Evolution of Neural Controllers for Robot Navigation in the elevated plus-maze environment

Nurgaliyev Shakh-Izat
School of Engineering and Physical Sciences
Heriot-Watt University
Edinburgh, United Kingdom
sn14@hw.ac.uk

Samir Mammadov
School of Engineering and Physical Sciences
Heriot-Watt University
Edinburgh, United Kingdom
sm109@hw.ac.uk

***Abstract*— In this paper, we presented the robot E-PUCK which operates in the simulated environment of Webots, using neuro-evolution techniques. The goal of this study is to evolve a neural controller for an E-PUCK robot which should move in the elevated plus-maze environment as fast as possible.**

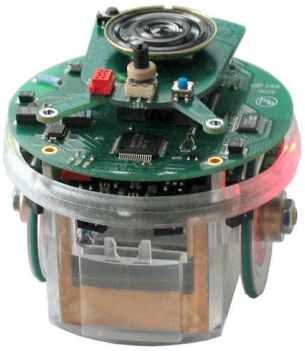
Keywords—neural networks; evolutionary algorithm; genetic algorithm; robot navigation; webots; e-puck

I. INTRODUCTION

The main feature of the robots is their autonomy. This quality can be achieved manually by human. However, currently investigated machine learning algorithms and other techniques used to generate autonomous behavior. In order to explore the possibilities of these algorithms and their specific qualities, it is often resorted to the use of robots. The robots are controlled by controllers, worked out by these methods and the behavior of these systems is analyzed by experimenters. In this section, we will look at the most popular models of mobile robots and the general classes of problems to which they apply.

One of the first robots have become widespread, it became Khepera robot ([11], [12]), developed by École Polytechnique Fédérale de Lausanne (EPFL). This is a two-wheeled robot, shaped like a small cylinder (diameter 13 cm and a height of 7), the wheels of which are controlled independently, which allows him to carry out rotations at any angle. Khepera is equipped with a number of sensors (8 infrared sensors receiving information about the distance to the object, as well as the degree of illumination, 2 infrared sensors, analyzing the surface of a robot for the task of following the line of ultrasonic sensors and 5). Due to its relatively low cost and ease of use, this robot is used in many scientific papers on evolutionary robotics, from the time of formation of this scientific field ([13], [14], [15]).

Robot E-PUCK, which is modeled in this paper is an updated modification Kheper'y [16]. Key differences from the point of view of the device include the presence of a color camera having a resolution of 640x480, three-axis accelerometer, LED-indicators located on the top of the cylinder in a circle, Bluetooth interface, 3 microphones and speakers, as well as long battery life (up to 2 hours). The default model is devoid of surface analyzing sensors, but they can be set separately.



E-PUCK robot

Autonomy of the robots is their key feature, as part of this work is necessary to give a short overview of the existing machine learning algorithms that provide opportunities for the development of autonomous behavior in any environment.

II. ALGORITHMS

A. *Reinforcement learning*

Reinforcement is widely used for mobile robots in the problems of development of autonomous behavior. Most often this method is used to produce strategies, including avoiding various obstacles and developing the optimum path to achieve the goal ([24], [25]), as well as in a group of robotics tasks. [26]

B. *Bayesian network*

Bayesian networks are commonly used in the tasks for which the answer is increasingly suggests it is a probabilistic assessment (elucidation of protein structures, genetic networks and diagnosis by symptoms), but the method is used, and in the problems of effective investigation surrounding the robot space ([28], [29], [30]).

C. *Artificial neural network(ANN)*

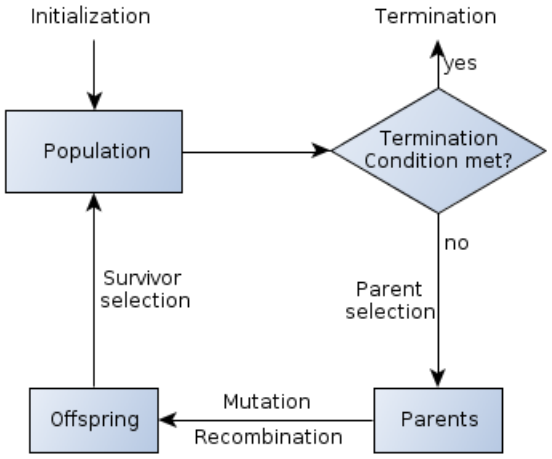
Application of neural networks in robotics is a widespread practice, because of their relatively easy implementation. In these tasks the robot is a "body" and the artificial neural network "brain" of an agent that interacts with the environment. Getting the most effective behavior associated with the problem of selecting the right parameters learning algorithm.

ANN can be used to solve the problems such as follow a certain trajectory ([31]) and avoiding obstacles ([32], [33]), as well as tasks such as master-slave ([34]).

D. Eevolutionary algorithms

Evolutionary algorithms -is a class of methods of artificial intelligence, simulating the processes of natural selection. The theory of biological evolution involves selection processes, mutation and reproduction, which are embodied in such algorithms.

The most widespread is genetic algorithms, the general principle of which is displayed in the diagram below:



The first step of the algorithm is to initialize a group of evolving agents, which in most cases are randomly. This group of agents is called a population. It should be noted that this population is the starting point for the method, and in most cases it's a good or bad choice has little effect on the convergence of the algorithm.

The population gradually evolves in accordance with the selection rules and objectives set in the environment. Depending on how well the agent reaches a specific purpose, it is assigned a certain value of fitness that affects the reproduction process (in most cases reproduce only the fittest individuals). Mutation and recombination provide a mechanism for the generation of new agents, contributing to finding the most effective strategy in a particular environment.

The result is a new population, devices that again evaluated in the framework of a given environment, and the process is repeated. The number of iterations may be limited by the criterion of convergence of the algorithm to a rather effective solution, or some limit the number of the population, or limiting the time allocated for the algorithm.

The successful application of this algorithm, two things are important:

- having a good genetic representation of agents (evolves genotype);
- Properly selected function adaptability to the environment.

Evolutionary algorithms are used in a variety of bioinformatics problems ([35]), economics, chemical kinetics, the theory of games, and many others. However, in this work we are most interested in its application for training artificial neural networks. The genetic method in which the genotype codes for the phenotype of the neural network, called neuro-evolution.

E. Neuro-evolutionary algorithms

This class of algorithms suggests that the neural network can be obtained efficiently enough to solve the problem as a result of his work. The advantages of this approach compared to the traditional learning are needed in the absence of any training data set. It is enough to have a function that evaluates the performance of the neural network.

In addition to mobile robotics well known classes of problems associated with finding the optimum route and avoiding obstacles, and problems of robotics group, neuro-evolutionary approach is applied in cases which require the formation of short-term memory ([39], [40]). This class of algorithms performed well in solving problems for quite biologically plausible neural networks with spiking activity ([38]). The most relevant ideas include an attempt to unify the process of evolution and learning in order to create an algorithm as efficiently coping with the task ([41], [42]).

III. DESCRIPTION OF THE ELEVATED PLUS-MAZE

Elevated plus maze is widely used to study the state of anxiety of rodents by evaluating the time spent of animals on open arm and two closed arm above the floor of the labyrinth. This test is based on the conflict between the innate fear that animal tests before the open space, and the simultaneous desire to explore an unfamiliar environment.

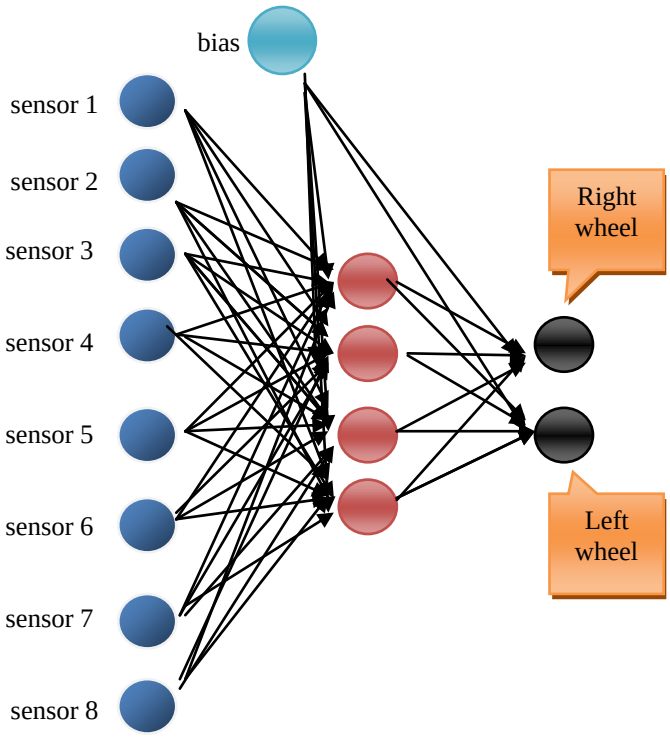
EPUCK robot should evolve a robot controller using evolutionary robotics techniques. The controller is modeled as an Artificial Neural Network (ANN) to control a single robot on an elevated plus-maze (EPM) arena depicted on image. The robot will move in the elevated plus-maze and mimic as close as possible the behaviour of the rat. Basically, the robot should avoid obstacles while moving around in the maze and explore it as quickly as possible.

IV. Develop an algorithm to control **EPUCK** mobile robot in the elevated plus-maze

The Robot E-PUCK has the shape of a small cylinder diameter of 7 cm and a height of 5 cm, and equipped with two independently controllable wheels, 8 infrared rangefinders arranged on a circle, and a color camera in the front.

A. Neural Controller

The E-PUCK robot is controlled by a neural network which transforms the sensory IR inputs received from the sensors of the e-puck into motor commands for the left and right wheels of the robot as shown in figure 1. Inputs are scaled to fit the range [0,1] and the activation function is chosen to be a hyperbolic tangent (tanh).



E-puck's neural controller.

B. Genetic Algorithm

A genetic algorithm is used to evolve the synaptic weights of the described neural controller. The synaptic weights, which represent the genes of the individuals, are coded using one float value. When put together, the genes form a genome.

A population of individuals is evolved using roulette-wheel selection or rank-based truncation selection, one-point crossover, weight mutation and elitism. The genomes of the first generation are initialized randomly in the range [-1, 1]. Each individual is evaluated based on the fitness function defined in the compute **fitness()** function defined in **epuck_controller.c**. After ranking the individuals according to their measured fitness values, the top individuals are copied to the new population (elitism). This allows evolution to make sure that good solutions are not destroyed because of mutation or crossover. The remaining population is generated from the crossover of two randomly paired individuals within the first ranks. One point crossover is applied to each pair with a certain probability and each gene is then mutated with a certain probability.

C. Programming

Evolution.c contains the code for the evolutionary algorithm, whereas **epuck_controller.c** contains the neural network and the controller that determines what the robot will do. At each generation, evolution sends the genomes to **epuck_controller.c** one by one. Then translates the genome it receives into a neural network controller, evaluates the controller and sends its fitness back to **evolution.c**.

The neural controller can be configured by modifying the following parameters in **epuck_controller.h.**

NB INPUTS - Number of inputs to the neural network. Possible values are: 2 (front IR senors), 4:(front IR sensors), 6 (front and side IR sensors) and 8 (all IR sensors).

NB HIDDEN NEURONS - Number of hidden neurons. Normally no hidden neurons are required, and we'll leave this value set to 0 for this exercise.

NB OUTPUTS - Number of outputs. For this exercise, the number of outputs is always equal to two. Note that changing the number of inputs and outputs of the neural controller will automatically modify the number of genes parameter NB GENES in **evolution.h**.

TRIAL_DURATION N - Evalutaion duration of one individual(ms).

SPEED_RANGE N - Wheel controllers receive commands between -SPEED_RANGE and SPEED_RANGE

OBSTACLE_THRESHOLD N – limitation to obstacle

The genetic algorithm can be configured by modifying the following parameters in **evolution.h**.

POP SIZE - Number of individuals per generation.

MUTATION PROBABILITY - Probability of mutating each weight in a genome.

MUTATION SIGMA - Determines the extent to which a value will be mutated.

ELITISM RATIO - Ratio of best individuals which are copied to the next generation without modification.

CROSSOVER PROBABILITY - Probability with which crossover occurs.

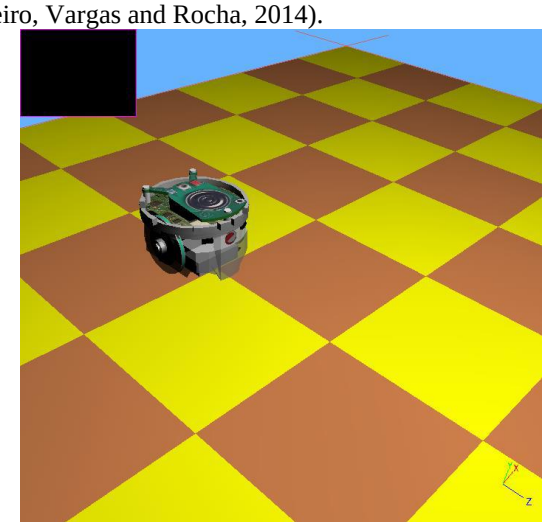
ROULETTE WHEEL - Choice of selection method. Set to 1 for roulette wheel selection (individuals are selected proportional to their fitness) or 0 for rank-based truncation selection (select best x% of individuals).

For these exercises we can use the Webots robot simulator

D. Webots robot simulator

Webots is development environment, designed for modelling, programming and emulation of mobile robots (Téllez and Angulo, 2007). With Webots, a user can design a scene with a different number of objects and the robots. Settings, such as weight, size, colour, and so on. Because of wide variety of sensors and materials, you can create any modifications to the robot. Robot controllers can be

programmed using the integrated development environment. Features of the robot can be tested in a physically realistic world. The controller of the robot, if necessary, can be loaded into the robot commercial e-puck as shown in figure 5 (Couceiro, Vargas and Rocha, 2014).



The main element in the development Webots - is the scene. It consists of various environment objects, setting the physical parameters of the virtual world, as well as robots. Physical parameters settings are quite flexible. They allow you to explore the robot's behavior under various conditions. For example, you can change the frictional force, change the state of the environment - to emulate the water bodies, adding wind, etc. For the design stage there is a built-in editor, it allows you to design the stage with a variety of geometric objects, and create unique models of robots. An important feature is the ability to Webots accelerated emulation, which allows a short time to collect important information, such as the behavior of the robot under different conditions. Integrated development environment allows you to create robot controllers in different programming languages (C ++, Java, etc.).

E. Figures and Tables

1) Positioning Figures and Tables: Place figures and tables at the top and bottom of columns. Avoid placing them in the middle of columns. Large figures and tables may span across both columns. Figure captions should be below the figures; table heads should appear above the tables. Insert figures and tables after they are cited in the text. Use the abbreviation "Fig. 1," even at the beginning of a sentence.

TABLE I. TABLE STYLES

Table Head	Table Column Head		
	Table column subhead	***Subhead***	***Subhead***
copy	More table copy[a]		

[a.] Sample of a Table footnote. *(Table footnote)*

[b.]

Fig. 1. Example of a figure caption. *(figure caption)*

We suggest that you use a text box to insert a graphic (which is ideally a 300 dpi resolution TIFF or EPS file with all fonts embedded) because this method is somewhat more stable than directly inserting a picture.

To have non-visible rules on your frame, use the MSWord "Format" pull-down menu, select Text Box > Colors and Lines to choose No Fill and No Line.

Figure Labels: Use 8 point Times New Roman for Figure labels. Use words rather than symbols or abbreviations when writing Figure axis labels to avoid confusing the reader. As an example, write the quantity "Magnetization," or "Magnetization, M," not just "M." If including units in the label, present them within parentheses. Do not label axes only with units. In the example, write "Magnetization (A/m)" or "Magnetization (A (m(1)," not just "A/m." Do not label axes with a ratio of quantities and units. For example, write "Temperature (K)," not "Temperature/K."

ACKNOWLEDGMENT *(Heading 5)*

The preferred spelling of the word "acknowledgment" in America is without an "e" after the "g." Avoid the stilted expression "one of us (R. B. G.) thanks ...". Instead, try "R. B. G. thanks...". Put sponsor acknowledgments in the unnumbered footnote on the first page.

REFERENCES

The template will number citations consecutively within brackets [1]. The sentence punctuation follows the bracket [2]. Refer simply to the reference number, as in [3]—do not use "Ref. [3]" or "reference [3]" except at the beginning of a sentence: "Reference [3] was the first ..."

Number footnotes separately in superscripts. Place the actual footnote at the bottom of the column in which it was cited. Do not put footnotes in the reference list. Use letters for table footnotes.

Unless there are six authors or more give all authors' names; do not use "et al.". Papers that have not been published, even if they have been submitted for publication, should be cited as "unpublished" [4]. Papers that have been accepted for publication should be cited as "in press" [5]. Capitalize only the first word in a paper title, except for proper nouns and element symbols.

For papers published in translation journals, please give the English citation first, followed by the original foreign-language citation [6].

[1] G. Eason, B. Noble, and I.N. Sneddon, "On certain integrals of Lipschitz-Hankel type involving products of Bessel functions," Phil. Trans. Roy. Soc. London, vol. A247, pp. 529-551, April 1955. (*references*)

[2] J. Clerk Maxwell, A Treatise on Electricity and Magnetism, 3rd ed., vol. 2. Oxford: Clarendon, 1892, pp.68-73.

[3] I.S. Jacobs and C.P. Bean, "Fine particles, thin films and exchange anisotropy," in Magnetism, vol. III, G.T. Rado and H. Suhl, Eds. New York: Academic, 1963, pp. 271-350.

[4] K. Elissa, "Title of paper if known," unpublished.

[5] R. Nicole, "Title of paper with only first word capitalized," J. Name Stand. Abbrev., in press.

[6] Y. Yorozu, M. Hirano, K. Oka, and Y. Tagawa, "Electron spectroscopy studies on magneto-optical media and plastic substrate interface," IEEE Transl. J. Magn. Japan, vol. 2, pp. 740-741, August 1987 [Digests 9th Annual Conf. Magnetics Japan, p. 301, 1982].

[7] M. Young, The Technical Writer's Handbook. Mill Valley, CA: University Science, 1989.