DOI: 10.17587/mau.21.470-479

A. R. Gaiduk, D.Sc , gaiduk_2003@mail.ru, **O. V. Martjanov,** C.Sc , martyanovov@fpi.gov.ru,
M. Yu. Medvedev, D.Sc , medvmihal@sfedu.ru, **V. Kh. Pshikhopov,** D.Sc , pshichop@rambler.ru,
N. Hamdan, Postgraduate student, dr.nizar.abou.hamdane@gmail.com,
A. Farhood, Postgraduate student, azhar.kadhum@stu.edu.iq,
Southern Federal University, Shevchenko str. 2, Taganrog, Russia

Corresponding author: **Medvedev Mikhail Yu.,** D.Sc , Southern Federal University,
Shevchenko str. 2, Taganrog, Russian Federation, e-mail: medvmihal@sfedu.ru

Accepted on March 27, 2020

Neural Network Based Control System for Robots Group Operating in 2-d Uncertain Environment

Abstract

This study is devoted to development of a neural network based control system of robots group. The control system performs estimation of an environment state, searching the optimal path planning method, path planning, and changing the trajectories on via the robots interaction. The deep learning neural networks implements the optimal path planning method, and path planning of the robots. The first neural network classifies the environment into two types. For the first type a method of the shortest path planning is used. For the second type a method of the most safety path planning is used. Estimation of the path planning algorithm is based on the multi-objective criteria. The criterion includes the time of movement to the target point, path length, and minimal distance from the robot to obstacles. A new hybrid learning algorithm of the neural network is proposed. The algorithm includes elements of both a supervised learning as well as an unsupervised learning. The second neural network plans the shortest path. The third neural network plans the most safety path. To train the second and third networks a supervised algorithm is developed. The second and third networks do not plan a whole path of the robot. The outputs of these neural networks are the direction of the robot's movement in the step k. Thus the recalculation of the whole path of the robot is not performed every step in a dynamical environment. Likewise in this paper algorithm of the robots formation for unmapped obstructed environment is developed. The results of simulation and experiments are presented.

Keywords: *path planning, group control, neural network, machine learning, obstructed environment*

Acknowledges. Research is supported by Russian Science Foundation (grant 16-19-00001), and was executed in Southern Federal University.

For citation:

Gaiduk A. R., Martjanov O. V., Medvedev M. Yu., Pshikhopov V. Kh., Hamdan N., Farhood A. Neural Network Based Control System for Robots Group Operating in 2-d Uncertain Environment, *Mekhatronika, Avtomatizatsiya, Upravlenie,* 2020, vol. 21, no. 8, pp. 470—479.

DOI: 10.17587/mau.21.470-479

УДК 004.896

DOI: 10.17587/mau.21.470-479

А. Р. Гайдук, д-р техн. наук, gaiduk_2003@mail.ru,
О. В. Мартьянов, канд. техн. наук, martyanovov@fpi.gov.ru,
М. Ю. Медведев, д-р техн. наук, medvmihal@sfedu.ru, **В. Х. Пшихопов,** д-р техн. наук, pshichop@rambler.ru,
Н. Хамдан, аспирант, dr.nizar.abou.hamdane@gmail.com, **А. Фархуд,** аспирант, azhar.kadhum@stu.edu.iq
ФГАОУ ВО Южный Федеральный университет, Таганрог, Россия

Нейросетевая система управления группой роботов в неопределенной двумерной среде

Описана разработанная система управления группой мобильных роботов, использующая нейронные сети. Система управления выполняет оценку состояния среды функционирования, поиск оптимального метода планирования пути, планирование пути, коррекцию траекторий движения по результатам взаимодействия роботов группы. Выбор оптимального метода планирования и планирование пути группы роботов реализуется тремя нейронными сетями глубокого обучения.

Первая нейронная сеть классифицирует состояние среды на два класса. Для первого класса оптимальным методом планирования пути является метод построения кратчайшего пути. Для второго класса оптимальным методом является метод планирования безопасного пути. Выбор метода планирования пути базируется на составном критерии, который включает в себя время движения в целевую точку, длину пути и минимальное расстояние от роботов группы до препятствий в ходе движения. Предложен новый алгоритм обучения нейронной сети, позволяющий итерационно сконструировать обучающую выборку и структуры нейронной сети. Алгоритм включает в себя как элементы обучения с учителем, так и без учителя.

Вторая нейронная сеть реализует алгоритм планирования кратчайшего пути. Третья нейронная сеть реализует алгоритм планирования безопасного пути. Для обучения второй и третьей нейронных сетей используется итерационный алгоритм обучения с учителем. При этом нейросетевые планировщики движения не планируют весь путь целиком. Выходом этих нейронных сетей является текущее направление движения робота группы на k-м шаге. В силу этого не требуется пересчет всей траектории движения на каждом шаге, что позволяет использовать нейросетевые планировщики в динамической среде.

Также в данной статье разработан алгоритм формирования строя группы мобильных роботов в некартографированной среде с препятствиями. При водятся результаты моделирования и экспериментальных исследований.

Ключевые слова: *планирования пути, групповое управление, нейронные сети, машинное обучение, среда с препятствиями*

Introduction

The most urgent fields of a mobile robotics are the autonomous control and the group control [1]. On one hand control problems for mapped environments are solved at a high level [2, 3]. On other hand in unmapped obstructed environments mobile robots are controlled by operator. Therefore the control methods and algorithms are to be improved.

The problem of mobile robots group motion control in unmapped obstructed environments is studying intensively. The papers [2—9] are reviews of the group motion planning methods. In article [2] the review of planning algorithms for 3-D environments is presented. A taxonomy and description of conventional and bio-inspired algorithms are given including the problem of the robots group path planning. The intelligent path planning methods including genetic, neural network based, bio-inspired, and hybrid algorithms is presented in paper [4]. Paper [5] includes review of the planning algorithms for stationary, dynamical, and uncertain environments. Both the global planning algorithms, based on map of environment, as well as the reactive algorithms, based on sensor information are studied. Special attention is paid to predictive control and sliding mode control. Article [6] is devoted to overview of the robots swarm's control methods. Report [7] is review of Particle Swarm Optimization methods. Paper [8] describes main methods of the robot's collective control. Taxonomy of the methods is given. Different task statements and approaches are described. In paper [9] the review of bio-inspired swarm control algorithms is presented. Studied algorithms are evaluated via the requirements for the hardware implementation. Novel algorithm of a swarm of robot configuration in a three-dimensional environment is suggested.

Reviews [2—9] and papers [10—12] notes the urgent problems of the group control methods. They are uncertainty, dynamics of the environment, and high computational requirements of the optimal path calculation.

In article [10] a hybrid algorithm of the global path planning, including a logical block and A* block, is described. Algorithm A* is used to compensate the uncertainty of the dynamical environment. Simulations and experiments demonstrate high performance of the optimal path searching.

According to paper [11] planning the movement of groups of robots in the uncertain environments is the problem solved by methods of artificial intelligence. An uncertainty is the main problem for conventional methods among them artificial potential fields, discrete searching (D*, A*), road maps, and Voronoi diagrams. To increase performance of the global searching procedure algorithm FireFly is proposed. A numerical and experimental study of the algorithm is developed for the mobile robot moving in obstructed environment.

The main disadvantages of the well-known path planning methods are errors in the improper situations and non-optimality of the planned trajectories [12]. In article [12] prediction method of mobile objects movement and integration of the prediction method with path planning algorithms are presented. Different methods are studied in the 2-D environment. Advantages of prediction for obstructed environments are demonstrated.

Thus development of autonomous robots on base of the artificial intelligent and bio-inspired methods is an urgent direction of investigations.

For instance, in the articles [13, 14] genetic algorithms are used to plan the path of a robots group. In paper [13] a hybrid planning method based on the combination of artificial potential fields and genetic global search procedure is proposed. The method increases performance of the feasible path searching. The path is evaluated by a composite criterion including length, safety, and smoothness. Results of comparative investigation of the developed method and other approaches are presented. The paper [14] proposes a method of the path planning of aircraft based on the division of space into cells by Voronoi diagrams and Delaunay triangulation. Then genetic algorithm searches the optimal path in the celled space. Investigation of the developed algorithms is performed by numerical simulations.

In article [15] the method of group control of robots using fuzzy logic is offered. The methods solved the problem of robots formation motion in the uncertain obstructed environment.

In the papers [16 — 18] hybrid algorithms based on unstable modes of movement is developed for the group of mobile robots. Offered control algorithms ensure movement of the group of robots in the obstructed 2-D and 3-D environment with moving obstacles. Delaunay triangulation and optimization procedure for the problem of robots formation motion are used. The developed algorithms allow set formation automatically without preliminary determination of the robot's place in the ranks. Algorithm of collision avoiding consists of a deliberative component and an unstable reactive component [16]. Feasibility of the algorithms is demonstrated by the experiments including fly of two copters in the environment with stationary obstacles.

In the articles [19—25] different bio-inspired algorithms are developed. In the paper [19] pigeon algorithm is proposed. In article [20] the problem of path the planning routing a group of robots in an uncertain dynamic environment using the wolf flock algorithm is solved. The proposed solution takes into account the distance between mobile robots and obstacles and the distance to the target point. The algorithms are verified by numerical simulation results. In report [21] the modification of an ant colony algorithm is presented. The algorithm is used to solve the problem of path planning of the group of robots. The problem of path planning of the ground robots group in the environment with stationary obstacles is studied. To solve the problem a bee algorithm is used. Performance of the solution is evaluated by the composite criterion including the distance to the target point and distances from robots to obstacles or other robots. Advantages of the method are demonstrated by numerical simulation results.

In the papers [22, 24] the Particle Swarm Optimization based algorithm is used to solve the problem of robots group path planning. In article [22] the problem of path planning in the plain environment with stationary dangerous obstacles and areas. Composition of the path length and safety is the optimization criterion. Different modifications of the Particle Swarm Optimization based algorithms are proposed. Advantages of the algorithms are demonstrated by numerical simulation results. In paper [24] the Particle Swarm Optimization based algorithm is used to solve the problem of robot's coalition formation. The proposed PSO algorithm improves convergence and weakens the problem of local minima. Comparative study is performed by numerical simulation.

Research [25] proposes the method of a group control for UAVs based on a multi-agent approach and relative state space conception. Advantages of the method are shown in comparison of the proposed control system and the control system with a leader.

Based on neural networks control systems are perspective field of robotics [26, 27]. Neural networks are effective adaptation mean. Therefore neural networks are applied in uncertain environments. In paper [26] the problem of adaptive coordinating control of the multi-agent systems is investigated. Proposed neural network operates in the system with variable topology, limited communications between robots, and actuators faults. Performance of the neural network control system is demonstrated by numerical simulation results. In article [27] dynamical neural network of Elman is designed. The network is training by flock algorithms.

The last success of neural networks application is based on deep learning neural networks [28—32]. Reports [29, 30] demonstrate the possibility of transferring knowledge from one area to another area using deep learning neural networks. In article [31] deep learning neural network is applied to manipulate objects by a human like robot. The reinforcement learning procedure is used. A two level planning system is developed. Experimental results are presented.

Paper [32] presents the motion control system of a wheeled mobile robot in an uncertain environment. Deep learning networks plan the robot path. The novel cascade topology of the neural network is developed. In different situations the different number of cascades is involved into planning process. Simulation results and experiments are presented for the wheeled robot in uncertain obstructed environment.

Paper [33] presents the onboard machine vision system with the aim of unmanned aerial vehicle navigation and control. New original methods are proposed for multiband images fusion based on diffuse morphology. The original methods are developed for deep machine learning.

This article is devoted to development of a neural network based control system of robots group. The control system performs estimation of an environment state, searching the optimal path planning method, path planning, and changing the trajectories on via the robots interaction. Set of the path planning method consists of two algorithms. The first algorithm is the shortest path searching (D*). The second algorithm is a safety path searching (modified D*).

The criterion of the planned path is evaluated as follows.

$$J_\Sigma = k_1 J_1 + k_2 J_2 + k_2 J_3 \to \max, \qquad (1)$$

where k_1, k_2, k_3 are weight coefficients, $k_1 + k_2 + k_3 = 1$, J_1 is a normalized time of the movement to ta target point taken with the opposite sign, J_2 is a minimal normalized distance between robots and obstacles, J_3 is a normalized path length taken with the opposite sign.

The paper is organized as follows. Section 2 presents a mathematical model of a mobile robot, description of environment, a navigation system, vision system, and communication system. Problem statement is given. Section 3 describes a block diagram of the control system and the global path planning algorithms. In section 4 neural networks based algorithms are developed. Likewise interaction algorithms between the robots are proposed. Section 5 describes results of numerical simulation and experiments.

Problem statement

A group of N_R mobile robots in a 2-D environment is considered. The mathematical model of the mobile robot is described as follows [1, 18].

$$\dot{y}_i(t) = R(y_i)x_i, \qquad (2)$$

$$M_i\dot{x}_i(t) = B_i u_i + F_{di}, \qquad (3)$$

where $y_i = [y_{1i}\, y_{2i}\, y_{3i}]^T$ is a vector of position (y_{1i}, y_{2i}) and orientation (y_{3i}) of i-th robot in the fixed frame $O_gY_{g1}Y_{g2}$ (Fig. 1), $x_i = [x_{1i}\; x_{2i}\; x_{3i}]^T$ is a vector of linear (x_{1i}, x_{2i}) and angular (x_{3i}) velocities of i-th robot in the moving frame OY_1Y_2, $R(y_i)$ is the matrix of kinematics, M_i is the matrix of inertia, F_{di} is a vector of the dynamical forces, u_i is a vector of the controls, B_i is the input matrix, $i = 1, 2, ..., N_R$, N_R is number of the robots.

The navigation system of the robot measures vectors y_i and x_i. The vision system of the robot detects obstacles located on distance from 0 to R_{vs}. Field of view of the vision system is α_{vs}. Likewise the communication system of the group robots ensures communication with each other.

Environment includes areas Ω_0, Ω_1, and Ω_2. These areas are presented in Fig. 2.

At start time moment $t = t_0$ the robots are within area Ω_0. Area Ω_2 is a target area. All robots are to be in the area Ω_2. Area Ω_1 is an uncertain environment with moving and stationary obstacles. At time moment $t = t_0$ information about the obstacles in area Ω_1 is absent at the control system of the robots.

Thus task of the group is movement from the area Ω_0 through area Ω_1 to area Ω_2. Also the robots have to locate in area Ω_0 such way to optimize a given criterion. The robots transmit each other vectors y_i, x_i, distances to obstacles $r_{obs}(i, j)$, and vector E_i, describing area Ω_1.

In the uncertain environment it is possible to use global and local planning algorithms [1, 18]. Aim of

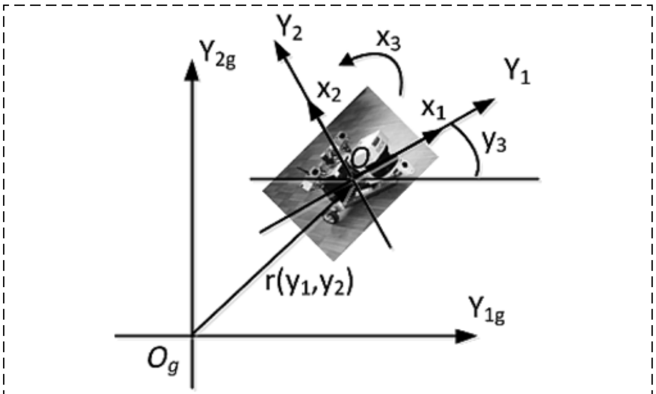

Fig. 1. The fixed and moving frames of the robot

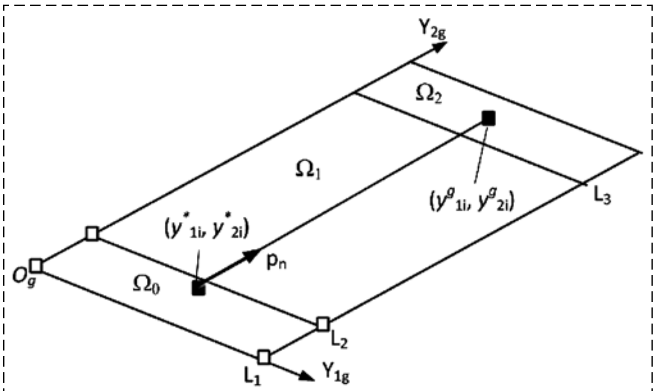

Fig. 2. Start, uncertain, and target areas

the global planning algorithm is calculation of the preliminary path. Aim of the local planning algorithm is correction of the path in the neighborhood of obstacles. There are two problems of the global path planning in the uncertain environments. First, in the uncertain environment it is hard to construct proper criterion of a mobile robot control system. Second, in the neighborhood of obstacle the target point should be moved. Therefore the position of the target point is function of the time and the path planning algorithm has to calculate all optimal trajectory every step.

Thus in the uncertain environment the criterion of motion and trajectory planning should be formed directly in the process of the robot moving. The global planner calculates a simple strategy based on the most general assumptions.

The global planning algorithms

Block diagram of the group control system is presented in Fig. 3.

The inputs of Global Planning block are a priory map (Map), data packet G from another robots of the group, and the target point y_i^*, x_i^* of i-th robot. Global planning algorithm calculates optimal path P_i^* of i-th robot. Local planning block corrects optimal path P_i^* using measured obstacle's position y_{ip}, x_{ip}. Corrected path P_i feeds to Motion Control System block.

The global planning is based on the following assumptions.

1) Starting locations of the robots are calculated as the problem of uniform distribution solution of area Ω_0.

2) The global path of i-th robot is a direct line between current location and final location at target area Ω_2.

Distribution of the robots in area Ω_0 includes Delaunay triangulation and optimization of i-th robot's location. This operation consists of the following steps.

Step 1. The aggregated group is formed. This group includes all mobile robots of the group, detected obstacles, and vertices of rectangle area Ω_0. In Fig. 2 the vertices are presented as white squares.

Step 2. Neighbors of i-th robot are calculated using Delaunay triangulation [14, 18]. The result of Delaunay triangulation is set M_{Ri} of adjacent objects to which i-th robot is connected by edges.

Step 3. The distances from i-th robot to the adjacent objects are calculates as follows.

$$r_{ij} = [(y_{1i} - y_{1j})^2 + (y_{2i} - y_{2j})^2]^{0.5},$$
$$i = \overline{1, n_i},\ j = \overline{1, N_G}, \tag{4}$$

where n_i is a number of the adjacent objects of i-th robot, r_{ij} is the distance from i-th robot to j-th object.

Step 4. Distances (4) are optimized. Free variables are y_{i1}, y_{i2}. The optimization criterion is as follows.

$$\left(\min_{i,j}(r_{ij})\right) \to \max_{y_{1i}, y_{2i}}. \tag{5}$$

The solution of problem (5) is optimal location of i-th robot.

$$[y_{1i}^*, y_{2i}^*] = \max_{y_{1i}, y_{2i}}\left(\min_{i,j}(r_{ij})\right). \tag{6}$$

Current direction of the robot movement is described by the following direction vector.

$$p_{ni} = [y_{1i} - y_{1i}^* \quad y_{2i} - y_{2i}^*]^T,\ i = \overline{1, N_R}. \tag{7}$$

Vectors (7) are the output of the global planning algorithm.

The local planning algorithms

State classification. Tasks of the local planning algorithms are:

— the state classification;

— optimal path planning algorithm searching;

— path planning.

The state classification uses data from the vision system of the robot. Vector E_i is introduced.

The first element of vector E_i is integer C_i. If obstacles are not detected by the vision system then integer C_i is equal to 0. If obstacles are detected by the vision system, and sector I (Fig. 4) is free then integer C_i is equal to 1. If sector I is occupied by obstacles then integer C_i is equal to 2. If all sectors

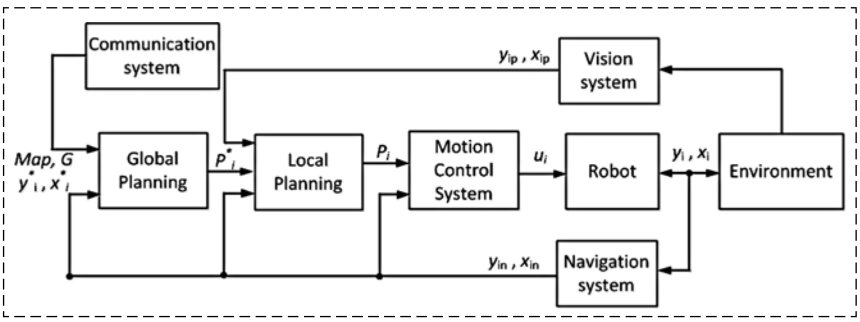

Fig. 3. Block diagram of the group control system

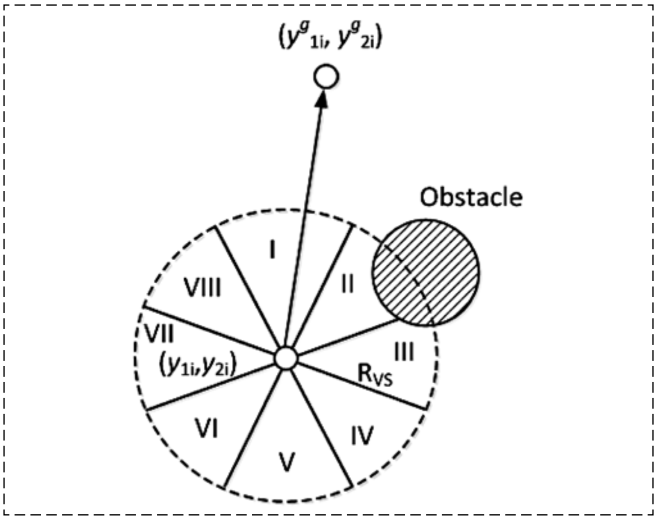

Fig. 4. State classification

are occupied by obstacles then integer C_i is equal to 3. Integer C_i is a characteristic of area Ω_1 complexity.

The second element of vector E_i is integer 1 or 2. If the optimal path planning algorithm is D* then the second element is equal to 1. If the optimal path planning algorithm is modified D* then the second element is equal to 2.

The first element of vector E_i is estimation of the state complexity [30]. To correct coordinates (y^*_{1i}, y^*_{2i}) weighting coefficients are introduced into expression (4) as follows.

$$r_{ij} = k_{ij}[(y_{1i} - y_{1j})^2 + (y_{2i} - y_{2j})^2]^{0.5},$$
$$i = \overline{1, n_i}, \ \ j = \overline{1, N_G}. \tag{8}$$

Weighting coefficients k_{ij} depends from the first element of vector E_i. If C_i is small then coefficients k_{ij} tends to 1. If C_i is big then coefficients k_{ij} tends to 0.

The optimal path planning algorithm is searched by a neural network. A block diagram of the learning system for the neural network is presented in Fig. 5. A supervised and unsupervised learning [28] could be performed on base of the presented block diagram. The inputs of the neural network are image P_i from the vision system and planning algorithm S_i. Based on the results of the control system simulation, the evaluation of the criterion (1) is formed. Criterion J_Σ (1) also enters the neural network. Planning algorithm S_i is D* or modified D*.

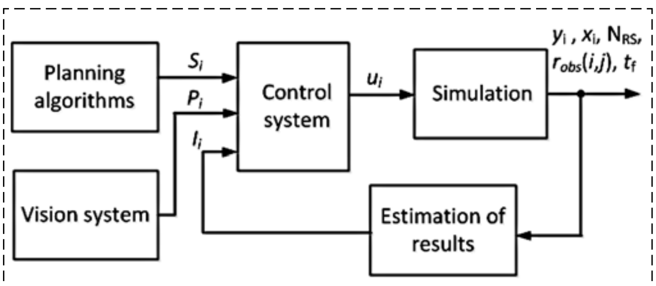

Fig. 5. Block diagram the training system

The optimal path is a reference input of the control system [1]. The control system includes the collision avoiding algorithm based on unstable modes [18]. Proposed block diagram allows learn the neural network in uncertain environment.

Path planning neural networks

The local planner is a deep learning neural network [25]. Algorithm D* is supervisor for the deep learning neural network. The map is split into cells as it is shown in Fig. 6. The resulting map is then presented as a matrix. Each element of the matrix corresponds to a specific cell of the map.

For example shown in Fig. 6 map is described by matrix P_i, 7×9, as follows.

$$P_i = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 \end{bmatrix}. \tag{9}$$

If a cell is free then corresponding element of matrix P_i is equal to 0. If a cell is occupied by obstacle then corresponding element of matrix P_i is equal to 1.

Current and target location of the robot are presented in other matrices. Thus matrix P_i, matrix of current location, and matrix of target location of the robot are description of the state. The state is clas-

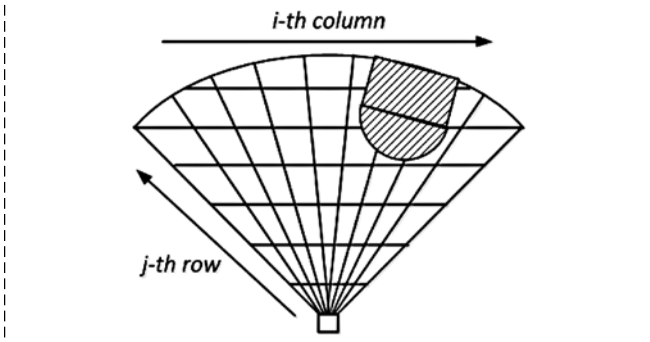

Fig. 6. Celled map

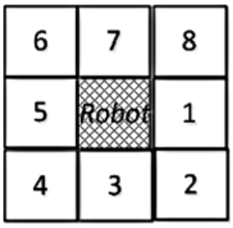

Fig. 7. Classes of the state

sified by algorithm D* into 9 classes. If planned by D* path passes through cell number $N = 1, 2, ..., 8$ (Fig. 7) then the state class is N. If there is no path from the current location to the target location of the robot then the state class is 9.

As a result, the neural network is trained to choose the current direction of motion. This direction is coinciding with the trajectory constructed by D*. The trained neural network does not plan whole path of the robot. Therefore the network could operate effectively in uncertain dynamical environments.

The training sample should contain all the situations required for training, and in sufficient quantity. In practice different samples occur with different frequency. Thus accumulation of rare situations is possible either through their special creation or a significant increase in the training sample. Special creation of the samples for training is complex, hard, and laborious process. In fact, it is necessary to determine the most important situations manually. Control of accumulation of the training sample is manual operation also. Random accumulation of the training sample leads to a significant increase in the training sample.

In this article the following iteration algorithm of creation of the training sample is proposed.

1) Sample P_k is generated in step k of the iterations. A dimension of sample P_k is N_k. Starting and target location of the robots, location, number and sizes of obstacles are random variables.

2) Algorithm D* calculates the path of the robot for every element of sample P_k. The path belongs to the class 1, 2, 3, 4, 5, 6, 7, 8, or 9 (Fig. 7).

3) If $k > 1$ then the element of sample P_k is classified by trained in the previous step neural network NN_{k-1}. If the output of neural network NN_{k-1} is equal to number of the path cell then the element of sample P_k is deleted from the sample. In other case the element of sample P_k is saved in the sample.

4) Neural network NN_k is trained by sample P_k. Let's note that every step the structure of the neural network NN_k is enhanced.

5) Iterations are broken if the given accuracy of training is achieved.

The proposed procedure allows select for training situations that the neural network classifies with errors.

The described algorithm is similar to reinforce training [31] because selection of the sample elements is based on the results of the trained network operating. In report [35] two networks are used to select elements of training sample, and one network trains another one. In our proposition the network trains itself.

Numerical simulation results

Studying of the training set development algorithm. Implementation of the algorithm above includes four iterations.

In iteration 1, the neural network NN1 was selected, including the input layer, 4 convolutional layers with 16 filters and 2 fully connected output layers containing 16 and 8 neurons, respectively. A starting training sample has been created, consisting of 4000 examples for each class. An example of the map, on the basis of which educational examples are created, is presented in Fig. 8 (see the second side of cover).

The results of the neural network NN1 training are presented in Table 1.

Table 1

The results of iterative training of the neural network

Iteration number	1	2	3	4
Sample size	4000	10 000	11 800	12 600
Estimation of training accuracy	79 %	84 %	89 %	94 %
Frequency of goal achievement without collisions according to the simulation results	50 %	79 %	85 %	92 %

In iteration 2, the neural network NN1 is used to filter out new training examples. Only those examples that the NN2 neural network has classified incorrectly are added to the training set. At the same time, at the second iteration, the NN2 neural network is trained, which includes the input layer, 5 convolutional layers with 32 filters and 3 fully connected output layers containing 32, 16 and 8 neurons. As a result of the second iteration, the training set was increased to 10000 examples per class. Estimation of the accuracy of training was equal to 84 %.

In iteration 3, the NN2 neural network is used to select training examples. The NN3 network is being studied, it includes an input layer, 9 convolutional layers with 40 filters and 3 fully connected output layers containing 32, 16 and 8 neurons.

Upon completion of the second iteration, the effect of a further increase in the sample by adding situations in which the network trained at the previous iteration makes the wrong decisions did not lead to an increase in the accuracy of training. In this regard, at the third and fourth iterations, all points of the paths on which the network trained at the second and third iteration allowed collisions were added to the training set. This approach has significantly improved the accuracy of the training network. As a result of the third iteration, the training set was increased to 11 800 examples for each class. At the same time, 1800 additional examples were created by filtering initial situations by the NN2 network. The estimation of accuracy for training on a filtered sample was 89 %.

In iteration 4, the NN2 neural network is used to select training examples. The NN4 network is being studied, it includes an input layer, 10 convolutional layers with 40 filters and 3 fully connected output layers containing 32, 16 and 8 neurons. As a result of the fourth iteration, the training set was increased to 12600 examples for each class. The estimation of accuracy for training on a filtered sample was 94 %.

The assessment of accuracy during training is formed as follows. The training sample is divided into two parts. The first part, which contains 80 % of the examples, is used for training. The second part, which contains 20 % of the examples, is used to test the trained network. Since the training and validation samples contain sections of the same trajectories, the accuracy of the trained network is additionally checked on a separately generated sample, which contains 2000 test situations.

The sufficiency of the obtained training accuracy is estimated by the frequency of reaching the target point without collisions. During the simulation, an additional evaluation sample was created, including 500 examples. For each example, a simulation of the movement of the robot with a neural network planner was carried out. An example is considered to be completed successfully if the robot will reach the target point without collisions and loops.

According to the simulation results, the probability estimation of achieving the goal is obtained: 50 % for the first iteration; 79 % for the second iteration; 85 % for the third iteration, 92 % for the fourth iteration.

The simulation results of a trained neural network, presented in Fig. 9 (see the second side of cover) allow us to conclude that a trained neural network is able to plan a route in a rather complex environment, for example, in a city. In Fig. 9 points show the route set by the D* algorithm, and asterisks indicate the route set by the NN3 neural network.

Further, to assess the effect of using the proposed iterative procedure, the modeling of the NN4 neural network trained on a simple (unfiltered) sample of 12 600 situations for each of the eight classes was simulated. The estimation of traning accuracy is 91.3 %. The probability of reaching the target point without collisions for a neural network trained on a regular sample was equal to 79 %. Thus, the increase in efficiency in the frequency of successful achievement of the goal was about 12 %.

Then, an increase in the simple training sample was applied to obtain the same frequency of successful achievement of the goal as in the sample created using the suggested iterative procedure at iteration 4. The shown indicator (92 %) was achieved on the volume of the training sample of 20,000 situations for each of the eight classes. Thus, the proposed procedure permits to reduce the training sample by 40 %.

Simulation results of the path planning algorithms. Considered group consists of 5 mobile wheeled robots, $N_R = 5$. Environment is presented in Fig. 2. Width of area Ω_0 is equal to 20 m, $L_1 = 20$ m. Length of area Ω_0 is equal to 5 m, $L_2 = 5$ m. Distribution of robots in area Ω_0 is described by solution problem (5). Vertices of area Ω_0 are [0; 0], [0; 5], [20; 5], and [20; 0]. Problem (5) with 10 free variables (desired coordinates of the robots) is solved by Matlab function fminimax.

In Fig. 10 (see the second side of cover) evolution of the desired locations of the robots are presented. In Fig. 10a the complexity of the environment is not taken into account. If Fig. 10b the complexity of the environment is taken into account.

Starting locations of the robots are indicated by circles. Target locations of the robots are indicated by clear squires. Obstacle is indicated by filled rectangular. The vision system detects obstacle on the firth step of simulation. Coefficients $k_{ij} = 1.0$ in Fig. 10a, but $k_{ij} = 0.5$ in Fig. 10b. Coefficients k_{ij} are used in expression (8). From Fig. 10 it is clear that the distances from the robots to the obstacle in the second case are more than in the first case. It should be noted that the group when moving independently forms a formation. The formation changes when the appearance or disappearance of obstacles.

In Fig. 11 (see the thired side of cover) simulation results are presented for environment with two obstacles. In this case the group is divided into two clusters while the obstacles avoiding. Coefficients $k_{i2} = 0.7$. Also, when approaching an obstacle at a dangerous distance, the algorithms of obstacle avoidance presented in [18] are used at the lower level.

Simulation of path planning and environment classification algorithms. Evaluation of the neural environment classification is based on criterion (1). Coefficients of criterion are $k_1 = 1$; $k_2 = k_3 = 0$. Thus criterion is the time of movement. Evaluation is performed on additional sample of 500 elements.

Velocity of the robot is calculated as follows.

$$V_R = \begin{cases} V_{\min} + (V_{\max} - V_{\min})\dfrac{r}{r_s}, & V_R \leqslant V_{\max}, \\ V_{\max}, & V_R > V_{\max}, \end{cases} \quad (10)$$

where r is the distance from the robot to the obstacle, $r_s = 3$ m is safety distance, $V_{\min} = 0.25$ m/s is minimal velocity of the robot, $V_{\max} = 1.0$ m/s is maximal velocity of the robot.

According to the results of numerical studies, the average time of movement is about 30 seconds. If the optimal algorithm is chosen then the time of movement is about 28,2—28,8 seconds. Thus the effect of environment classification is 4—6 %.

Simulation results of five robots movements in the uncertain environment with four obstacles are presented in Fig. 12 (see the thired side of cover).

Starting locations of the robots are [5; 4], [15; 2], [17; 1], [30; 3], and [35; 3]. Vertices of area Ω_0 are [0; 0], [0; 5], and [50; 5], [50; 0]. Aim of the group is movement to the target area with vertices [0; 45], [0; 50], [50; 50], and [50; 45]. Aim is achieved if all robots of the group are in the target area.

In figure 12, the three main maneuvers performed by the group during the movement are clearly visible. The first time the group performs a maneuver when it detects an obstacle located in the area of the first robot. The second time the group performs a maneuver when it detects obstacles in the right part of the environment. The last maneuver is forced by the obstacle located in the target area.

Experiments results. The first aim of experiments is evaluation of computational efficiency of neural networks in comparison with optimal algorithm D* (modified D*). The second aim of experiments is evaluation of feasibility of the proposed algorithms.

Experiments are performed with the group of two mobile robots equipped by wheeled platform, onboard computer Raspberry PI 3B +, all-around looking lidar RP Lidar A2, controller Arduino Uno, and optical encoders.

Implementation of the planning and classification algorithms are based on the neural network described as follows.

Computational efficiency is evaluated in 512×512 cell environment. The results of the evaluation are as follows.

The time for path planning of D* algorithm is 0.3—3.0 seconds. The average time is 1.3 seconds. The time for path planning of the proposed neural network is about 0.4 seconds.

The obtained effect is achieved due to the fact that the neural network calculates only the current direction of movement. The D* algorithm calculates the entire path of the robot. In a stationary environment this advantage is not important. In a dynamical environment or for a moving target point this advantage is very important.

In Fig. 13 (see the thired side of cover) results of experiments are presented. Planned points are indicated by squires, and paths of the robots are shown by lines. Step of the grid is 0.5 m. There are two stationary obstacles in the Fig. 13.

Mean error of the robot's path is about 0.12 m.

Conclusion

The algorithms of the group control of mobile robots for 2-D uncertain environment are proposed. The novelty of the algorithms is described as follows.

Algorithm of robot's formation constructing is proposed. The algorithm includes Delaunay triangulation and optimization of i-th robot's location. This is development of algorithms presented at [18], but in this article a maximin optimization problem is used.

Coefficients k_{ij} as function of the state complexity are introduced. These coefficients shift the robots from obstacles to free areas.

Neural path planning algorithm is proposed. The algorithm is characterized by the use of a neural network that produces the current direction of movement. The algorithm decreases time of the path planning by three times. The effect is achieved for uncertain dynamical environments.

Novel iterative algorithm of neural networks training is proposed. The algorithm allows select for training situations that the neural network classifies with errors. The algorithm reduces the amount of training sample to 40 %. Also the proposed algorithm reduces the time of creating a training sample.

Further development of the results is to take into account the limitations imposed on the planned path of dynamic and kinematic properties of robots. Due to inertia the robot may not get to the planned point, so taking into account the inertial properties of the robot is an important problem. Also, when driving at high speeds, it is important to specify not only the next target point, but also a certain number of path points. The solution to this problem will increase the possible speed.

References

1. **Pshikhopov V., Medvedev M.** Position-Path Control of a Vehicle, Path Planning for Vehicles Operating in Uncertain 2D Environments, Elsevier, Butterworth-Heinemann, 2017, pp. 1—23.
2. **Yang L., Qi J., Song D., Han J., Xia Y.** Survey of Robot 3D Path Planning Algorithms, *Journal of Control Science and Engineering*, 2016, vol. 2016, 22 p.
3. **La Valle S.** Planning Algorithms. Cambridge University Press, 2006, 842 p.
4. **Mac T. T., Copot C., Tran D. T., De Keyser R.** Heuristic approaches in robot path planning: A survey, *Robotics and Autonomous Systems*, 2016, vol. 86, pp. 13—28.
5. **Hoy M., Matveev A. S., Savkin A. V.** Algorithms for collision-free navigation of mobile robots in complex cluttered environments. A survey, *Robotica*, 2015, vol. 33 (3), pp. 463—497.
6. **Bayindir L.** A review of swarm robotics tasks, *Neurocomputing*, 2016, available at: https://doi.org/10.1016/j.neucom.2015.05.116.
7. **Shi Y.** Particle swarm optimization: Developments, applications and resources, *Evolutionary Computing, 2001 IEEE International Conference on*, 2001, vol. 1, pp. 81—86.
8. **Uny Cao Y., Fukunaga A. S., Kahng A. B.** Cooperative Mobile Robotics: Antecedents and Directions, *Autonomous Robots*, 1997, vol. 4, pp. 1—23.
9. **Shlyakhov N. E., Vatamaniuk I. V., Ronzhin A. L.** Review of the Methods and Algorithms of a Robot Swarm Aggregation, *Mekhatronika, Avtomatizatsiya, Upravlenie*, 2017, vol. 18(1), pp. 22—29 (in Russian).
10. **Yu X.- Y., Fan Z.- Y., Ou L.-L., Zhu F., Guo Y.-K.** Optimal Path Planning Satisfying Complex Task Requirement in Uncertain Environment, *Robotica*, 2019, vol. 37 (11), pp. 1956—1970.
11. **Patle B. K., Pandey A., Jagadeesh A., Parhi D. R.** Path planning in uncertain environment by using firefly algorithm, *Defence Technology*, 2018, vol. 14(6), pp. 691—701.
12. **Kamil F., Hong T. S., Khaksar W., Moghrabiah M. Ya., Zulkifli N., Ahmad S. A.** New robot navigation algorithm for arbitrary unknown dynamic environments based on future prediction and priority behavior, *Expert Systems with Applications*, 2017, vol. 86, pp. 274—291.
13. **Nazarahari M., Khanmirza E., Doostie S.** Multi-objective multi-robot path planning in continuous environment using an enhanced genetic algorithm, *Expert Systems with Applications*, 2019, vol. 115, pp. 106—120.
14. **Qu Y., Zhang Y., Zhang Y.** A Global Path Planning Algorithm for Fixed-wing UAVs, *Journal of Intelligent and Robotic Systems: Theory and Applications*, 2018, vol. 91(3—4), pp. 691—707.
15. **Belogalzov D., Finaev V., Shapovalov I., Soloviev V., Medvedev M.** Decentralized control of a group of robots using fuzzy logic, *Journal of Engineering and Applied Sciences*, 2017, vol. 12 (9), pp. 2492—2498.
16. **Pshikhopov V., Medvedev M.** Decentralized control of the mobile robots group in an obstructed 2-D environment, *Mekhatronika, Avtomatizatsiya, Upravlenie.* vol. 17(5), pp. 346—353 (in Russian).
17. **Pshikhopov V., Medvedev M.** Mobile Objects Control in Three-Dimensional Area Using the Hybrid Decentralized Algorithm, *5th International Conference on Control, Decision and Information Technologies*, CoDIT, 2018, pp. 452—457.
18. **Pshikhopov V., Medvedev M.** Group control of autonomous robots motion in uncertain environment via unstable modes, *SPIIRAS Proceedings*, 2018, vol. 60(5), pp. 39—63 (in Russian).
19. **Duan H., Qiao P.** Pigeon-inspired optimization: A new swarm intelligence optimizer for air robot path planning, *International Journal of Intelligent Computing and Cybernetics*, 2014, vol. 7 (1), pp. 24—37.
20. **Radmanesha M., Kumar M., Sarim M.** Grey wolf optimization based sense and avoid algorithm in a Bayesian framework for multiple UAV path planning in an uncertain environment, *Aerospace Science and Technology,* 2018, vol. 77, pp. 168—179.
21. **Fan X., Luo X., Yi S., Yang S., Zhang H.** Optimal path planning for mobile robots based on intensified ant colony optimization algorithm, *International Conference on Robotics, Intelligent Systems and Signal*, Changsha, China (2003), pp. 131—136.
22. **Zhang Y., Gong D., Zhang J. H.** Robot path planning in certain environment using multi objective particle swarm optimization, *Neurocomputing*, 2013, vol. 103, 172—85.
23. **Liang H. H., Lee C.** Efficient collision-free path planning of multiple mobile robots system using efficient artificial bee colony algorithm, *Advanced Engineering Software*, 2015, vol. 79, 47—56.
24. **Xu B., Yang Z., Ge Yu., Peng Z.** Coalition Formation in Multi-agent Systems Based on Improved Particle Swarm Optimization Algorithm, *International Journal of Hybrid Information Technology*, 2015, vol. 8(3), pp. 1—8.
25. **Munasypov R. A., Muslimov T. Z.** UAVs Group Control Based on the Relative State Space Method, *Mekhatronika, Avtomatizatsiya, Upravlenie*, 2018, vol. 19, no. 2, pp. 120—125 (in Russian).
26. **Wang X., Yang G.-H.** Adaptive reliable coordination control for linear agent networks with intermittent communication constraints, *E-LETTER on Systems, Control, and Signal Processing*, 2018, vol. 362.
27. **Hong-Wei G., Wen-Li D., Feng Q., Lu W.** A Dissimilation Particle Swarm Optimization-Based Elman Network and Applications for Identifying and Controlling Ultrasonic Motors, *Advances in Soft Computing*, 2007, vol. 41, pp. 1345—1360.
28. **Le Cun Ya., Yoshua B., Geoffrey H.** Deep learning, *Nature*, 2015, vol. 521.7553, pp. 436—444.
29. **Zhu J., Park T., Isola P., Efros A. A.** Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks, *In IEEE International Conference on Computer Vision (ICCV)*, 2017.
30. **Gholami B., Rudovic O., Pavlovic V.** Probabilistic Unsupervised Domain Adaptation for Knowledge Transfer Across Visual Categories, *The IEEE International Conference on Computer Vision (ICCV),* 2017, pp. 3581—3590.
31. **Li Z., Zhao T., Chen F., Hu Y., Su C.-Y., Fukuda T.** Reinforcement Learning of Manipulation and Grasping Using Dynamical Movement Primitives for a Humanoidlike Mobile Manipulator, *IEEE/ASME Transactions on Mechatronics*, 2018, vol. 23(1), pp. 121—131.
32. **Pshikhopov V., Medvedev M., Vasileva M.** Neural network control system of motion of the robot in the environment with obstacles, *Lecture Notes in Computer Science*, 2019, vol. 11606, pp. 173—181.
33. **Knyaz V. A., Vishnyakov B. V., Gorbatsevich V. S., Vizilter Yu. V., Vygolov O. V.** Intelligent data processing technologies for unmanned aerial vehicles navigation and control, *SPIIRAS Proceedings*, 2016, vol. 45(2), pp. 26—44 (in Russian).
34. **Karkishchenko A. N., Pshikhopov V. Kh.** On finding the complexity of an environment for the operation of a mobile object on a plane, *Automation and Remote Control,* 2019, vol. 80(5), pp. 897—912.
35. **Zoph B., Vasudevan V., Shlens J., Le Q. V.** Learning Transferable Architectures for Scalable Image Recognition, *Proc. of 2017 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2017, July, 22—25, Honolulu, Hawaii.