A colony of robots using vision sensing and evolved neural controllers

A. L. Nelson, E. Grant, G. J. Barlow
Center for Robotics and Intelligent Machines
Department of Electrical and Computer Engineering
North Carolina State University
Raleigh, NC 27695-7911

T. C. Henderson
School of Computing
University of Utah
50 S. Central Campus Drive, Room 3190
Salt Lake City, UT 84112

Keywords: Evolutionary robotics, Robot colonies, Mobile robots, Evolutionary neural computing, Behavioral robotics, Vision, Robot vision

Abstract--This paper describes the development and testing of a new evolutionary robotics research test bed. The test bed consists of a colony of small computationally powerful mobile robots that use evolved neural network controllers and vision based sensors to generate team game-playing behaviors. The vision based sensors function by converting video images into range and object color data. Large evolvable neural network controllers use these sensor data to control mobile robots. The networks require 150 individual input connections to accommodate the processed video sensor data. Using evolutionary computing methods, the neural network based controllers were evolved to play the competitive team game Capture the Flag with teams of mobile robots. Neural controllers were evolved in simulation and transferred to real robots for physical verification. Sensor signals in the simulated environment are formatted to duplicate the processed real video sensor values rather than the raw video images. Robot controllers receive sensor signals and send actuator commands of the same format, whether they are driving physical robots in a real environment or simulated robots agents in an artificial environment. Evolved neural controllers can be transferred directly to the real mobile robots for testing and evaluation. Experimental results generated with this new evolutionary robotics research test bed are presented.

1. Introduction

Evolutionary robotics (ER) is an emerging field under the general rubric of behavioral robotics. The field of ER applies evolutionary computing methods to automate the development of autonomous robot controllers. In a typical application, autonomous mobile robot controllers are evolved to produce robot behaviors such as homing in on a light source (Phototaxis) [1][2] or avoiding obstacles [3][4]. Artificial evolution is applied to a population of randomly initialized controller structures. Typically these structures are neural networks although genetic programming (GP) constructs have also been used [5][6]. Each controller in such a population is tested and ranked according to how well it can control a robot to produce a desired behavior. The best performing controllers are selected and the poorer controllers are discarded. The best controllers are copied and slightly altered using genetic operators such as mutation and recombination (crossover). The altered controllers then take the places of poorer controllers in the population and the process is repeated.

Recently, the field of ER has been reviewed in several publications [7][8][9]. Pertinent issues raised in those works include 1) the feasibility of applying current ER methods to more sophisticated and general problems; 2) the coupling of training simulation to reality; and 3) methods of performance evaluation. We address the question of scalability of ER methods to complex problems by evolving complex neural network based controllers to generate game playing behaviors in teams of mobile robots. Controllers are evolved using a competitive relative fitness selection metric (fitness function). The metric bases controller fitness on the results of tournaments involving all individuals in an evolving population.

The main focus of this paper is the description of a coupled real and simulated ER research platform with vision-based sensors. A versatile vision based sensing system that is amenable to simulation but can still provide extensive sensor information to the neural controllers is presented. The platform generates large evolvable neural networks that support very large arrays of processed video sensor inputs. In this research, neural controllers for autonomous mobile robots using on the order of 150 processed video inputs were evolved to play a competitive team robot game. In other ER work, simpler sensing systems such as IR, photo detectors or sonar have been used. Such sensor systems provide limited information about the robots environment. Sensor data complexity can be viewed as a double-edged sword. Simpler sensor systems make the task of evolving controllers more tractable. However, limiting the resolution and quality of sensor information may put an upper limit on the complexity of evolvable behavior.

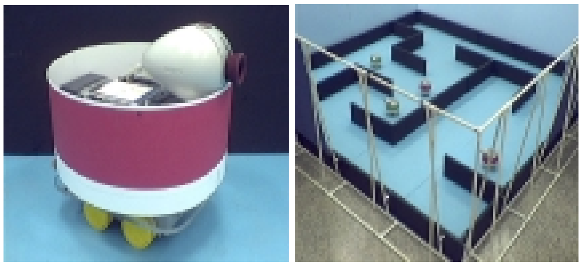
There has been very little ER work done in which video signals, processed or otherwise, were used in conjunction with evolved neural controllers. Exceptions include [10]. In that work, a ccd-camera array was used, but it was functionally sampled by averaging values within a very small number of photo-receptive fields, thus limiting sensor resolution to that of several photo receptors. Recently, in [11] research involving evolved neural networks that made use of video images fed into a 5 by 5 array of neurons was presented.

In the research described in this paper, video images are processed into a generalized form of substance type (color), range and angle numerical data and provide a considerable wealth of information to the neural controllers. Unlike other work involving video sensors, in this work numerical data from the vision system are not tagged or prioritized, but are fed directly to the neural controllers. The controllers must evolve to make use of correlations between numerical sensor data input and actuator outputs in order produce fit behavior. They are given no a priori knowledge of the physical meanings of numerical sensor data.

2. The evolutionary robotics physical research platform

This research utilizes a recently developed, computationally powerful colony of small mobile robots. These robots have been named EvBots from EVolutionary roBOTs [12].

The robots make up a colony of eight small fully autonomous mobile robots. Each robot is 5 in. wide by 6.5 in. long by 6 in. high and is constructed on a two track treaded wheel base. Each robot is equipped with a PC/104 based onboard computer. A custom Linux distribution derived from RedHat Linux 7.1 is used as the operating system and is capable of supporting MATLAB in addition to other high-level software packages. The robots are linked to one another and to the Internet via a wireless network access point. Each robot also supports video data acquisition (up to 640x480 live motion resolution) through a USB video camera mounted on each robot. A photograph of a fully assembled EvBot is shown in Figure 1 (a)



(a) (b)

Figure 1. A fully assembled EvBot (a), The real maze environment with several EvBots (b)

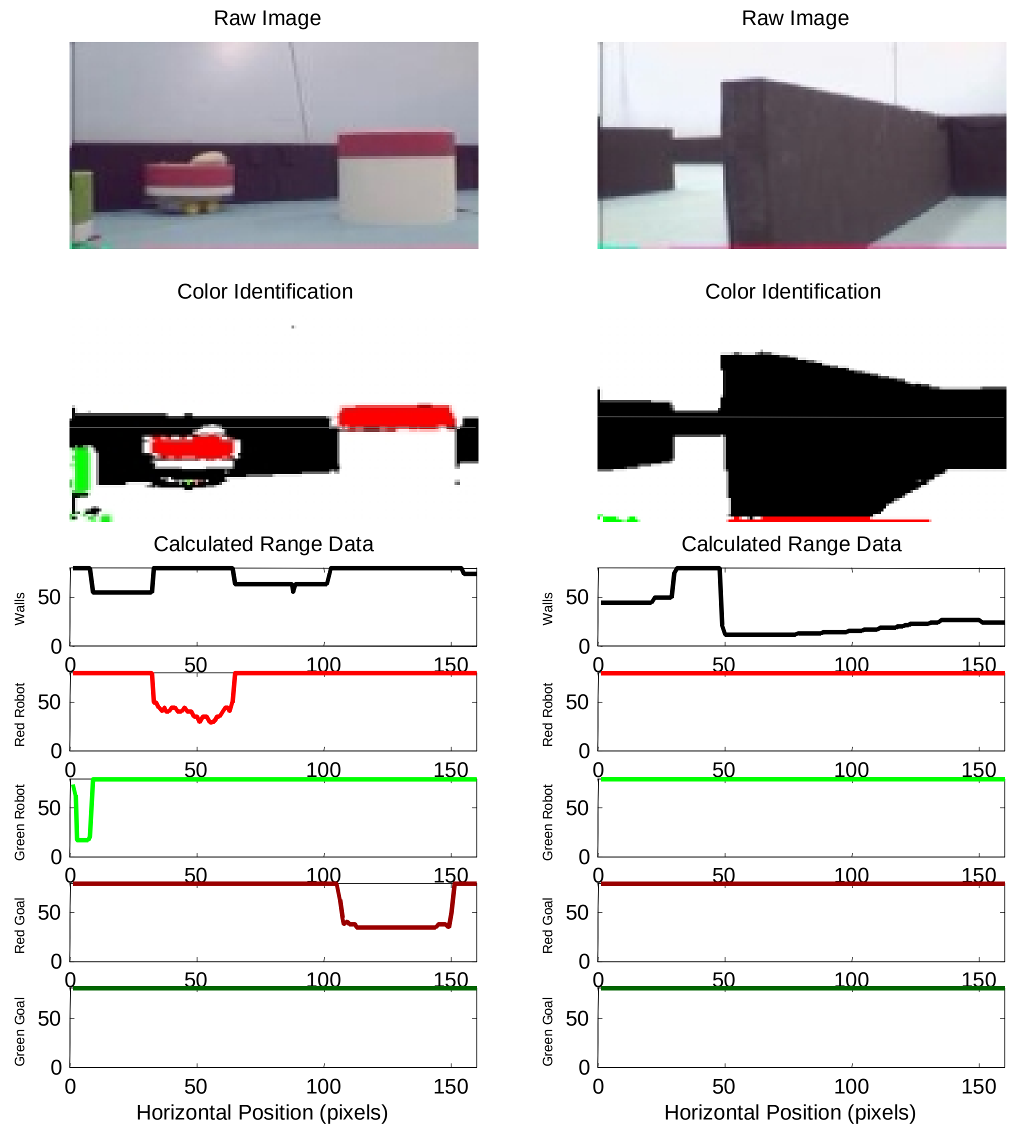
Each robot in the colony is fully autonomous and capable of performing all computing and data management on board. At each time step during controller operation, a single video image is acquired and processed. The data from the processed image are then given to the neural network application, which in turn calculates a set of drive motor actuator commands. The robots have two parallel driving wheel sets and maneuver using differential steering.

A physical reconfigurable maze environment was constructed for the mobile robot colony. To facilitate vision-based sensing, the maze was surrounded by a blue backdrop. Robots and other objects in the environment were also fitted with colored skirts. The entire maze environment is viewable from a video camera mounted above the environment. Figure 1 (b) shows the physical maze environment with several EvBots.

3. Video range-finding emulation sensors

In the experiments presented in this paper, all robotic sensing of environments was accomplished via video. The goal of this work is not to develop sophisticated vision systems, but rather to make use of simple methods to extract useful information in a form that would be presentable to a neural network based controller.

Initially, the motivation for developing the vision-based object range detection system was to emulate laser-range-finding sensors on the real robots. Subsequently it was found that video emulation of range finding sensors provides an advantage over real range finders in that object color can be used to identify object type (or 'substance' type) in addition to distance. This range finding emulation system provides an important unifying crossover point between the simulated and real environments. Simulation of the emulated range finding sensors is a much more tractable task than direct simulation of video images.



(a) (b)

Figure 2. Examples of image decomposition into vectors of range data to be fed into neural network controller inputs. One vector of length equal to the horizontal resolution in pixels of the image is produced for each ‘substance’ type in the physical robot environment.

The vision system takes advantage of fixed geometric elements and color properties within the physical maze environment to calculate the ranges and angles of walls, robots, and other objects. Each robot camera is attached at a fixed angle and altitude. Maze walls are of a constant height so distance can be calculated from a monocular image taken from a set altitude within the maze environment. In addition, each robot is fitted with a skirt that has a colored band of fixed width. Robot distances can be calculated from an image by determining the relative width of the colored band within the image. Likewise, stationary goal objects are also fitted with colored bands of fixed width. The vision system can detect five object or substance types. These are walls, red robots, green robots, red goal objects and green goal objects. Range values are reported over a spread of 48 degrees centered on the forward direction of the robot body frame of reference.

The system works by successively decomposing a video image of fixed resolution. First, each pixel is identified as being red, green, black or other (all ‘other’ colors are ignored). The image is then converted to a 2D numerical array where the index of each element is its xy-location in the original image, and its value is an identifying integer depending on the determined color of that pixel. The matrix is subdivided along the horizon into upper and lower regions to distinguish between goal objects and robots. The vertical sum of pixels, Σp, of each object type is calculated and stored in a set of arrays spanning the horizontal spread of the image. These numerical arrays are then fed element by element through a simple distance formula to produce the final vectors of ranges *d* for each object type:

$$d = K \frac{H}{\sum p} \tag{1}$$

Here, *H* is the physical height of each object type and *K* is an empirically derived constant. Each array element, *d*, represents the distance of a substance or object type associated with each vertical slice of the processed image. We will call these 'object elements' because groups of them can be interpreted by humans as making up whole objects. No such interpretation is given to the robot neural controllers. The final form of the data is (for each object type) a vector of numbers spanning the horizontal angular spread of the original image. Each number element represents the distance of the closest object element type associated with each direction (or angular position). If no object element is detected at an angular position, the maximum sensor range is returned. The angle of a detected object element is implicit in the location of each numerical distance value within each data array. Each array spans the horizontal spread of the robot camera's field of view, and each successive element represents an incremental angular step from left to right across the horizontal field of view. Figure 2 shows two example robot-eye-view images and their successive decomposition into range data vectors. The object range data vectors shown in Figure 2 were further reduced in length by extracting the minimum distance over successive groups of horizontal elements. The end results are sets of data similar to those that would be obtained from groups of 30 laser range finding sensors that were selective for a particular object type. This makes a set of 150 total process sensor inputs.

Controllers are only given the resulting numerical data vectors. All associations relating numerical values to physical distances, angles, and object types must be learned by the neural networks.

4. The evolutionary neural network architecture

Neural networks are the most commonly used controller structures in ER. This is mainly due to their flexibility and their close association with the research field of evolutionary computing.

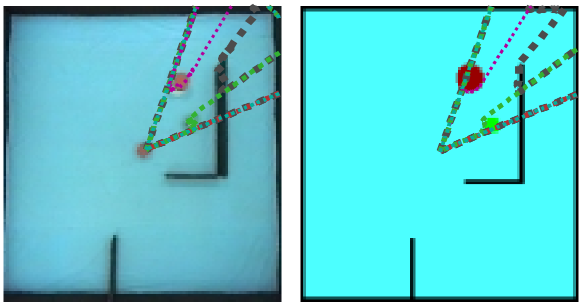
In general, behavioral robotics tasks are not well characterized. Hence, it is not always possible to select the best neural network architecture for a particular behavioral robotics application. Much of the ER work to date used very simple network topologies and restricted weight values [13][14][15][16]. Such restrictions limit the scalability of the methods studied. We have developed a generalized evolvable neural network architecture capable of implementing a very broad class of network structures. Networks are not limited to any particular layered structure and may contain feed forward and feedback connections between any of the neurons in the network. Networks may contain mixed types of neurons, and a variable integer time delay may be set on the inputs of any neuron in the network. Internal neuron activation function types include sigmoidal, linear, step-threshold, and Gaussian radial basis functions.

5. Results

5.1 Simulated vs. real sensors

Figure 3 (a) shows an image of the real maze environment with a graphical representation of real sensor readings superimposed on the image. Here, the sensor data were gathered by the robot in the center of the maze. In part (b) of Figure 3, the environment and object configuration is duplicated in simulation. Again, sensor data were taken from the center of the simulated maze and from the same orientation as the real robot in the real maze. The simulated sensor data were also superimposed onto the simulated maze graphic. The robot and environment simulator used in this work is derived from, and similar to, the one developed in [17].

To investigate and quantify the fidelity of the video-range emulation sensor system, sets of real and simulated sensor readings were compared. 10 images similar to the one shown in Figure 3 (a) were taken of the real maze environment with real robots. The images were then overlaid with sensor data produced by the robot in the center of the maze. The cone of dashed lines on the image is a graphic representation of the sensor readings. The physical maze environment configurations were then duplicated in the simulation environment and simulated sensor readings were recorded. Over the set of 10 test configurations, the real vision based sensors produced an error of about 12.5 percent when compared to simulated sensor values.



(a) (b)

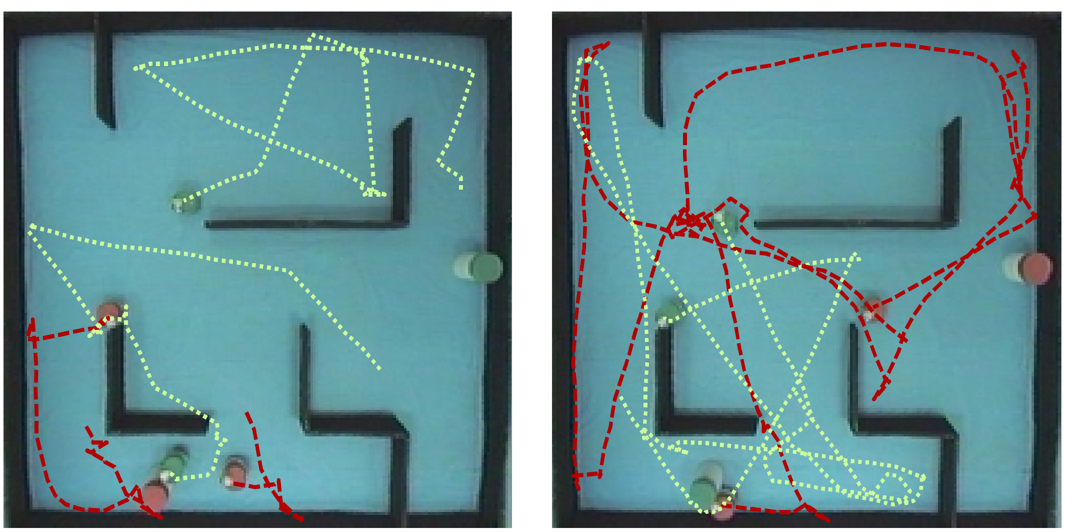
Figure 3. Real sensor readings are plotted on an image of the real maze environment (a) These are compared to simulated sensor readings generated in the simulation environment. Both the real and simulated worlds were configured similarly.

5.2 Evolved controller performance validated in real robots

In this section, we present results of a population of robot controllers evolved to play robot capture-the-flag. In this game, there are two teams of robots and two goal objects. All robots on team #1 and one of the goal objects are of one color (red). The other team members and their goal object are of another color (green). In the game, robots of one team must try to come within a certain distance of the other team's goal object while protecting their own. The robot which first comes within one robot body diameter's distance of an opponent's goal wins the game for its team.

A population of robot controllers using video range-emulation sensors was evolved in simulation and then transferred to real robots in a real environment for validation. The evolution process used a form of relative competitive performance evaluation for selection. In the evolutionary process, each generation consisted of a tournament of games played between the controllers in the evolving population. Robot controllers were selected and propagated based on whether they won or lost games in the course of a tournament.

Evolved controllers were transferred to real robots and tested in a physical maze environment. In order to demonstrate that evolved controller had gained a level of proficiency, they were placed in competition with knowledge-base controllers coded to play robotic Capture the Flag. Figure 4 shows the results of two games played with teams of real robots in a physical maze environment. In the games, the best evolved ANN controller from the population and the hand coded knowledge-based controllers were used. These were transferred to teams of green (lighter colored) and red (darker colored) robots respectively. In the figure, robots are shown in their final positions at the end of each game. The darker dotted lines indicate the paths followed by the green robots while the lighter lines indicate the paths followed by the green robots. In the simulations and in the real environment, robots displayed several learned



(a) (b)

Figure 4. Two example games involving real robots in a physical maze environment. In each panel, the green robots are controller by evolved neural networks while the red robots are controlled by the knowledge-based controller. The dashed lines indicate the paths taken by each of the robots during the course of each game. The first game was won by the evolved neural network controllers, while the second was won by the knowledge-based controller.

behaviors. These include wall avoidance, homing on an opponent's goal, and avoidance of other robots. These results show that behaviors relying on a vision based sensing can be evolved in simulation and transferred to real robots. This paper's main focus was on the design and development of the real and simulated vision based robot neural controller evolution platform. A more in-depth analysis of the evolved behaviors is given in [17].

6. Conclusion and future research

In this paper, a new evolutionary robotics research environment and test bed was described and related experimental results were presented. Robots relied entirely on processed video data for sensing of their environment. This is a departure from the simpler IR and sonar sensors employed on other ER research robots. The video sensing system was modeled in a coupled simulation environment. The simulation environment was used to evolve neural controllers for teams of small mobile robots. For the evolutionary training of the neural controllers, a tournament training performance evaluation function was implemented. This fitness function was used to evolve controllers for teams of robots to play a benchmark competitive game, 'Capture the Flag'. The fitness function was not based on game specific factors and could be used on other multi-robot tasks that can be formulated into competitive games. The use of competitive performance evaluation allows for the improvement of behavior without the need for an absolute performance measure.

Although the work presented here used only vision-based sensors, it may be beneficial to incorporate other sensing modalities into the robots and controllers. Additional sensors might include tactile sensors, sound sensors, and laser range sensors. The robot platform is fully extendable and allows for the incorporation of additional sensor types. The work will be extended by investigating sensor fusion at the neural controller level. This will be accomplished by providing the evolving neural controllers with a larger variety of sensor inputs and processed sensor data. The evolutionary process will be used to select controllers that make advantageous use of available sensor data.

References

[1] R. A. Watson, S. G. Ficici, J. B. Pollack, Embodied Evolution: Distributing an Evolutionary Algorithm in a Population of Robots, Robotics and Autonomous Systems, Vol. 39 No. 1, pp 1-18, Volume 39, April 2002.

[2] N. Jakobi, P. Husbands, I. Harvey, Noise and the reality gap: The use of simulation in evolutionary robotics. In F. Moran, A. Moreno, J. Merelo, and P. Chacon, editors, Advances in Artificial Life: Proc. 3rd European Conference on Artificial Life, pp. 704-720. Springer-Verlag, Lecture Notes in Artificial Intelligence 929, 1995.

[3] D. Floreano, F. Mondada , Evolution of homing navigation in a real mobile robot. IEEE Transactions on Systems, Man, Cybernetics Part B: Cybernetics Vol. 26 No. 3, pp. 396-407, 1996.

[4] W. Lee, Evolving Autonomous Robot: From Controller to Morphology, IEICE Trans. Inf. & Syst., Vol E83-D, No. 2, pp. 200-210, 2000.

[5] T.E. Revello, R. McCartmey, A Cost Term In an Evolutionary Robotics Fitness Function, Proceedings of the 2000 Congress on Evolutionary Computation, IEEE, Vol. 1, pp. 125-132, 2000.

[6] W. Lee, J. Hallam, H. Lund, Applying Genetic Programming to Evolve Behavior Primitives and Arbitrators for Mobile Robots, Proceedings of the 1997 IEEE International Conference on Evolutionary Computation, pp 495-499, 1997.

[7] M. Mataria, D. Cliff, Challenges in evolving controllers for physical robots, Robotics and Autonomous Systems , Vol. 19, No. 1, pp 67-83, November 1996.

[8] I. Harvey, P. Husbands, D. Cliff, A. Thompson and N. Jakobi, Evolutionary robotics: the Sussex approach, Robotics and Autonomous Systems, Vol. 20 No. 2-4, pp. 205-224, 1997.

[9] L. A. Meeden, D. Kumar, Trends in Evolutionary Robotics, Soft Computing for Intelligent Robotic Systems, L.C. Jain and T. Fukuda editors, Physica-Verlag, New York, NY, pages 215-233, 1998.

[10] I. Harvey, P. Husbands, D. Cliff: Seeing the light: artificial evolution, real vision, Appears in D. Cliff, P. Husbands, J.-A. Meyer and S. Wilson editors, From Animals to Animates 3, Proc. of 3rd Intl. Conf. on Simulation of Adaptive Behavior, SAB94, pp. 392-401. MIT Press/Bradford Books, Boston MA, 1994.

[11] D. Marocco, D. Floreano, D., Active Vision and Feature Selection in Evolutionary Behavioral Systems. In Hallam, J., Floreano, D. Hayes, G. and Meyer, J. editors, From Animals to Animats 7, Cambridge, MA. MIT Press. 2002.

[12] J.Galeotti, S. Rhody, A. Nelson, E. Grant, G. Lee, EvBots – The Design and Construction Of A Mobile Robot Colony for Conducting Evolutionary Robotic Experiments, Proceedings of the ISCA 15th International Conference: Computer Applications in Industry and Engineering (CAINE-2002), pp. 86-91, San Diego Ca, Nov. 7-9, 2002.

[13] S. Luke, C. Hohn, J. Farris, G. Jackson, J. Hendler. Co-evolving soccer softbot team coordination with genetic programming, Proceedings of the First International Workshop on RoboCup, pp. 115--118, Nagoya, Japan, August 1997.

[14] M. Quinn, Evolving cooperative homogeneous multi-robot teams, Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2000), Takamatsu Japan, Vol.3, pp. 1798 –1803, 2000.

[15] A. J. Ijspeert, Synthetic Approaches to Neurobiology: Review and Case Study in the Control of Anguiliform Locomotion, In Proceedings of the Fifth European Conference on Artificial Life, ECAL99, pp. 195-204, Springer Verlag, 1999

[16] J. Xiao Z. Mickalewicz, L. Zhang, K. Trojanowski, Adaptive Evolutionary Planner/Navigator for Mobile Robots, IEEE Transactions on Evolutionary Computing, Vol. 1, No. 1 pp. 18-28, 2000.

[17] A.L. Nelson, E. Grant, T.C. Henderson, Competitive relative performance evaluation of neural controllers for competitive game playing with teams of real mobile robots, Measuring the Performance and Intelligence of Systems: Proceedings of the 2002 PerMIS Workshop, NIST Special Publication 990, pp 43-50, Gaithersburg MD, Aug. 13-15, 2002.

Corresponding Author E-mail:
alnelso2@eos.ncsu.edu egrant@unity.ncsu.edu