the original first approximation path using the RRT method. Next, a dynamic planner is used, which changes the nodes by using repulsive forces from obstacles and inertial forces. To account for moving obstacles, this method uses a forecasting procedure. The effectiveness of this method is confirmed by the results of numerical experiments. In this work, a heuristic optimization function is used, the choice of which depends on various requirements and constraints. The correct formation of such function is the central problem of the presented method.

In [31], the problem of constructing an optimal path with a visit to the set of given target points is considered [32]. To solve this problem in an environment with obstacles, a forest of trees is built [33], where the trees start from the given target points. A forest of trees is built before they are connected to each other. This work is a development of [34], in which the construction of a forest of trees is optimized due to the priorities of the queue of nodes. The proposed method is studied in a two-dimensional and three-dimensional environment using numerical simulation methods. The reduction in computational costs and memory requirements is due to the fact that each tree is built separately from each other. In this regard, the nodes of individual trees are stored in separate arrays.

In [35], a modification of the RRT algorithm is considered, taking into account the limitations imposed by the vision system and the speed limitations. An algorithm based on computer vision that functions under conditions of an unknown image depth is proposed. A model of a robot in the form of kinematics of a material point is considered. The main contribution of the method proposed in [31] is the presence of a virtual target, which is located so that the robot is on a line that provides its specified orientation angle at the target point. The advantages of the proposed algorithm are confirmed by the results of the experiments. This article shows the efficient operation of the RRT algorithm in real conditions. However, the experiments were carried out in a simple environment.

1.2. Contribution of the article. The performed analysis allows us to conclude that currently the typical issues of path planning using the RRT algorithm are:

1) Consideration of restrictions in terms of the formation of feasible paths that take into account the dynamic properties of robots.

2) Optimization of computational complexity and convergence of algorithms to optimal trajectories.

3) Consideration of the uncertainty associated with the lack of a complete map of the environment and optimization of the processes of replanning paths in dynamic environments.

The purpose of this article is to develop planning algorithms that reduce the length of the planned trajectory, reduce its calculation time and take into account the restrictions imposed by the robots' properties. We propose a two-stage procedure, during which the MPN-RRT method is used to find the first approximation path which is optimized using the developed iteration algorithm. To take into account the dynamic restrictions on the turns of a mobile robot, it is proposed to add virtual obstacles to the map, which block unfeasible sectors of the movement. This makes it possible to take into account these limitations without changing the developed MPN-RRT algorithm.

In summary, the contributions of this paper are listed as follows.

1) In this work, virtual obstacles are introduced on the map, located in such a way that they overlap the unfeasible sectors of the mobile robot's movement. This makes it possible to take into account the restrictions on maneuvers that depend on the linear speed of the robot and its maximum permissible angular accelerations and velocity. At the same time, the virtual obstacle method does not change the planning algorithm and allows the use of any map-oriented planning algorithm.

2) This article proposes the use of several parent nodes when building a random tree using the RRT algorithm. On the one hand, this complicates the structure of the constructed graph. On the other hand, it gives more possibilities to optimize the resulting motion trajectory. As will be shown below, the introduction of additional parent nodes makes it possible to reduce the length of the planned path with increasing computational complexity.

3) The proposed method includes global and local planning stages. The use of a two-stage procedure makes it possible to reduce the computational costs for finding an optimal solution compared to a one-stage global search. A proposed MPN-RRT algorithm with two-parent nodes is used as a global planner. An optimization algorithm is used as a local planner, taking into account the path length and deviation from the points planned by the global planner.

4) An algorithm of local optimization and path smoothing described by segments connecting the points of the first approximation path is proposed. A quality criterion is proposed, which is a convolution of two indicators, which are the path length and the deviation from the global path points. This procedure allows optimizing the path’s length while limiting the deviation from the original trajectory of the first approximation. This method is based on the outcomes of [36, 37]. Its difference lies in the application of a new optimization functional. The function that was used in these papers included the probability of successful passage of a section of the trajectory and the deviation from the trajectory of the first approximation.

2. Problem statement. Let us consider a rectangular stationary two-dimensional environment of size $L_x \times L_y$, shown in Figure 1, which simulates a low-density urban development.

The lower-left corner of the map coincides with the origin of the fixed coordinate system OX_gY_g. The length of the environment along the OX_g axis is L_x, and along the OY_g axis is L_y. The position of the robot in the environment is determined by the coordinate vector (x, y) and the orientation angle φ. The obstacles are the rectangles which are r_{pi} wide and b_{pi} high. The position of an obstacle is determined by the coordinates of its lower-left corner (x_{pi}, y_{pi}). The target position is specified by a vector of coordinates (x_t, y_t).

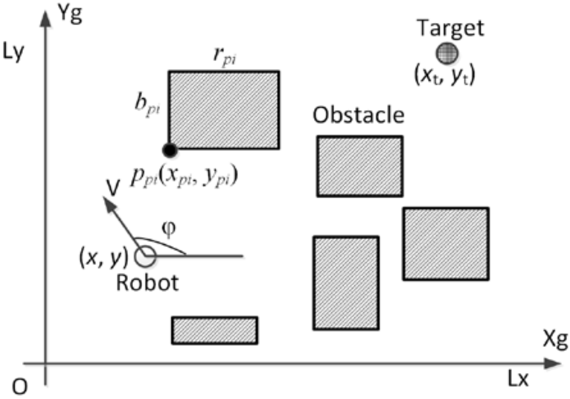

Fig. 1. 2-D obstructed environment

The mobile robot is described by the following equations [16, 38]:

$$\dot{x} = V\cos\varphi, \quad \dot{y} = V\sin\varphi, \quad \dot{\varphi} = \omega. \tag{1}$$

$$\dot{V} = -\alpha_v(V - V^*), \quad \dot{\omega} = -\alpha_\omega(\omega - \omega^*), \tag{2}$$

where x, y are the robot's coordinates in the fixed frame OX_gY_g; V is the robot's linear velocity; ω is the robot's angular velocity; φ is the robot's yaw angle; α_v, α_φ are constants; V*, φ* are references for V and φ, respectively.

The mobile robot's design features, inertial properties, and requirements for stability and controllability do not allow it to move to an arbitrary point in space [39]. In this regard, the following restriction is introduced on the maximum turning angle depending on:

$$|\varphi^* - \varphi|_{max} = \Delta\varphi_{min} + \left(\pi - \Delta\varphi_{min}\right)e^{-\frac{V - V_{min}}{\alpha(V_{max} - V_{min})}}, \tag{3}$$

where $\Delta\varphi_{min}$ is the minimum rotation angle, independent of the robot's velocity, determined by the rolling resistance; V_{min}, V_{max} are the minimum and maximum values of the robot's velocity, respectively; α is the coefficient depending on critical velocity, robot length, friction, and rolling resistance. The function (3) is shown in Figure 2.

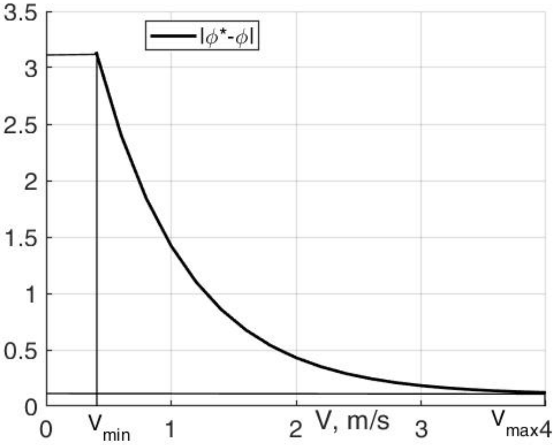

Fig. 2. Restriction on the maximum angle of rotation of the robot

The reference influence on the velocity V* changes in accordance with the expression:

$$\begin{gathered} V^* = V_{max}\frac{r}{r_s}k_\varphi k_{obs}, \quad \forall\, V^* \leq V_{max}, \\ V^* = V_{max}, \quad \forall\, V^* > V_{max}, \end{gathered} \tag{4}$$

where r is the straight distance to the target; r_s is the distance that determines the moment when the robot starts braking; k_φ is the function of velocity reduction when turning around; k_{obs} is the function of decreasing velocity when approaching obstacles.

The function k_φ is shown in Figure 3, according to which the robot's velocity decreases linearly with an increase in the difference between the current φ and the desired φ* angle of the movement of the robot.

The k_{obs} function is shown in Figure 4, according to which, when the robot approaches an obstacle at distance smaller than r_s, its velocity begins to decrease in proportion to the specified distance:

$$
\begin{aligned}
k_{obs} &= \frac{r_{obs}}{r_s}, \quad \forall r_{obs} < r_{s,} \\
k_{obs} &= 1, \qquad \forall r_{obs} \geq r_{s.}
\end{aligned} \tag{5}
$$

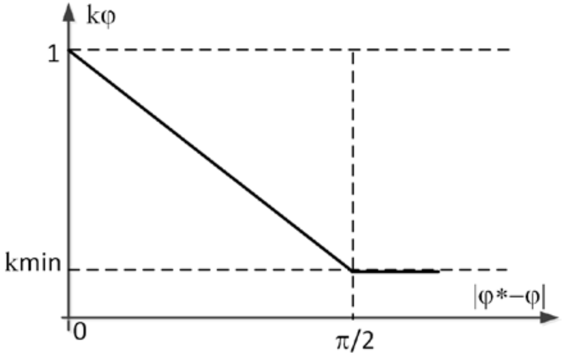

Fig. 3. Graph of the function that determines the decrease in the velocity of the robot when turning

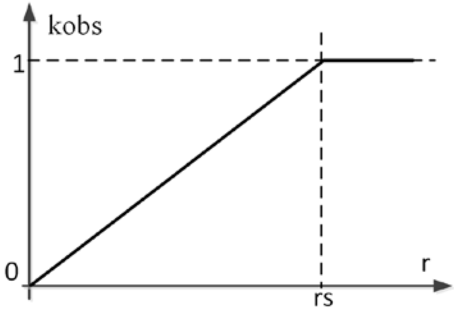

Fig. 4. Graph of the function that determines the decrease in the speed of the robot when approaching obstacles

It is required to develop an algorithm for planning the movement of a mobile robot from the current point (x, y) to a given target point (x_t, y_t), taking into account the restrictions on the rotation angle (3) and ensuring the absence of collisions. There are no requirements for the orientation of the mobile robot at the target point. The velocity of the robot at the target point should be zero, but at intermediate points of the trajectory, the velocity is determined by the expression (4). The planned trajectory must be smooth, i.e. have no breaks. The quality criterion of the planned trajectory is its length, i.e.:

$$D_{path} = \sum_{i=1}^{N} \sqrt{(x_i - x_{i-1})^2 + (y_i - y_{i-1})^2} \rightarrow min.$$

3. Planning system algorithms. This section considers the planning procedure, which includes the following:

1) Introduction of a virtual obstacle into the map to take into account the constraints (3).

2) Construction of a random tree for planning the first approximation path using the modified RRT method.

3) Finding the shortest first approximation path using the ant colony algorithm.

4) Local optimization and smoothing of the first approximation path.

3.1. Adding virtual obstacles to the map. As follows from the expression (3), if the velocity is $V = V_{min}$, then the robot can turn at any angle from –π to +π. As the robot's velocity increases, the sector available for its maneuver begins to decrease in accordance with (3). To consider these limits, a virtual obstacle is added to the environment map as shown in Figure 5.

Figure 5 shows the evolution of a virtual obstacle in the case when $V_3 > V_2 > V_1$. The virtual obstacle is V-shaped. At low speeds, this obstacle is located at a convex angle to the robot. As the velocity increases, this angle decreases to unfold and becomes sharp. The presence of a virtual obstacle makes it possible to increase the safety of the robot's movement without using additional algorithms to avoid collisions. As shown in Figure 6, if the robot moves in a narrow passage between two obstacles and receives a target point located behind it, then a collision may occur if the robot’s velocity is restricted by inequalities 0<Vmin≤V≤Vmax.

The advantage of virtual obstacles for eliminating such collisions is that they do not increase the computational complexity of the used planning algorithm. The procedure for introducing a virtual obstacle is based on the expression (3), which does not present additional computational complexity. At the same time, it should be noted that the parameters $\Delta\varphi_{min}$ and α, which determine the turning radius in the expression (3), depend on the type of the robot and the conditions of its movement. In this regard, these parameters must be set, and the robot must evaluate the movement resistance coefficients.

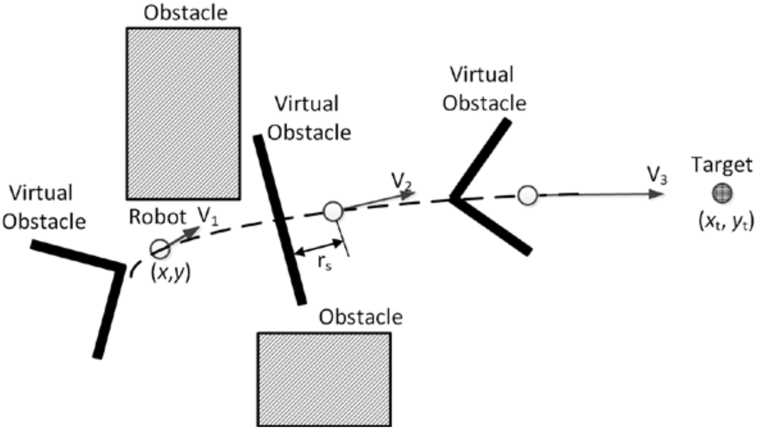

Fig. 5. Graph of the function that determines the decrease in the robot’s velocity when approaching obstacles

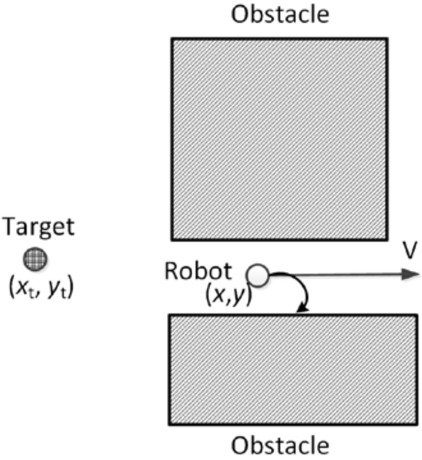

Fig. 6. An example of a collision due to the inertial properties of a mobile robot

3.2. Two-parent nodes RRT algorithm. The pseudocode of Algorithm 1 for constructing a graph using the RRT method is distinguished by the search for two-parent nodes for each new node.

Algorithm 1 for constructing a graph for planning the trajectory of a mobile robot:

Input: Robot's position (x, y); environment map Map; target point (x_t, y_t); robot's velocity V; yaw angle φ; parameter L_s; trajectory optimization parameter N_{add}.

Output: List of nodes Tree; Relationship matrix Matrix_tr.

```
[Tree, Matrix_tr] ← Init_Tree([x,y], [xt,yt]);
Map←Save_Vir_Obs(V, φ);
FL_0←Str_Line(Tree(last),[xt, yt], Map, Ls);
IF FL_0=1
  WHILE FL_1<Nadd
    X_rand←Rand_Node(Map, Tree, Ls)
    X_nod1←Parent_Node(Tree, Map,rs, []);
    X_nod2←Parent_Node(Tree, Map, rs, X_nod1);
    Tree←Expand_Tree(X_rand);
    Matrix_tr← Expand_Matr(X_nod1, X_rand);
    Matrix_tr← Expand_Matr(X_nod2, X_rand);
    FL_2←Str_Line(X_rand, [xt,yt], Map, Ls);
      IF FL_2=1
        FL_1= FL_1+1;
        Matrix_tr← Expand_Matr(X_rand,[ xt,yt]);
      END IF
  END WHILE
ELSE
  Matrix_tr← Expand_Matr([x,y],[xt,yt]);
END IF
Path_opt=Aco(Matrix_tr, Tree);
```

The Init_Tree () function initializes the tree and the matrix of paths' weights Matrix_tr. During the initialization phase, the starting point of the path (x, y) and the target point (x_t, y_t) are added to the list of nodes. The first tree node in the list is always the starting position of the robot (x, y). The last node in the tree list is always the target point (x_t, y_t). The path matrix Matrix_tr is initialized as a zero 2×2 matrix.

The Save_Vir_Obs function adds a virtual obstacle to the map based on the current velocity and the robot's movement direction. In this case, the environment map is a matrix of dimension $N_x \times N_y$, where the number of matrix cells is determined by the expressions:

$$Nx = \lceil Lx/Sx \rceil, \qquad Ny = \lceil Ly/Sy \rceil, \tag{6}$$

where S_x, S_y are the dimensions of the space cell along the axes OX_g and OY_g, respectively; $\lceil * \rceil$ is the operation of rounding to the nearest integer from the top.

The value of a cell of the Map matrix equals 0 if it is free, 1 if it contains the current position of the robot, 2 if it contains the target point, and 3 if it contains an obstacle. Obstacles are recorded in the Map in accordance with Figure 7. According to this figure, if the cell with coordinates (x_c, y_c) is occupied by an obstacle, then the number 3 is written into the cell of the matrix Map (y_c + 1, x_c + 1). Thus, the x_c coordinates are the column numbers of the Map matrix, and the y_c coordinates are the row numbers. Adding 1 to the coordinates is due to the fact that the rows and columns are numbered starting from one, and the coordinates of the cells are numbered from zero.

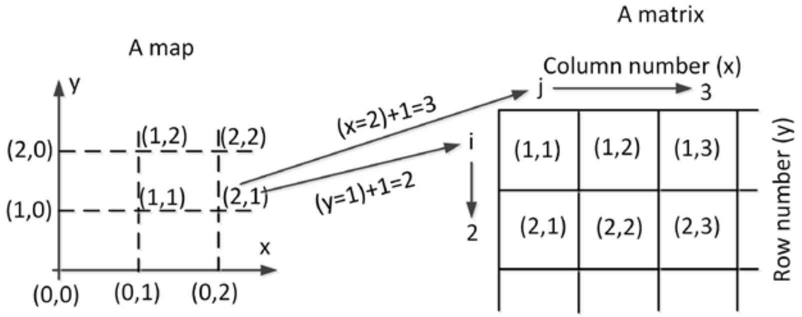

Fig. 7. Recording the environment map to the Map matrix

Line 3 of Algorithm 1 checks for the presence of a straight line connecting points (x, y) and (x_t, y_t) without collisions. Such a check is carried out using the Str_Line () function. It returns the FL_0 flag, which is equal to 1 in the presence of a collision, which is determined when the robot's trajectory approaches any obstacle closer than the distance L_s.

The main tree building loop begins with the generation of a random node X_rand for which the Rand_Node () function is used. This function generates a node so that it is located at a distance of at least L_s from the obstacles and the other nodes of the tree. Next, the search for parent nodes is carried out using the Parent_Node () function. The presence of two-parent nodes is a key difference between the used algorithm and the traditional version of RRT. The main planned effect of this introduction of two-parent nodes is a decrease in the length of the planned path, which occurs at cost of increasing the tree construction time. This effect is due to the fact that the presence of two-parent nodes increases the number of links in the graph under construction, which makes it possible to optimize the path length more effectively.

If the parent nodes are found successfully, then a new node X_rand is added to the tree and the corresponding links are added to the Matrix_tr matrix using the functions Expand_Tree () and Expand_Matr ().

Next, using the function, Str_Line () is checked for a direct collision-free path from the new X_rand node to the target point (x_t, y_t). If such a path exists, then the FL_1 counter is incremented.

The search cycle continues until it finds N_{add} nodes that are added to the tree and gives a new path to the target point.

The search for a path on the constructed graph is carried out using the ant colony optimization algorithm [40]. The choice of the evolutionary algorithm is due to the need of optimizing the search time. The search for the shortest path (line 21) using the ant colony optimization algorithm is implemented by Aco () function that takes Tree and the matrix of coupling Matrix_tr as an input. The result of the operation of the Aco () function is a list of points of the found path Path_opt. This list defines a piecewise broken path that needs to be smoothed. It should also be noted that the RRT algorithm converges rather slowly, so it is necessary to further optimize its length.

3.3. Path optimization and smoothing. Path_opt is the first approximation path. The task is to reduce the length of the curve Path_opt, assuming that the optimized curve is found in a certain bounded neighborhood, or in a certain corridor relative to the original trajectory. To solve this problem, the following functional can be generated:

$$G(Tr) = \delta_1 \sum_{i=2}^{N-1} \left[\left(x_i - x_i^{\prime}\right)^2 + \left(y_i - y_i^{\prime}\right)^2 \right] \\ + \delta_2 \sum_{i=2}^{N-1} [(x_{i+1} - x_i)^2 + (y_{i+1} - y_i)^2], \tag{7}$$

where the first term is the sum of the squares of the deviations of the points (x_i, y_i) of the constructed trajectory from the points of the original trajectory (x'_i, y'_i); the second term is the trajectory length T_r; the coefficients $0 \le \delta_1, \delta_2 \le 1$ are the weights of each term.

The stationary point of the functional (6) with respect to the variables {x_i, y_i}, i = 2, 3, ..., N-1 is found from the system of equations:

$$\begin{cases} \partial G(Tr)/\partial x_i = 0, \\ \partial \dfrac{G(Tr)}{\partial} y = 0, \qquad i = 2, \dots, N-2 \end{cases} \tag{8}$$

System (8) is reduced to two independent systems of linear equations with respect to vectors $X = [x_2 \dots x_{N-1}]^{T}$ and $Y = [y_2 \dots y_{N-1}]^{T}$. The solution of these two systems has the form:

$$\begin{gathered} X_{zt} = [x_{st2} \dots x_{stN-1}]^T = A^{-1}B_X, \\ Y_{zt} = [y_{st2} \dots y_{stN-1}]^T = A^{-1}B_Y, \end{gathered} \tag{9}$$

$$B_x = [\delta_1 x_2' + \delta_2 x_1', \delta_1 x_3', \dots, \delta_1 x_{N-2}', \delta_1 x_{N-1}' + \delta_2 x_N']^T, \tag{10}$$

$$B_y = [\delta_1 y_2' + \delta_2 y_1', \delta_1 y_3', \dots, \delta_1 y_{N-2}', \delta_1 y_{N-1}' + \delta_2 y_N']^T, \tag{11}$$

$$A = \begin{bmatrix} \delta_1+2\delta_2 & -\delta_2 & 0 & 0 & \dots & 0 & 0 & 0 \\ -\delta_2 & \delta_1+2\delta_2 & -\delta_2 & 0 & \dots & . & 0 & 0 \\ 0 & -\delta_2 & \delta_1+2\delta_2 & -\delta_2 & \dots & 0 & . & 0 \\ 0 & 0 & -\delta_2 & \dots & \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots & . & . & 0 & 0 \\ 0 & . & 0 & \dots & . & \delta_1+2\delta_2 & -\delta_2 & 0 \\ 0 & 0 & . & \dots & 0 & -\delta_2 & \delta_1+2\delta_2 & -\delta_2 \\ 0 & 0 & 0 & \dots & 0 & 0 & -\delta_2 & \delta_1+2\delta_2 \end{bmatrix}. \tag{12}$$

Let us show that at the found stationary point (9) the sufficient condition for a local maximum with respect to minors of the Hessian matrix is satisfied for any values of δ_1, δ_2 such that, $0 \le \delta_1, \delta_2 \le 1$, $\delta_G + \delta_L = 1$. Indeed, it is easy to observe that the Hessian matrix for functional (6) is the block-diagonal matrix:

$$Gess(G) = \begin{bmatrix} A & 0_{N-2\times N-2} \\ 0_{N-2\times N-2} & A \end{bmatrix}, \tag{13}$$

where $0_{N-2\times N-2}$ is a zero matrix of dimension N-2×N-2.

It is known that if each of the blocks of such a matrix is a positive definite matrix, then the resulting block diagonal matrix will also be positive definite [41]. Also, according to the corollary to the Gershgorin theorem [41], a sufficient condition for any symmetric matrix with positive entries on the main diagonal to be positive definite is the condition of strict dominance of diagonal entries. In this case of symmetric tridiagonal Toeplitz matrix of type (9), this condition takes the form:

$$|\delta_1 + 2\delta_2| > 2|\delta_2|, \tag{14}$$

Since $0 \le \delta_1,\ \delta_2 \le 1$, then the expression (14) takes the form:

$$\delta_1 + 2\delta_2 > 2\delta_2 \Rightarrow \delta_1 > 0. \tag{15}$$

Thus, an additional constraint is a requirement that the coefficient δ_1 does not vanish.

Note that the global minimum of functional (6) is simultaneously attained at point (8) due to the uniqueness of the stationary point for this functional.

For practically important cases of minimization of the resulting curve, two conditions can be used. The first condition is the natural requirement of not colliding with obstacles. The second is not exceeding a certain maximum deviation from the original curve, expressed, for example, in the form of the limiting root-mean-square discrepancy σ_{kr} between the corresponding nodal points. This condition is written as:

$$\sqrt{\sum_{i=2}^{N-1}[(x_i - x_i^{'})^2 + (y_i - y_i^{'})^2]\,/(N-2)} \le \sigma_{kr}. \tag{16}$$

The following procedure is used to find the offset path of the minimum length, taking into account the first and second constraints. First, the values of the coefficient δ_1 are sorted out in the range [$\delta_{1\min}$, $\delta_{1\max}$] with a step $\Delta\delta_1 << \delta_{1\max}$. The value $\delta_1=\delta_{1\min}$ corresponds to the case when the optimized trajectory is the closest in terms of the standard deviation to the straight line connecting the starting and target points. If there are no collisions and the deviation from the initial trajectory does not exceed a given one, then the optimal trajectory for $\delta_1=\delta_{1\min}$ is the best. If at least one of these conditions is not met, it is necessary to move to the next larger value of the coefficient δ_1 and re-check the specified conditions.

This procedure is repeated iteratively until there is such a value δ_1 from the corresponding finite sampling sequence of the interval $[\delta_{1\min}, \delta_{1\max}]$, for which both of the above constraints are satisfied, or until the search is completed. To exclude the last case, it is necessary to choose the value of the step $\Delta\delta_1$ in accordance with the value of the upper admissible boundary σ_{kr} of the limiting root-mean-square discrepancy between the corresponding nodal points, and the less is the $\Delta\delta_1$ the less should be the step.

After finding the optimized curve, it has to be smoothed. For this, the method of approximation by straight-line segments, smoothly conjugate to arcs of the circles, is applied. Moreover, these arcs are constructed for each of the nodal points of the resulting curve.

The resulting trajectory in the form of a list of points that the robot must pass is fed to the motion control function, which calculates the desired direction of motion and setpoints for the linear and angular velocities of the robot. The motion control system is implemented in the form of an adaptive multi-loop system presented in [38].

4. Numerical modeling results. In this section, a case study of the developed algorithms is carried out. The simulation was carried out in MATLAB. During the simulation, the following parameters of the mobile robot were set: $a_v = 1$; $a_w = 5$; $V_{max} = 4$ m/s; $w_{max} = 2$ rad/s. The safe distance at which the robot starts braking is $r_s = 6$. The minimum distance to the obstacles is $L_s = 1$ m. The dimensions of the environment are $L_x = 100$ m, $L_y = 100$ m. The maximum number of obstacles is 24. The maximum width and length of each obstacle is 11 m. An urban environment is simulated with a maximum building rate of 30 %.

4.1. Investigation of the influence of the number of parent nodes on the trajectory length and the time of the trajectory calculation. The study of the influence of the number of nodes on the trajectory length and the time for calculating the trajectory was carried out for various values of the parameter N_{add} = 20, 40, 80. For each number of nodes and the parameter N_{add}, 1000 tests were carried out, the results of which are summarized in Table 1 and presented in Figure 8 and Figure 9. The confidence probability is accepted as 95%. The confidence interval for the path length is 0.026, and the confidence interval for the path calculation time is 0.011.

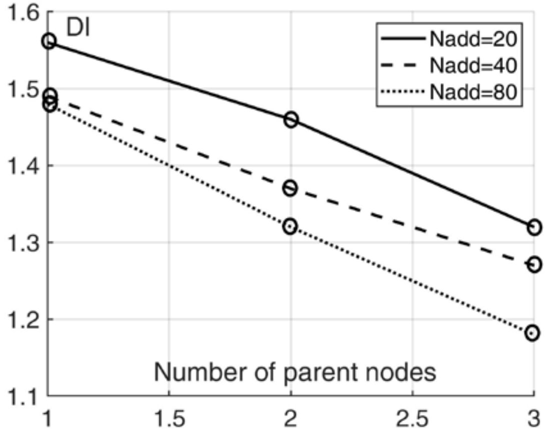

Fig. 8. Dependence of the average relative path length on the number of parent nodes

The N_{add} parameter determines the degree of optimality of the first approximation path. The larger this parameter, the higher the probability of the first approximation path approaching the optimal trajectory.

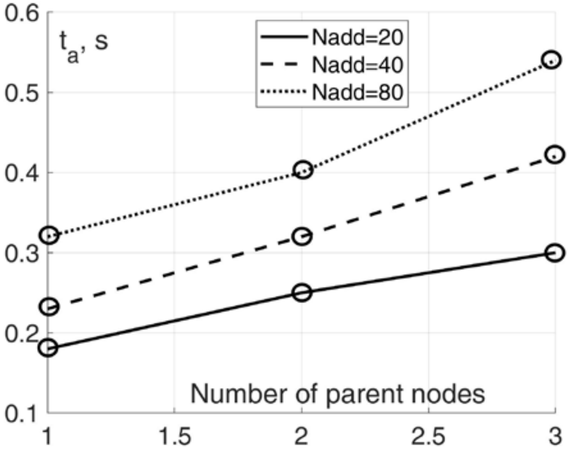

Fig. 9. Dependence of the average time for calculating the path on the number of parent nodes

Table 1. Influence of the number of parent nodes on the length of the path and the time of calculation of the trajectory

Number of nodes	Parameter N_{add}	Average relative path length	Average calculation time
1 node	80	1.48	0.32
	40	1.49	0.23
	20	1.56	0.18
2 nodes	80	1.32	0.4
	40	1.37	0.32
	20	1.46	0.25
3 nodes	80	1.18	0.54
	40	1.27	0.42
	20	1.32	0.3

The average relative path length is calculated according to the expression:

$$Dl = \frac{1}{1000}\sum_{i=1}^{1000}(Dl_i/r_{rti}), \tag{17}$$

where Dl_i is the length of the l-th path in the i-th experiment; r_{rti} is the distance from the starting point of the robot's position to the target point in the i-th experiment.

The calculation time was measured using the tic and toc MATLAB functions. The table shows the average calculation time calculated in accordance with the expression:

$$t_a = \frac{1}{1000}\sum_{i=1}^{1000} t_i\,, \tag{18}$$

where t_i is the time of calculating the path in the i-th experiment.

As can be seen from Figure 8 and Table 1, the path length decreases with an increase in the number of parent nodes in the RRT algorithm. This is explained by the fact that with an increase in the number of parent nodes, the number of links in the graph under construction increases. This entails increasing the different path options from point (x, y) to the target point (x_t, y_t).

It follows from the simulation results that the introduction of an additional parent node results in 6.4 to 10.8% reduction of the planned path length, depending on the N_{add} parameter. The introduction of two additional nodes makes it possible to reduce the average path length from 14.8 to 20.3%, depending on the N_{add} parameter.

Reducing the length of the planned path increases the computational cost of the algorithm. Thus, Figure 9 shows the dependence of the average time for calculating the trajectory from a different number of nodes and different values of the parameter N_{add}. From the graphs shown in Figure 9 and Table 1, it can be seen that the increase in calculation time is from 28 to 40% for one additional parent node and from 40 to 45% for two additional parent nodes.

At the same time, we note that an increase in calculation time is associated with both the procedure for constructing a random graph and further searching for a path. When using one parent node, the path from a potential new node only to the nearest graph node is checked. When using additional parent nodes, the path to several nearest nodes of the graph is checked, which leads to additional computational costs.

At the same time, it should be noted that using the ant colony algorithm for finding the shortest path allows parallelizing the computational process [42], which reduces computational costs.

4.2. Study of the optimization and smoothing algorithm. The used method of local optimization and trajectory smoothing is based on the optimization of functional (6). In this case, the main issue in the formation of this functional is the rational choice of the admissible average deviation σ_{kr} of the optimal trajectory from the first approximation path. This conclusion follows from the fact that when the condition $\delta_1+\delta_2=1$ is satisfied, there is only one independent weighting coefficient. The coefficient δ_1 was chosen as such a coefficient, which is sought using the procedure described in Section 3. Thus, the weighting coefficients of functional (6) are chosen automatically; therefore, it is necessary to choose only the parameter σ_{kr}. The influence of this parameter on the trajectory length is shown in Figure 10 and Figure 11.

Figure 10 shows the result of modeling the procedure for finding a displaced trajectory of the minimized length, taking into account the limiting conditions for the following input data: σ_{kr} = 10 m, the length of the original curve is 148.5 m. As a result of applying the procedure of local optimization and smoothing described in Section 3, a trajectory with a length of 71.63 m is obtained. The parameter change interval is δ_1 [0.02 0.98], and step is $\Delta\delta_1 = 0,02$. The optimal value of the parameter is δ_1 = 0.1. In this case, the standard deviation at the nodes of the original and displaced curves is 9.85 m, which is less than the maximum allowable value. Similar simulation results for σ_{kr} = 5 m are shown in Figure 11, taking into account the limiting conditions and for the same initial trajectory of the first approximation as in the previous case. The interval for changing the parameter δ_1 and the step are the same. As a result, the length of the optimized curve is 90.8 m with the optimal value $\delta_1 = 0.6$.

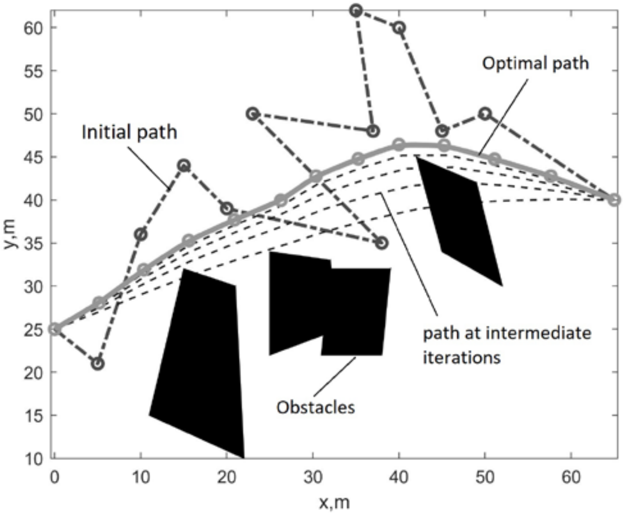

Fig. 10. First approximation path and optimal trajectory for σkr=10 m

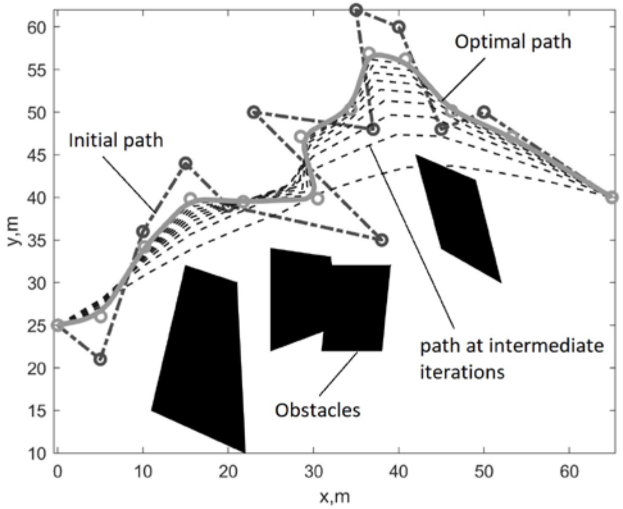

Fig. 11. First approximation path and optimal trajectory for σkr=5 m

Thus, in the first case with σ_{kr} = 10 m, the length of the first approximation path is reduced by approximately 51.8%. In the second case, with σ_{kr} = 5 m, the length of the first approximation path is reduced by about 38.8%.

Figure 10 and Figure 11 show the first approximation path obtained using the RRT algorithm and the ant colony algorithm, the trajectories of intermediate iterations, and the optimal trajectory of the movement. The nodes of the optimized trajectory according to the considered algorithm are shown separately.

Based on the simulation, it can be concluded that with the tightening of restrictions on deviations from an initial curve, i.e. as the value of σ_{kr} decreases, the length of the optimized curve increases. Thus, solving the problem of optimizing the original trajectory, taking into account the restrictions on avoiding collisions and not exceeding the maximum permissible standard deviation from the original curve, requires prioritization when answering the question of whether it is more important to significantly reduce the length of the curve, or to prevent deviation from the original curve by more than a given value σ_{kr}.

4.3. Study of a trajectory planning and motion control system. The results of the simulation of the mobile robot control system are shown in Figure 12 - 14. Figure 12 shows the obstacles, the virtual obstacle at the initial moment of time, the initial position of the robot with the symbol "*", the target position of the robot with the symbol "+", the path obtained using the RRT algorithm with two-parent nodes, and the smoothed path obtained as a result of smoothing the initial trajectories. To plan the path in Figure 12, the RRT algorithm generated a graph that includes 341 nodes. As a result of the search for the shortest path using the ant colony algorithm, a path was found containing the following 8 nodes, including the starting and target nodes:

$$Tr = \begin{bmatrix} 18 & 26 & 33 & 41 & 50 & 50 & 48 & 93 \\ 84 & 90 & 96 & 91 & 98 & 94 & 93 & 52 \end{bmatrix}. \qquad (19)$$

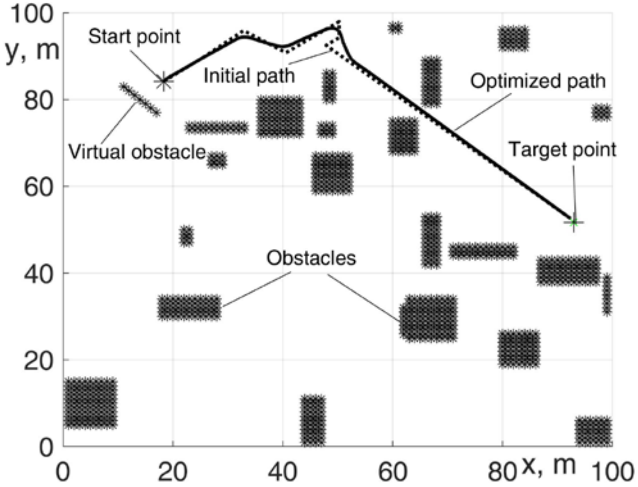

Fig. 12. The trajectory of a mobile robot in an environment with obstacles

The path length obtained as a result of the application of the RRT algorithm is approximately 107 m. The length of the optimized trajectory is 99.9 m. Thus, the reduction in the path is about 6.6%. The results of simulation on a sample of 1000 random situations showed that the smoothing and optimization procedure gives an effect of approximately 6 - 8% on the path's length. In this case, the greater the number of intermediate points between the starting and target points, the more decrease in the length of the path. For the first approximation paths consisting of two and three points, the use of the local optimization procedure does not make sense.

Figure 13 shows the change in the velocity of the mobile robot in time and the reference velocity. In the time interval from 0 to 5 s, the robot accelerates to V_{max}. Then, in the interval from 5 to 13 seconds, the motion control system reduces the velocity four times. The first three velocity maneuvers are due to turns, and the last speed reduction is due to passing close to an obstacle. When approaching the target point, the robot also slows down.

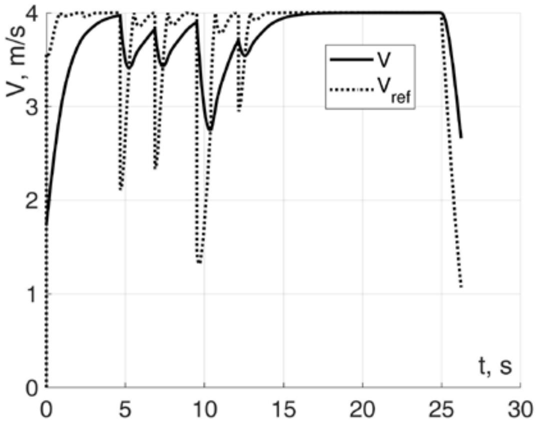

Fig. 13. The velocity of a mobile robot in an environment with obstacles

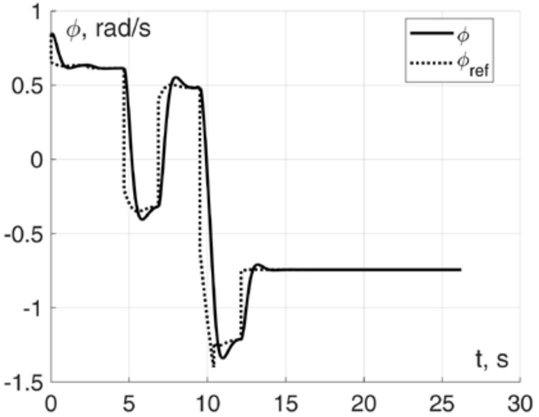

Fig. 14. Yaw angle of a mobile robot in an environment with obstacles

Figure 14 shows the change in the yaw angle of the robot when moving to the target point. It can be seen from this graph that the robot makes turns at the initial moment, and then three turns when moving between the nodes of the planned path.

Figure 15 shows the simulation results demonstrating the movement of the robot in an environment with obstacles, including the movement in a narrow corridor formed by the obstacle and the boundary of the area.

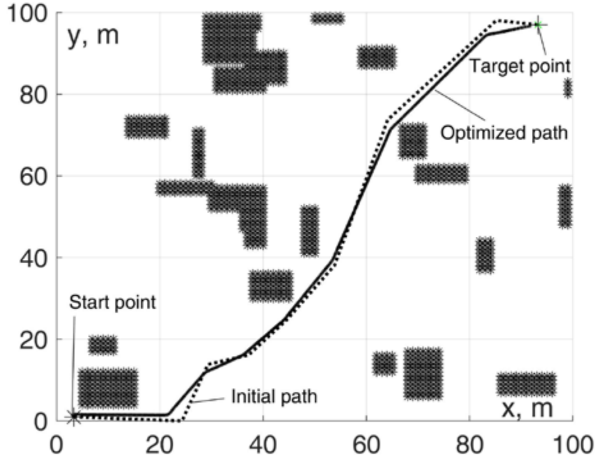

Fig. 15. The robot's trajectory in the presence of a narrow corridor

Studies have shown that the presence of narrow corridors can significantly increase the planning time of the robot's trajectory. In this regard, the RRT algorithm uses a variant of generating a new potential node in a certain limited area relative to the last node added to the graph. This way of building a tree gives an additional effect in terms of reducing the length of the planned trajectory. So, for the MPN-RRT algorithm with two-parent nodes, an additional 1000 experiments were carried out with the generation of a new potential node within a radius of 20 meters from the last node added to the tree. An additional reduction in the trajectory length is obtained, ranging from 5 to 10%, depending on the value of the parameter N_{add}. However, in the MPN-RRT algorithm, there is a significant increase in the trajectory search time, which is on average 30% compared to the variant of random generation of a new node over the entire area of operation. This increase in the computation time is observed only on average. For situations with narrow corridors, the algorithm with a local area for generating a new potential node does not lead to an increase in the computation time. In this regard, it is possible to recommend the use of a unit for evaluating the functioning environment in the robot control system. Such a block can be built using intelligent technologies. Thus, [43] presents a neural network that estimates the geometric complexity of the environment. This estimate is used to determine the clearest areas for planning a route for a group of robots.

Note that generating a new potential node in a certain bounded area relative to the constructed tree leads to trajectories consisting of a larger number of points. However, the application of the above-described iterative local optimization and smoothing procedure allows obtaining a smooth trajectory with a minimized length. In this case, the effect of reducing the path length becomes higher and amounts to approximately 10% for the simulated movement conditions. A more significant decrease in the trajectory length is due to an increase in the optimization efficiency with an increase in the number of nodal points of the first approximation trajectory. Figure 16 shows the results of one of the numerical experiments in which a new potential node is generated within a radius of 20 meters relative to the last added node.

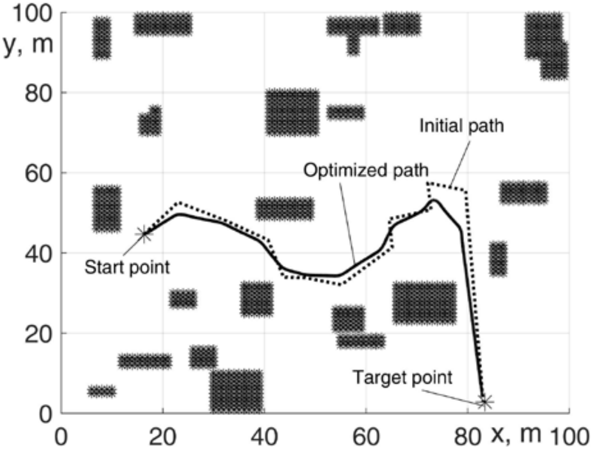

Fig. 16. The robot’s trajectory when a new potential node is generated within a radius of 20 meters relative to the last added node

7. Discussion and conclusion. Some discussions about the highlights of the approach proposed in this article are summarized as follows.

1) Taking into account the dynamic and design constraints of mobile robots when planning a path usually leads to the complication of planning algorithms. In this article, it is proposed to take into account the restrictions on the angle of rotation of the robot by introducing virtual obstacles. Such obstacles are located in such a way that they cover sectors into which the mobile robot cannot turn at a given time. As a result, the planned trajectory is physically realizable. Thus, compared to the existing

methods of considering dynamic constraints, the proposed approach does not affect the planning method if it uses a map to calculate the trajectory.

2) The potential of using probabilistic methods for path planning is limited by a large number of iterations to obtain the optimal solution. This article proposes an improved MPN-RRT algorithm that uses multiple parent nodes. This algorithm allows creating a graph with a large number of connections, resulting in an increase in the optimality of the planned path.

3) To reduce the number of iterations in the path search, a two-stage path planning method is proposed. At the first stage, the first approximation path is searched using the proposed MPN-RRT algorithm. At the second stage, the resulting trajectory is optimized and smoothed by an iterative local search algorithm. Compared to the one-stage MPN-RRT algorithm, the two-stage method, which includes MPN-RRT and an iterative optimization and smoothing procedure, reduces the time it takes to find a trajectory of the same length by 7–13%. This effect is due to the slow convergence of algorithms based on the probabilistic method.

4) The numerical experiments performed have shown that for a simulated urban environment, the MPN-RRT algorithm with two-parent nodes finds a trajectory of the same length as the original RRT algorithm approximately 30–40% faster. Adding a third parent node reduces the path search time by roughly 25%. With a fixed trajectory calculation time, the introduction of an additional parent node reduces the trajectory length by 6–11%. The introduction of two additional parent nodes reduces the length of the trajectory by 15–20% compared to the original RRT algorithm.

As a further development of the work, it is planned to study the proposed algorithms in a dynamic environment. It is planned to re-plan the rest of the path preservation of those nodes that the robot has not yet passed. Another important issue not covered in this article is the consideration of the uncertainties of the mobile robot model and their influence on the procedure of introducing virtual obstacles. In addition, it can be noted that it is possible to use a large number of the local optimization algorithms at the second stage of planning. Studying their effectiveness and finding the optimal moment for switching from global planning to local optimization and smoothing is also the subject of a separate study.

References

1. K.A. Kazakov, V.A. Semenov, “Obzor sovremennyh metodov planirovaniya puti,” TRUDY ISP RAN, T. 28(4), 2016, s. 241-294.
2. Sánchez-Ibáñez, J.R.; Pérez-del-Pulgar, C.J.; García-Cerezo, A. Path Planning for Autonomous Mobile Robots: A Review. Sensors 2021, 21, 7898. https://doi.org/10.3390/s21237898

3. L.E. Kavraki; P. Svestka; J.-C. Latombe; M.H. Overmars, “Probabilistic roadmaps for path planning in high-dimensional configuration spaces,” IEEE Transactions on Robotics and Automation, v. 12 (4), pp. 566-580, 1996.

4. А.Г. Сухарев, “Оптимальные стратегии поиска экстремума,” СССР. Вычислительная математика и математическая Computational физика, Т. 11(4), с. 910-924, 1971.

5. F. Samaniego, J. Sanchis, S. García-Nieto, R. Simarro. Recursive Rewarding Modified Adaptive Cell Decomposition (RR-MACD): A Dynamic Path Planning Algorithm for UAVs. Electronics, v. 8 (3), 306, 2019.

6. R. Toodesh, and S. Verhagen. "Adaptive, variable resolution grids for bathymetric applications using a quadtree approach," Journal of Applied Geodesy, v. 12(4), 2018, pp. 311-322.

7. P.E. Hart, N.J. Nilsson, B.A Raphael, “Formal Basis for the Heuristic Determination of Minimum Cost Paths,” IEEE Transactions on Systems Science and Cybernetics, v. 2, pp. 100 – 107, 1968.

8. A. Stentz, “Optimal and efficient path planning for partially known environments,” In Intelligent Unmanned Ground Vehicles, Springer, Boston, MA, USA, 1997, pp. 203–220.

9. Q. Wang, Y. Hao, F. Chen. “Deepening the IDA* algorithm for knowledge graph reasoning through neural network architecture,” Neurocomputing, v. 429, 2021, pp. 101-109.

10. R. Zhou, E.A. Hansen, “Memory-Bounded {A*} Graph Search,” The Florida AI Research Society Conference - FLAIRS, pp. 203–209, 2002.

11. R. Holte, M. Perez, R. Zimmer, A. MacDonald, “Hierarchical A*: Searching abstraction hierarchies efficiently,” Proceedings of the thirteenth national conference on Artificial intelligence, v. 1, pp. 530–535, 1996.

12. B. Liu, X. Xiao and P. Stone, "A Lifelong Learning Approach to Mobile Robot Navigation," in IEEE Robotics and Automation Letters, v. 6(2), 2021, pp. 1090-1096.

13. B.Y. Chen, X.-W. Chen, H.-P. Chen, W.H.K. Lam, “Efficient algorithm for finding k shortest paths based on re-optimization technique,” Transportation Research Part E: Logistics and Transportation Review, v. 133, 2020, Article number 101819.

14. X. Zhang, B. Wylie, C. Oscar, C.A. Moore, “Time-Optimal and Collision-Free Path Planning for Dual-Manipulator 3D Printer,” IEEE Transactions on Systems, Man, and Cybernetics: Systems, 2020-October, 9283493, pp. 2389-2396.

15. O. Khatib, “Real-Time Obstacles Avoidance for Manipulators and Mobile Robots,” International Journal of Robotics Research, vol. 5(2), pp. 90–98, 1986.

16. V.Kh. Pshikhopov (Ed.), D. Beloglazov, V. Finaev, V. Guzik, E. Kosenko, V. Krukhmalev, M. Medvedev, V. Pereverzev, A. Pyavchenko, R. Saprykin, I. Shapovalov, V. Soloviev. Path Planning for Vehicles Operating in Uncertain 2D Environments, Elsevier, Butterworth-Heinemann, 2017. 312 p, ISBN: 9780128123058.

17. S.S. Ge, Y.J. Cui, “New potential functions for mobile robot path planning,” IEEE Transactions on Robotics and Automation. v. 16 (5), pp. 615 – 620, 2000.

18. A.C. Woods, H.M. La, “A Novel Potential Field Controller for Use on Aerial Robots,” IEEE Transactions on Systems, Man, and Cybernetics: Systems, 49 (4), 7932539, pp. 665-676, 2019.

19. Y. Koren, J. Borenstein, “Potential field methods and their inherent limitations for mobile robot navigation,” International Conference on Robotics and Automation v. 2, pp. 1398 – 1404, 1991.

20. В.Х. Пшихопов, М.Ю. Медведев. “Групповое управление движением мобильных роботов в неопределенной среде с использованием неустойчивых режимов”, Труды СПИИРАН. 2018. Вып. 60. С. 39-63.

21. Zhou, C., Huang, B. & Fränti, P. A review of motion planning algorithms for intelligent robots. Journal of Intelligent Manufacturing, 2021. https://doi.org/10.1007/s10845-021-01867-z.

22. S. Chakravorty, S. Kumar, “Generalized Sampling-Based Motion Planners,” IEEE Transactions on Systems, Man, and Cybernetics – Part B: Cybernetics, vol. 41(3), 2011.

23. A. Ravankar, Ab. Ravankar, T. Emaru, Y. Kobayashi, “HPPRM: Hybrid Potential Based Probabilistic Roadmap Algorithm for Improved Dynamic Path Planning of Mobile Robots,” IEEE Acsess, vol. 8, pp. 221743 – 221766, 2020.

24. S.M. LaValle, J. Kuffner, “Randomized kinodynamic planning,” Int. Journal of Robotics Research, vol. 20(5), pp. 378–400, 2001.

25. S.M. LaValle, J.J. Kuffner, “Rapidly-exploring random trees: Progress and prospects,” 2000 Workshop on the Algorithmic Foundations of Robotics, pp. 293–308, 2000.

26. X. Wang, X. Li, Y. Guan, J. Song, R. Wang, “Bidirectional Potential Guided RRT* for Motion Planning,” IEEE Acsess, vol. 7, pp. 95046 – 95057, 2019.

27. A. Qureshi, Y. Ayaz, “Potential functions based sampling heuristic for optimal path planning,” Autonomous Robot, vol. 40, pp 1079–1093, 2016.

28. S. Karaman, E. Frazzoli, “Sampling-based algorithms for optimal motion planning,” The International Journal of Robotics Research, 30(7), pp. 846–894, 2011.

29. L. Chen, Y. Shan, W. Tian, B. Li, and D. Cao, “A Fast and Efficient Double-Tree RRT *Like Sampling-Based Planner Applying on Mobile Robotic Systems,” IEEE/ASME Transactions on Mechatronics, vol. 23(6), pp. 2568 – 2578, 2018.

30. J. Wang, M.Q.-H. Meng, O. Khatib, “EB-RRT: Optimal Motion Planning for Mobile Robots,” IEEE Transactions on Automation Science and Engineering, v. 17(4), pp. 2063-2073, 2020.

31. J. Janos, V. Vonasek, R. Penicka, “Multi-Goal Path Planning Using Multiple Random Trees,” IEEE Robotics and Automation Letter. v. 6(2), 2021, pp. 4201-4208.

32. S.N. Spitz and A.A.G. Requicha, “Multiple-goals path planning for coordinate measuring machines,” in Proc. IEEE Int. Conf. Robot. Automat., vol. 3, 2000, pp. 2322–2327.

33. D. Devaurs, T. Siméon, and J. Cortés, “A multi-tree extension of the transition-based RRT: Application to ordering-and-pathfinding problems in continuous cost spaces,” in Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst., 2014, pp. 2991–2996.

34. V. Vonásek and R. Pˇeniˇcka, “Space-filling forest for multi-goal path planning,” inProc. 24th IEEE Int. Conf. Emerg. Technol. Factory Automat., 2019, pp. 1587–1590.

35. R. Wang, X. Zhang, Y. Fang, B. Li, “Virtual-Goal-Guided RRT for Visual Servoing of Mobile Robots With FOV Constraint,” IEEE Transactions on Systems, Man, and Cybernetics: Systems, 2021.

36. V. Kostjukov, V. Pshikhopov, M. Medvedev, “Optimization of mobile robot movement on a plane with finite number of repeller sources,” SPIIRAS Proceedings. v. 19(1), pp. 43-78, 2020.

37. V. Kostjukov, M. Medvedev, V. Pshikhopov, “Method for Optimizing of Mobile Robot Trajectory in Repeller Sources Field,” Informatics and Automation. v. 20(3), pp. 690-726, 2021.

38. V. Pshikhopov, M. Medvedev, “Multi-Loop Adaptive Control of Mobile Objects in Solving Trajectory Tracking Tasks,” Automation and Remote Control, v. 81(11), pp. 2078–2093, 2020.

39. B.A. Güvenç, L. Güvenç, S. Karaman, “Robust Yaw Stability Controller Design and Hardware-in-the-Loop Testing for a Road Vehicle,” IEEE Transactions on Vehicular Technology, v. 58(2), pp. 555-571, 2009.

40. M. Dorigo, L.M. Gambardella, “Ant Colony System: A Cooperative Learning Approach to the Traveling Salesman Problem,” IEEE Transactions on Evolutionary Computation, v. 1(1), pp. 53-66, 1997.

41. J.H. Wilkinson, “The algebraic Eigenvalue Problem,” Oxford, Clarendon Press, 1965.

42. W. Chen and J. Zhang, “An Ant Colony Optimization Approach to a Grid Workflow Scheduling Problem With Various QoS Requirements," IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews), v. 39(1), pp. 29-43, 2009.

43. A.R. Gaiduk, O.V. Martjanov, M.Yu. Medvedev, V.Kh. Pshikhopov, N. Hamdan, A. Farhood, “Neural network based control system for robots group operating in 2-d uncertain environment,” Mekhatronika, Avtomatizatsiya, Upravlenie. v. 21(8), pp. 470 – 479, 2020.

Pshikhopov Viacheslav — Ph.D., Dr.Sci., Professor, Director, R&d department of robotics and control systems, Southern Federal University (SFedU). Research interests: planning and control of autonomous robots and manipulators with a focus on time-optimal control, control in obstructed unmapped environments, and intelligent and group control. The number of publications — 300. pshichop@rambler.ru; 2, Shevchenko St., 347922, Taganrog, Russia; office phone: +7(863)437-1694.

Medvedev Mikhail — Ph.D., Dr.Sci., Associate Professor, Leading researcher, Southern Federal University (SFedU). Research interests: planning and control of autonomous robots with a focus on adaptive and robust control, path planning methods, and neural network planning. The number of publications — 260. medvmihal@gmail.com; 2, Shevchenko St., 347922, Taganrog, Russia; office phone: +7(8634)371-694.

Kostjukov Vladimir — Ph.D., Associate professor, Southern Federal University (SFedU). Research interests: applied electrodynamics, dynamical systems modeling, path planning methods, aerodynamical modeling. The number of publications — 65. wkost-einheit@yandex.ru; 2, Shevchenko St., 347922, Taganrog, Russia; office phone: +7(8634)371-694.

Houssein Firas — Junior researcher, Southern Federal University (SFedU). Research interests: path planning algorithms, optimization algorithms, machine learning algorithms, and dynamical systems modeling. The number of publications — 0. info@sfedu.ru; 2, Shevchenko St., 347922, Taganrog, Russia; office phone: +7(8634)681-894.

Kadhim Azhar — Ph.D., Assistant, Department of mechanical techniques, Technical Institute of Nasiriyah. Research interests: path planning algorithms, machine learning methods. The number of publications — 0. info@nust.edu.iq; 26, Baghdad St., 64001, Nasiriyah, Iraq; office phone: +9647800778836.

Acknowledgements. The study is supported by the Russian Science Foundation, grant 22-29-00533, executed at Joint-stock Company “Scientific-Design bureau of Robotics and Control Systems”.

УДК 681.5 **DOI** 10.15622/ia.21.3.1

В.Х. Пшихопов, М.Ю. Медведев, В.А. Костюков, Ф. Хуссейн, А. Кадим

АЛГОРИТМЫ ПЛАНИРОВАНИЯ ТРАЕКТОРИЙ В ДВУМЕРНОЙ СРЕДЕ С ПРЕПЯТСТВИЯМИ

Пшихопов В.Х., Медведев М.Ю., Костюков В.А., Хуссейн Ф., Кадим А. **Алгоритмы планирования траекторий в двумерной среде с препятствиями.**

Аннотация. В данной статье предложены алгоритмы планирования и управления движением мобильного робота в двухмерной стационарной среде с препятствиями. Задача состоит в том, чтобы сократить длину запланированного пути, учесть динамические ограничения робота и получить плавную траекторию. Для учета динамических ограничений мобильного робота на карту добавляются виртуальные препятствия, перекрывающие невыполнимые участки движения. Такой способ учета динамических ограничений позволяет использовать картографические методы без увеличения их сложности. В качестве алгоритма глобального планирования используется модифицированная версия алгоритма быстрого исследования случайных деревьев (Multi parent nodes RRT – MPN-RRT). В этом алгоритме, в отличие от оригинальной версии, используется несколько родительских узлов, что уменьшает длину запланированной траектории по сравнению с исходной версией RRT с одним узлом. Кратчайший путь на построенном графе находится с помощью алгоритма оптимизации муравьиной колонии. Методами численного моделирования показано, что использование двух родительских узлов позволяет уменьшить среднюю длину пути для городской среды с низкой плотностью застройки. Для решения проблемы медленной сходимости алгоритмов, основанных на случайном поиске и сглаживании путей, алгоритм RRT дополнен алгоритмом локальной оптимизации. Алгоритм RRT ищет глобальный путь, который сглаживается и оптимизируется итеративным локальным алгоритмом. Алгоритмы управления нижнего уровня, разработанные в этой статье, автоматически уменьшают скорость робота при приближении к препятствиям или повороте. Общая эффективность разработанных алгоритмов продемонстрирована методами численного моделирования с использованием большого количества экспериментов.

Ключевые слова: мобильный робот, планирование движения, управление движением, движение робота, планирование пути.

Пшихопов Вячеслав Хасанович — д-р техн. наук, профессор, директор, нии робототехники и процессов управления, Южный федеральный университет (ЮФУ). Область научных интересов: управление мобильными роботами в неопределенных средах, оптимальное управление роботами, анализ и синтез систем группового управления, интеллектуальное управление и планирование в робототехнике. Число научных публикаций — 300. pshichop@rambler.ru; ул. Шевченко, 2, 347922, Таганрог, Россия; р.т.: +7(863)437-1694.

Медведев Михаил Юрьевич — д-р техн. наук, доцент, ведущий научный сотрудник, Южный федеральный университет (ЮФУ). Область научных интересов: адаптивное и робастное управления мобильными роботами, оценивание возмущений, методы анализа и синтеза систем автоматического управления. Число научных публикаций — 260. medvmihal@gmail.com; ул. Шевченко, 2, 347922, Таганрог, Россия; р.т.: +7(8634)371-694.

Костюков Владимир Александрович — канд. техн. наук, доцент, Южный федеральный университет (ЮФУ). Область научных интересов: математическое моделирование, мобильные роботы, алгоритмы планирования пути, ветроэнергетические установки. Число научных публикаций — 65. wkost-einheit@yandex.ru; ул. Шевченко, 2, 347922, Таганрог, Россия; р.т.: +7(8634)371-694.

Хуссейн Фирас — младший научный сотрудник, Южный федеральный университет (ЮФУ). Область научных интересов: алгоритмы планирования пути, алгоритмы оптимизации, машинное обучение, моделирование динамических системе. Число научных публикаций — 0. info@sfedu.ru; ул. Шевченко, 2, 347922, Таганрог, Россия; р.т.: +7(8634)681-894.

Кадим Азар — Ph.D., ассистент, кафедра механических технологий, Технический институт Насирии. Область научных интересов: алгоритмы планирования пути, методы машинного обучения. Число научных публикаций — 0. info@nust.edu.iq; ул. Багдадская, 26, 64001, Эн-Насирия, Ирак; р.т.: +9647800778836.

Поддержка исследований. Работа выполнена при финансировании Российского научного фонда (грант 22-29-00533, выполняемый в ОАО «НКБ робототехники и систем управления»).

Литература

1. K.A. Kazakov, V.A. Semenov, “Reviwes of moderm path planing methods,” Proceedings of ISP RAS, v. 28(4), 2016, pp. 241-294.
2. Sánchez-Ibáñez, J.R.; Pérez-del-Pulgar, C.J.; García-Cerezo, A. Path Planning for Autonomous Mobile Robots: A Review. Sensors 2021, 21, 7898. https://doi.org/10.3390/s21237898
3. L.E. Kavraki; P. Svestka; J.-C. Latombe; M.H. Overmars, “Probabilistic roadmaps for path planning in high-dimensional configuration spaces,” IEEE Transactions on Robotics and Automation, v. 12 (4), pp. 566-580, 1996.
4. A.G. Sukharev, “Optimal strategies of the search for an extremum,” U.S.S.R. Computational Mathematics and Mathematical Physics, v. 11(4), pp. 910-924, 1971. Translated from Russian, Journal Vychisl. Mat. i Mat. Fiz.
5. F. Samaniego, J. Sanchis, S. García-Nieto, R. Simarro. “Recursive Rewarding Modified Adaptive Cell Decomposition (RR-MACD): A Dynamic Path Planning Algorithm for UAVs,” Electronics, v. 8 (3), 306, 2019.
6. R. Toodesh, and S. Verhagen. "Adaptive, variable resolution grids for bathymetric applications using a quadtree approach," Journal of Applied Geodesy, v. 12(4), 2018, pp. 311-322.
7. P.E. Hart, N.J. Nilsson, B.A. Raphael, “Formal Basis for the Heuristic Determination of Minimum Cost Paths,” IEEE Transactions on Systems Science and Cybernetics, v. 2, pp. 100 – 107, 1968.
8. A. Stentz, “Optimal and efficient path planning for partially known environments,” In Intelligent Unmanned Ground Vehicles, Springer, Boston, MA, USA, 1997, pp. 203–220.
9. Q. Wang, Y. Hao, F. Chen. “Deepening the IDA* algorithm for knowledge graph reasoning through neural network architecture,” Neurocomputing, v. 429, 2021, pp. 101-109.
10. R. Zhou, E.A. Hansen, “Memory-Bounded {A*} Graph Search,” The Florida AI Research Society Conference - FLAIRS, pp. 203–209, 2002.

11. R. Holte, M. Perez, R. Zimmer, A. MacDonald, “Hierarchical A*: Searching abstraction hierarchies efficiently,” Proceedings of the thirteenth national conference on Artificial intelligence, v. 1, pp. 530–535, 1996.
12. B. Liu, X. Xiao and P. Stone, "A Lifelong Learning Approach to Mobile Robot Navigation," in IEEE Robotics and Automation Letters, v. 6(2), 2021, pp. 1090-1096.
13. B.Y. Chen, X.-W. Chen, H.-P. Chen, W.H.K. Lam, “Efficient algorithm for finding k shortest paths based on re-optimization technique,” Transportation Research Part E: Logistics and Transportation Review, V. 133, 2020, Article number 101819.
14. X. Zhang, B. Wylie, C. Oscar, C.A. Moore, “Time-Optimal and Collision-Free Path Planning for Dual-Manipulator 3D Printer,” IEEE Transactions on Systems, Man, and Cybernetics: Systems, 2020-October, 9283493, pp. 2389-2396.
15. O. Khatib, “Real-Time Obstacles Avoidance for Manipulators and Mobile Robots,” International Journal of Robotics Research, vol. 5(2), pp. 90–98, 1986.
16. V.Kh. Pshikhopov (Ed.), D. Beloglazov, V. Finaev, V. Guzik, E. Kosenko, V. Krukhmalev, M. Medvedev, V. Pereverzev, A. Pyavchenko, R. Saprykin, I. Shapovalov, V. Soloviev. Path Planning for Vehicles Operating in Uncertain 2D Environments, Elsevier, Butterworth-Heinemann, 2017. 312 p, ISBN: 9780128123058.
17. S.S. Ge, Y.J. Cui, “New potential functions for mobile robot path planning,” IEEE Transactions on Robotics and Automation. v. 16 (5), pp. 615 – 620, 2000.
18. A.C. Woods, H.M. La, “A Novel Potential Field Controller for Use on Aerial Robots,” IEEE Transactions on Systems, Man, and Cybernetics: Systems, 49 (4), 7932539, pp. 665-676, 2019.
19. Y. Koren, J. Borenstein, “Potential field methods and their inherent limitations for mobile robot navigation,” International Conference on Robotics and Automation v.2, pp. 1398 – 1404, 1991.
20. V. Pshikhopov, M. Medvedev, “Group control of autonomous robots motion in uncertain environment via unstable modes,” SPIIRAS Proceedings. v. 60(5), pp. 39-63, 2018.
21. Zhou, C., Huang, B. & Fränti, P. A review of motion planning algorithms for intelligent robots. Journal of Intelligent Manufacturing, 2021. https://doi.org/10.1007/s10845-021-01867-z.
22. S. Chakravorty, S. Kumar, “Generalized Sampling-Based Motion Planners,” IEEE Transactions on Systems, Man, and Cybernetics – Part B: Cybernetics, vol. 41(3), 2011.
23. A. Ravankar, Ab. Ravankar, T. Emaru, Y. Kobayashi, “HPPRM: Hybrid Potential Based Probabilistic Roadmap Algorithm for Improved Dynamic Path Planning of Mobile Robots,” IEEE Acsess, vol. 8, pp. 221743 – 221766, 2020.
24. S.M. LaValle, J. Kuffner, “Randomized kinodynamic planning,” Int. Journal of Robotics Research, vol. 20(5), pp. 378–400, 2001.
25. S.M. LaValle, J.J. Kuffner, “Rapidly-exploring random trees: Progress and prospects,” 2000 Workshop on the Algorithmic Foundations of Robotics, pp. 293–308, 2000.
26. X. Wang, X. Li, Y. Guan, J. Song, R. Wang, “Bidirectional Potential Guided RRT* for Motion Planning,” IEEE Acsess, vol. 7, pp. 95046 – 95057, 2019.
27. A. Qureshi, Y. Ayaz, “Potential functions based sampling heuristic for optimal path planning,” Autonomous Robot, vol. 40, pp 1079–1093, 2016.
28. S. Karaman, E. Frazzoli, “Sampling-based algorithms for optimal motion planning,” The International Journal of Robotics Research, 30(7), pp. 846–894, 2011.
29. L. Chen, Y. Shan, W. Tian, B. Li, and D. Cao, “A Fast and Efficient Double-Tree RRT *-Like Sampling-Based Planner Applying on Mobile Robotic Systems,” IEEE/ASME Transactions on Mechatronics, vol. 23(6), pp. 2568 – 2578, 2018.

30. J. Wang, M.Q.-H. Meng, O. Khatib, “EB-RRT: Optimal Motion Planning for Mobile Robots,” IEEE Transactions on Automation Science and Engineering, v. 17(4), pp. 2063-2073, 2020.

31. J. Janos, V. Vonasek, R. Penicka, “Multi-Goal Path Planning Using Multiple Random Trees,” IEEE Robotics and Automation Letter. v. 6(2), 2021, pp. 4201-4208.

32. S.N. Spitz and A.A.G. Requicha, “Multiple-goals path planning for coordinate measuring machines,” in Proc. IEEE Int. Conf. Robot. Automat., vol. 3, 2000, pp. 2322–2327.

33. D. Devaurs, T. Siméon, and J. Cortés, “A multi-tree extension of the transition-based RRT: Application to ordering-and-pathfinding problems in continuous cost spaces,” in Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst., 2014, pp. 2991–2996.

34. V. Vonásek and R. Pˇeniˇcka, “Space-filling forest for multi-goal path planning,” inProc. 24th IEEE Int. Conf. Emerg. Technol. Factory Automat., 2019, pp. 1587–1590.

35. R. Wang, X. Zhang, Y. Fang, B. Li, “Virtual-Goal-Guided RRT for Visual Servoing of Mobile Robots With FOV Constraint,” IEEE Transactions on Systems, Man, and Cybernetics: Systems, 2021.

36. V. Kostjukov, V. Pshikhopov, M. Medvedev, “Optimization of mobile robot movement on a plane with finite number of repeller sources,” SPIIRAS Proceedings. v. 19(1), pp. 43-78, 2020.

37. V. Kostjukov, M. Medvedev, V. Pshikhopov, “Method for Optimizing of Mobile Robot Trajectory in Repeller Sources Field,” Informatics and Automation. v. 20(3), pp. 690-726, 2021.

38. V. Pshikhopov, M. Medvedev, “Multi-Loop Adaptive Control of Mobile Objects in Solving Trajectory Tracking Tasks,” Automation and Remote Control, v. 81(11), pp. 2078–2093, 2020.

39. B.A. Güvenç, L. Güvenç, S. Karaman, “Robust Yaw Stability Controller Design and Hardware-in-the-Loop Testing for a Road Vehicle,” IEEE Transactions on Vehicular Technology, v. 58(2), pp. 555-571, 2009.

40. M. Dorigo, L.M. Gambardella, “Ant Colony System: A Cooperative Learning Approach to the Traveling Salesman Problem,” IEEE Transactions on Evolutionary Computation, v. 1(1), pp. 53-66, 1997.

41. J. H. Wilkinson, “The algebraic Eigenvalue Problem,” Oxford, Clarendon Press, 1965.

42. W. Chen and J. Zhang, “An Ant Colony Optimization Approach to a Grid Workflow Scheduling Problem With Various QoS Requirements," IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews), v. 39(1), pp. 29-43, 2009.

43. A.R. Gaiduk, O.V. Martjanov, M.Yu. Medvedev, V.Kh. Pshikhopov, N. Hamdan, A. Farhood, “Neural network based control system for robots group operating in 2-d uncertain environment,” Mekhatronika, Avtomatizatsiya, Upravlenie. v. 21(8), pp. 470 – 479, 2020.