IEEE International Workshop on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications
6-8 September 2007, Dortmund, Germany

Autonomous Mobile Robot Control Using Fuzzy Logic and Genetic Algorithm

Gintautas Narvydas, Rimvydas Simutis, Vidas Raudonis
Kaunas University of Technology, Studentu 48-327, Kaunas, Lithuania,
gintautas.narvydas@ktu.lt, rimvydas.simutis@ktu.lt, vidasraudonis@yahoo.com

Abstract *– Design of efficient control algorithms for autonomous mobile robot movement in unknown and changing environment with obstacles and walls is a difficult task. There exist different strategies to design control systems to perform the robot movement. In this article possibility to use FUZZY logic, IF-THEN rules, and Genetic Algorithm for autonomous mobile robot control is presented. Control using fuzzy logic is softer and better then control using IF-THEN rules because of absence of motors speed jumps. It is shown that the rough control system can be designed using an expert knowledge. Then genetic algorithm can be used to improve the quality of the control system.*

Keywords *– autonomous mobile robots, intelligent hybrid control systems, FUZZY logic, Genetic Algorithm*

I. Introduction

Research and applications of the autonomous mobile robots (AMR) is growing every day. During the last decade the science of the AMR design has been developing very fast [1-4]. Evolutionary robot techniques and evolutionary programming methods provide new opportunities for evolutionary AMR control systems design [5, 6]. Stationary, wheeling, crawling, creeping, walking, swimming, and flying robots are designed [7]. They are able to see, hear, communicate among themselves, and even talk to people. Moreover, they are able to plan their actions. Robots are able to wash rooms, launder, make food, search for certain things, transport them to specific places, orient in stable and shifting environments, interact with them, help men to control sophisticated technologies, perform dangerous tasks, even the ones in an environment where men cannot get into or stay within. They can serve as guards of a special territory or simply to be sources of intelligent entertainment.

However, AMR cannot move, communicate, recognize and orient in the environment in the same way as the living creatures can. The intelligent control systems are neither universal nor powerful enough to guarantee a good working of AMR. In the robotics it is necessary to improve both the hardware and the software as well as to pay exclusive attention to the intelligent hybrid control systems and evolutionary algorithms. In order to design the effective control systems we have to implement two stages:

1) to create hybrid intelligent control system to perform a specific task;

2) to use evolutionary programming methods to improve the results of the first stage.

This article analyzes both stages: the first one related to the creation of the framework of a FUZZY control system (FCS), and the second one related to the methodology how to improve the control system using the Genetic Algorithm (GA).

In the experiments the AMR Khepera II (Fig. 1 left) was taught to follow the walls of the maze (Fig. 3) using the control system based on the FUZZY logic (FL) and IF-THEN rules. The ability to follow the walls can be easily transformed into the ability to avoid even moving obstacles and to circuit them on the right or on the left. It is shown that the FL can be used as the control system for AMR. Two inputs system with FL (Fig. 4) and eight inputs system based on the IF-THEN rule have been applied in the experiment. The major goal was to seek optimal parameters for FCS which gives the best result for the wall following task.

II. Robot Khepera II

All experiments were made using the miniature mobile robot Khepera II [8]. The robot and its view from the top are shown in figure 1.

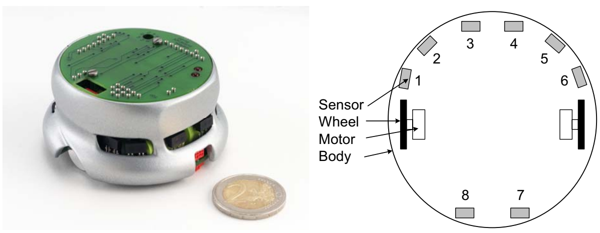

Fig. 1. The miniature mobile robot Khepera II and its view from the top [8].

The robot's diameter is 70mm, height 30mm, and weight about 80g. It is designed to move on the flat surface. It has two lateral wheels that can rotate in both directions at appointed speed. Two rigid pivots in the front and in the back of the robot are installed. They do not let the robot sway forward and backward.

Various additional turrets can be added on the top of the robot basis: linear vision module, gripper, vision turret,

1-4244-1348-6/07/$25.00 ©2007 IEEE

communication turrets, etc. Eight infrared proximity sensors are located around the body: six sensors from the front and two sensors on its back. Using Khepera's sensors it is possible to detect obstacles and to estimate distances up to them. Values of each sensor are integer numbers from 0 to 1023. The higher the value of the sensor, the nearer the obstacle is. The maximum sensitivity distance is about 100 mm. The robot can be attached to the computer through a serial cable and RS232 connector, which can be used to transfer data and to supply power. The miniature size of the robot gives a few advantages using it in experiments. First, it is possible to build various complex environments directly on the table, for example, mazes. Second, a little working surface let us keep a close watch on the behavior of the robot during experiments. However, one of the most important advantages of the miniature robot is its size, weight, and ability to move fast. Therefore, when the robot hits a wall or an obstacle it is neither damaged nor broken. The application possibilities of the robot are wide. It can be used to design and to develop the movement, recognition, orientation in the environment, path planning, control, cooperation, and other algorithms.

III. Genetic Algorithm

Genetic Algorithm [9, 10] is a useful instrument of evolutionary computation for optimization. In general, it is a search algorithm which uses principles inspired by the Nature. GA differs from a direct search procedure. This method is appropriate when we do not need to find the best solution and it is enough to find a near solution to the best one from a huge quantity of possible alternatives. The major weakness of the GA is that it usually tends to be computationally expensive in the real systems. The evaluation of fitness function is an expensive process and takes much time. In the figure 2 the scheme of GA is presented.

GA operates with randomly generated population of individuals encoded by binary strings (chromosomes) which correspond to particular solutions. This population is a small part of all possible solutions. Population evolves toward better solutions while genetic operators such as selective reproduction, mutation, and crossover are applied. In every generation better individuals (solutions) generate offspring which inherit better characteristics and replace worse individuals during generations. The fitness function is a performance criterion that evaluates the performance of each individual. Individuals with higher fitness values are better. Selective reproduction is based on natural selection. Members of the population are chosen for reproduction on the basis of their fitness defined according to some specified criteria. The best individuals are given a greater probability of reproducing in proportion to the value of their fitness. There are a few basic methods to implement selection: roulette wheel, truncation selection, selection based on tournament. Often it is useful to preserve the best individual(s) for the next generation. This strategy is known as elitism.

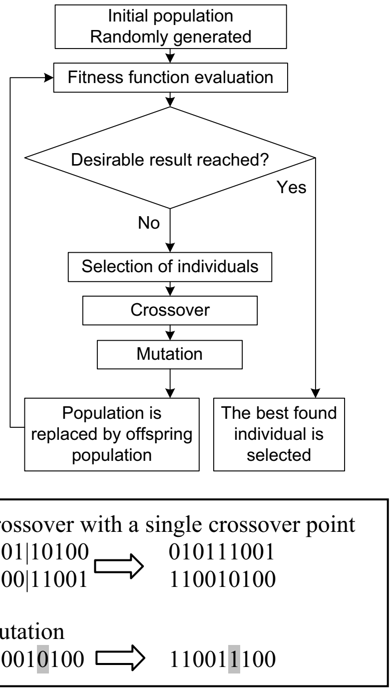

Fig. 2. Scheme of the Genetic Algorithm.

Crossover lets two members of the population exchange genes. There are many ways to implement crossover. It is possible to have a single crossover point or many crossover points. These crossover points are selected randomly. The last procedure of GA generation is mutation. In this procedure a randomly selected gene is replaced by a particular chromosome (solution). Thus, a 0 is changed to a 1 or vice versa. During generations of GA mutation occurs with a little probability (less than 0.05).

IV. Fuzzy Logic

Fuzzy Logic is an attractive method to solve problems of the control and simulation of complex system [11]. The strategy of the FL is commonly used for the simulation of non-linear systems, which have indefinite states. Normally FL is treated as the "data connector" between input and output, which are usually the vectors with different dimensions. Such data connection is defined by linguistic rules. If a model, which describes a certain process, is based on FL and on linguistic rules, it is called a FL model. The linguistic rules are usually designed according to the information which is collected from the process. Therefore, such a heuristic model is also called a "grey-box" model. The knowledge about the process, input and output data as well as the relationship between them can be used in such "grey-box" models. It is the main advantage of FL models in comparison with polynomial or regression models [12].

The human decisions are based on particular IF-THEN rules, which are commonly used in the computer science. These rules reflect the ideas and connect one event with another. The FL systems, which tend to duplicate a human reasoning, work in the same way. Therefore, the inputs and outputs of the system are transformed into fuzzy variables or so-called linguistic variables and the decisions are made according to the linguistic rules in the fuzzy systems.

The basic concept underlining FL is a linguistic variable, characterized by words rather than numbers. Such linguistic variables are defined by membership function which varies from 0 to 1.

A FL system consists of three operations: fuzzification, inference engine, and defuzzification. The input data are transmitted to the FL system where physical quantities are interpreted as linguistic variables with appropriate membership functions. These linguistic variables are then used in the antecedents (IF - part) of a set of fuzzy IF-THEN rules within an inference engine to result in a new set of fuzzy linguistic variables or consequent (THEN - part) [13].

A commonly used so-called Mamdani type rule can be written as follows:

$$\text{IF } x_1 \text{ is } A_1^i \text{ AND } x_2 \text{ is } A_2^i \text{ THEN } y^i \text{ is } B^i, \quad i = 1,2,\ldots,l. \qquad (1)$$

There x_1 and x_2 real values of the process, A_1^i and A_2^i are the fuzzy sets representing the $i^{th}-$ antecedent pairs, and B^i are the fuzzy sets representing the $i^{th}-$ consequent, and *l* is the number of rules.

V. Description of the Experiment

The aim of the evolutionary experiment was to create intelligent control system for AMR using the FL, IF-THEN rules, and GA that allows AMR to follow the wall of the maze bearing right hand rule. For the experiment, i.e. for the evolution of the control system we used a maze 60x40 cm shown in the figure 3.

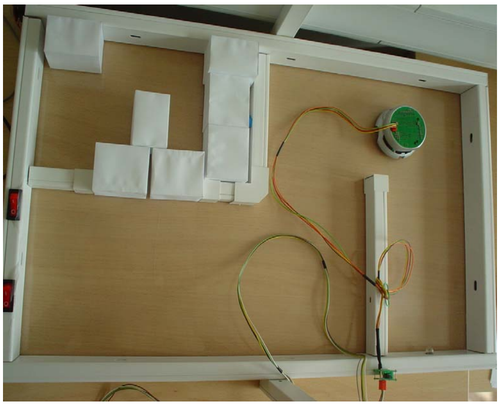

Fig. 3. The maze used in the experiment.

We created FCS (Fig. 4) with two inputs: 4th and 5th sensors of the robot, two outputs: speed for the left and right wheels, and 9 rules. Input sens5 has five, input sens4 has three, and outputs SL and SR has by five membership functions. The input sens4 has only three membership functions because the 4th sensor of the robot only has react to the obstacle in front. Optimal parameters of the FCS were found using GA.

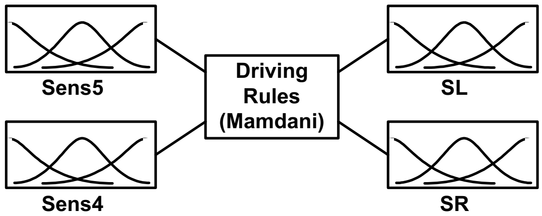

Fig. 4. The scheme of the FCS used in the experiment.

After the evolution process the robot must be able to circuit the maze following the wall on the right side.

To evaluate control systems ability to control the robot the fitness function is used:

$$fitness = (0.05a + 0.05b + 0.2c + 0.7d) \cdot \left(\frac{t}{T}\right)^2 \rightarrow \max \qquad (2)$$

$$a = 1 - \frac{1}{40N}\sum_{i=1}^{N}\left\| S_{Li} \right| - \left| S_{Ri} \right\|, \qquad (3)$$

$$b = 1 - \frac{1}{40N}\sum_{i=1}^{N}\left|\Delta_{Li}\right| + \left|\Delta_{Ri}\right|, \qquad (4)$$

$$c = 1 - \frac{1}{N(1023 - A)}\sum_{i=1}^{N}\left|S_{6i} - A\right|, \qquad (5)$$

$$d = \frac{1}{40N}\sum_{i=1}^{N} S_{Li} + S_{Ri}. \qquad (6)$$

There *N* is a number of measurements made during one drive which usually took up to 40 seconds. About 14-16 measurements per second are made (it depends on computer quality). S_L and S_R are speeds of the left and right wheels. Δ_L and Δ_R are the deviations of the current wheel speeds from the regulation speeds. These deviations emerge due to acceleration and inertia. S_6 is a current value of the sixth infrared sensor (Fig. 1 right) because it is the nearest to the wall. *A* is a constant which shows desirable distance of the robot from the wall or obstacle. Therefore *a* (2, 3) is a measure of snaking, *b* (2, 4) is a measure of twitch, *c* (2, 5) is a measure of the deviation from the regulation distance from the walls, and *d* (2, 6) is a measure of average speed. *t* is a time which robot runs controlled by control system. T is a maximum possible time of the run while control system with one collection of the parameters controls the robot. In our experiment T corresponds to 40 seconds. Over this time the robot can circuit the maze approximately twice. If a robot can follow the wall, it runs all 40 seconds and t is 40 too. However, if a robot cannot follow the wall (it turns from the wall, the speed of its wheels is very slow or negative or it sticks, it stops earlier), then t is less than 40. The proportion of t and T is squared to distinguish individuals with better skills. The architecture of the robot lets us get all the data for the calculating of these measures. It is evident that the robot must snake and twitch less, keep constant distance from the wall on the right side, and move as fast as it is possible following the wall. Since speed and distance from the wall are the most important criteria, these two components have bigger weight (2).

Our created intelligent control system is intelligent hybrid control systems. While the robot follows the wall FL controls it, and when the robot approaches too closely to the wall and can not go forward, IF-THEN rule undertake the control. When IF-THEN rule controls the robot it does not move, it only pivots anticlockwise.

GA with population of 50 individuals is used in our experiment. Initial population was created randomly. Each individual is encoded in binary chromosome with 136 genes. 98 genes are responsible for the optimal input parameters and others 38 are responsible for the optimal output parameters of the FCS. In fives best individuals from the population were preselected in each GA generation. Other individuals were selected using roulette wheel principle. Crossover was implemented using a single randomly selected crossover point with the probability 0.8. Mutation usually occurred with the declining probability during the generations from 0.04 till 0.0005 for each gene.

VI. RESULTS

The main measure – fitness that shows quality of the robot movement was observed. In the figure 5 shown below we can see average fitness of the population of 50 individuals and the fitness of the best individual from each generation of GA. Fitness value 0.8406 at the maximum was reached. The falls of the fitness were caused by the changes of the external conditions. The sensors of the robot were affected by the fluorescent lamps. These lamps have sharp influence to the sensors. That we can see in the 78th generation.

In the figure 6 the range of the fitness function components from (2) during generations of GA are shown: snaking a – gray solid line, twitch b – gray dotted line, deviation from the regulation distance from the walls c – black dotted line, and average speed d – black solid line. We can see that during evolution capacity to go fast and to move precisely fluctuate similarly, where as the snaking fluctuates contrarily. It is normal result. Our maze is not big and there are relatively many places where robot has turn. The twitch is almost constant.

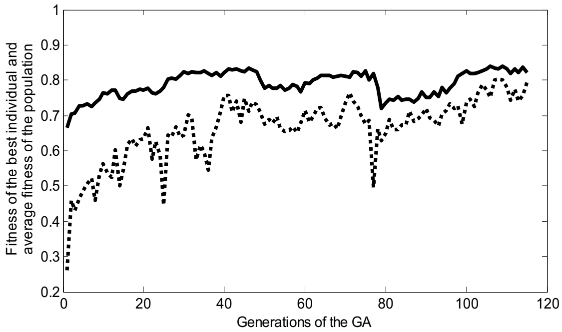

Fig. 5. Fitness functions of the best individual and average fitness of the population during GA generations.

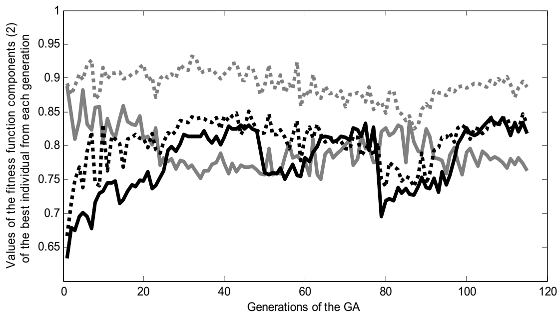

Fig. 6. Fitness functions components (2) of the best individual of the population during GA generations.

The percentage of the time when IF-THEN rule controlled the best individual of the population during GA generations is shown in the figure 7. The less this percentage is, the better the control system works. We mentioned above that when IF-THEN rule controls the robot it does not move, it only pivots anticlockwise and fitness function value recede.

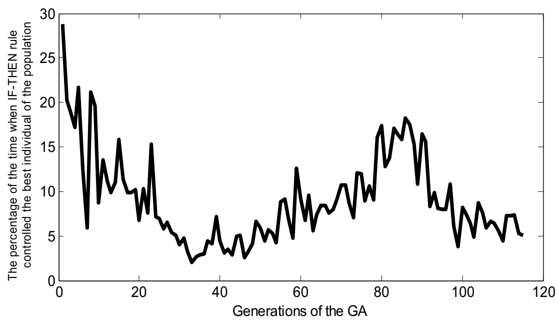

Fig. 7. The percentage of the time when IF-THEN rule controlled the best individual of the population during GA generations.

We made similar experiment when the robot was controlled by State-Based control system [3] with 7 states (IF-THEN rules). The optimal parameters also were found using GA. It reached 0.7721 fitness value. In the figure 8 the results of the State-Based control system are shown.

We can see that FCS gives better results than State-Based control system. Also the population, when FCS controls is more stable and has better fitness value.

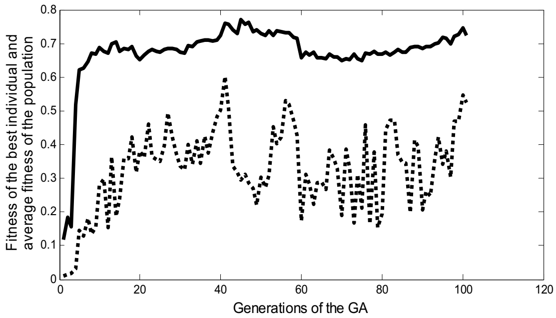

Fig. 8. Fitness functions of the best individual and average fitness of the population during GA generations when only the IF-THEN rules controlled the robot. Depicted from [3].

VII. CONCLUSIONS

In this paper the usage of the FL, IF-THEN rule and GA for the design of hybrid intelligent control systems for AMR is presented. We showed that the framework of the control system can be designed and then evolutionary algorithms can be used to find optimal parameters of the control system. The most difficult problem in our task is to find such parameters where fast speed and precise movement match. When the speed increases, the robot can follow the wall, but it cannot react to the wall ahead on time. If the robot had to follow a straight wall, it would be a simpler task. But when we have environment with inner and outer corners, the task becomes quite complicated. The part of the experiment which took the most of the time was the robot Khepera II movement and evaluation of the fitness function. The ability to follow the walls can be easily transformed into the ability to avoid even moving obstacles and to circuit them on the right or on the left. We measured fitness when the robot Khepera II was controlled by human expert operator via joystick. He reached 0.73 fitness value, and this is worse than our designed control system. Control system with FL is pretty better because it change speed of the wheels continuously while in the State-Based control system each state has its own speed of the wheels and the jumps from one state to another cause the jumps of the wheels speed.

If we compared average populations fitness of the systems, we would see that the system with FL tends to have better population. The curve of the population average fitness shows more stability and higher values.

Our designed control systems can be a part of Behaviour-Based control systems [14].

ACKNOWLEDGMENT

We thank our colleague dr. Arunas Lipnickas. A part of ideas in this paper are based on discussions with him.

REFERENCES

[1] S. Nolfi, D. Floreano, *Evolutionary Robotics: The Biology, Intelligence, and Technology of Self-Organizing Machines.* The MIT Press, 2000.

[2] R. Siegwart, I.R. Nourbakhsh, *Introduction to Autonomous Mobile Robots.* The MIT Press, 2004.

[3] G. Narvydas, R. Simutis, V. Raudonis, *Autonomous Mobile Robot Control using IF-THEN Rules and Genetic Algorithm.* 3dr international conference, Mechatronic Systems and Materials 2007. Kaunas, in press.

[4] Johan M. Holland, *Designing Autonomous Mobile Robots; Inside the Mind of an Intelligent Machine.* Elsevier, 2004.

[5] D. Floreano, F. Mondada, *Evolution of Homing Navigation in Real Mobile Robot.* IEEE Transactions on Systems, Man, and Cybernetics, 1996.

[6] D. B. Fogel, *Evolutionary computation: Toward a New Philosophy of Machine Intelligence.* 2nd edition, Piscataway, NJ: IEEE Press, 1998.

[7] G. A. Bekey, *Autonomous Robots: From Biological Inspiration to Implementation and Control.* The MIT Press, 2005.

[8] K-Team. *Khepera II User Manual.* EPFL, Lausanne, 2002, http://ftp.kteam.com/khepera/documentation/ Kh2UserManual.pdf.

[9] John H. Holland, *Adaptation in Natural and Artificial Systems.* MIT Press, 1975, 1992.

[10] Z. Michalewicz, *Genetic Algorithms + Data Structures = Evolution Programs.* Berlin: Springer, 1996.

[11] T.J. Ross, *Fuzzy logic with engineering applications.* 2nd edition, Hoboken, NJ: John Wiley, 2004.

[12] Nikola K. Kasanov, *Foundations of Neural Networks, Fuzzy Systems, and Knowledge Engineering.* A Bradford Book, The MIT Press, Second printing, 1998.

[13] M. Jamshidi, A. Cilouchian, *Intelegent Control system using Soft Computing Methodologies.* CRC Pres, Boca Raton, FL, 2001;

[14] R. C. Arkin, *Behaviour-Based Robotics.* Cambridge, The MIT Press, 1998.