# ALGORITHM FOR THE OPTIMAL TRAJECTORY OF OBJECTS MONITORING

**Anonymous authors**
Paper under double-blind review

## ABSTRACT

The article considers the task of developing a system for modeling the trajectories of a robot monitoring in a closed area from the ground: inspection of premises, survey of industrial facilities, assessment of the condition of fruit trees, etc. During monitoring, some obstacles may arise in the robot's path: temporary and removable (people, small furniture, household appliances, emergency zone) or permanent (walls, permanently installed equipment, fixed furniture). The robot is controlled using a program developed by the authors, which stores a database of routes successfully overcome by the robot earlier, for the most accurate determination of the trajectory of the robot, taking into account obstacles encountered along the constructed path. The article considers an algorithm for constructing a graph model of the monitoring area for the subsequent search for the shortest route of the robot, which consists in discretization the area, identifying possible ways to move the robot, analyzing existing obstacles and setting distances between the objects of study. It is shown that after obtaining such a graph, it is possible to apply one of the algorithms for finding the Hamiltonian path in a graph. It connects the vertices of the graph corresponding to the monitoring points. The result of applying the algorithm will be the shortest route of the robot, or a message about the impossibility of monitoring (partial or complete insolubility of the problem) if a number of obstacles do not allow you to plot a route passing through some (or all) vertices. Choosing the most appropriate algorithm, as well as using artificial intelligence methods to classify obstacles and make adjustments to routes are areas for further research.

## 1 INTRODUCTION

Monitoring and analyzing the condition of various land-based facilities is an urgent task in many fields of activity from industry to agriculture. The use of human resources in this case is often difficult or impossible, for example, when examining several hectares of fruit tree plantings or crops, when monitoring a large area indoors or with environmental conditions harmful to humans, when protecting residential premises, examining infectious disease departments of hospitals, etc.

In all the cases discussed above, there are a number of obstacles in the surveyed space, which can be divided into permanent (trees, structures, fixed furniture) and temporary (people, animals, cargo). The device (robot) conducting monitoring must take these obstacles into account when calculating the optimal way to conduct monitoring, and plan its movement route taking into account these obstacles.

The known approaches in common do not consider any obstacles. The classical statement of the task takes into account the number of objects to be visited. In Sundarrajan et al. (2023) approach to solve the coverage problem as a TSP by incorporating tree selection algorithms to improve efficiency and accuracy. This approach represents the study area as a graph, with each vertex represents a coverage point and the edges are the distances between them. To reduce computational complexity, the authors use various tree selection algorithms to select the best possible starting point for the TSP algorithm. The simulation results show that, compared to conventional TSP algorithms, the suggested technique greatly increases coverage quality while decreasing computing costs. This approach is rather good but it is used to solve the task for drone, so the presence of obstacles is not considered there.

Some ideas to find the path passing through all points of interest is deeply studied in Chentsov (2023) and Chentsov & Chentsov (2022) where the authors consider so-called megalopolis (a nonempty finite set) in order to perform some work. This work in terms of our research may be monitoring held. The choice of the order in which the tasks are performed is subject to precedence conditions, which are localized for the mentioned two subsets of the total set of tasks. The cost functions used in the formation of the additive criterion may involve a dependence on the list of tasks. To construct a solution, a two-stage procedure based on dynamic programming is proposed. Nevertheless, these algorithms do not consider obstacles neither constant nor moving. The effectiveness of this approach may be examined if obstacles are considered to be the megalopolis as objects under monitoring but the results obtained in these articles do not consider the dynamical situation, so we need an approach to take into account this type of obstacles.

One more approach for solving the same task are considered in Wu et al. (2023) but this article also does not take obstacles into account.

Articles considering obstacles, like Yong et al. (2024), concentrate on physical aspects, for example, searching for a turning point based on free segments, nevertheless, the objectives are the same – to define the shortest safe path.

The considered in this paper task has common points with the so-called drone-and-truck problem Chung et al. (2020), which also considers a monitoring device (drone) and a device for drone maintenance (truck). The differences in the considered task are that the robot must be able to overcome an obstacle, and in some cases a truck is not required.

This paper considers an algorithm for constructing the shortest route for a robot monitoring from the ground. The robot is controlled using a program developed by the authors, which stores a database of routes successfully overcome by the robot for the most accurate determination of the trajectory of the robot, taking into account obstacles encountered in its path. This work is a bit similar to Cannon et al. (2012) where real-time heuristic search algorithms for kinodynamic motion planning with dynamic obstacles are presented. But the task formalization and algorithms differ, and this paper be a good analogue for compare in the future research.

## 2 STATEMENT OF THE PROBLEM AND CONSTRAINTS

Let it is necessary to monitor a group of objects $M_i, i = 1, \dots K$ located in some area $F$. The area $F$ is a polygon, generally non-convex. The coordinates of the $N$ corners of the polygon $(x_j, y_j)$, $j = 1, \dots, N$ are given as a vector, the coordinates are indicated clockwise, starting with the smallest. The objects $M_i$ are entirely located in the area of $F$. The distances $L(M_i, M_j)$, $i, j = 1, \dots, K$, $i \neq j$ between all pairs of objects are known. The task is to monitor these objects using a minimum of resources of the monitoring device (robot or drone). In this case, resources are understood as: battery consumption (in general, recharging is required during monitoring), the size of the memory card (in general, the amount of memory for monitoring all objects is not enough and a replacement card is required). Recharging the battery or replacing the memory card can be held out both on site and after returning the monitoring device to its parking lot. In the first case, a so-called ¡¡truck¿¿ will be required, for which a route for the delivery of a battery or memory card is being built. In the second case, it is necessary to take into account the time to return to the parking lot before the battery charge is consumed and build the shortest route.

Thus, the monitoring device, let's call it ¡¡robot¿¿, is located in the parking lot with coordinates $(P_x, P_y)$. The task is to get the optimal route the robot from the point $(P_x, P_y)$, called parking, to carry out the work (monitoring) of objects $M_i$, $i = 1, \dots, K$. In general, objects $M_i$ are defined by the coordinate vectors of the polygons bounding them. The purpose of monitoring is to obtain images using a robot to study the condition of objects $M_i$ and carry the further analysis.

Ideally but not realistic, monitoring is carried out without constraints. To solve such a problem, it is enough to consider one of the special cases of solving the travelling salesman problem (TSP) Petunin et al. (2022). However, when moving between objects, various kinds of obstacles may arise that the robot cannot overcome for certain reasons.

Obstacles $S_i, i = \overline{1, s}$ can be divided into movable and stationary ones. Movable ones include people and animals, dead animals can also be included here, as well as equipment (cars, tractors, etc.) left somewhere on the way and temporary storage (for example, a haystack, a container with parcels). These are *temporary* obstacles, information about the location of which should not be stored and used for analysis when building a trajectory in the future. The remaining obstacles are *stationary* or sedentary (new walls, structures, furniture). The presence of these obstacles is recorded in the database for route analysis, since their location is permanent or long-term.

To define the trajectory of the robot, it is necessary to solve the following tasks.

1. Create a database of routes: ideal (without any obstacles), taking into account the impact of external factors (temperature, wind force, etc.), routes with insurmountable obstacles (only routes with stationary obstacles such as fences, furniture, etc., routes with movable obstacles (people, animals) are added to the database they are not added to the database). By an obstacle, we will understand any object that stands in the way of the robot's movement and prevents further following a given route.

2. Determine the solvability of the routing problem (is it possible to build a route for monitoring objects of interest in the presence of existing restrictions and obstacles).

In general, there are $N$ objects for monitoring, the coordinates of each of them are known. The photo or video of this object is taken and then is being sent to the server for further analysis and decision making. The robot takes pictures at the specified coordinates and moves on to the next object. When surveys have been carried out at all specified coordinates or the impossibility of further monitoring has been revealed, the robot returns to the parking lot.

If the work resource is exhausted, it either returns to the parking lot (it is necessary that there is enough resource to return), or sends a message to the operator about the need to replace the obstacle. In the latter case, it is necessary to solve the problem of building a route for a service ¡¡truck¿¿ delivering replenished resources.

## 3 ROBOT ROUTING ALGORITHM SCHEME

The diagram of the robot's motion algorithm during monitoring is shown in figure 1. This diagram reflects not only the sequence of actions during monitoring, but also the stages of project implementation.

In this diagram, it is assumed that the route $P_i$ already takes into account all external influences, and it takes into account all recharging of the robot and changing the memory card if the resource is not enough to complete a full monitoring cycle. Movement by $P_i$ means moving the robot one position along the route. It is believed that the entire space explored by the robot is divided into elementary segments of the same length and width.

The opportunity to continue driving is considered to be the opportunity to reach the central point of the next elementary segment. If an obstacle is found on the robot's path to this point, further continuation of movement along the previously selected path is considered impossible and a detour route is searched. If at one of the monitoring stages it is not possible to find a workaround route, the robot sends a message about the impossibility of monitoring to the operator and shuts down (returns to the parking lot for recharging).

Thus, based on the constructed algorithm scheme, the following tasks can be determined.

1. Constructing a route $P_i$ taking into account external influences.

2. Determination of the possibility of continuing moving using computer vision algorithms.

3. Search for a detour route, taking into account external influences and existing obstacles.

4. Updating information in the database about obstacles and detours.

The main database tables are shown in the table 1.

### 3.1 DISCRETIZATION OF THE AREA

All the area used for robot moving is divided into small square segments using discretization algorithm Makarovskikh et al. (2024).

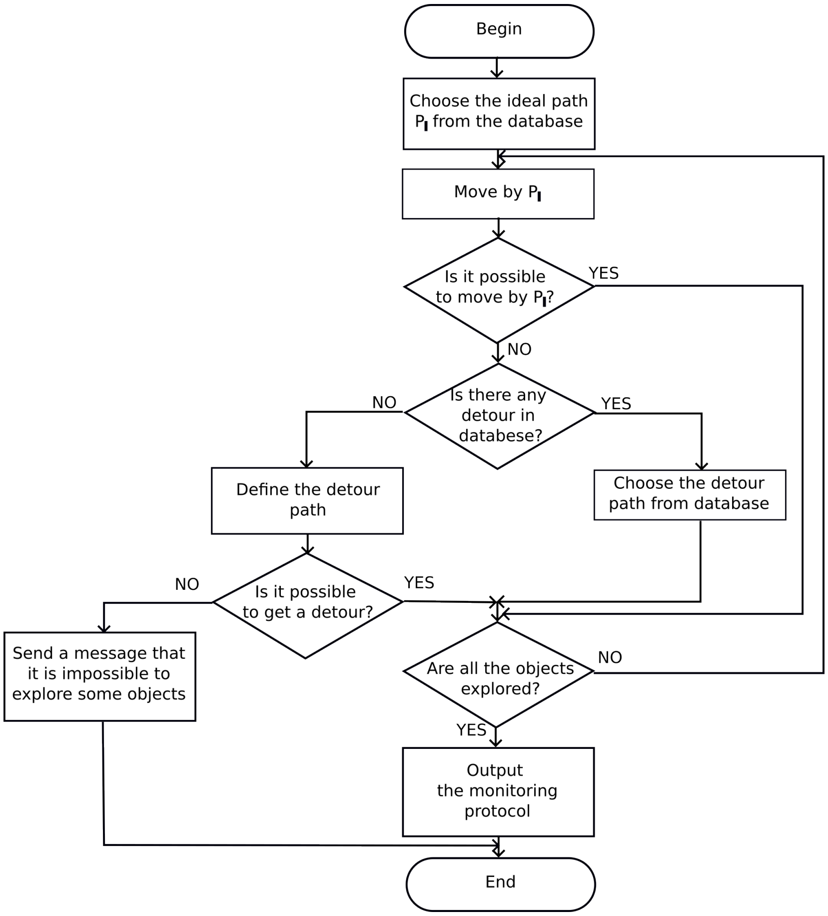


Figure 1: The scheme of the algorithm for moving the robot in the monitoring area

Table 1: Description of the database tables

| Table | Description |
|---|---|
| Users | List of robot users |
| UserRoles | A list of user roles indicating the ID of the space in which the user with the role ID has access to launch the robot |
| Roles | A list of user roles and a description of the access rights corresponding to each role |
| Areas | List of areas for monitoring |
| Robots | Information on available monitoring devices |
| AreaRobotsHome | Parking spots of a robot with a specific ID in an area with a given ID |
| Routes | Routes stored in the database |
| RoutsPath | A coordinator who saves the route points taking into account obstacles, as well as the status for each route point, so that in the future he knows possible obstacles on the route. The table shows the expected time to complete the route, taking obstacles into account |
| RobotsObjectionPath | Fragments of the route that take into account temporary obstacles |

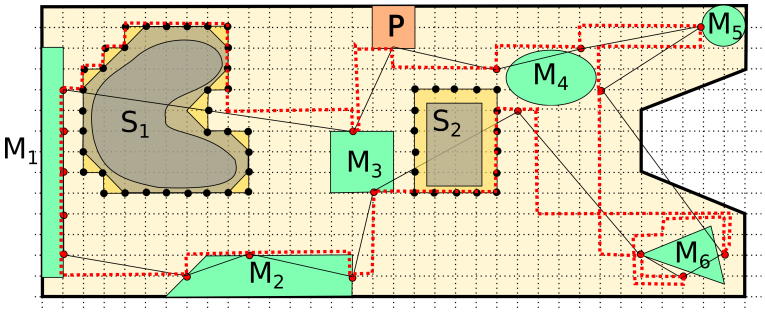


Figure 2: An example of monitoring map

**Algorithm DISCRETIZATION**

**Input:** The coordinates of the corners of the polygon $F_i$ on the map selected by the user.

**Output:** The vector $R_c$ of the rectangles $R_j$, $j = 1, \dots M$ centers available for moving.

**Step 1.** Find the minimum and maximum values of $x_{min}$, $x_{max}$ and $y_{min}$, $y_{max}$ using the given coordinates of the polygon $F_i$.

**Step 2.** Iterate over the values of $x$ and $y$ within the minimum and maximum values in increments equal to the width and length of the survey area.

**Step 3.** Calculate the coordinates of the corners of the rectangle $R_j$, $j = 1, \dots M$ (where $M$ is the number of rectangles) for each position.

**Step 4.** If at least one corner of the rectangle $R_j$, $j = 1, \dots M$ is inside the polygon $F_i$, then calculate the coordinates of the center of the rectangle $R_j$.

**Step 5.** Add the coordinates of the center of the rectangle $R_j$ to the resulting vector $R_c$.

**Step 6.** Repeat steps 2–5 until all $R_j$ are processed.

**Step 7.** Return the resulting vector $R_c$ containing the coordinates of the centers of the rectangles intersecting with the polygon.

The length of each tile size is equal to the distance of robot moving between the information exchange moments. Usually this value is 30–50 cm. Let this value is equal to $l$. Robot can move between the corners of a tile horizontally or vertically (then the way length between 2 points is equal to $l$), or diagonally (in this case robot moves $\sqrt{2}l$ before the next information exchange).

After finishing the discretization we can get the plane graph $G = (V, E)$ corresponding to the investigated area, the set of vertices $V$ of which be the points where robot exchanges the information with the applied software, and edges $E$ be the connections between the nearest neighbours. As soon as points of discretization are placed in distance less than 50 cm, the number of such points is rather big, so $|V|$ has thousands of vertices. Hence, it is unreasonable to use all these vertices to define the optimal route. Besides, this graph is sparse, because $\forall v \in V\colon 2 \leq \deg(v) \leq 8$, and $|V|$ be equal to $10^3 - -10^7$ vertices. The edges $e \in E$ weights be equal $l$ or $\sqrt{2}l$. To save the data on such a graph we may use the adjacency list for each vertex.

For the practical use let's divide all the graph vertices $V = V_p \cup V_m \cup V_d \cup V_t$ into the following subsets:

- parking vertices ($p_i \in V_p$) be the vertices where monitoring device has some service;
- monitoring vertices ($m_i \in V_m$) be the vertices where device holds monitoring;
- detour vertices ($d_i \in V_d$) be the vertices belonging to a detour route around an obstacle;
- transit vertices $v \in V_t$ be all the other vertices of a graph that can be included to a route.

Let us consider an example in figure 2. In this figure we have a non-convex area bounded by the bold line. Vertices belonging to boundary are not used in the route; $P$ be a parking vertes; $M_1 \dots M_6$ be

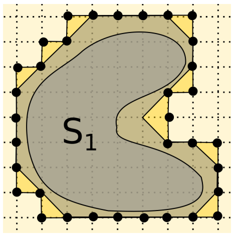


Figure 3: An example of covering the obstacle

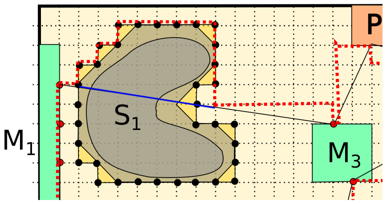


Figure 4: A segment of path goes through the obstacle

the objects of monitoring; $S_1, S_2$ be some obstacles. Vertices marked by red are points (vertices) of monitoring, black vertices be the detour vertices. Transit vertices are not specified in this figure as soon as their number is large (they are places in the crosses of straight lines used for discretization). The same designations are used for all the further examples.

### 3.2 ROUTING ALGORITHM

The task of obtaining the optimal route for holding the monitoring can be divided into the following steps.

**Step 1.** Define the ideal route $P_{ideal}$ which is the minimal length Hamiltonial cycle in a subgraph $G_M = (V_M, E_M), V_M = \{v | v \in V_m \cup V_p\}$. The weights $w(e)$ of $e \in E$ are defined as path lengths using the city metric for the defined discretization step. It is a thick black line for graph in figure 2. There are lots of known algorithms to solve this task. Let us use the nearest neighbour method.

**Step 2.** Define the detour routes for each obstacle $S_i$. At the current stage we suppose that all the obstacles are placed on the map and their positions are known. Each $S_i$ is covered by square tiles of size equal to discretization step. So, each tile is attached to the transit vertices by its corners. The tile is included to a cover is it has any points inside of the obstacle contour. After all the tiles are defined the vertices belonging the outer contour of the obtained rectangle are placed to the set $V_d$. The example of building such a cover is shown in figure 3. The same approach is used for defining the detour routes for the inspected area. If an obstacle is a stationary one then its detour route is saved to a database. If it is a movable one, the detour route may be defined in real time using computer vision equipment of the monitoring device. In this case the algorithm of obtaining vertices $V_d$ will be different (it is a topic of a single research).

**Step 3.** Define the detour path for a robot. In this stage let us analyse the ideal route $P_{ideal}$ and get the real route $P_real$ taking in account all the obstacles and moving through $v \in V_t$. If there is a segment of route belonging to the obstacle (like in figure 4) then let us replace this segment of a route by a shortest path consisting only of vertices $v \in V_d$. In this step we also define the solvability of monitoring task. Example, shown in figure 2 this task is solvable, and red point line shows the route for monitoring. In figure 5 we see the partially solvable task. Object $M_6$ cannot be included to the monitoring route as soon as all the ways to is are closed by obstacle $S_3$. If $S_3$ is the stationary one then $M_6$ can never be inspected. If it is moving one, then some decisions for evacuation of this obstacle need to be made. Here we have the red dotted route passing through all the monitored objects except of $M_6$.

Unsolvable task is shown in figure 6. The same reasoning on obstacle $S_3$ as above is true in this

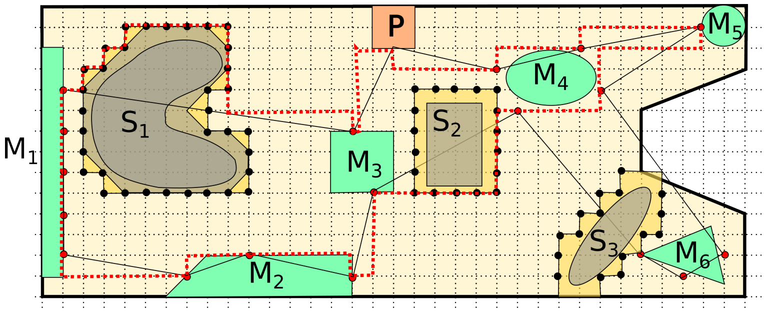


Figure 5: An example of monitoring map with partially solvable routing task

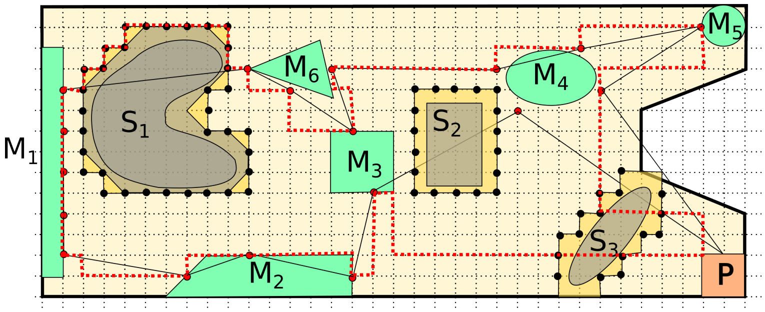


Figure 6: An example of monitoring map with unsolvable routing task

case.

Speaking on the computing complexity of the presented algorithm we see that it is defined by the computing complexity of algorithm used to solve TSP in step 1 of our approach. Step 2 can be held in polynomial time $O(|V|^2)$ because in the worst case we need to look through all the vertices of graph $G$. Even in this case the time we need to solve this task can be large because $|V|$ may contain more than $10^3$ elements. To speed up this process the parallel computing technologies may be used, but it is a topic of a separate research.

If an obstacle is a convex area then it is divided into 2 parts by the ideal path, so defining the shortest detour consists on checking only two paths. If obstacle is not convex area then the number of paths to be checked is a finite set as soon as $|V_{d_i}|$ for $S_i$ is a finite set.

## 4 SOFTWARE FOR MONITORING

So, the basic idea of the monitoring system works to track and inspect objects through robots and drones is shown in figure 7.

Hence, to control the monitoring device, the Robot Test and Receive Data application has been developed (figure 8), which is designed to control the robot (Send control panel) and transmit messages in console mode, communicate with the robot via WiFi via the robot's IP address. When the robot is turned on, data is transmitted via the ESP card. Data is transmitted on the distance between the robot and the nearest obstacle and the number of steps to reach this obstacle in a straight line, the file name of the obstacle photo, the coordinates of the obstacle and the resolution of the image. Information about the robot's status is also transmitted (moving, standing, parked). The received data is recorded in a CSV file. The robot is controlled on the Send control panel by analogy with using the control panel. While the mouse button is pressed on one of the control buttons, the robot moves according to the trajectory set by the operator. Information about the commands transmitted to the robot is transmitted to the console panel. The robot can also be controlled by sending commands in the console panel.

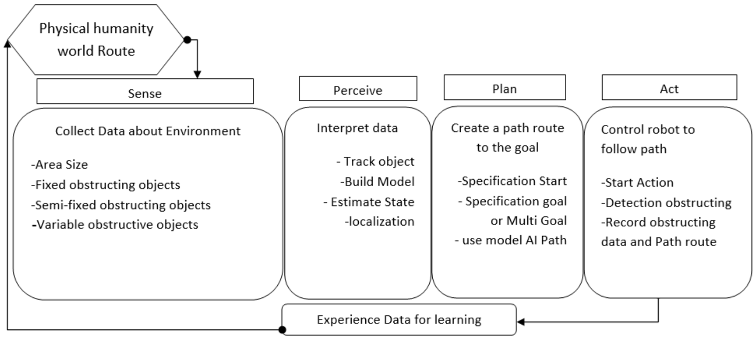
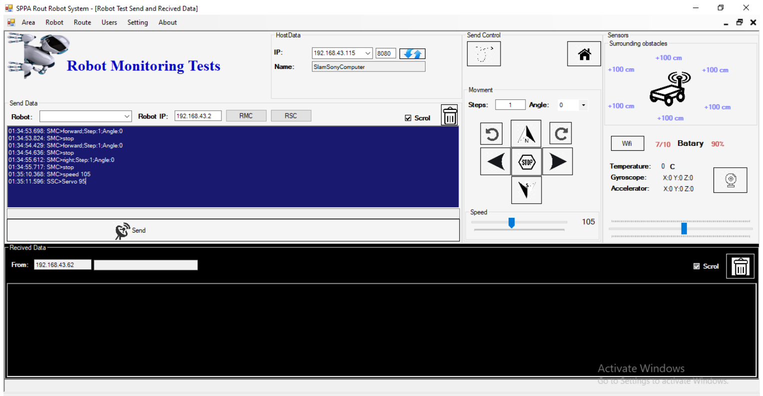


Figure 7: The basic idea of the system functioning



IP: 192.168.43.115 8080
Name: SlamSonyComputer
Robot:
Robot IP: 192.168.43.2
☑ Scrol

```
01:34:53.698: SMC>forward;Step:1;Angle:0
01:34:53.824: SMC>stop
01:34:54.429: SMC>forward;Step:1;Angle:0
01:34:54.636: SMC>stop
01:34:55.612: SMC>right;Step:1;Angle:0
01:34:55.717: SMC>stop
01:35:10.368: SMC>speed 105
01:35:11.596: SSC>Servo 95
```

Steps: 1
Angle: 0
Speed: 105
+100 cm
+100 cm
+100 cm
+100 cm
+100 cm
+100 cm
Wifi 7/10
Batary 90%
Temperature: 0 C
Gyroscope: X:0 Y:0 Z:0
Accelerator: X:0 Y:0 Z:0
From: 192.168.43.62
☑ Scrol

Figure 8: The starting window of the program for controlling a robot that monitors objects

## 5 CONCLUSION

In our article, we considered the task of planning the monitoring path with obstacles in the explored area. We defined the algorithm allowing improve the defined ideal path (that is the solution of TSP problem) according to existing stationary and movable obstacles. The information on stationary obstacles is saved to a database, and can be obtained in one moment. The movable obstacles are supposed to have the known placement on the map. This approach can be used for monitoring different objects from land.

The tasks of the future research are the following. First of all we need define the most suitable for the given task algorithm for defining the ideal path. The movable obstacles can appear dynamically. So, they can appear after defining the initial (ideal or not) route, and in this case algorithm needs use artificial intelligence unit to recognise the obstacle, define its contours and put it to the map. It is reasonable to save information on such obstacles in a database, but the information on their presence needs to be updated each predefined period of time. If obstacle is evacuated then it is placed to archive table, and extracted from it once it is recornized somewhere later.

## REFERENCES

J. Cannon, K. Rose, and W. Ruml. Real-time motion planning with dynamic obstacles. In *Vol. 3 No. 1 (2012): Fifth Annual Symposium on Combinatorial Search*, volume 3. 2012.

A.G. Chentsov. A bottleneck routing problem with a system of priority tasks. *Izvestiya IMI UdGU*, 61:156–186, 2023.

A.G. Chentsov and P.A. Chentsov. An extremal two-stage routing problem and procedures based on dynamic programming. *Trudy Instituta Matematiki i Mekhaniki UrO RAN*, 28(2):215—248, 2022.

S.H. Chung, B. Sah, and J. Lee. Optimization for drone and drone-truck combined operations: A review of the state of the art and future directions. *Computers & Operations Research*, 123, 2020.

T. Makarovskikh, A. Panyukov, M. Abotaleb, V. Maksimova, O. Dernova, and E. Raschupkin. Optimal route for drone for monitoring of crop yields. In *Olenev, N., Evtushenko, Y., Jaćimović, M., Khachay, M., Malkova, V. (eds) Advances in Optimization and Applications. OPTIMA 2023. Communications in Computer and Information Science*, volume 1913. Springer, Cham, 2024.

A.A. Petunin, A.G. Chentsov, and P.A. Chentsov. Optimal routing in problems of sequential bypass of megacities in the presence of restrictions. *Chelyab. Phys.-Math. journal*, 7:2:209–233, 2022.

M. Sundarrajan, A. Jothi, D. Prabakar, and S. Kadry. The smart coverage path planner for autonomous drones using tsp and tree selection. In *Kadry, S., Prasath, R. (eds) Mining Intelligence and Knowledge Exploration. MIKE 2023. Lecture Notes in Computer Science()*, volume 13924. Springer, Cham, 2023.

J. Wu, M. Li, C. Gao, Z. Liu, B. Zhang, and X. Zhang. Research on deployment scheme and routing optimization algorithm of distribution cable condition monitoring devices. *Energies*, 16, 2023.

P. D. Yong, Zh. Ya. Ya, C. Yue, Ze W. H. Hao, M. L., L. C. Ri, and Zhen Y. L. Animal science journal. *Path planning optimization for swine manure-cleaning robots through enhanced slime mold algorithm with cellular automata*, 95(1), 2024. doi: 10.1111/asj.13992.