F.S.A.I., Vol. 10, Nos. 1–2, pp. 31– 40, 2004

Genetic Adaptation of Weightless Neural Networks For Obstacle Avoidance in Mobile Robot Control

Ionel Bostan*, Emil Sofron*, Silviu Ioniţă*, Valeriu Ionescu*

* Faculty of Electronics and Electromechanics, University of Pitesti

Abstract: This paper presents a solution for using a genetic algorithm in the determination of the content of the RAM memories found in the structure of a weightless neural network. The solution aims to develop the behaviour of obstacle avoidance for an autonomous mobile robot that has behavioural control architecture.

1 Introduction

The architectures for mobile robot control, although started from the modularity concept, have different approaches concerning the connections between the resulting modules [1].

The hierarchical decomposition approach assumes that highest level poses an abstract model of the world. Based on this model, each action of the mobile robot is a three-step process: perception – reasoning – control. This control architecture is not well suitable for real time response [1].

The behavioral decomposition approach does not use [1]abstract models. The behavioral modules are connected in parallel and the robot's answer is not a result of a decisional process – but rather a conditioned answer to the sensorial signals, that is *a reflex* [1].

The hybrid architecture combines the advantages of both previous architectures: is behavioral at lower levels (where a real time response is needed) and decisional at top level (where a decisional process is needed) [1].

Artificial evolution can be used in different ways in the field of behavioral robot control, and this area of research is called *Evolutionary Robotics*. Even if this term was recently introduced by Cliff, Harvey and Husbands [2], the idea of representing the command system through an artificial chromosome, subject to the laws of natural selection, appeared in the early '80.

The research's effort in the field of evolutionary robotics is headed towards developing the genetic behavioral modules implemented with neural networks.

The basic idea in this field consists in the random generation of an artificial chromosome population that interacts in the test environment [3], [4]. Each chromosome represents (in an encoded form) the robot's command system, and its morphology. Each robot, real or simulated, is acting free in its environment, according to its mission. During the testing period, the robot is evaluated according to several performance criteria, in order to establish the fitness (a scalar meaning how well a robot performs a certain task). After all individuals have been tested, the chromosomes with the highest fitness are reproduced, in order to achieve even

---

A preliminary version of this paper has been presented in the ICCC2004 Conference and has been printed in the Proceedings of ICCC2004

better individuals. This process is repeated over many generations until an individual that better satisfies the performance criteria is obtained.

2. Weightless Neural Networks

The digital artificial neural networks are often used in practical applications, due to their advantages: faster training, higher processing speed, ease to implement in hardware.

A particular case in digital neural networks is obtained when the connections between the nodes have all weights equal to 1. These networks are called Weightless Neural Networks, Boolean Neural Networks, or Neural Networks with RAM Memory. Because of the lack of weights, these networks work with binary inputs and outputs, and the neuron's function is stored in a look-up table that can be easily implemented in software or in hardware.

In order to determine the exact content of the RAM memory for each neuron of the neural network, one of the following methods can be used:

1. Training before actual use – the neural networks are subject to a learning process for the common circumstances that can appear in normal robot's navigation.
2. Training before actual use and continuous refining of the memory contents during robot's use.
3. Using an artificial evolution algorithm with a role in determining the content of the RAM memories.

In robot control, using Weightless Neural Networks is especially attractive because these ensure a great flexibility, a good modularity, a parallel implementation of the control functions and a fast learning rate. All these characteristics lead to simpler control architectures that can be implemented with common electronic circuits.

2.1 Weightless Neural Networks Model

A typical model for a RAM neuron is presented in Figure 1.a. We can see that the actual neuron is formed of a small RAM memory. The neuron's N inputs are actually an address and the output is given by the content of the addressed memory. The RAM memory size is 2^N, so it is given by the number of inputs.

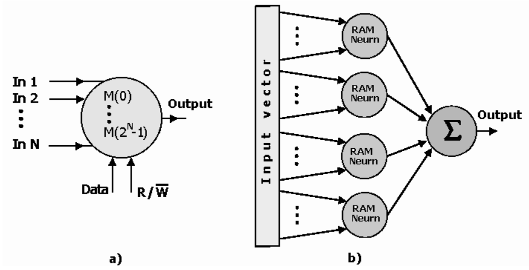


Figure 1.-Weightless neural networks: a) Model of a RAM neuron; b) Diagram of WIESARD discriminator.

Thus, there is possible to implement any binary function that depends on the inputs combination.

The *R/W* is used to differentiate the training stage – case where data is written in neuron's memory, and the interrogation stage – case where data is read form memory. The *Data* input is used in the training stage, for accepting the data that will be learned.

2.2 The WIESARD discriminator

Typically, a Weightless Neural Network has a single layer structure and must have enough neurons to cover the binary vector from the network's input. Studies showed that there is a relationship between the input vector's size K, the number of inputs per neuron N, and the number of neurons used J:

$$J * N \geq K \tag{1}$$

If the outputs of such network are connected to an adding circuit, with the role of counting the active neurons, a WIESARD discriminator is created. The typical diagram of such a discriminator is presented in Figure 1.b.

A system for input data classification can be achieved if several WIESARD discriminators (one for each class of objects that needs to be recognized) are grouped together with a decisional block, working on the principle "the winner takes all". According to this principle, the discriminator with the highest output value is considered winner and the input vector is classified as being of the type represented by the winning discriminator. The principle diagram of a WISARD system is presented in figure 2.

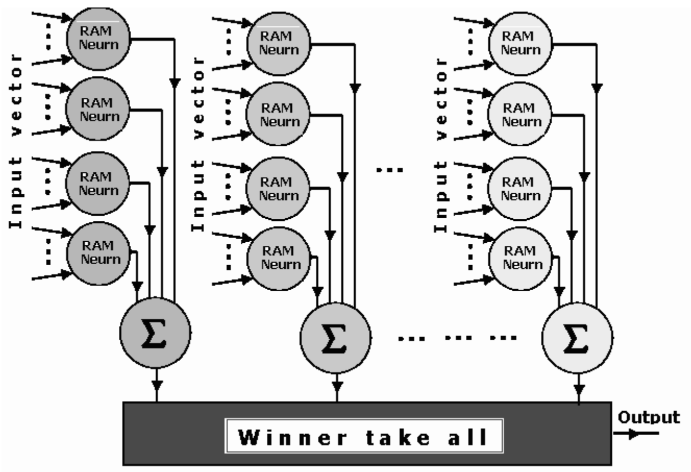


Figure 2.- Tipicaly diagram of WISARD system

We notice that by combining simple elements (that are not capable of generalization) it is possible to obtain a system capable to classify never-seen-before input information, based on the knowledge acquired in the previous training step.

3. Designing of the control system

In mobile robot control, the WISARD discriminator is useful if we take into account that the control process can be considered as a classification of sensorial information in order to assign an action from a predefined set.

The actual usage of a weightless neural network, with genetic adaptability, in mobile robot control for obstacle avoidance will be presented in the next. The robot has 8 infrared sensors and 2 independent tractor wheels. The application is „obstacle avoidance" because of its simplicity, and because it is an important task of any mobile robot control architecture. For this purpose we use a weightless neural network, with genetic adaptability, in mobile robot control for obstacle avoidance. The robot has 8 infrared sensors and 2 independent motor wheels (differential drive). We chose „obstacle avoidance" behavior because of its simplicity, and because it is an important part of any mobile robot control architecture.

3.1 Control architecture

The control architecture used it is a behavioral one, and it is composed of 3 distinct modules, connected as seen in Figure 3. Each module has a well defined role and is encoded in the chromosome by a certain sequence of bits having a preliminary defined length.

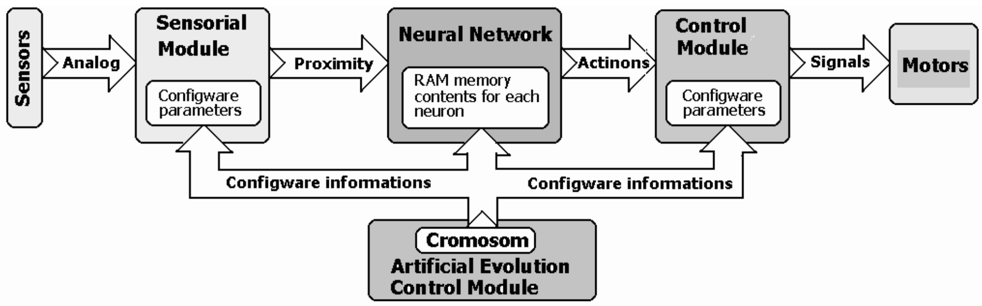


Figure 3.- The control system architecture for a autonomous mobile robot

3.2 Genetic representation of the sensorial system

The placement of the 8 infrared proximity sensors and the genetic representation adopted are presented in Figure 4. There are two bits assigned to each sensor: one that shows the presence/absence of the obstacle, and one used by the evolutionary algorithm for validating/blocking the sensor's working. The validation bits allow the evolutionary algorithm to establish the minimum number of sensors needed and their placement according to the robot's motion direction.

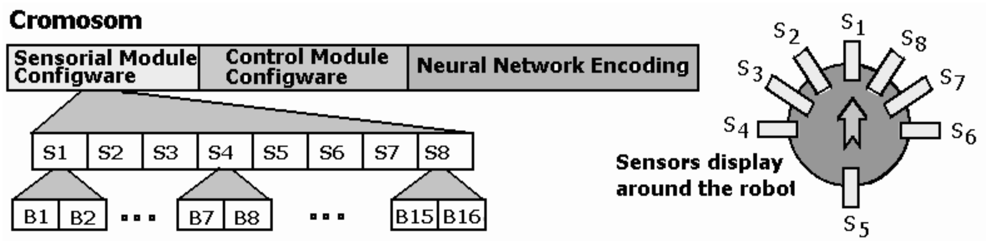


Figure 4 - Genetic representation of the sensorial system and the placement of the onboard sensorial system

3.3 Genetic representation of the neural network

The neural network is responsible with the generation of the elementary actions, according to the sensorial context where the robot is in.

For mobile robot control, a wide range of motion types can be obtained by combining 8 elementary actions: forward move- slow; forward move –medium; forward move – high speed; steer left wide; steer left tight; steer right wide; steer right tight and stop. As the mater of fact, the moving has different levels of speed. One way to achieve the command system based on a weightless neural network is presented in Figure 5. There are 8 WISARD discriminators used (one for each elementary action).

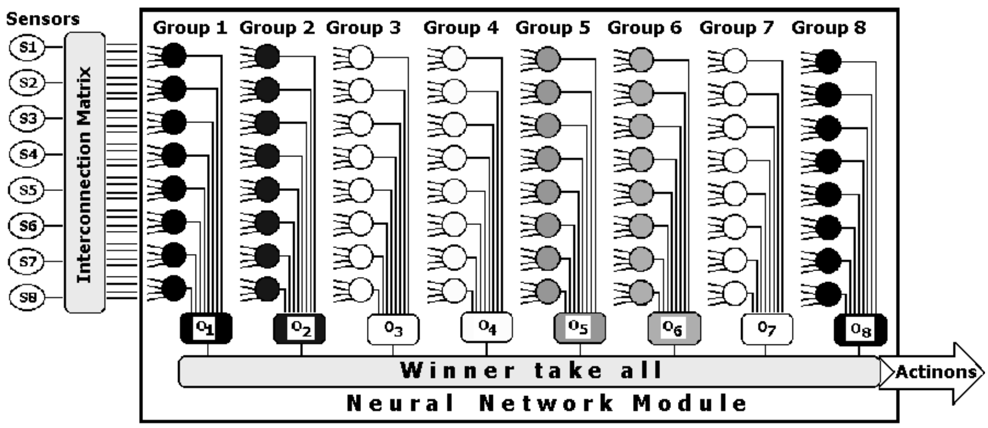


Figure5 - Neural Network Module (for 8 elementary actions)

In order for such a network to be useful it is necessary to address certain requests:

1. Good coverage of the input range – meaning that it needs a high enough number of inputs, some of the sensorial system outputs being connected to the inputs of several neurons present in the network.

2. Saturation avoidance – learning to associate correctly the sensorial input to the required action. For this each group must have a high number of neurons

For hardware and software implementation ease in the following experiments, the following particular connections were made:

– all neurons have the same number of inputs;
– all neurons with similar position in the groups are connected to the same inputs

Connecting the outputs of the sensorial system is done through a connection matrix, randomly initialized at the beginning of the evolutionary process. The modular character of this network allows rapid extension of the elementary action set, or the extension of the sensorial system.

The genetic representation of the neural network presented in Figure 5, is performed by adding the content of the RAM memory of each neuron, in their discriminators order, as seen in Figure 6.

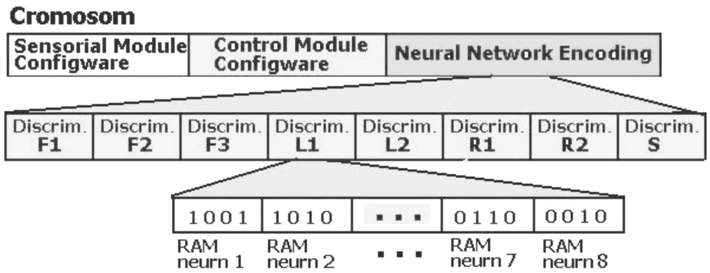


Figure 6 - Genetic encoding of the weightless neural network for obstacle avoidance development

3.4. Genetic encoding of robots actions

We have already shown that the neural network's role is to choose the best action from the predefined set for each sensorial context. Because there are three different degrees for speed, a supplementary module called control module was needed. That will transform in numerical values the meaning of the terms: slow, medium and fast.

For the genetic coding of this module there are 10 bits assigned, because we chose to divide the engine's speed range in 10 distinct intervals. The genetic coding of this module is presented in Figure 7. This encoding is adopted for ensuring that fast speed is indeed faster than medium, and the latter is faster than slow speed:

– fast speed is computed by adding all the 10 bits resulting in a number in the 0-10 range;
– medium speed results from adding the first 6 bits;
– slow speed results by adding the first 3 bits resulting in a number in the 0-3 range.

For any sensorial context, the neural network is responsible for choosing the best action and the control modules is responsible for establishing the actual execution speed. For this to happen, it computes a speed index ranging from 0 to 3 for slow speed, 3 to 6 for medium speed and 6 to 10 for fast speed.

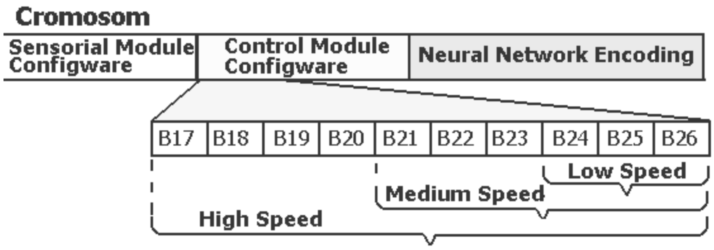


Figure 7 - Genetic encoding of the speed level for each elementary action

It is important to note that all three speed indexes are maintained unaltered during of entire individual's life. These indexes are modified only by the evolutionary process, so they will only be modified from one generation to another.

The indexes for speed and the chosen action are used to establish the references for the speed of the two electrical motors.

3.5. The connection matrix

The neural network has an input number considerably greater than the number of sensors that is why each sensor can be connected to one or more of the network's inputs. Where the actual connection will be done is specified through a connection matrix, randomly generated at the beginning of the evolutionary process.

3.6. The Genetic Algorithm

As was already shown, in the chromosome are encoded the neural network and elements form the robot's morphology (number and sensor position, speed execution levels). Each of the three sections of the chromosome is formed of binary sequences.

Defining the genetic operators did not raise any problem, because the chromosome resulting from the control system's encoding is also a binary sequence. All this are leading to the possibility of using the standard form, proposed by Holland.

Determining a simple and efficient evaluation function is the key step in the successful solving of any problem using artificial evolution. In the case o evolutionary robots, the evaluation function is responsible for the robot's behavior, and must be designed in a way that satisfies the requests that sometimes are even contradictory.

The major difficulties that must be solved in the process of evaluation function design have different causes, the most important being [4]:

- the need to use only local sensorial information (aquired by the robot's sensors);
- the local character of the sensorial information; the complexity of the missions that the autonomous robot has to perform; the limited time to test the individuals;
- the risk of learning the geometry of the test lab and not the desired behaviors; etc.

In many cases, it is easier to design this function than create a program that must control the entire application.

Unlike the standard case, where the evaluation is performed in the Cartesian space given by the [4], [5], [6] and [7], we chose the evaluation in the sensorial space. The method works this way: for each possible combination in the sensorial space, a comparison is made between the action proposed by the tested individual and the action given by a reference control system. If the actions are identical, the individual's fitness value is raised by 1. For a system with 8 proximity sensors, the sensorial space consists of 256 possible combinations. In this case, the fitness value cannot exceed this value.

Testing all the sensorial combinations allows the evolutionary algorithm to develop a neural network that gives the robot the same behavior in any environment, independent of the obstacle's number and position or the environment's geometry. This method has a global character because there are not any other situations but those verified.

The reference control system contains the robot's wanted answer. In our simulations the reference system is formed of a manually selected rule base.

Such rule base can be organized as follows:

1. **IF** (S1=1 and S6=1) **THEN** command = steer left tight
2. **IF** (S1=1 and S4=1) **THEN** command = steer right tight
3. **IF** (S6=1 and the rest is 0) **THEN** command = steer left
4. **IF** (S4=1 and the rest is 0) **THEN** command = steer right
5. **IF** (nr_left<nr_right) **THEN** command = steer left
6. **...**

4. Simulation results

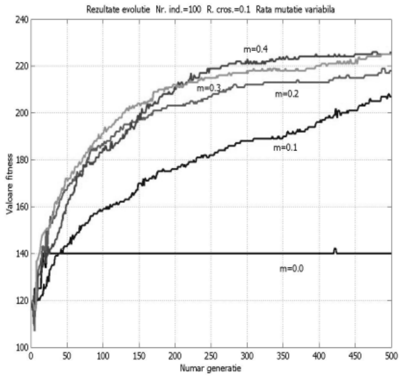
The simulation results were obtained for a weightless neural network characterized by:

- 8 neuron groups (8 WIESARD discriminators);
- each neuron group is formed of 8 neurons;
- each neuron has 4 inputs, meaning a RAM 16x1 bits memory.

4.1 The effect of genetic operators

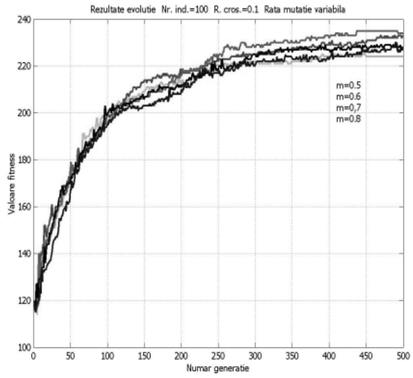
It is known that there are no universal methods of determining the use rate for the genetic operators. That is why, a wide set of simulations had the purpose to show a dependence between the convergence rate and the various speeds of the mutation/crossover rate.

For the crossover rate we chose the 0.1-0.9 range and for the mutation we choose the 0.0–0.9 range. The sampling rate in both cases was 0.1. Due to the fact that crossover is the main operator in a genetic algorithm, we chose not to include 0 for this operator (0 meaning it is not applied). For each combination of the crossover/mutation rate we made 10 runs of the genetic algorithm and mediated the outputted results.



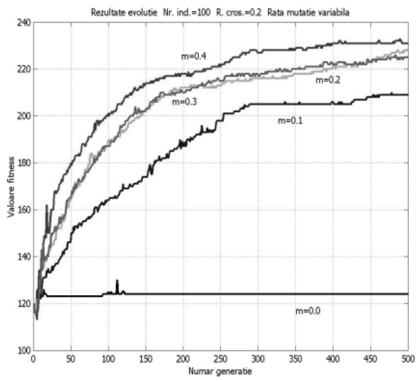
a) Crossover rate r=0.1

Mutation rate variable from 0.0 to 0.4



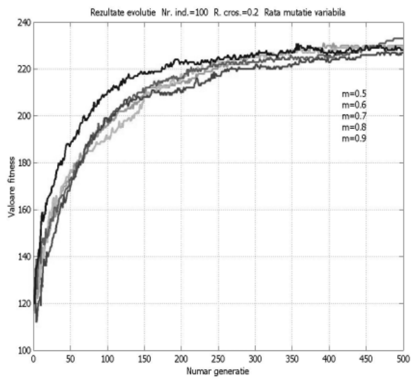
b) Crossover rate r=0.1

Mutation rate variable from 0.5 to 0.9



c) Crossover rate r=0.2

Mutation rate variable from 0.0 to 0.4



d) Crossover rate r=0.2

Mutation rate variable from 0.5 to 0.9

Figure 8 - Performance dependence of the genetic algorithm to the mutation rate at low crossover rates

After data processing and graphical representation, the following conclusions are to be drawn:

1. Lack of mutation operator drops the performance of the genetic algorithm (Figure 8.a and Figure 8.c).
2. For low crossover rates (under 0.3) the convergence speed depends on the mutation rate. This dependence is accentuated for mutation rates between 0.1.and 0.4 (Figure 8.a and Figure 8.c) but is less visible for rates above 0.6 (Figure 8.d)

A possible explanation is that the performance of the genetic algorithm depends on the diversity of the individuals in the population.

For the genetic coding used for this application, the genetic diversity can be ensured as follows:

– if one of the operations has a high enough use rate (so there is a reduced dependence on the other operator);
– if both operators have small use rates (in this case a small in the use rate leads to great convergence rate).

4.2. Error tolerance check

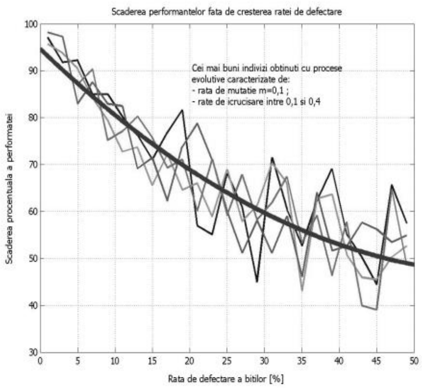
For the control systems for mobile robots, a great importance has their tolerance to errors. Overall, weighted neural networks are known to be tolerant to errors because of the redundant way the control function is memorized. In order to see if these characteristics are found also in weightless networks, we designed a series of tests that will determine the degree of performance drop for the control system when the errors occur. In these simulations, the error consists in the permanent altering of the state of a variable number of bits from the RAM neurons memory.

For the tests were selected the best individuals resulted form the genetic algorithm executed before. For each individual, we increase the percent of altered bits, which are chosen in a random way. For each percent of altering the content of the neural network the individual was tested for how many sensorial inputs it can react correctly.

The results are illustrated in Figure 9 and highlight the following aspects:

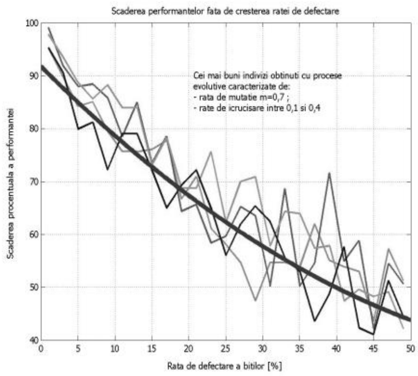
– the performance of the robot decrease with the increase of number of altered bits in the RAM neurons memory but, the command system is not entirely out of function, even for high number of altered bits!
– the performance decrease is linear, especially for the first part of the characteristic, especially for error rates under 25%.

Taking into account all these results, we can conclude that weightless neural networks are a true alternative for creating robust control systems for autonomous mobile robot control.



a) Constant mutation rate r=0.1

Variable crossover rate from 0.1 to 0.4



b) Constant mutation rate r=0.7

Variable crossover rate from 0.1 to 0.4

Figure 9 - Performance decrease for the command and control system when the error rate in the neurons RAM memory increases. The thick line shows a medium value for the results presented in thin line.

5. Conclusion

This paper presents the results of an experiment of artificial evolution with the purpose of determining the content of the neurons RAM memory in the structure of a weightless neural network, so that a mobile robot develops obstacle avoidance behavior.

Choosing such a network is justified by the ease of implementation in present digital systems, by the real time response, by the tolerance to errors, and by the fact that is less studied in the context of artificial evolution.

The genetic algorithm used in this experiment is similar to the standard, without the encoding and evaluation of the individuals. Simulation results show that it can develop a weightless neural network capable of similar behavior to the reference system over 300-400 generations, (if we find a proper parameterization for genetic algorithm).

The performance of the genetic algorithm is strongly dependent on the used rate of the genetic operators. There are also no significant improvements in using high rates – the system will only be wasting computation time.

The tolerance to errors shows that the performance decreases with the increasing of number of altered bits in the RAM neurons memory but, the control system is not entirely out of function, even for high number of altered bits. All in all, weightless neural networks are a true alternative for creating robust control systems for autonomous mobile robot control.

References

[1] DAM J., DEV A., DORST L., *"Organization and Design of Autonomous Systems",* WINS, University of Amsterdam, Aug. 1996.
[2] CLIFF D., HARVEY I., HUSBANDS P., *Explorations in evolutionary robotics*, Adaptive Behaviors, no. 2, pp. 73-110, 1993.
[3] FLOREANO D., URZELAI J., *Evolutionary Robots: The Next Generation*, Swiss Federal Institute of Technology, 2000.
[4] NOLFI S., FLOREANO D., *Evolutionary Robotics: The Biology, Intelligence, and Technology of Self-Organizing Machines*, Cambridge, MIT Press/Bradford Books, 2000.
[5] NOLFI S., FLOREANO D., MIGLINO O., MONDADA F., *How to evolve autonomous robots: Different approaches in evolutionary robotics*, Proceeding of the Fourth Workshop on Artificial Life, pp. 190-197, Boston, MA, 1994. MIT Press.
[6] MIGLINO O., LUND H., NOLFI S., *Evolving Mobile Robots in Simulated and Real Environments,* Artificial Life, No.2, pp. 417-434, 1996.
[7] FLOREANO D., URZELAI J., *Evolutionary Robots: The Next Generation*, Swiss Federal Institute of Technology, 2000.