Autonomous Mobile Robot Self-Learning In Motion Planning Problem

Sergey V. Manko, Valery M. Lokhin, Sekou A. K. Diane, and Alexander S. Panin
MSTU MIREA, Moscow, Russia
Email: cpd@mirea.ru

***Abstract*—Increasing robotic systems autonomy is a major challenge in ensuring both their performance and ease of application in numerous areas of human activity. The present study attempts to combine several artificial intelligence methods to design self-learning control system for the task of mobile robot motion planning in a complex environment. We propose two main components of the control system: a dynamic path planner based on D* algorithm and terrain type prediction subsystem based on classification trees. Efficiency of both of the subsystems grows with time due to knowledge accumulation, leading the robot to a better maneuvering in the environment filled with obstacles.**

***Index Terms*—autonomous mobile robot, motion planning, graph route planning, classification trees, machine learning**

I. Introduction

One of the key goals of autonomous mobile robotics is route planning and execution. Control systems (CS) that solve this task are usually designed in a hierarchical way [1].

At the level of strategy planning a simplified route is generated with a limited accounting for information on robot construction, available sensors and software.

The level of tactical planning is responsible for selection of control mode most appropriate for driving the robot along a current route segment. Hereby, specific details of robot functioning environment may be taken into account, e.g. terrain passability, obstacle presence, robot onboard equipment failures, etc.

Widely used expert-logic approach to intelligent control systems (ICS) design is often limited due to the complexity (for a human expert) of composing an all-inclusive control rule database, accounting for all situations that may occur to robot.

One example of the opposite approach is suggested in [2], where a machine-learning process is used to refine autonomous mobile robot (AMR) behavior during its functioning. The aim of the present work is to develop a machine-learning method to refinement of AMR functioning rules through changes in its knowledge database.

In our research we model the robot equipped with a vision sensor. Information about distant terrain regions gathered through robot's camera is used for continual amendment of its motion planning graph. The requirement for the vision system correct operation is to undergo learning process, which is described further. In addition to the camera robot is also equipped with a collision sensor, which is limited in range but is a priori operable.

While robot explores the environment, the proposed algorithm learns to predict terrain passability based on camera image analysis. The data used in the learning routine is generated by correlating collision sensor readings with images of the terrain in front of the robot.

Manuscript received July 1, 2014; revised June 11, 2015.

II. Virtual Modeling Software

As a part of our research we developed mobile robot simulation software that supports 2D-environment models and rigid body physics modeling for objects of various shapes. The choice of 2D-graphics was made due to convenience of virtual scenes setup and clearness of experiments visualization as well as for simplification of the algorithms. The main programming language used for development is C#. Physical modeling is based on the Box2DX library.

A fragment of simulation at Fig. 1 shows several objects composing a virtual environment: mobile robot (at point R); obstacles (black areas); areas of low passability (grey areas).

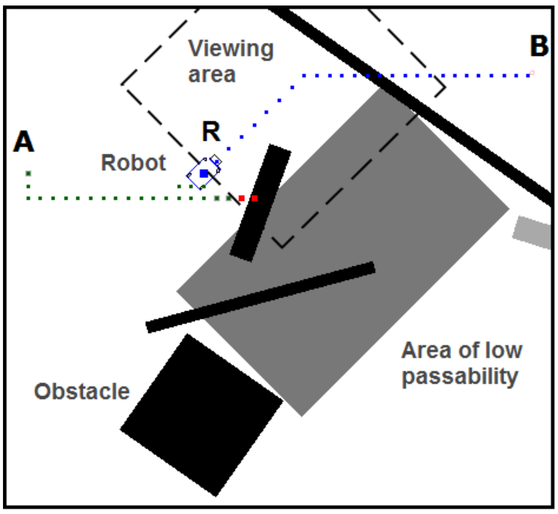

Figure 1. The model of robot functioning environment

The figure also shows some non-physical elements that indicate functioning aspects of robot control system:

doi: 10.18178/ijmerr.4.3.238-241

initial and terminal points of the route (A and B); passed segment of route AR; dynamically planned route RB; robot viewing area. It should be noted that planned route may cross obstacles if they are not yet percepted by the robot, and hence are not stored in cartographical database. Thus in Fig. 1. route RB crosses an obstacle due to the fact that it is not in the range of collision sensor and the camera sensor has not yet undergone learning process.

Mobile robot kinematic model is a 4-wheel platform with forward steering wheels as shown in Fig. 2. Robot length and width are set to values L=2.5 m and W = 1.7 m correspondingly.

Software model of the AMR sensory subsystem includes the following components:

- Global positioning subsystem that determines robot coordinates (x, y) and rotation degree α;
- Collision sensor, located in front of the robot hull;
- Vision sensor, that allows to get an image of environment in robot local coordinates. The size of the viewing area is 20×20 m, and the resolution of the image is 100×100 pixels.

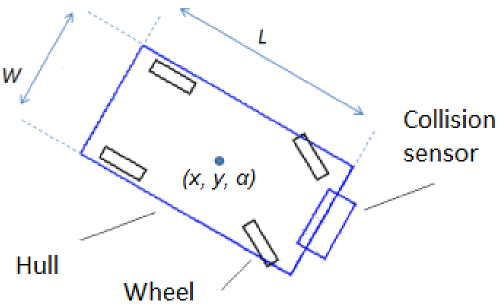

Figure 2. The model of mobile robot

III. Route Planning and Execution

AMR route planning at a strategic control level may be implemented with use of Dijkstra's algorithm, A* search and a variety of other methods [3], [4]. The requirement is that motion environment is represented as a graph *G*, whose vertices v_i, $i \in \mathbb{N}$ correspond to cells of discretized environment, and edges r_j $j \in \mathbb{N}$ correspond to possible transitions between adjacent cells.

The complexity of robot functioning environment and the incompleteness of percepted information lead to the necessity of repetitive route computation. The most appropriate algorithm in terms of computational efficiency is dynamic A* search or D* [4]. Its main distinctions from well-known Dijkstra algorithm are the usage of heuristic in goal distance estimation, which ensures a more directed search through the graph, and skip of computation for those route segments that did not change since last graph update process.

One more complication of path planning is that AMR kinematics limits the choice of directions possible for robot transition between adjacent graph vertices. Taking this into account only 6 of 8 possible transitions are allowed: forward, forward-left and forward-right, backward, backward-left and backward right accordingly.

The planning process also accounts for the fact that the linear size of the robot exceeds the distance between adjacent graph vertices. Edges of the graph *G* located in dangerous proximity to the obstacles are marked as low-passible allowing the planner to change the route in advance to possible collision. The collision estimation is given by inequality $d<R$, where d – distance from robot to obstacle in meters, $R=2$ – robot safety radius.

An execution of generated motion route is implemented as a simplistic motion control algorithm using a PID acceleration controller through distance and a PD steer controller through angular error. The target point for both controllers is set to a path node, closest to robot's position.

IV. Terrain Type Prediction

The AMR sensory information analysis functions based on classification and regression trees (CART) framework [5], [6]. This technology is widely used in data mining, medical diagnostics, banking analytics, web-marketing, etc. At the same time, the capabilities of classification trees make them applicable in the field of intelligent robotic control. In our research, classification trees are used for AMR vision subsystem automated online learning and prediction of terrain properties.

The CART algorithm can be described as a repetitive partitioning of learning samples set $\mathbf{S}=\{\mathbf{s}_1,\ldots,\mathbf{s}_n\}$. In the specific terrain type prediction problem each sample $\mathbf{s}_i=\{\mathbf{x}_i, y_i\}$ consists of input vector $\mathbf{x}_i$ containing terrain visual features and classification label y_i that is generated automatically based on robot velocity and collision sensor readings.

The classification tree composition algorithm uses dichotomization procedure, that is based on the entropy estimation. For a given set of training samples $\mathbf{S}$, an entropy value is given by the formula:

$$H(\mathbf{S}) = -\sum\nolimits_{y\in D(y)} P(\mathbf{S},y)\log_2 P(\mathbf{S},y)$$

where $D(y)$ is a set of discrete values of the parameter y; $P(\mathbf{S},y)$ is the statistical probability of observing a value *y* on the set of examples $\mathbf{S}$.

Let us define a logical predicate d on the set $\mathbf{S}_x=\{\mathbf{x}_1,\ldots,\mathbf{x}_n\}$. Then the set of examples can be divided into two subsets: a subset on which d is true and a subset, where d is false. For these two subsets in the same way as for the original set, the values of entropy are calculated as follows:

$$H(\mathbf{S}^d) = -\sum\nolimits_{y\in D(y)} P(\mathbf{S}^d,y)\log_2 P(\mathbf{S}^d,y),$$

$$H(\mathbf{S}^{\bar{d}}) = -\sum\nolimits_{y\in D(y)} P(\mathbf{S}^{\bar{d}},y)\log_2 P(\mathbf{S}^{\bar{d}},y).$$

In accordance to the property of dichotomization, the obtained subsets will satisfy the criterion:

$$P(\mathbf{S}^d\,|\,\mathbf{S})H(\mathbf{S}^d) + P(\mathbf{S}^{\bar{d}}\,|\,\mathbf{S})H(\mathbf{S}^{\bar{d}}) \le H(\mathbf{S})$$

In other words, after the dichotomy the verity or the falsity of parameter y in resulting subsets will be more evident than in the original set. The described dichotomization process may be iteratively applied to every derived subset, ending up with the subsets of minimal entropy. The nodes of a resulting classification tree correspond to the steps of the repetitive dichotomization and store appropriate logical predicates for efficient multi-stage data classification.

V. Active Machine Learning Design

Mobile robot intelligent control system should be designed with an accounting for uncertainties of surrounding environment such as incompleteness of robot's cartographical database and changes in environment state with time due to the motion of nearby objects.

Elimination of these uncertainties is possible through incorporation of machine-learning subsystem into ICS of the robot. The functioning of this subsystem may be organized as two complementary processes.

Firstly, the classification tree should constantly be reconstructed using the newly obtained samples, enabling robot to estimate terrain properties not only with collision sensor, but also with the camera.

Secondly, the system should update cartographical knowledge through changes to motion planning graph based on results of visual classification. Thus, second stage of the learning is dependent on the first one.

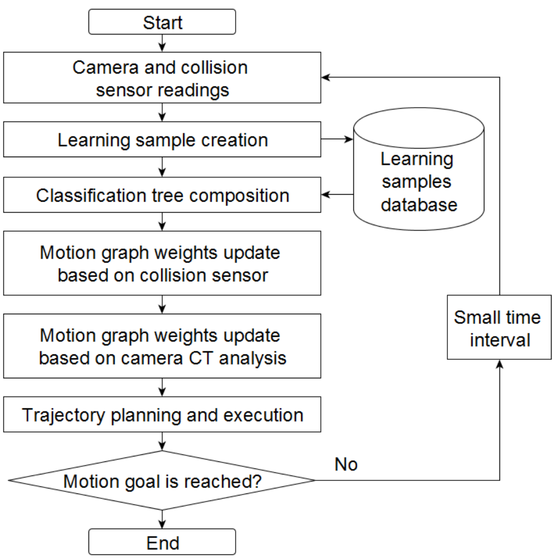

Figure 3. Mobile robot functioning algorithm

Fig. 3 shows a block diagram of robot learning process. One notable fact is that, a complex and effective visual sensory information classification subsystem is being taught with a simple small ranging collision sensor.

As it was stated earlier, the analysis of camera images is based on classification trees technology. The visual image is split into rectangular cells c_k, $k \in \mathbb{N}$, of dimensions $l_x = [W / N]$, $l_y = [H / N]$, where W and H – width and height of image in pixels correspondingly; N is a number of cells along every direction. In our research, these parameters were set to constant values: $W = H = 100$, $N = 20$, yielding $l_x = l_y = 5$ pixels. Classification tree task is then to determine a presence of an obstacle in a region of virtual environment, which corresponds both to the cell of robot camera image and to a vertex of motion planning graph.

VI. Experiments

The evaluation of proposed algorithms was possible through carrying out experimental research aimed at comparison of robot motion efficiency with and without using machine-learning. Fig. 4 shows results of environment exploration in the two robot functioning modes. Without learning robot perception range is limited to one sensory channel of low efficiency, leading to small situation awareness, incomplete mapping (Fig. 4, a) and hence non-optimal route planning. Incorporation of learning algorithms allows involving additional sensory channel (camera) with better characteristics, thus introducing the possibility to map distant obstacles and low passability regions (Fig. 4, b).

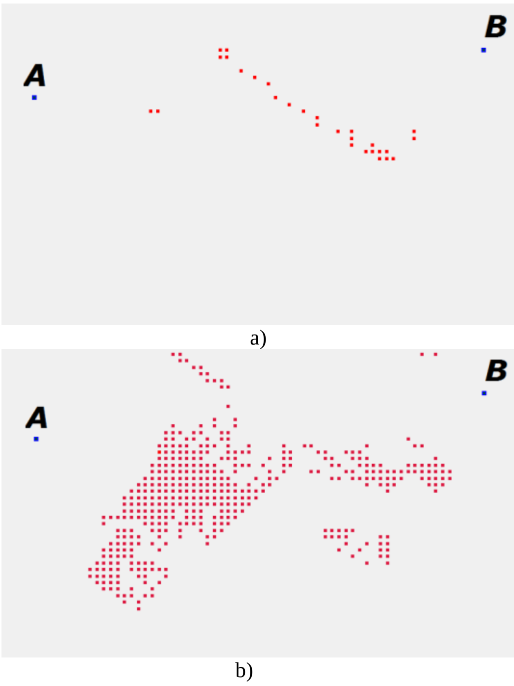

Figure 4. Results of AMR environment mapping: a - with disabled visual system learning b - with enabled visual system learning

VII. Summary

Adaptive intelligent control systems design is one of the key goals of modern robotics. The main distinction of such control systems from the widely used expert systems is their ability of adaptation and prediction via machine-learning methods. The control system of autonomous mobile robot described in our research combines these two features.

The prediction is implemented both on the level of visual data analysis (terrain passability estimation) and inherently at the stage of motion planning on the graph. AMR self-learning process is also two-fold. The visual subsystem learns to classify terrain types, and robot graph motion planning algorithm also learns through the collection of cartographical knowledge.

ACKNOWLEDGMENT

This work was financially supported by the Ministry of Education and Science of the Russian Federation under grant № 2014/112 (project № 994).

REFERENCES

[1] G. Baldassarre and M. Mirolli, (Eds.), *Computational and Robotic Models of the Hierarchical Organisation of Behaviour*, Berlin: Springer-Verlag, 2013.

[2] X. Song, H. Fang, X. Jiao, and Y. Wang, “Autonomous mobile robot navigation using machine learning,” in *Proc. IEEE 6th International Conference on Information and Automation for Sustainability*, 2012.

[3] S. Russell and P. Norvig, *Artificial Intelligence: A Modern Approach*, 3rd Edition, USA: Prentice Hall Press Upper Saddle River, NJ, 2009.

[4] X. Sun, W. Yeoh, and S. Koenig, “Moving target D* lite,” in *Proc. International Joint Conference on Autonomous Agents and Multiagent Systems*, 2010, pp. 67-74.

[5] J. Quinlan, “Induction of decision trees,” *Machine Learning*, vol. 1, pp. 81–106, 1986.

[6] A. Schmidt and K. Walas, “The classification of the terrain by a hexapod robot,” in *Advances in Intelligent Systems and Computing*, 2013, vol. 226, pp. 825-833.

Sergey V. Manko was born in Moscow, Russia on October, 10, 1956. In 2000 he obtained the Ph.D. in Control Engineering from Moscow State Institute of Radio Engineering, Electronics and Automation (MIREA).
Now, he works in position of Professor at MIREA, Moscow. The scope of his current scientific interests includes multi-agent systems and intellectual control in robotics applications.
Mr. Manko is an author of several books on intellectual control systems design and machine learning.

Valery M. Lokhin was born in Moscow on August, 9, 1943. In 1986 he obtained the Ph.D. in Control Engineering from Moscow State Institute of Radio Engineering, Electronics and Automation.
Now, he holds a position of Professor at MIREA, Moscow. The main areas of his scientific activity are automatic control systems and fuzzy inference systems.
Mr. Lokhin is an author of numerous articles in the field of automatic control. He is also an editor of a wide range of books in the field of intellectual control.

Sekou A. K. Diane was born in Moscow on November, 24, 1990. He is a postgraduate student of MIREA working at position of Junior Research Scientist. His major field of study is multi-robot systems. He is also interested in machine learning technologies.
As a part of his scientific work Mr. Diane is responsible for software development and conducting experiments.

Alexander S. Panin was born in Mytishchi, Russia on September, 27, 1995. He is a 3rd year student at MIREA interested in machine learning, specifically neural networks and classification trees.
Mr. Panin works closely with Mr. Diane in the tasks of software design and algorithm implementation.