ISSN 2071-1077(print)
ISSN 2663-2411(online)

doi: 10.32620/oikit.2026.108.17

UDC 621.865:004.896

Igor Nevliudov,
Vladyslav Yevsieiev,
Svitlana Maksymova,
Serhii Tesliuk

Numerical simulation of mobile robotic platform route construction in dynamic space using QR tags

Kharkiv National University of Radio Electronics, Ukraine

In the modern conditions of Industry 5.0 development, mobile robotic platforms play a key role in ensuring the automation of logistics and production processes. It requires the development of effective navigation methods in a dynamic environment with static and moving obstacles. The task of adaptive formation of a movement route based on sensory information, in particular QR tags, is of particular relevance. It allows for the prompt determination of the coordinates of target points and provide flexibility in controlling the mobile platform. The object of the study is the process of moving a mobile platform in a discrete dynamic workspace. The subject of the study is a method for constructing a mobile platform route using an incremental replanning algorithm and a model for determining target coordinates based on QR tags. The purpose of the study is a mathematical model development and software implementation of a method for constructing an optimal mobile platform route in a dynamic environment, taking into account changes in the space configuration and prompt determination of the coordinates of target points based on QR tags. The study used methods of mathematical modeling, graph theory, numerical integration, incremental pathfinding algorithms and computer modeling using a discrete occupancy map. The scientific novelty of the work lies in the development of mathematical support for the method of constructing a mobile platform route with the integration of a target observation model based on QR tags. This allows for adaptive replanning of the trajectory in real time when the state of the environment changes. The obtained results of numerical modeling confirm the effectiveness of the proposed method, which ensures safe movement of the mobile platform, adaptive avoidance of obstacles and stable achievement of target points with high computational efficiency. The developed model demonstrates the potential for implementation in intelligent control systems for mobile robots and can be used as a basis for further improvement of navigation algorithms in dynamic environments.
Key words: mobile robotic platform, path planning, dynamic environment, D* Lite algorithm, occupancy grid, numerical simulation, QR codes, incremental replanning, autonomous navigation, Industry 5.0

Introduction

Modern logistics and production systems, oriented towards the concepts of Industry 4.0 and Industry 5.0, require highly efficient mobile robotic platforms capable of autonomously moving in a dynamic environment with the presence of both static and moving obstacles [1-3]. The task of adaptive route planning in conditions of variable configuration of the workspace, where the position of obstacles and target points can change in real time, which makes it impossible to use only static navigation methods [4-6], is of particular relevance. One of the promising approaches is the use of numerical modeling based on a discrete occupancy map and incremental replanning algorithms, such as D* Lite, which allow for effective trajectory updating without complete route recalculation [7-9]. Additionally, the use of QR tags as a source of coordinates of key points provides a flexible and technologically simple mechanism for integrating sensory information into the navigation system of a mobile platform [10-13]. This allows for the implementation of dynamic changes in target coordinates without interfering with the control system program code and increases the level of autonomy

 ISSN 2071-1077(print)
ISSN 2663-2411(online)

of the robotic platform. The development of mathematical software and a software model of such a process is necessary for studying the properties of navigation algorithms, assessing their computational efficiency and testing their performance in a changing environment. Numerical modeling allows for the study of the behavior of a mobile platform, optimizing route construction algorithms and ensuring safe interaction with dynamic objects. Thus, the study of methods for numerical modeling of the mobile platform route construction using QR tags is relevant and appropriate for creating adaptive, efficient and intelligent robotic transport systems.

1. Related works

In the article Shen Y., Shen Y., et al. [14], a generalized review of the progress of global and local path planning methods and tracking control methods for garden mobile robots in complex scenarios is proposed. it allows a systematic comparison of approaches to navigation in conditions of narrow aisles, plant obstacles and partial observability of the environment. However, direct application of the solutions is limited, since the review is focused on the specifics of orchard environments and the task of trajectory following.

In the paper Shanmugaraja M., Thangamuthu M., et al. [15], a comprehensive review of hybrid path planning algorithms for autonomous mobile robots is developed with a focus on combining graph, sampling-based, reactive and optimization strategies, which allows a reasonable choice of combinations of methods for the requirements of speed, adaptability and obstacle avoidance. However, direct transfer is limited because the paper is of a review nature and does not provide a complete scheme of "QR-target tracking → update → incremental replanning".

In the study of Jin P., Li W., et al. [16], a 3D obstacle avoidance algorithm for underwater vehicles (UUV) based on improved D* Lite combined with artificial potential field (APF) and software design is proposed. This makes it possible to improve the comprehensive planning performance in three-dimensional space compared to the basic D* Lite-APF options. Unfortunately, direct use is limited because the environment and motion model are 3D and specific to UUV, while the mobile platform uses 2D discretization, occupancy map and event-based goal setting via QR tags with real-time requirement for route recalculation on the grid.

The paper by Wu H., Zhong Y., et al. [17] proposes a hierarchical planning framework combining A* and D* Lite for heterogeneous sea ice scenarios with the integration of global and local planning levels. It allows to improve the suitability of routing under different map resolutions and changing ice conditions. However, direct application is limited, since the setting is focused on a macroscale environment with multi-level maps and coordinate transformations. Furthermore, in this study the system is critical for frequent local updates of occupancy cells due to dynamic obstacles and operational change of the endpoint via QR tags without external geospatial models.

In the paper, Mlinarček D., Jánoš R., et al. [18] investigated the quantitative tuning of costmap for autonomous navigation and performed a comparison in simulation and real experiments on the Hiwonder JetAcker platform. It allows to justify the parameters of the navigation stack and improve the stability of obstacle avoidance in practical conditions. However, direct transfer is limited, since the emphasis is on empirical tuning of costmap in a specific implementation of navigation software. And the study on QR tags requires a formalized target observation channel and incremental route replanning in a discrete environment with metrics of step time and number of replannings.

ISSN 2071-1077(print)
ISSN 2663-2411(online)

In the study of Liang Z., Wang L., et al. [19], an approach to autonomous obstacle avoidance and path planning for mobile robots in orchard environments is proposed, combining map construction and positioning methods. This allows obtaining safe, smoothed, and kinematically consistent trajectories in complex long-duration missions. In our study, direct application is limited, since the main complexity there is related to mapping and localization in orchard environments. In our formulation the key is the mechanism of setting sequential goals through QR tags and fast replanning of D* Lite on the occupancy map under dynamic obstacles.

In the article [20] Abu-Jassar A., Al-Sukhni H., et al. proposed a mobile robot route construction based on BRRT and A*(H-BRRT) using A* as an optimizer and implemented in Python, which makes it possible to form routes in complex environments and consider combined search strategies. Within the framework of this topic, direct use is limited, since BRRT/A*(H-BRRT) are focused on route construction mainly in a more static setting and do not contain an integrated mechanism for event-based goal change through QR tags and incremental value updates with frequent changes in occupancy, which is a basic requirement for a mobile platform in a dynamic space.

The general conclusion is that modern publications demonstrate the rapid development of hybrid and incremental planning methods, hierarchical schemes and practically oriented tuning of navigation subsystems, however, the issue of operational assignment of consecutive key points through QR tags with guaranteed real-time replanning on a discrete occupancy map remains insufficiently covered. That is why research aimed at numerical modeling of mobile platform routing in dynamic space with QR tags as a source of route endpoints is relevant for Industry 5.0 logistics tasks, since they combine flexible “

The purpose of the study is a mathematical model development and software implementation of a method for constructing an optimal mobile platform route in a dynamic environment, taking into account changes in the space configuration and prompt determination of the coordinates of target points based on QR tags.

2. Development of mathematical support for the method of constructing a route for moving a mobile robotic platform

To simplify understanding, before describing the mathematical models of the method for constructing a mobile platform movement route in dynamic space using QR tags, let us introduce the following abbreviations:: W, H – map size; Δt – simulation update step; O_s – static obstacle; M – static obstacle; p_i – state vector; v_i – velocity; r_i – i-th dynamic obstacle radius; $O_d(t)$, $O(t)$ – dynamic and full occupation; s_{start} – start point; s_{goal} – finish point (goal); $c(s,s')$ – cost of transition; $g(s), rhs(s)$ – functions D* Lite; $h(\cdot)$ – heuristic function, when $w \geq 1$, it is its weight; k_m – key correction; U – priority queue of vertices for update; ϕ – converting QR data into goal coordinates.

In the first step, we develop a model of the mobile platform environment in the form of a discrete map and a set of static and dynamic obstacles. Let the working area be represented by a grid of size $W \times H$:

$$\mathcal{G} = \{ (x,y) \mid x \in \{0,\dots,W-1\}, y \in \{0,\dots,H-1\} \}, \quad (1)$$

where: $\mathcal{G}$ – grid, i.e. the complete set of all valid map cell indices. This is the simulation coordinate system in which the planner (D* Lite) searches for a route; x – cell

 ISSN 2071-1077(print)
ISSN 2663-2411(online)

coordinates along the axis X ; y – cell coordinates along the axis Y ; $\{0,\dots,W-1\}$, $\{0,\dots,H-1\}$ – restricting indices according to the dimension of the map (matrix $W\times H$).

Let us describe static obstacles, let $O_s \subset \mathcal{G}$, as a set of static obstacles:

$$O_s = \{ (x,y) \in \mathcal{G} \mid static[y,x]=1 \}, \tag{2}$$

where: $static[y,x]$ – static *interference* occupancy matrix, this is a two-dimensional array of size $W\times H$, and is described by the following system:

$$static[y,x] = \begin{cases} 1, if\ \ the\ cell\ is\ occupied\ by\ static\ interference \\ 0, if\ \ the\ cell\ is\ free \end{cases}. \tag{3}$$

Expression 2 defines forbidden regions that do not change in time.

Let us represent dynamic obstacles, let there are M moving obstacles. For i-th obstacle:

– continuous state:

$$p_i(t) = \begin{bmatrix} x_i(t) \\ y_i(t) \end{bmatrix}, v_i(t) = \begin{bmatrix} v_{x,i}(t) \\ v_{y,i}(t) \end{bmatrix}, r_i > 0; \tag{4}$$

– grid occupation zone:

$$O_{d,i}(t) = \left\{ (x,y) \in \mathcal{G} \middle| \sqrt{(x-x_i(t))^2 + (y-y_i(t))^2} \le r_i \right\}; \tag{5}$$

– total dynamic employment:

$$O_d(t) = \bigcup_{i=1}^{M} O_{d,i}(t), \tag{6}$$

where: $p_i(t)$ – obstacle center coordinates; $v_i(t)$ – velocity; r_i – obstacle radius, in cells; $(x_i(t), y_i(t))$ – i-th dynamic obstacle center coordinates in time t; $v_{x,i}(t)$ – horizontal component of obstacle velocity; $v_{y,i}(t)$ – vertical component of obstacle velocity; $O_{d,i}(t)$ – the area of occupation of the i-th dynamic obstacle in time t, is the set of map cells that are considered occupied by a moving object; $O_d(t)$ – definition of a new set of occupations.

To form a complete picture of the workspace state at each moment of time, taking into account both static and dynamic obstacles, which ensures the correct determination of movement permissible areas of the mobile robotic platform, we will develop a model of the general occupation map. This will be used by the D* Lite algorithm to check the passability of cells, timely replanning of the trajectory and guaranteeing safe robot navigation in a changing dynamic environment.

$$O(t) = O_s \cup O_d(t),\ \ free(t) = \mathcal{G} / O(t), \tag{7}$$

where: $free(t)$ – set of free cells at a point in time t.

Let us describe a model of dynamic obstacle motion with the possibility of "repulsion" from static obstacles, which will simulate the movement of people and other mobile cargo robots in the working zone.

– basic kinematics based on integration of speeds for a step (k) discretization Δt:

$$p_i^{k+1} = p_i^k + v_i^k \Delta t. \tag{8}$$

 ISSN 2071-1077(print)
ISSN 2663-2411(online)

Model (8) predicts the next obstacle position.

– we define the conditions for preventing passage through static obstacles, through the obstacle disk collision predicate O_s:

$$collide_s\left(p_i\right)=\begin{cases}1,\exists\left(x,y\right)\in O_s:\left(x,y-p_i\right)\le r_i\\ 0,other\end{cases},\quad(9)$$

where: $collide_s\left(p_i\right)$ – is a collision check function for the i -th dynamic obstacle, and determines whether the moving object is in contact with the static obstacle. The value of this function is used to change the obstacle's speed (reflection) and prevent it from entering the forbidden area. If $collide_s\left(p_i^{k+1}\right)=1$, then velocity reflection is applied.

Model (9) ensures that dynamic obstacles do not pass through O_s, but are reflected.

The next step is to develop a model of a mobile platform on a grid, that is, we will implement the simulation of the robot's movement in cells along the found route.

– robot state:

$$s_R^k=\left(x_R^k,y_R^y\right)\in\mathcal{G},\quad(10)$$

where: s_R^k – state vector of the mobile robotic platform at a point in time k, which determines the current position of the robot in the workspace and is used by the *D** Lite algorithm as the initial vertex of the graph to build the optimal route to the target point; x_R^k – coordinate of the robot along the axis X at a point in time k; y_R^y – coordinate of the robot along the axis Y at a point in time k.

The robot state model (10) describes the current position of the mobile platform in the digital environment and is a key parameter for implementing navigation, route replanning, and ensuring safe movement in a dynamic workspace using the *D** Lite algorithm and QR-defined target points.

– discrete control, describes at each step the robot moves to the neighboring cell when using 8-connectivity:

$$s_R^{k+1}=s_R^k+u^k,$$
$$\mathcal{U}=\left\{\left(\pm1,0\right),\left(0,\pm1\right),\left(\pm1,\pm1\right)\right\},\quad(11)$$

where: s_R^{k+1} – robot state vector at the next time point $k+1$; u^k – discrete control vector at a point in time k; $\mathcal{U}$ – the set of permissible control actions, determines all possible directions of robot movement (8-connected robot movement (up, down, left, right and diagonally)) on a discrete map.

Model (11) provides a mathematical connection between the current and next state of the robot, allows you to simulate the trajectory execution in time and is used to update the robot position, check the achievement of the target point and form the actual trajectory of movement in a dynamic environment.

Let us develop the formulation of the planning problem based on graph theory [21]. Let us construct a state graph:

$$\mathcal{V}=free\left(t\right)\subseteq\mathcal{G},\ \mathcal{E}=\{(s,s')\,|\,s\in Nbr(s)\cap\mathcal{V}\},\quad(12)$$

where: $\mathcal{V}$ – the set of vertices of the state graph that corresponds to all permissible (free) coordinates of the workspace that can be used by the mobile robotic platform for movement and route construction; $free(t)$ – the set of free cells in

 ISSN 2071-1077(print)
ISSN 2663-2411(online)

the workspace at a time t; $\mathcal{E}$ – the set of edges of the state graph that defines all possible transitions between permissible states of the robot and forms the graph structure for the trajectory planning algorithm; (s,s')– an ordered pair of graph vertices that defines a possible transition of the robot from state s to a neighboring state s' according to the permissible directions of movement; $Nbr(s)$ – set of neighboring states for a state s, which defines all possible coordinates to which the robot can move in one step according to the discrete control model.

– transition cost:

$$c(s,s')=\begin{cases} 1,\|s-s_1'\|=1(orthogonal) \\ \sqrt{2},\|s-s_1'\|=2(diagonally) \\ \infty, ifs\ or\ s\in O(t) \end{cases}, \tag{13}$$

where: $c(s,s')$ – local step price; $O(t)$ – actual obstacles.

Using the *D** Lite method, we implement an incremental search for the shortest path [22]. Let s_{start}, be the initial position of the robot, and accordingly s_{goal} – is a goal, provided that D* Lite supports two functions: $g(s)$ – current estimate of the cost of the best path from s to *goal*; $rhs(s)$ – “one-step lookahead” assessment.

– definition rhs:

$$rhs(s)=\begin{cases} 0, s=s_{goal} \\ \min\limits_{s'\in Succ(s)}\left[c(s,s')+g(s')\right], s\neq s_{goal} \end{cases}, \tag{14}$$

where: $Succ(s)$ – the set of successor states that are adjacent to state s and can be reached in one movement step according to the discrete control model;

$\min\limits_{s'\in Succ(s)}$ – the minimum operator, which determines the smallest possible total cost of movement to the goal through all permissible neighboring states, ensuring the choice of the optimal direction of movement.

Model (14) provides the calculation of a local estimate of the cost of movement to the target point and is used by the D* Lite algorithm to determine the consistency of states, update the priority queue and incrementally replan the optimal route of a mobile robotic platform in a dynamic environment.

– let us describe the consistency of states s:

$$g(s)=rhs(s). \tag{15}$$

Expression (15) describes the condition that D* Lite updates only “inconsistent” vertices. By “inconsistent” vertices we mean such graph states for which the value of the cost function $g(s)$ is not equal to the value of the one-step estimate $rhs(s)$, which indicates the presence of outdated or incorrect information about the optimal cost of reaching the target point.

– to increase the speed of search and reduce the replanning time, it is proposed to use the weighted Chebyshev heuristic [23]:

$$h(s_{start},s)=w\cdot max(|x-x_s|,|y-y_s|),\ w\geq 1, \tag{16}$$

where: w – heuristic weighting factor; $h(\cdot)$ – heuristic evaluation.

 ISSN 2071-1077(print)
ISSN 2663-2411(online)

– priority key, allows you to determine the order of processing vertices in the priority queue. D* Lite uses the priority queue U with the key:

$$k(s)=\left[k_1(s),k_2(s)\right], \tag{17}$$

where:

$$k_1(s)=min\,min\left(g(s),rhs(s)\right)+h\left(s_{start},s\right)+k_m,\ k_2(s)=min\left(g(s),rhs(s)\right), \tag{18}$$

where: $k(s)$ – vertex priority key vector s, which is used by the D* Lite algorithm to determine the order of processing states in the priority queue and ensures correct and efficient re-planning of the route of the mobile robotic platform; $k_1(s)$ – the first component of the key, which determines the priority of the vertex taking into account the current minimum estimate of the cost of achieving the goal and the heuristic distance from the current position of the robot to the state s, used to determine the most promising direction for route search; $k_2(s)$ – the second key component, which determines the current lowest estimate of the cost of reaching the target point from the state$s$ without taking into account heuristics and is used as an additional criterion for ordering vertices.

– update when the dynamics of the environment change, necessary for calculating when the map changes (cells become occupied/free), edges change $c(s,s')$. Let the set of changed cells be:

$$\Delta O=\left(O(t)\setminus O(t-\Delta t)\right)\cup\left(O(t-\Delta t)\setminus O(t)\right), \tag{19}$$

where: ΔO – a set of changed cells of the occupancy map, which defines all coordinates of the *workspace* whose state has changed between two consecutive points in time, and is used by the D* Lite algorithm for local replanning of the trajectory of a mobile robotic platform; $O(t)$ – set of occupied cells at the current time t, which includes all cells occupied by static and dynamic obstacles after their movement; $O(t-\Delta t)$ – set of occupied cells at the previous point in time $t-\Delta t$, which describes the state of the occupancy map before updating the position of dynamic obstacles; $O(t)\setminus O(t-\Delta t)$ – a set of newly occupied cells that were free at the previous time point but became occupied due to the movement of dynamic obstacles; $O(t-\Delta t)\setminus O(t)$ – a set of cells that were occupied at a previous point in time but became free after the movement of dynamic obstacles.

Model (19) provides the definition of local changes in the structure of the workspace, which allows the D* Lite algorithm to update only those vertices of the graph that have been affected by environmental changes and effectively perform adaptive replanning of the route of a mobile robotic platform in a dynamic environment.

To obtain a sequence of cells (path) for the robot's movement, policy extraction is not necessary. After D* Lite convergence, the route from s_{start} is built greedily [24]. Greedy route recovery means that after the D* Lite algorithm is completed, the trajectory is built by sequentially selecting the next state that has the minimum total cost estimate of the transition to the target point according to the cost function. At each step, the robot chooses the most optimal neighbor based on the locally best value, without reviewing all possible alternative paths, which allows for a quick formation of the final route. This approach uses the already calculated values of the function $g(s)$ and provides efficient recovery of the optimal trajectory in a dynamic environment:

 ISSN 2071-1077(print)
ISSN 2663-2411(online)

$$s_{K+1} = arg\,min_{s' \in Succ(s_k)} \left[c(s,s') + g(s) \right], s_0 = s_{start}, \quad (20)$$

Let's imagine QR tags as a model for determining the goal (s_{goal}), that is, QR tags will be "anchors" on the basis of which the mobile platform's movement route in dynamic space will be built.

– target observation, let the camera give the z dimension (text from the QR code), which is converted into *goal* coordinates:

$$z \xrightarrow{\phi} \hat{s}_{goal} = \left(\hat{x}_g, \hat{y}_g \right), \quad (21)$$

where: z – observation vector obtained from the sensor system (laptop camera), which contains QR tag data and is used as input information to determine the coordinates of the target point of the mobile robotic platform's route; ϕ – an observation decoding function that processes QR tag data and converts the information message z in the coordinates of the target state of the robot in discrete space; $\hat{s}_{goal}$ – estimated (determined based on observation) state of the target point, which is used by the planning algorithm as a new final vertex of the graph for route construction; $\hat{x}_g$ – estimated coordinate of the target point along the axis X, obtained as a result of decoding a QR tag, which determines the horizontal position of the target in the workspace; $\hat{y}_g$ – estimated coordinate of the target point along the axis Y, obtained by decoding a QR tag, which determines the vertical position of the target in the workspace.

Model (21) provides integration of QR tag sensor data into the motion planning system, allowing the mobile robotic platform to determine new target coordinates in real time and adaptively rebuild the route according to the received information.

– "keypoint" logic, implements sequential movement between key points, which are set via QR code or manually by the operator. That is, after reaching the current target:

$$s_R = s_{goal} \Rightarrow to\ expect\ a\ new\ ones_{goal}, \quad (22)$$

where: s_R – the current state of the mobile robotic platform, which determines its actual position in a discrete workspace and is used to control the route execution process.

The logic in (22) ensures the sequential execution of navigation tasks by controlling the achievement of the current goal and the automatic transition of the system to the new coordinate waiting mode, which allows for the implementation of step-by-step movement of the mobile robotic platform between specified control points in a dynamic environment.

The developed mathematical software allows for the formalization of the process of building a mobile platform route in a dynamic environment by integrating a discrete space model, a dynamic occupancy map and the incremental D* Lite algorithm, which provides adaptive trajectory replanning when the position of obstacles changes. The use of a target observation model based on QR tags provides the ability to quickly set new key coordinates in real time without the need for prior route programming, which increases the flexibility and autonomy of the system. The introduction of the functions $\boldsymbol{g}(s), rhs(s)$ and the priority key $k(s)$ allows for efficient local updating of only those areas of the graph that have undergone changes, which significantly reduces computational costs and increases the speed of the method. The general occupancy map model ensures correct consideration of both static and dynamic obstacles, which increases navigation safety and eliminates the construction

 ISSN 2071-1077(print)
ISSN 2663-2411(online)

of trajectories through impassable areas. Taken together, this provides high adaptability, computational efficiency and suitability of the method for practical application in automated logistics systems and Industry 5.0 environments [25-26].

3. Conducting numerical simulations of the route construction for a mobile robotic platform and analyzing the results obtained

The purpose of the experiment is to verify the performance and effectiveness of a numerical model of mobile platform navigation in a dynamic space with static and dynamic obstacles based on *D** Lite incremental replanning and operational goal setting via QR tags.

Tasks that need to be solved to achieve the goal:

1. Generate a discrete map of the working area $\mathcal{G}$ and static occupancy map O_s with correct traversability constraints.

2. Model dynamic obstacles as moving circular regions $O_d(t)$ with physically correct interaction with static obstacles (without passing through them) and reflection from boundaries.

3. Construct a graph of possible states $\mathcal{V} = free(t) \subseteq \mathcal{G}$ and a set of transitions $\mathcal{E}$ and implement route search to the destination *D** Lite with functions $g(s), rhs(s)$ and a key $k(s)$.

4. Provide incremental replanning when the set of occupied cells changes ΔO with updating only locally influential vertices.

5. Implement a cycle of consecutive key points "achieving the goal → choosing the input method → obtaining a new s_{goal} → replanning", including QR mode with frame fixation until the goal is reached.

6. Provide real-time route and trajectory animation and collection of performance metrics to assess the speed and stability of work;

Expected results: the robot stably builds and executes the route to sequentially specified goals without entering occupied cells, when moving dynamic obstacles, correct local replanning occurs without complete recalculation from scratch. Dynamic obstacles do not cross static areas thanks to the collision and reflection model, and the collected metrics demonstrate an acceptable step calculation time.

Input data for numerical modeling:

– W=70,H=42 – width and height of the map in cells;

– $speed$ =9 – random generator seed for reproducibility of obstacle configuration and starting point;

– dt =0.25 – simulation time step for updating interference dynamics and replanning frequency;

– $heuristic\_weight$ = 1.3 – heuristic weight in *D** Lite for a trade-off between speed and optimality;

– $movers = (v_x, v_y, r)$ – parameters of each dynamic obstacle that specify the speed of movement and occupancy radius;

– $goal\_input$ – way of forming an observation z and converting it into coordinates $\hat{s}_{goal}$.

Description of hardware for conducting the study: Microsoft Surface Pro 9 with the following parameters: CPU Deca-core Intel Core i7-1255U (1.7 – 4.7 GHz), GPU Iris Xe Graphics, RAM 16Gb, SSD 512.

Software: Windows 11 Pro (version 24H2) OS type 64-bit operating system,

ISSN 2071-1077(print)
ISSN 2663-2411(online)

processor based on x64 architecture.

Development environment for the program for numerical modeling PyCharm 2025.1.1.1 and programming language Python 3.13.7 [27-29].

The general view of the numerical simulation program interface for building a mobile robotic platform route in dynamic space using QR tags is presented in Figure 1. In the form of blue squares, static obstacles that model storage areas or production equipment appear, as circles with a red dot appear as dynamic objects, which are people or other mobile robots. The QR Camera window allows you to read QR codes and build and display the route of a mobile robotic platform in real time.

The obtained numerical simulation results are presented in Figures 2-5. Table 1 also presents a step-by-step simulation experiment of building a mobile platform route in dynamic space with obtaining the s_{goal} coordinates from the QR code.

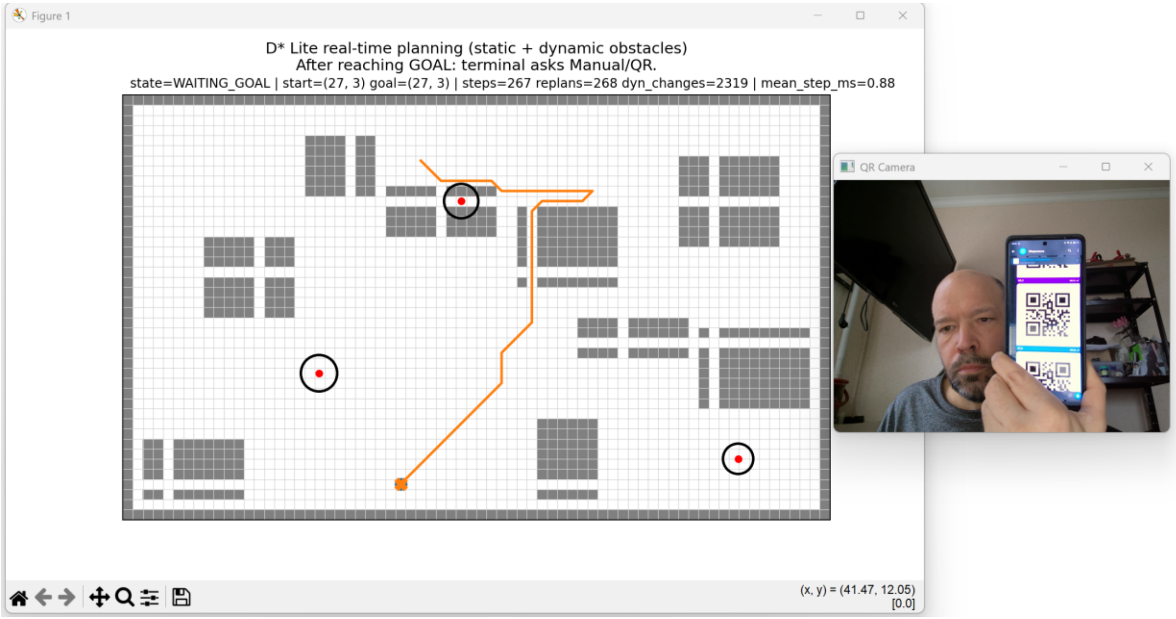


Fig. 1. General view of the program for numerical simulation in the mode of reading the s_{goal} coordinates from the QR code

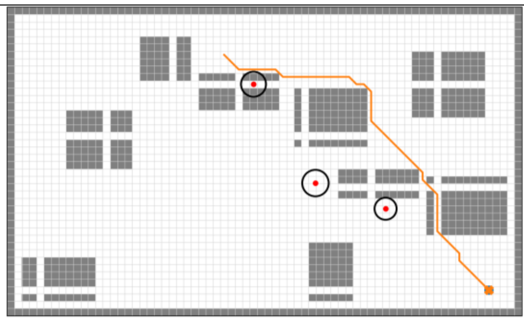
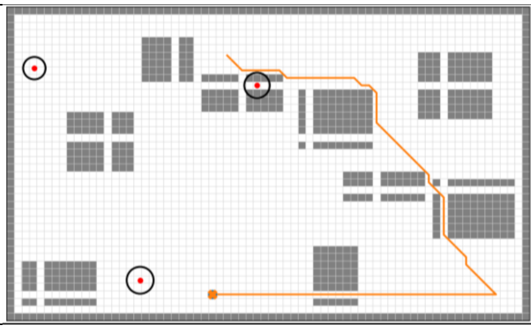
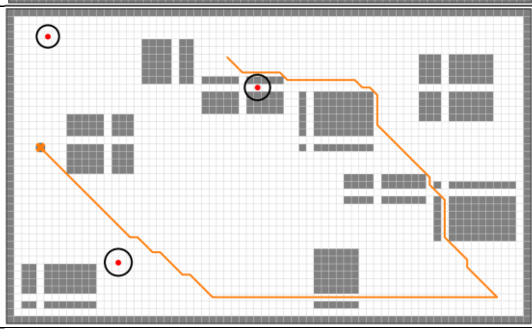
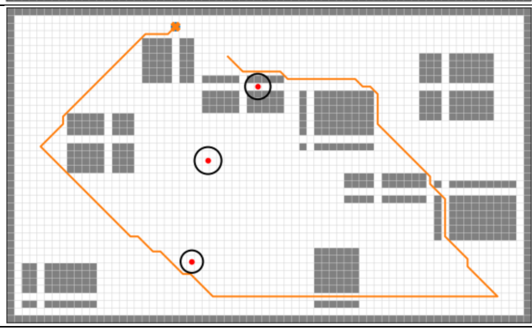
The obtained simulation results (Table 1) confirm that the mobile platform stably builds an acceptable trajectory to each target point s_{goal}, obtained from the QR code, with automatic adaptation of the route to the configuration of static and dynamic obstacles without collisions.

Qualitative analysis shows that the trajectory passes through topologically optimal corridors of free space and maintains a safe distance to obstacles, while the route structure changes according to the spatial location of the targets (65,3), (27,3), (4,23), (22,39), (65,3), (27,3), (4,23), (22,39), (65,3), (27,3), (4,23), (22,39), which confirms the correctness of the work of the incremental replanning. Numerically, the length of the trajectories correlates with the Euclidean distance to the target and the configuration of obstacles, demonstrating the efficient use of the available space without excessive deviations, and the consistent achievement of all key points confirms the convergence of the algorithm and its suitability for navigation in a dynamic environment. This indicates the high accuracy of determining the target coordinates through QR tags and the computational efficiency of the D* Lite method when updating the occupancy map in real time.

ISSN 2071-1077(print)
ISSN 2663-2411(online)

Table 1

Step-by-step presentation of building a mobile platform route in dynamic space with obtaining s_{goal}coordinates from a QR code

Point number	Coordinates (x, y)	QR code	Route
1	(65,3)		
2	(27,3)		
3	(4,23)		
4	(22,39)		

The fully constructed and executed route by the mobile platform, as a result of the simulation, is presented in Figure 2.

The presented executed trajectory demonstrates that the mobile platform successfully reached the target point, while the route is formed taking into account the geometry of static obstacles and passes through available free space corridors without crossing occupied areas. Qualitative analysis shows that the trajectory has a globally optimal nature with local direction adjustments that correspond to the topology of the environment and ensure safe obstacle avoidance while maintaining a sufficient margin

 ISSN 2071-1077(print)
ISSN 2663-2411(online)

of distance. Numerically, the length of the actual trajectory exceeds the Euclidean distance to the target, which is an expected result in an environment with constraints. However, the shape of the route indicates the efficient use of the available space and the absence of unnecessary oscillations or inefficient cycles. This confirms the convergence and computational efficiency of the D* Lite algorithm. This confirms the correctness of the replanning model and its suitability for mobile platform navigation in a structured dynamic environment.

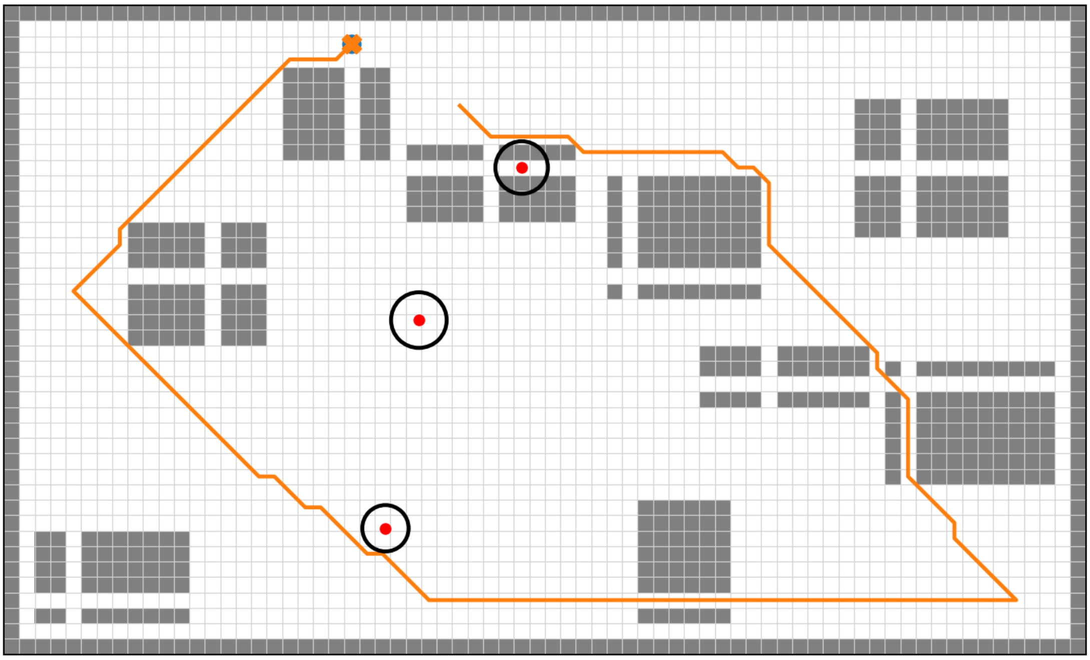


Fig. 2. Executed trajectory (final snapshot)

The calculation time per step graph (Fig. 3) demonstrates the stable operation of the D* Lite algorithm with a low average processing time, which is mainly in the range of approximately 0.5-2 ms, which indicates the high computational efficiency of incremental replanning. The presence of individual peaks up to 10-18 ms corresponds to the moments of significant changes in the occupancy map or the need to update a larger number of graph vertices. However, their limited number confirms the local nature of the recalculation and the absence of a complete replanning. Overall, the results confirm that the method provides a speed sufficient for real-time operation and demonstrates stable performance even in a dynamic environment with moving obstacles.

The graph (Fig. 4) shows that the length of the planned route decreases monotonically as the mobile platform approaches the target point, which numerically confirms the correctness of the D* Lite algorithm and the convergence of the planning process. The maximum values of the route length are approximately in the range of 24-37 cells and correspond to the initial position of the robot relative to the target, while a decrease to values of about 1-2 cells indicates the achievement of the target state. Qualitatively, this behavior confirms that the algorithm performs optimal replanning without the appearance of unstable or inefficient routes, ensuring a consistent and effective reduction of the distance to the target in a dynamic environment.

ISSN 2071-1077(print)
ISSN 2663-2411(online)

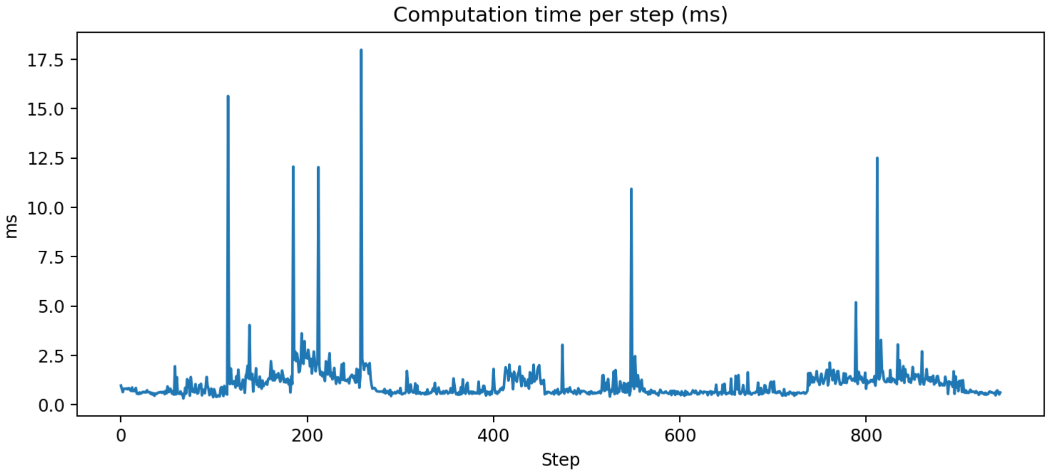


Fig. 3. Computation time per step

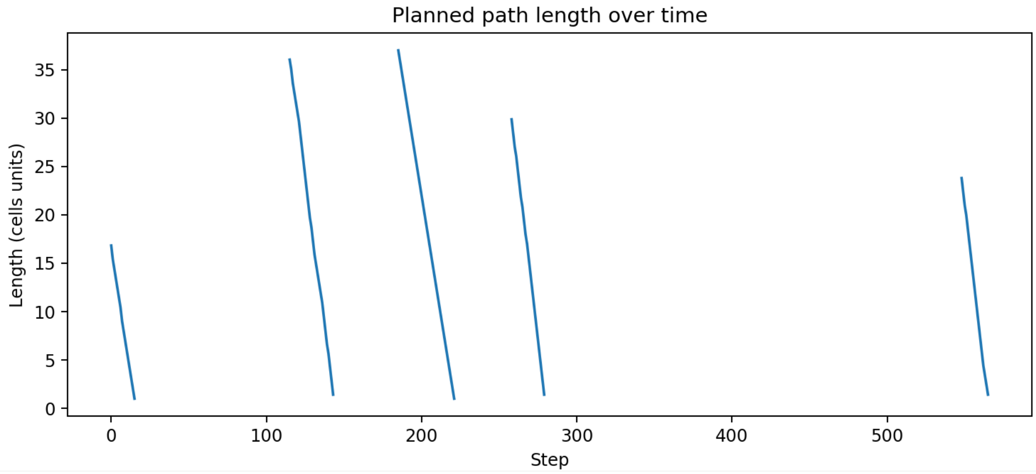


Fig. 4. Planned path length over time

```
==================== SUMMARY ====================
Last goal : (22, 39)
Total steps: 127
Executed path length: 150.196
Replans: 949
Mean step ms: 1.072
Max  step ms: 17.990
Dynamic cell changes (total): 8506
=================================================
```

Fig. 5. Results of numerical simulation calculations

The simulation results (Fig. 5) demonstrate that the mobile platform successfully reached the final goal (22, 39)(22, 39)(22, 39), completing 127 movement steps with a total trajectory length of 150,196 cells, which corresponds to the complex geometry of the environment with constraints and the need to avoid obstacles. The average calculation time of one step is 1,072 ms, which indicates the high computational efficiency of the algorithm and its suitability for real-time operation, while the maximum value of 17,990 ms corresponds to moments of intensive replanning during significant

ISSN 2071-1077(print)
ISSN 2663-2411(online)

changes in the occupancy map. The total number of replannings, which is 949, and a significant number of cell state changes (8,506) confirm the active dynamics of the environment and at the same time demonstrate the ability of the algorithm to stably adapt the route without losing convergence. Qualitative analysis confirms that the trajectory is formed adaptively taking into account the current state of the environment, ensuring safe and continuous movement of the platform to the target point. This confirms the effectiveness of the developed mathematical software and its suitability for use in intelligent logistics robotic systems.

4. Conclusions

As a result of the research, mathematical and software support for the a mobile platform route constructing method in a dynamic environment using QR tags was developed. It allows to formalize the navigation process based on a discrete occupancy map and the incremental D* Lite algorithm. The obtained numerical simulation results confirmed the operability of the proposed approach and its ability to provide adaptive route replanning in a changing environment. The mobile platform successfully reached the target point after completing 127 steps with a total trajectory length of 150,196 cells, which corresponds to a complex configuration of the workspace. It was established that the average calculation time of one step is 1.072 ms, which ensures the possibility of the system functioning in real time, and the maximum values of the calculation time are associated with local route replanning when changing the obstacle configuration and do not affect the overall stability of the algorithm. A significant number of changes in the state of map cells and route replanning confirms the high adaptability of the method to dynamic changes in the environment, while the algorithm provides effective updating of only locally changed areas of space, which allows to significantly reduce computational costs. Qualitative analysis of the constructed trajectories showed that the routes are formed through topologically optimal areas of free space and ensure safe avoidance of obstacles without collisions, which confirms the correctness of the developed mathematical model and the effectiveness of its software implementation. The results obtained confirm that the use of QR tags as a source of coordinates of target points allows for flexible and operational control of the mobile platform without the need for prior route programming, which increases the level of system autonomy. The developed method can be used as a basis for creating intelligent control systems for mobile robotic platforms in logistics and production environments. A promising direction for further research is the integration of dynamic obstacle motion prediction methods, the use of sensor data fusion, and the application of artificial intelligence methods to increase the efficiency of route planning and ensure a higher level of autonomy of mobile robotic systems.

Conflict of interest

The authors declare that they have no conflict of interest, in particular financial, personal, authorial or any other nature, which could affect the research, as well as the results published in this article.

Funding

Funding was provided within the framework of the implementation of the state theme "Hardware and software complex for detection and neutralization of explosive objects based on intelligent robotic platforms" at the Department of Computer

 ISSN 2071-1077(print)
ISSN 2663-2411(online)

Integrated Technologies, Automation, Robotics and Safety Engineering (CITARBI), Kharkiv National University of Radio Electronics.

References

1. Flores Gonzalez, J. M., Coronado, E., & Yamanobe, N. (2025). ROS-Compatible Robotics Simulators for Industry 4.0 and Industry 5.0: A Systematic Review of Trends and Technologies. Applied Sciences, 15(15), 8637. https://doi.org/10.3390/app15158637.
2. Alonso, R., Sánchez, T. F., Alfaro, D. A., Cruz, Y. J., Villalonga, A., & Castaño, F. (2025). Automation and robotics pilot lines in the context of industry 5.0. Applied Sciences, 15(5), 2510. https://doi.org/10.3390/app15052510.
3. Murthy, M. N., Lakshminarayanan, S., & Alagarsamy, S. (2026). Nurse training in the era of Industry 5.0: Evidence from a systematic literature review (2015–2024). Nurse Education in Practice, 92, 104769. https://doi.org/10.1016/j.nepr.2026.104769.
4. Nevliudov, I., Yevsieiev, V., Maksymova, S., Gopejenko, V., & Kosenko, V. (2025). Development of mathematical support for adaptive control for the intelligent gripper of the collaborative robot manipulator. Advanced Information Systems, 9(3), 57–65. https://doi.org/10.20998/2522-9052.2025.3.07.
5. Yao, L., Li, G., Zhang, T., Hussien, A. G., & Lu, Y. (2026). Adaptive multi-step path planning for multi-robot in dynamic environments based on hybrid optimization approach. Expert Systems with Applications, 298, 129699. https://doi.org/10.1016/j.eswa.2025.129699.
6. Wang, Y., & Wang, G. (2026). An adaptive and efficient path planning algorithm for UAV navigation in complex environments. Computers & Operations Research, 185, 107296. https://doi.org/10.1016/j.cor.2025.107296.
7. Pang, Z., Qian, L., Hong, M., & Li, D. (2026). An intelligent route planning algorithm integrating Q-learning and A* for dynamic marine environments. Journal of Marine Science and Technology, 1-28. https://doi.org/10.1007/s00773-026-01107-0.
8. Choi, J., & Choi, Y. (2025). Path Planning for Unmanned Aerial Vehicle: A-Star-Guided Potential Field Method. Drones, 9(8), 545. https://doi.org/10.3390/drones9080545.
9. Nevliudov, I., Omarov, M., Yevsieiev, V., Maksymova, S., & Jabrayilzade, E. (2026). Mathematical modeling of trajectories construction, movement of the gripping device of a collaborative robot. Advanced Information Systems, 10(1), 11–20. https://doi.org/10.20998/2522-9052.2026.1.02.
10. Akram, M. W., Sood, K., Hassan, M. U., & Thiruvady, D. (2026). ALFA: A Safe-by-Design Approach to Mitigate Quishing Attacks Launched via Fancy QR Codes. arXiv preprint arXiv:2601.06768. https://doi.org/10.48550/arXiv.2601.06768
11. Rios, O., Gauci, M. O., Debortoli, C., Alunni, V., & Savoldelli, C. (2026). A 3D-Printed Craniofacial Fracture Library with QR-Integrated Pedagogical Content: A technical note for Maxillofacial Trauma Teaching. Journal of Stomatology Oral and Maxillofacial Surgery, 102736. https://doi.org/10.1016/j.jormas.2026.102736.
12. Improvement of the Encoding Information Method for Pharmaceutical Products QR-Codes During Sorting on a Robotic Conveyor Line / I. Nevliudov, O. Klymenko, V. Yevsieiev, S. Maksymova. // Bulletin of the National Technical University "KhPI", series: Techniques in a machine industry. – 2025. – No. 1(11). – P. 128-134. – DOI : https://doi.org/10.20998/2079-004X.2025.1(11).15.
13. Liao, J. W., Wang, W., Wang, T. S., Peng, L. X., Weng, J. H., Chou, C. F.,

ISSN 2071-1077(print)
ISSN 2663-2411(online)
& Chen, J. C. (2025, February). DiffQRCoder: Diffusion-based aesthetic qr code generation with scanning robustness guided iterative refinement. In 2025 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV) (pp. 5916-5925). IEEE. https://doi.org/10.1109/WACV61041.2025.00577.

14. Shen, Y., Shen, Y., Zhang, Y., Huo, C., Shen, Z., Su, W., & Liu, H. (2025). Research Progress on Path Planning and Tracking Control Methods for Orchard Mobile Robots in Complex Scenarios. Agriculture, 15(18), 1917. https://doi.org/10.3390/agriculture15181917.

15. Shanmugaraja, M., Thangamuthu, M., & Ganesan, S. (2025). Hybrid Path Planning Algorithm for Autonomous Mobile Robots: A Comprehensive Review. Journal of Sensor and Actuator Networks, 14(5), 87. https://doi.org/10.3390/jsan14050087

16. Jin, P., Li, W., Zhan, J., & Shan, C. (2026). 3D Obstacle Avoidance Path Planning Algorithm and Software Design for UUV Based on Improved D* Lite-APF. Journal of Marine Science and Engineering, 14(4), 373. https://doi.org/10.3390/jmse14040373.

17. Wu, H., Zhong, Y., Zhang, K., Zhou, L., Zhang, Y., Mei, X., ... & Sun, T. (2026). Improved A∗+ D∗ Lite hierarchical framework path planning algorithm for heterogeneous sea ice scenarios. Ocean Engineering, 343, 123387. https://doi.org/10.1016/j.oceaneng.2025.123387.

18. Mlinarček, D., Jánoš, R., Málik, M., Svetlík, J., Semjon, J., & Ondočko, Š. (2026). Costmap Tuning for Autonomous Navigation: A Simulation and Real-World Study on the Hiwonder JetAcker. Applied Sciences, 16(4), 1923. https://doi.org/10.3390/app16041923.

19. Liang, Z., Wang, L., Wang, H., Zhang, B., & Liu, C. (2026). Autonomous obstacle avoidance and path planning for mobile robots in orchard environments combining with map construction and positioning methods. Computers and Electronics in Agriculture, 244, 111514. https://doi.org/10.1016/j.compag.2026.111514.

20. Building a Route for a Mobile Robot Based on the BRRT and A*(H-BRRT) Algorithms for the Effective Development of Technological Innovations / Amer Abu-Jassar, Hassan Al-Sukhni, Yasser Al-Sharo, S. Maksymova, V. Yevsieiev, V. Lyashenko // International Journal of Engineering Trends and Technology. – 2024. – V. 72(11). – P. 294-306. https://doi.org/10.14445/22315381/IJETT-V72I11P129.

21. Yuan, C., Zhao, K., Kuruoglu, E. E., Wang, L., Xu, T., Huang, W., ... & Rong, Y. (2025). A survey of graph transformers: Architectures, theories and applications. arXiv preprint arXiv:2502.16533. https://doi.org/10.48550/arXiv.2502.16533.

22. Ye, J., Niu, L., Xu, X., & Wang, Y. (2025, June). Indoor navigation path planning method based on improved D* Lite algorithm for complex environments. In IET Conference Proceedings CP931 (Vol. 2025, No. 23, pp. 94-100). Stevenage, UK: The Institution of Engineering and Technology. https://doi.org/10.1049/icp.2025.2854.

23. Yadav, H. (2025). Evaluating A* Heuristics Across Diverse Map Types for Robotic Navigation. Authorea Preprints.

24. Kahar, M. A. B., Bingi, K., Ibrahim, R., & Omar, M. (2025, June). Review on Global Path Planning Algorithms for Autonomous Navigation of Multiple Mobile Robots. In 2025 IEEE International Conference on Automatic Control and Intelligent Systems (I2CACIS) (Vol. 1, pp. 202-207). IEEE. https://doi.org/10.1109/I2CACIS65476.2025.11100872.

25. Abu-Jassar, A. T., Attar, H., Amer, A., Lyashenko, V., Yevsieiev, V., & Solyman, A. (2025). Development and Investigation of Vision System for a Small-Sized Mobile Humanoid Robot in a Smart Environment. International Journal of Crowd

ISSN 2071-1077(print)
ISSN 2663-2411(online)

Science, 9(1), 29-43. https://doi.org/10.26599/IJCS.2023.9100018.

26. I. Nevliudov, V. Yevsieiev, S. Maksymova, N. Demska, K. Kolesnyk, Olha Miliutina, "Mobile Robot Navigation System Based on Ultrasonic Sensors", 2023 IEEE XXVIII International Seminar/Workshop on Direct and Inverse Problems of Electromagnetic and Acoustic Wave Theory (DIPED), vol.1, pp.247-251, 2023. https://doi.org/10.1109/DIPED59408.2023.10269500.

27. Moumoula, M. B., Kabore, A. K., Klein, J., & Bissyande, T. F. (2025). Evaluating programming language confusion. arXiv preprint arXiv:2503.13620. https://doi.org/10.48550/arXiv.2503.13620.

28. Rouhandeh, H., & Behroozsarand, A. (2025). Simulation and optimization of methanol production process via bi-reforming of methane: A novel genetic algorithm-based approach in Python. International Journal of Hydrogen Energy, 101, 1161-1171. https://doi.org/10.1016/j.ijhydene.2025.01.003.

29. Yevsieiev, V., Gurin, D., Kulish, S., & Voloshyn, Y. (2025). Development of a partially supervised Markov decision-making model for a 3-link collaborative robot-manipulator. Radioelectronic and Computer Systems, 2025(4), 83-94. doi: https://doi.org/10.32620/reks.2025.4.06.

Надійшла до редакції 5.02.2026, розглянута на редколегії 14.04.2026

Чисельне моделювання побудови маршруту мобільної роботизованої платформи в динамічному просторі з використанням QR-міток

У сучасних умовах розвитку Industry 5.0 мобільні роботизовані платформи відіграють ключову роль у забезпеченні автоматизації логістичних і виробничих процесів, що потребує розроблення ефективних методів навігації в динамічному середовищі зі статичними та рухомими перешкодами. Особливої актуальності набуває задача адаптивного формування маршруту переміщення на основі сенсорної інформації, зокрема QR-міток, які дозволяють оперативно визначати координати цільових точок і забезпечують гнучкість керування мобільною платформою.

Об'єктом дослідження є процес переміщення мобільної платформи у дискретному динамічному робочому просторі.

Предметом дослідження є метод побудови маршруту мобільної платформи з використанням інкрементального алгоритму перепланування та моделі визначення координат цілі на основі QR-міток.

Метою дослідження є розроблення математичної моделі та програмної реалізації методу побудови оптимального маршруту мобільної платформи в динамічному середовищі з урахуванням змін конфігурації простору та оперативного визначення координат цільових точок.

У дослідженні використано методи математичного моделювання, теорії графів, чисельного інтегрування, алгоритмів інкрементального пошуку шляху та комп'ютерного моделювання з використанням дискретної карти зайнятості. Наукова новизна роботи полягає у розробленні математичного забезпечення методу побудови маршруту мобільної платформи з інтеграцією моделі спостереження цілі на основі QR-міток, що дозволяє забезпечити адаптивне перепланування траєкторії в режимі реального часу при зміні стану середовища.

ISSN 2071-1077(print)
ISSN 2663-2411(online)

Отримані результати чисельного моделювання підтверджують ефективність запропонованого методу, який надає безпечне переміщення мобільної платформи, адаптивне обходження перешкод та стабільне досягнення цільових точок при високій обчислювальній ефективності. Розроблена модель демонструє потенціал для впровадження в інтелектуальні системи керування мобільними роботами та може бути використана як основа для подальшого вдосконалення алгоритмів навігації в умовах динамічного середовища.

Ключові слова: мобільна роботизована платформа, планування маршруту, динамічне середовище, алгоритм D* Lite, карта зайнятості, чисельне моделювання, QR-мітки, інкрементальне перепланування, автономна навігація, Industry 5.0.

Відомості про авторів:

Невлюдов Ігор Шакирович – д-р техн. наук, проф., зав. каф. комп'ютерно-інтегрованих технологій, автоматизації, робототехніки та безпекової інженерії (КІТАРБІ), Харківський національний університет радіоелектроніки, Харків, Україна, e-mail: igor.nevliudov@nure.ua, ORCID: 0000-0002-9837-2309.

Євсєєв Владислав В'ячеславович – д-р техн. наук, проф., проф. каф. комп'ютерно-інтегрованих технологій, автоматизації, робототехніки та безпекової інженерії (КІТАРБІ), Харківський національний університет радіоелектроніки, Харків, Україна, e-mail: vladyslav.yevsieiev@nure.ua, ORCID: 0000-0002-2590-7085.

Максимова Світлана Святославівна – к-т. техн. наук, доц., доц. каф. комп'ютерно-інтегрованих технологій, автоматизації, робототехніки та безпекової інженерії (КІТАРБІ), Харківський національний університет радіоелектроніки, Харків, Україна, e-mail: svitlana.milyutina@nure.ua, ORCID: 0000-0002-1375-9337.

Теслюк Сергій Ігорович – ст. викл. каф. комп'ютерно-інтегрованих технологій, автоматизації, робототехніки та безпекової інженерії (КІТАРБІ), Харківський національний університет радіоелектроніки, Харків, Україна, e-mail: serhii.tesliuk@nure.ua, ORCID: 0000-0003-0711-9250.

About the authors:

Igor NEVLIUDOV – Doctor of Technical Science, Professor, Head of Department of Computer-Integrated Technologies, Automation, Robotics and Safety Engineering (CITARSE), Kharkiv National University of Radio Electronics, Kharkiv, Ukraine, e-mail: igor.nevliudov@nure.ua, ORCID: 0000-0002-9837-2309.

Vladyslav YEVSIEIEV – Doctor of Technical Science, Professor, Professor Department of Computer-Integrated Technologies, Automation, Robotics and Safety Engineering (CITARSE), Kharkiv National University of Radio Electronics, Kharkiv, Ukraine, e-mail: vladyslav.yevsieiev@nure.ua, ORCID: 0000-0002-2590-7085.

Svitlana MAKSYMOVA – Candidate of Technical Sciences, Associate Professor, Associate Professor of Computer-Integrated Technologies, Automation, Robotics and Safety Engineering (CITARSE), Kharkiv National University of Radio Electronics, Kharkiv, Ukraine, e-mail: svitlana.milyutina@nure.ua, ORCID: 0000-0002-1375-9337.

Serhii TESLIUK – Senior Lecturer, Department of Computer-Integrated Technologies, Automation, Robotics and Safety Engineering (CITARSE), Kharkiv National University of Radio Electronics, Kharkiv, Ukraine, e-mail: serhii.tesliuk@nure.ua, ORCID ID: 0000-0003-0711-9250.