Mobile Robot Navigation using Reinforcement Learning based on Neural Network with Short Term Memory

Andrey V. Gavrilov [1] , Artem Lenskiy [2]

[1] Dept. of Production Automation in Machine Engineering
Novosibirsk State Technical University, Karl Marx av., 20,
Novosibirsk, 630092, Russia
Andr_gavrilov@yahoo.com

[2] School of Electrical, Electronics & Communication Engineering,
Korea University of Technology and Education,
1800 Chungjeol, Byeongcheon, Dongnam Cheonan, 330-708, Korea
a.a.lensky@gmail.com

Abstract. In this paper we propose a novel bio-inspired model of a mobile robot navigation system. The novelty of our work consists in combining short term memory and online neural network learning using history of events stored in this memory. The neural network is trained with a modified error back propagation algorithm that utilizes reward and punishment principal while interacting with the environment.

Keywords: neural networks, mobile robots, reinforcement learning.

1 Introduction

For the past decades a number of neural network based approaches have been suggested for mobile robots navigation. The early works are dated back to 1975, 1986 and were conducted by N.M.Amosov [1] and R.Brooks [2] correspondingly. Short review of this topic may be found in [3]. The key issue in mobile robot navigation is to design a system that allows robot to autonomously navigate in unstructured, dynamic, partially observable, and uncertain environments. The robot navigation problem can be divided into the following tasks: map building, localization, path planning, and obstacle avoidance. Some of these problems can be solved by applying approaches based on neural networks. One of the most popular approaches is based on a multilayer perceptrons (MLP) that is trained with the error back propagation (BP) learning algorithm. The disadvantages of approach based on the BP learning algorithm are in its complexity, slow training and orientation on supervised learning. Moreover, in the case when a part of MLP should be retrained, the whole training processes should be repeated. Janglova [4] attempted to overcome some of these shortcomings by proposing a multilayer hybrid neural network with preprocessing that utilize principle component analysis (PCA). His solution reduces the time needed for MLP training. However, it does not resolve the remaining disadvantages. A.Billard and G.Hayes [5] suggested a model for mobile robot navigation (DRAMA) that is based on recurrent neural network with delays. This is probably the first model where an attempt to develop a universal neural network as a part of control system to navigate in uncertain dynamic environment was made. However, the model was mainly oriented on quite simple binary sensors for events detection.

An unsupervised learning model for map building based on adaptive resonance theory (ART) [7] was proposed by Araujo [6]. Another model for robot navigation was suggested in [8]. The robot model was able to receive commands on a natural language and analyze graphical image of the environment to take decisions for further movement. The system was extracting simple image features sensitive to spatial transformations. Therefore, the same object observed from different viewpoints generated distinct features. As a result the number of features representing the same object was too high, leading into a great number of neurons. To overcome this drawback a multi-channel cognitive model was proposed [9] and evaluated for solving a minefield navigation task. To classify objects it was proposed to create a separate ART models. However, similar problem affects this approach. With the increasing number of object groups the number of ART models increases slowing down the performance.

On other hand there are methods that are specifically designed for visual navigation. One recent approach [10-12] proposed to extract image features that are robust to spatial transformations. Thus, feature vectors remain the same for greater variations in a view angle, consequently keeping the number of neurons low.

Another way to reduce the number of neurons in ART model was proposed in [13-16]. The idea behind this method consist in preprocessing input vectors the multilayer perceptron which goal is to reduce the sensitivity of ART-2 model to spatial image transformations. However, the simulation experiments showed that this model (MLP-ART2) is more suitable for finding differences in image sequences. It also very depended on parameters of ART-2 and MLP, and selection of them is a complex and nontrivial task.

Gavrilov et. al [17] suggested to combine MLP-ART2 with the reinforcement learning which is based on modified error back propagation algorithm or Generalized Error Back Propagation (GEBP).

In this paper we propose a novel bio-inspired model of a mobile robot navigation system that combines MLP, short term memory and reinforcement learning with GEBP. The difference between the model presented in [17], and the proposed model is in the history of events that is used in learning of MLP when the robot gets either a reward or a punishment. Also the in the proposed model there is no ART-2 model that is associated with specific learning problems. It is also easier to train the proposed model. It may be seen as universal bio-inspired reinforcement learning in contrast to popular probabilistic approaches which are more abstract and mathematical based.

2 Proposed Architecture of Control System

The proposed control system for navigation of a mobile robot is shown in fig. 1. The proposed system consists of the following blocks: critic, short-term memory, FFNN trained with GEBP algorithm. The short-memory block store a few recent pairs of input and output vectors of the network. The critic block utilizes pairs of input and output vectors stored in short-term memory to train the network. Those vectors that represent collisions with obstacle are used as negative samples (punishment) and those that represent visible target are positive samples (reward).

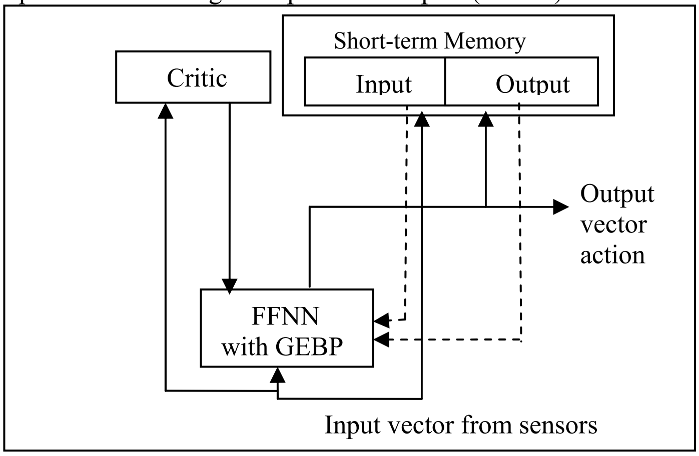


Fig. 1. Proposed Architecture.

Below example of navigation algorithm learning to achieve target in unknown environment is shown using this architecture and generalized error back propagation algorithm. There *Estimation* is produced by critic as *none*, *reward* or *punishment*, *direction_to_turn* for control of motion is produced by neural network or simple logical algorithm (at short distance to obstacle). Functions WorkNN and LearnNN aims to produce *direction_to_turn* and to learn neural network with respect to *Estimation* correspondingly. Random behavior sometimes is needed to reduce possibility of cycling in motion, in particular, using untrained neural network in first time. Sensor vector and selected decision (randomly or by neural network) is stored in short-term memory and when critic generates reward or punishment this information is used for training of neural network. At that the influence of stored decisions depends on order in time. Procedure Move provides turn and one step of forward motion.

Algorithm of robot behavior

```
While target_is_not_achieved
        Get values_from_sensors;
        Calculate distance_to_obstacle_in_front;
        Assign none to Estimation ;
        If target_is_visible
          Then
```

```
            Assign reward to Estimation;
            Direction_to_turn := Direction_to_target;
      End if
      If distance_to_obstacle_in_front < Threshold distance
          Then
            Assign punishment to Estimation;
            Calculate direction_to_turn from obstacle by simple
                logical algorithm;
          Else
               Calculate random_value;
               If random_value < Probability of random behavior
               Then
                       Assign random direction to Direction_to_turn;
                Else
                Direction_to_turn:=WorkNN(values_from_sensors);
               End if
      Storing of sensor vector and selected direction
                   (input-output) in short-term memory;
       End if
       Move(Direction_to_turn);
       Current_situation := last situation in memory;
       $r$ := 1;
       If Estimation is reward or punishment then
            While not_all_memory_is_tested
                 LearnNN(Current_situation, Estimation);
                 $r$ := $r/2$;
                 Current_situation := previous situation in memory;
            End while
       End if
 End while
 End of algorithm of robot behavior
```

The procedure LearnNN uses Generalized Error Back Propagation learning algorithm described below.

3 Generalized error back propagation (GEBP) algorithm providing learning based on positive and negative samples

Our Generalized Error Back Propagation (GEBP) is based on two modes of EBP – positive or negative respecting for attraction and repulsion of target output vector. Positive mode of this model is classic EBP. The negative mode provides update of weights with opposite sign. Thus updates of weights in GEBP are described as follows:

$$\Delta w_{ij} = ar\varphi_j x'_i \, , \qquad (1)$$

where:
w_{ij} is weight of connection between *i*th neuron and *j*th neurons;
a is value of reward, 1 or -1;
r is a rate of learning;
φ_j is error propagation for *j*th neuron;
x'_i is derivative of active function of *i*th neuron.

Function φ_j for calculation of error propagation for output layer differs from same function in usual EBP algorithm. For case a=1 it is same as in EBP classic algorithm:

$$\varphi_j = y_j(1-y_j)(d_j - y_j)\,, \tag{2}$$

where y_j and d_j are actual and desirable output of neuron respectively.

For case a=-1 the function φ_j is determined as

$$\varphi_j = ky_j(1-y_j)\exp[-\frac{1}{2\sigma^2}(d_j - y_j)^2] \tag{3}$$

The expression $y_j(1-y_j)$ of this formula represents the derivative of neuron's state like in usual error back propagation. The exponential function in this formula provides maximal value of φ_j at equality of actual and desirable states of *j*th neuron. Value σ represents the sensitivity in neighborhood of danger (undesirable) output vector. Coefficient k may be interpreted as a level of timidity and may be connected with simulation of emotions.

Another variant of calculation of φ_j is possible.

For $d_j \neq y_j$ function φ_j may be defined as

$$\varphi_j = \frac{ky_j(1-y_j)}{d_j - y_j} \tag{4}$$

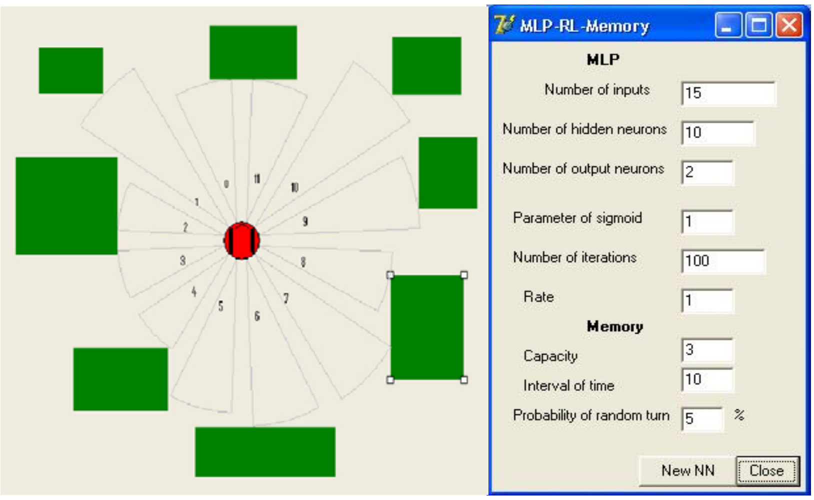
For $d_j = y_j$ φ_j may be determined as constant value k.

Unlike classical EBP with positive reward the punishment in GEBP provides adaptation of weights for repulsion of target output vector (the vector associated with any danger or collision). Particular case is learning to predict of events in time. In this case MLP may be replaced by recurrent neural network dealing with sequences of patterns, e.g. Elman model [18] with EBP through time.

4 Simulation

The simulation of the proposed algorithm and navigation system has been done using the Mobile Robot Simulator (MRS) developed for 2D-simulation of mobile robots in simplified environment with rectangles as obstacles, discrete time and step-type motion of robot [16]. Task solving by the robot is to find the path to a target without knowing the map.

In our experiments we use 12 range sensors (Fig. 2a) for estimating distances to surrounding obstacles. Besides them input of the neural network includes direction of robot motion and its coordinates. Therefore, input vector consists of 15 components. We use a neural network with two outputs. Direction of the robot movement is calculated as a difference between two outputs.



MLP

Number of inputs 15

Number of hidden neurons 10

Number of output neurons 2

Parameter of sigmoid 1

Number of iterations 100

Rate 1

Memory

Capacity 3

Interval of time 10

Probability of random turn 5 %

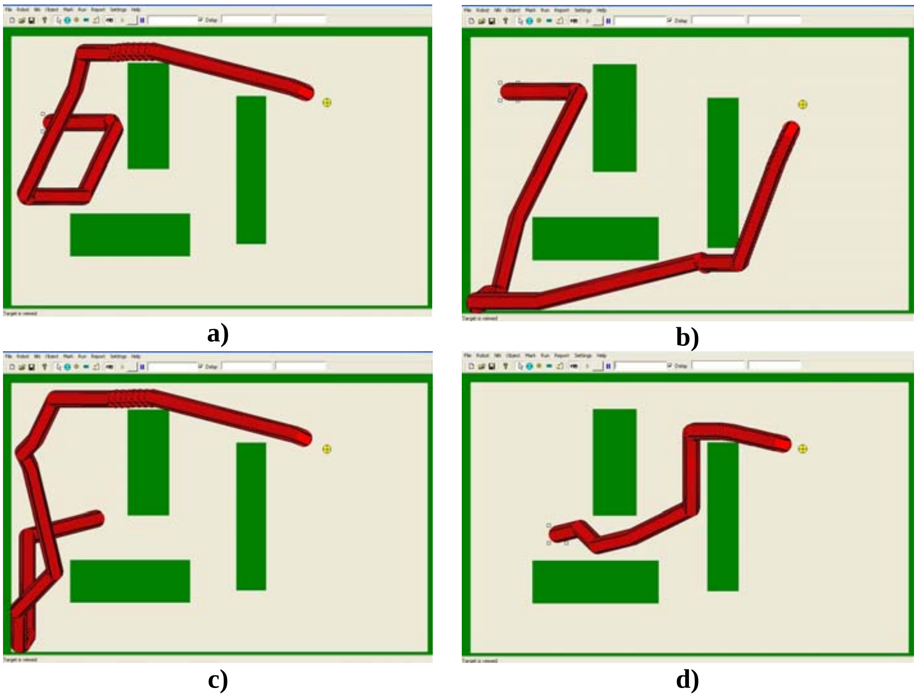
a) **b)**

Fig. 2. a) Range sensors in MRS and b) parameters of simulation.

In our previous simulations we used a network with one output, which often led to the problem of getting into a loop, especially in the beginning of the traverse when the network is untrained.

Fig. 2b shows a form with main parameters of the neural network and the short-term memory that are used in the experiments. The parameter “Interval of time” determines the number of steps of motion that is necessary to make a decision either by the neural network or using short-term memory that is limited by the parameter “Capacity”.

Some experiments are shown in Fig. 3. There are series of experiments with the same environment and same neural network that is trained once and permanently..



a) b) c) d)

Fig. 3. Simulation examples, a),b),c) and d) are first, second, third and fourth series. The robot's trajectory is show in red, obstacles are green and the target is yellow.

5 Conclusion

Conducted experiments show that based on the proposed navigation algorithm, robot behaves appropriately in an unknown indoor environment. The advantage of this model compared to classical reinforcement learning is the absence of a necessity of preliminary knowledge on the environment and discretization of space.

Our model is based on universal bio-inspired reinforcement learning in contrast to popular probabilistic approaches which are more abstract and mathematics based.

We are planning to test our model in real environment with different parameters. Besides we are planning to combine the proposed navigation algorithm with the algorithm suggested in [19, 20] that uses natural language to train and control a mobile robot.

References

1. Amosov N.M., Kussul E.M. Fomenko V.D.: Transport robot with a neural network control system. In: Advance papers of the Fourth Intern. Joint Conference on Artificial intelligence, vol. 9, pp. 1--10, (1975).

2. Brooks R.: A robust system layered control system for a mobile robot. IEEE Trans. on robotics and automation, RA-2, pp. 14--23, (1986).
3. An-Min Zou, Zeng-Guang Hou, Si-Ya Fu, Min Tan: Neural Network for Mobile Robot Navigation. A Survey. In: Proc. of Int. Symp. on Neural Networks ISNN-2006, LNCS, vol. 3972, pp. 1218--1226. Springer-Verlag, Berlin Heidelberg New York (2006).
4. Janglova D.: Neural Networks in Mobile Robot Motion. Int. J. of Advanced Robotic Systems, vol. 1(1), pp. 15--22, (2004).
5. Billard A. Hayes G.: DRAMA, a Connectionist Architecture for Control and Learning in Autonomous Robots. Adaptive Behavior, 7(1), pp. 35--63 (1999).
6. Rui Araujo: Prune-able fuzzy ART Neural Architecture for Robot Map Learning and Navigation in Dynamic environment. IEEE Trans. on Neural Networks, vol. 17(5), pp. 1235--1249, (2006).
7. Carpenter G.A. Grossberg S.: Pattern Recognition by Self-Organizing Neural Networks, Cambridge, MIT Press, MA (1991).
8. Gavrilov A.V., Gubarev V.V., Jo K.-H. Lee H.-H.: An architecture of hybrid control system of mobile robot. Mechatronics, Automation, Control., 8, pp. 30--37 (2004).
9. Ah-Hwee Tan: FALCON: A fusion architecture for learning, cognition and navigation. In: Proc. of IEEE Int. Joint Conf. on Neural Networks IJCNN04, vol. 4, pp. 3297--3302, (2004).
10. Lenskiy, A.A. and J.-S. Lee. Rugged terrain segmentation based on salient features in International Conference on Control, Automation and Systems 2010 2010. Gyeonggi-do, Korea
11. Lenskiy, A.A. and J.-S. Lee. Terrain images segmentation in infra-red spectrum for autonomous robot navigation in IFOST 2010. 2010. Ulsan, Korea.
12. Lenskiy, A.A. and J.-S. Lee, Machine learning algorithms for visual navigation of unmanned ground vehicles, in Computational Modeling and Simulation of Intellect: Current State and Future Perspectives, B. Igelnik, Editor 2011, IGI Global.
13. Gavrilov A.V. Hybrid neural network based on models Multi-Layer perceptron and Adaptive Resonance Theory. In: Proc. of 9th Int. Russian-Korean Symp. KORUS-2005, pp. 604--606, NSTU, Novosibirsk, (2005).
14. Gavrilov A.V., Lee Y..-K., Lee S.-Y.: Hybrid Neural Network Model based on Multi-Layer Perceptron and Adaptive Resonance Theory. In: Proc. of Int. Symp. on Neural Networks ISNN06, LNCS, vol. 3972, pp. 707--713. Shpringer-Verlag, Berlin Heidelberg New York (2006).
15. Gavrilov A.V., Lee S.-Y.: An Approach for Invariant Clustering and Recognition in Dynamic Environment. In: Advances and Innovations in Systems, Computing Science and Software Engineering (Ed. Khalet Elleithy), pp. 47-52. Springer, Heidelberg (2007).
16. Gavrilov A.V., Lee S.-Y.: Usage of Hybrid Neural Network Model MLP-ART for Navigation of Mobile Robot. In: Int. Conf. on Intelligent Computing ICIC'07, China, LNAI, vol. 4682, pp. 182-191. Springer-Verlag, Berlin, Heiderberg, (2007).
17. Gavrilov A., Lee S.-Y.: Unsupervised hybrid learning model (UHLM) as combination of supervised and supervised models. In: IEEE Int. Conf. SMC UK&RI, Dublin, (2007).
18. Elman J.L.: Distributed representations, simple recurrent networks, and grammatical structure. Machine Learning, 7(2/3), pp. 195-226 (1991).
19. Gavrilov A. V.: Context and Learning based Approach to Programming of Intelligent Equipment. In: The 8th Int. Conf. on Intelligent Systems Design and Applications ISDA'08, pp. 578-582. Taiwan, (2008).
20. Gavrilov A.V.: New Paradigm of Context based Programming-Learning of Intelligent Agent. In: Proc. of 1st Workshop on Networked embedded and control system technologies. In conjunction with 6th Int. Conf. on Informatics in Control, Automation and Robotics ICINCO-2009, Milan, pp. 94-99, (2009).