Self-Organizing Path Planning Control System for a Vehicle

Valentin Dimakov and Vladimir Golovko

Department of Computers and Mechanics,
Brest polytechnic institute, Moscowskaja 267,
224017 Brest, BELARUS
E-mail: VDimakov@mail.ru, cm@brpi.belpak.brest.by,
Fax: +375 162 42 21 27, Phone: +375 162 42 10 81

Abstract

Autonomous mobile robots typically require a preconceived and very detailed navigational model (map) of their intended operating environment. It requires the presence of a priori information. The creating world model of environment is a difficult problem requiring a detailed description of all possible routes of the robot motion. Therefore, it is better to describe a behaviour strategy that robot can create the world model of environment itself during exploration of an unknown territory to achieve efficiently the target from any start position in the future. This strategy assumes full self-organization and self-adaptation to the environment. This paper describes an architecture of such system closely connected with neural network solving the shortest path problem. Such interconnection allows determining the global strategy of the robot behaviour parallel with local strategy formed by reactive navigational system.

1: Introduction

Building navigation systems for mobile robots divided on several fundamental problems: perception and processing of sensor information, reactive control by vehicles and global planning of the behavior strategy.

Building system of the processing of the sensor information assumes the error correction of raw information acting from cheap transducers and combining them in the local environment map. Using the neural network technique allows solving quite effective such problem [1,2].

Together with it, the various neural network models take place at the reactive control by the vehicles. Often the systems of sensor information processing are combined together with reactive control architectures in a single whole [3-5]. Such approach is quite efficiently due to generalization ability of the neural networks.

On other hand, the global strategy of the behavior is oriented on traditional approaches using graph algorithms and on the building Voronoi diagram [6]. Such approaches assume creating detailed world model of the robot safe motion. As algorithm solving shortest path problem the Dijkstra's algorithm and dynamic programming method are used [7,8].

The purpose of this article is to present the key ideas and algorithms underlying research of self-organizing autonomous system to effective control by mobile robots. Existing experience allows defining concepts basing on hybrid technique. In this paper the original approach of building world model of the operation environment is also presented.

2: Principles of the map construction

The planning system consists of two basic subsystems: subsystem of the world model construction and the planning subsystem. The world model is based on indicators placed during exploration of the unfamiliar territory. The navigation system creates indicator as a landmark of the constructed map. As usually, each indicator contains the following information:

- Indicator co-ordinates;
- Serial number of the indicator;
- A set of intervals describing the motion directions formed by the reactive navigation system. Thus,

each interval can contain itself parameters, motion direction and the distance to the next indicator (Figure 1). Hence, the robot moves from indicator to indicator until achievement of the target (Figure 2).

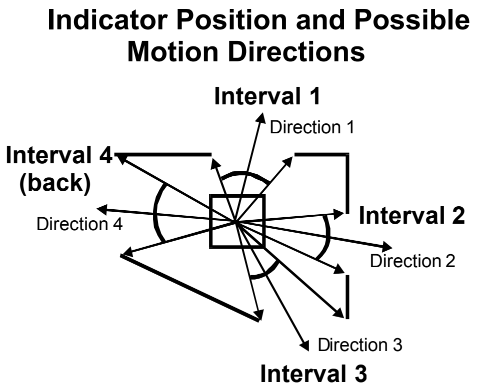
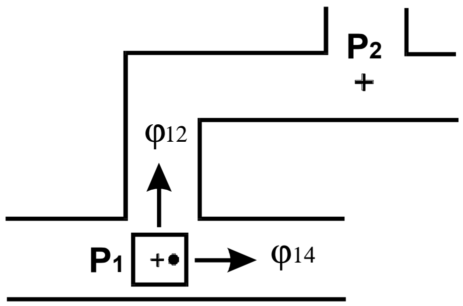


Figure 1.



P_1 – current robot position

P_2 – an indicator

j_{12} – motion direction to the indicator P_2.

Figure 2.

In general the algorithm of territory map formation and planning consists of the following steps:

1. If there are two possible motion direction (for example, forward and back), the mobile robot is controlled only by the reactive navigation system (robot moves along a corridor in the labyrinth).
2. If there are more then two possible motion directions and there is no any indicator in the current robot position, the planning system creates a new indicator and describes the link to previous indicator as a traversed distance, motion direction to achieve the previous indicator etc. The similar link is created for the previous indicator to achieve the indicator in current place of the robot from this previous indicator in the future. The navigation system chooses a direction according to the attraction force directed on the target.
3. If there is an indicator on "intersection of roads" in current robot position and there is at least one unknown motion direction, the navigation system tries to examine this direction because there is possibility to achieve the target using shorter route, but the robot does not known about it yet.
4. If there is an indicator on "intersection of roads" in current robot position and all possible directions are known, the neural networks, solving the shortest path problem, forms an optimal route to the indicator, which is nearest to the target.
5. The indicator is also placed at the presence of the dead-end situation in the labyrinth.
6. If between two known indicators there is no a free path, the links between them are removed, i.e. territory map is continuously modified.
7. All items are repeated on each step of motion.

There is also a problem of interval definition round the indicator (Figure 3) because the robot does not move exactly from indicator to indicator and it is oriented on an attraction region of the indicator. Thus, the robot can perceive the operation environment state being different from described world model for the given indicator. Hence, the navigation system must "foresee" such situation to avoid continuous modifications of the world model.

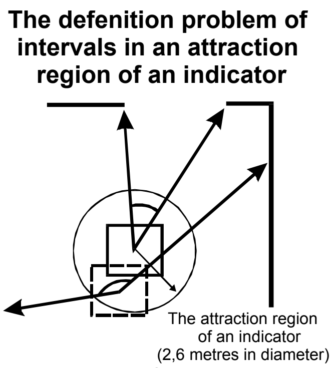


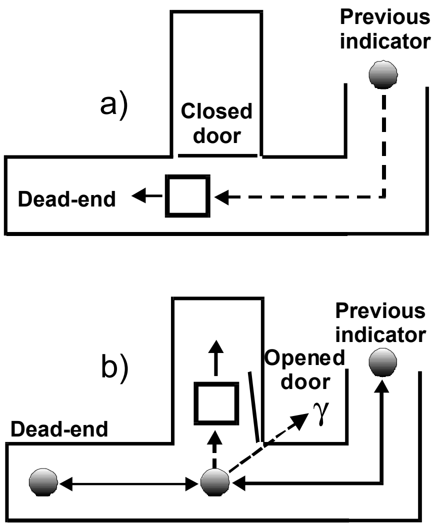
Figure 3. The problem of the interval definition round the indicator.

3: Dead-end situation processing

The main disadvantage of reactive navigation systems is inability to foresee the dead-end situations. Accordingly, the global route planning system must solve this problem. If the robot has already completed world model, the optimal path planning algorithm allows avoiding dead-ends. On other hand, the uncompleted world model does not ensure the successful achievement of the target. Hence it is necessary to have algorithm allowing solving such problem.

In this case we can define the following actions:

1. In normal work mode, the navigation system forms actions directed on the achievement of the target or of the next indicator in the shortest path.
2. While the reactive system detected dead-end situation, the planning system should form actions to achieve the previous indicator.
3. While the robot comes back from dead-end, it can detect new "intersection of roads" (Figure 4) and therefore it will place a new indicator. In this case there can be several possible motion direction and the robot, choosing one of them, may not achieve the previous indicator. Hence the robot must come back to the new indicator and choose other of possible directions.
4. Coming back from dead-end the navigation system can form new routes if environment state is modified continuously.



a) Before detection of the dead-end situation
b) After detection of the dead-end situation and after placing of a new indicator
γ - direction to come out from dead-end situation

Figure 4.

4: The shortest path formation

The neural network solving effectively the shortest path problem has been developed (Figure 5). Conceptually it consists of n layers, where n is a number of the indicators memorized by planning system. Here the 1st layer is selecting: it selects a best route of $n-1$ routes-candidates. All remaining layers form the routes-candidates consisting of $2,3,4 \ldots n$ indicators separately from each other. The experiments with proposed neural network have shown that use of all n layers for a solution of the shortest path problem is an extreme case, because with increase of total number of the indicators, the number of the involved indicators in the path is essentially decreased. Therefore we used the following equation for determination of number of layers of the neural network:

$$E = \min\left\{\left\lfloor 2 \times \sqrt{n} \right\rfloor + 1, n\right\} \qquad (1)$$

where n is a total number of the indicators (key points).

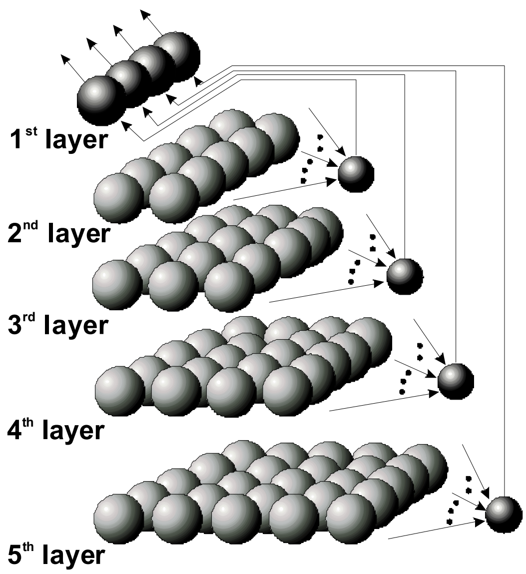


Figure 5. The architecture of the neural network solving the shortest path problem of 5 cities.

One of layers of the network generating a path-candidate is shown in a Figure 6.

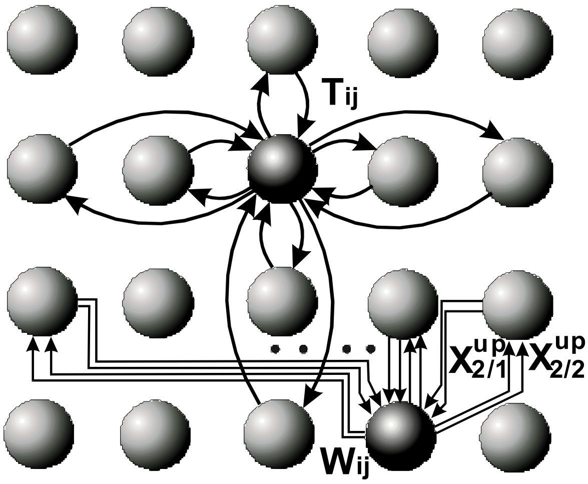
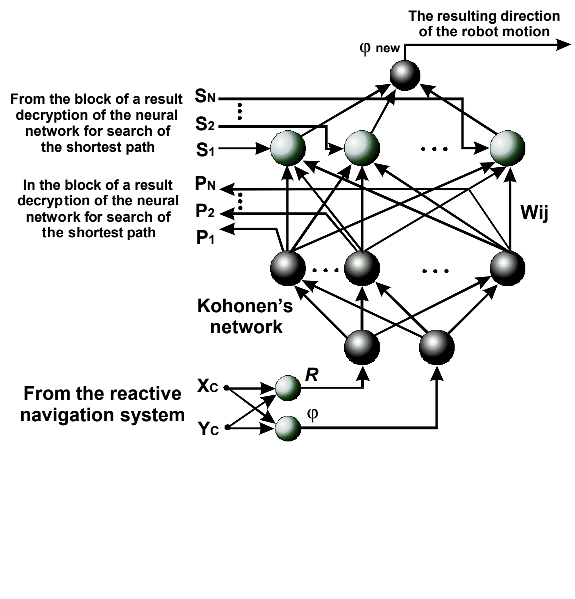


Figure 6. The structure of the layer forming the path-candidate: a connecting schema of neurons by braking ($_{ij}$) and exiting connections (W_{ij}).

It is the relaxation network and forms a best route of the robot motion (according to the Figure 6 it forms a route of 4 indicators), where two of them are fixed in the first and the last row of the layer. Here the row number of the layer determines a position of some indicator in the route, and the column number determines a number of the active indicator. At that only one neuron must be active in each row and in each column.

The system shown in a Figure 7 (simplified schema) for memorizing the indicator location on territory was used. It has a Kohonen's network, which is a single-level memory of the storage system of the key points of the territory, and forms also a final motion direction. Here $\{X_C, Y_C\}$- coordinates of current place of the robot, R and j are polar coordinates of $\{X_C, Y_C\}$, W_{ij} defines a motion direction j_{ij} (Figure 2) to achieve the next indicator from current robot position according to the obtained best route.



R and j are polar coordinates

Figure 7.

The given subsystem is also controlled by the decryption scheme of the shortest route of motion (Figures 8 and 9), which determines the order of indicators following in the route. Here $Y_{W<row><column>}$ defines an indicator chosen by the neural network solving the shortest path problem, $P_1...P_N$ define an indicator in current place of robot $\{X_C, Y_C\}$, $S_1...S_N$ define a next indicator in the shortest route, and $R_{ij} = 1$.

In general the behavior of the planning system has two-level nature: the memory scheme forms an internal form of the operation environment examined by the robot, and the neural network solving shortest path problem forms the global strategy of the robot motion for achievement of the target in real time.

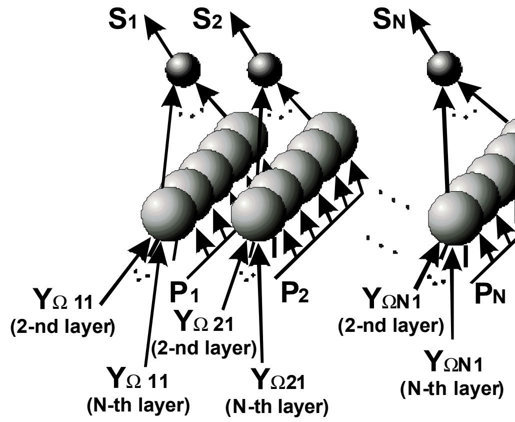


Figure 8. The decryption block of results of the neural network: the connection scheme to the neural network and to the interface scheme.

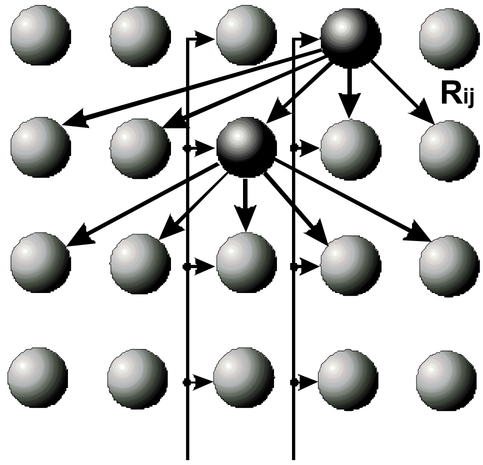
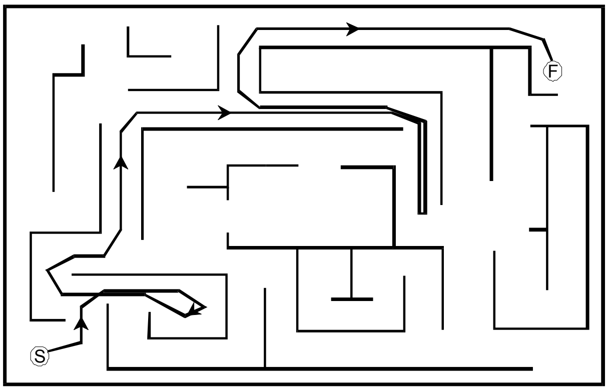


Figure 9. The decryption block of results of the neural network: the scheme of decryption of results.

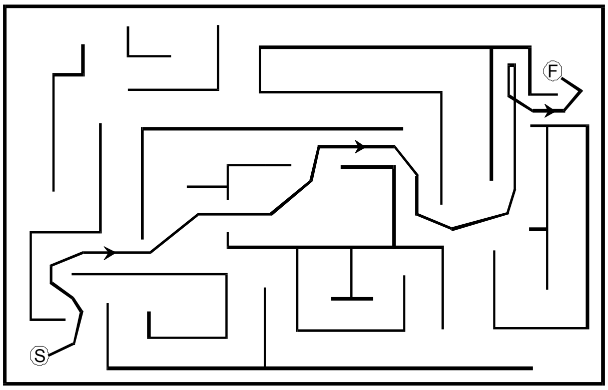
5: Experimental results

The software combining both the reactive system and self-organizing planning system to test the behavior model was used. As an example the labyrinth image shown in Figures 10-12 was taken. On the first stage the world model formed by the navigation system is clear. At first the robot tries to achieve the final point from start position anyhow (Figure 10). It does not have any a priori information about environment and knows about obstacle arrangement within the limits of sensor device detection (about three meters). During examination of the unfamiliar environment the navigation system places indicators on "intersection of roads". After first travel the only one possible path detected to achieve the target and it does not cover all possibilities to search an optimal route. During motion in the next time (Figure 11) the robot extends itself knowledge about environment and therefore it can form shorter route like a person in an unfamiliar city. In the end the navigation system creates a "web" of indicators describing fully the operation environment (Figure 12). If there are changes of the obstacle arrangement, they are emerged and the navigation system modifies itself the environment representation.

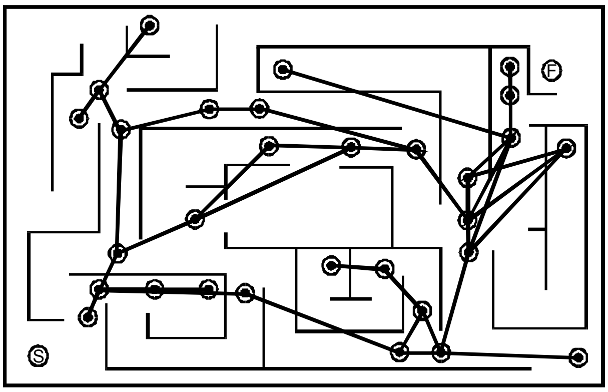


S - start point of motion;
F - final point of motion
Figure 10.

The minimal hardware requirement for real-time simulation for the neural system is PC-computer on the basis of Intel 486DX-100MHz microprocessor. However because of increase of the territory map size the computation delay at the optimal path planning is also increased because of simulation of parallel computations.



S - start point of motion;
F - final point of motion
Figure 11.



S - start point of motion;
F - final point of motion
Figure 12.

6: Conclusions

In this paper the self-organizing optimal route planning system for control of an autonomous mobile robot was considered. It allows to form territory map of an unfamiliar environment, to carry out modification of itself representation of examined environment, to plan optimal route and to control a vehicle in real-time. The proposed approach allows to control by a vehicle in difficult operation environments and at that the system defines the robot behavior without a teacher. At present time the research is carried out to increase robustness of the proposed navigation system.

Acknowledgments

The given work is carried out within the framework of the project INTAS-BELARUS-97-2098 "Intelligent neural system for autonomous control of a mobile robot". The author thanks European Union for the informational and financial support.

References

1. A. Harner and P. Gaudiano. A Neural Model of Attentive Visual Search // Proceedings of the International Conference on Vision, Recognition, Action: Neural Models of Mind and Machine, Boston, MA, May 1997.
2. S. Martens, G. A. Carpenter, P. Gaudiano. Neural sensor fusion for spatial visualization on a mobile robot// SPIE Proceedings, 1998, Vol:3523, pp. 100-111.
3. J. R. Millan. Reinforcement learning of goal-directed obstacle-avoiding reaction strategies in an autonomous mobile robot// Robotics and Autonomous Systems, Elsevier, Vol 15, No. 4, Oct 1995, pp. 275-299.
4. S. Thrun. An approach to learning mobile robot navigation// Robotics and Autonomous Systems, Elsevier, Vol 15, No. 4, Oct. 1995, pp. 301-319.
5. P. Gaudiano and S. Grossberg. Vector Associative Maps: Unsupervised Real-Time Error-Based Learning and Control of Movement Trajectories// Neural Networks, 4, 1991, pp. 147-183.
6. S. Thrun, A. Bucken, W. Burgard, D. Fox, T. Frohlinghaus, D. Henning, T. Hofmann, M. Krell, T. Schmidt. Map Learning and High-Speed Navigation in RHINO // AI-based Mobile Robots: Case studies of successful robot systems, MIT Press, D. Kortenkamp, R.P. Bonasso, and R.R. Murphy (eds), invited paper.
7. M.N.S. Swamy and K. Thulasiraman. Graphs, Networks, and Algorithms// A Wiley Interscience Publication, John Wiley&Sons, New York-Chichester-Brisbane-Toronto, 1981.
8. Y.M. Korshunov. Mathematical principles of cybernetics// Energoizgat, Moscow, 1987.