EVOLUTIONARY NEURAL CONTROLLERS FOR MOBILE ROBOT COLONIES

Edward Grant, CRIM, North Carolina State University, USA, egrant@ncsu.edu
Leonardo Mattos, CRIM, North Carolina State University, USA, lsmattos@ncsu.edu
Greg Barlow, CRIM, North Carolina State University, USA, gjbarlow@ncsu.edu
Andrew L. Nelson, ECE Department, University of South Florida, USA, asnelson@ece.usf.edu
Kyle Luthy, CRIM, North Carolina State University, USA, kaluthy@ncsu.edu
Blaine Levedahl, CRIM, North Carolina State University, USA, bleveda@ncsu.edu
Gordon Lee, ECE Department, San Diego State University, USA, glee@kahuna.sdsu.edu

ABSTRACT

This paper demonstrates how a well designed autonomous mobile robot platform can be used in a variety of robot applications, and how this platform can provide a generic test-bed for robot colony research. The paper describes research into robot architectures, integrated sensing, robot colonies, evolutionary neural controllers, and distributed sensing and communication.

KEYWORDS: autonomous mobile robot architecture, integrated sensing, evolutionary control, robot colony learning.

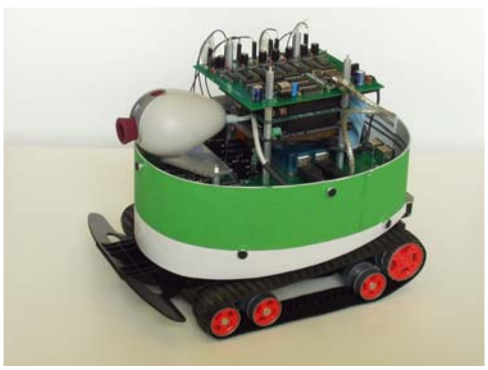
1. INTRODUCTION

The EvBotII autonomous mobile robot platforms were initially developed to test high-force and high-torque solid state motors, but quickly became the focus for other research. Traditionally, in most of the contemporary applications of mobile robots, the research focused on navigation, localization, path planning, and adaptive controller development. Mobile robot colony research focused on cooperation and collaboration, but with very specific goals. Here, it is the generic nature of robot learning [1-3] that is explored and showing that real technology [4-5] can be used for other research domains, e.g., directional control based on sound, distributed sensing and communication.

Here, the research concentrated on developing a colony of miniature robotic platforms that are inexpensive, robust, and is computationally powerful enough to be autonomous and display behaviors [4-7], including avoidance [7-10]. The robot architecture was designed to integrated data from "plug and play" sensors, to run complex evolved controllers onboard, and to have RF communication. Neural controllers evolved in simulation were ported onto a single robot, or to a colony of robots based on the same architecture. The autonomous nature and onboard RF communications enabled these robots to perform complex interactions within a test-bed. To be truly useful, comprehensive development tools must be available for robot colonies, in particular those for supporting rapid controller construction through evolution in robot simulation [11]. This paper presents the design and development of a new robotic platform capable of advanced robotics experimentation, colony behaviors and evolutionary robotics. This robot platform is called the "EvBot II". The research leading to the design and development of the EvBot II is presented. Later, software design will be explained in depth and demonstrations showing the current potential of the EvBot II will be demonstrated. Lastly, thought will be given to how the EvBot platform can be expanded and enhanced in the future.

2. THE EvBot II ROBOTIC PLATFORM

This research utilizes a small a fully autonomous mobile robot called EvBot-II for the validation and transferability experiments. This robot is part of a colony of nine mobile robots recently developed by the Center for Robotics and Intelligent Machines, which are intended for experimentation with evolutionary robotics. Each robot in the colony is 9 in. wide by 12 in. long by 9 in. tall, and is constructed on a two track treaded wheel base. The EvBot II robot features powerful motors, wheel encoders and an utility board that hosts two microcontrollers, which can drive up to four DC or servo motors in closed loop. Each robot is also equipped with a PC/104 onboard computer that is responsible for all the computations, data acquisition and high-level control. A custom Linux distribution derived from RedHat Linux 7.1 is used as the operating system and is capable of supporting high-level software packages like Matlab. The EvBots in the colony are linked to one another and to the internet via a wireless network access point. In addition, each robot features a USB hub with four input ports and supports video data acquisition (up to 640 x 480 live motion resolution) through an USB video camera. A photograph of one EvBot-II featuring an optional acoustic array sensor is shown in Figure 1.



(a)

(b)

Figure 1(a) The EvBot-II fitted with an acoustic array sensor and Figure 1(b) The EvBot II colony.

Installing EvBot II robot controllers begins with starting Matlab and running the file startup.m which is the entry point for the robot controllers. By default, the controller called by startup.m is demo_drive_robot.m, located in the directory /matlab/demo_controller/. This controller moves the robot randomly. The controller called by startup.m can be changed by changing the file startup.m itself. Other controllers are already installed on the flash card, including a random controller and a neural network controller that use more complex code based on structures. These controllers also make use of the USB camera, so code is included for it. To run a random controller a modified version of startup.m is included on the flash card. A new controller can also be installed by removing the flash card from the EvBot II and installing it in a laptop computer running Linux. Using the laptop you can modify all the files in the flash card and also include new controllers. The EvBot II has two BasicX-24 microcontrollers, which are connected to the MZ104 CPU using the COM1 serial port. The RS232 settings of such microcontrollers are: Emulation: ANSI (use auto detect when running Hyperterminal); Baud rate: 19200; Data bits: 8; Parity: None; Stop bits: 1; Flow control: None. If desired, you can connect any computer to the Master BasicX (BasicX 1) using the connector CN5. This connector also allows for onboard reprogramming. The Slave BasicX does not have connection to talk to an external device. It is directly connected to the Master BasicX, from which it and receives commands. The Slave BasicX needs to be taken out of the circuit board for programming.

2. THE EvBot II RANGE SENSORS

The vision-based range-finding sensor systems on the robots used fixed geometric properties of the physical maze environment to calculate the ranges and angles of materials. Using color and position, the vision system could detect walls, robots, and goals. The goals are stationary cylinders and were used in the robotic version of the 'Capture the Flag' game.

At each sensor update interval, and for each object type, range and angle values were calculated over the horizontal field of view of the robot's camera. A vector of range values was produced for each object type. Angular data was implicitly encoded in the order of the range values reported in each object range data vector. Object type information was not explicitly given to the robot neural controllers. Controllers were only given these resulting numerical data vectors. All associations relating distances, angles, and object types must be learned by the neural networks.

In separate experiments the navigation and directional control of the EvBot-II robot was based solely on data acquired from an onboard passive sonar system. This sensor system makes use of an acoustic array formed by eight microphones distributed on a fixed 3-D arrangement around the robot. It uses data collected from the array to perform beamforming and to find the direction of sound sources.

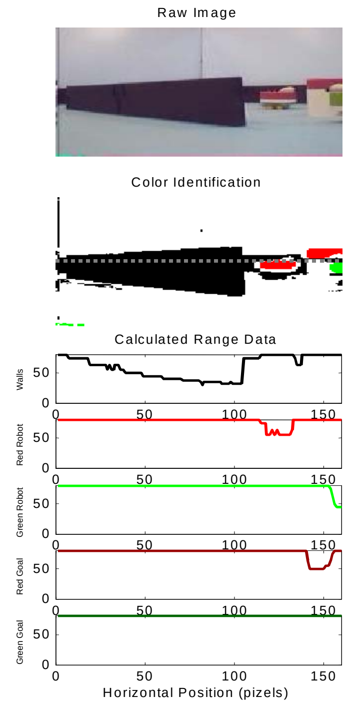


Figure 2 Vision Range Data for Neural Network Training

Audio direction data is obtained by a passive sonar system, through developing a custom USB data acquisition board to sample all eight sensors were simultaneously. The sampled signals are then transferred to the EvBot II on-board PC/104 computer and processed by a beamforming algorithm. In essence, the beamforming algorithm performs microphones sampling and integrating the sampled input data signals. The algorithm then calculates the magnitude of all the sounds coming from all around the robot, all 360^0. The resulting information is represented in the form of a polar plot of sound intensities, see Figure 3.

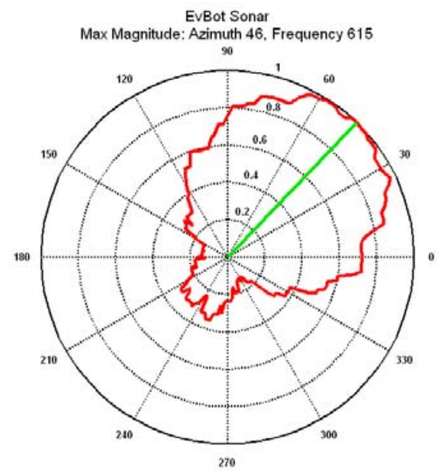


Figure 3: Plot of the data generated by the passive sonar.

3. EVOLVED NEURAL CONTROLLER EXPERIMENTS

Neural network controllers evolved through playing a game called capture-the-flag were transferred to the EvBot II robots and tested in a physical maze environment [11]. In capture-the-flag, there are two teams of robots and two goals. The goal and the two robots on a team are the same color. Team 1 is red and Team 2 is green. In the game, robots on one team must try to come within a certain distance of the other team's goal while protecting their own goal from their opponents. The first robot to arrive at their opponents' 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. Evolved controllers were placed in competition with knowledge-base controllers coded to play robotic “capture-the-flag.” Figure 5 shows the results of 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 and red robots respectively. In the simulations and in the real environment, robots displayed several learned 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.

In this research, a generalized evolvable neural network architecture capable of implementing a very broad class of network structures was used. The networks may contain arbitrary feed forward and feedback connections between any of the neurons in the network. Networks contain neurons with heterogeneous activation functions including sigmoidal, linear, step-threshold, and Gaussian radial basis functions. Neurons include a variable time-delay associated their inputs. This give networks the potential to evolve temporal processing abilities.

In Figure 4, the inputs to the network (left) are supplied by the robot’s video range emulation sensors. The outputs of the network (right) are interpreted as wheel motor speed commands. Figure 4 shows an example of a graphical representation of an evolved neural network. The connectivity and weighting relationships are specified by a single two-dimensional matrix **W** of real valued scalar weights, neuron types and time delays are given in a vector structure **N** with one formatted field per neuron. **W** and **N** form the basis of the genetic encoding for each network.

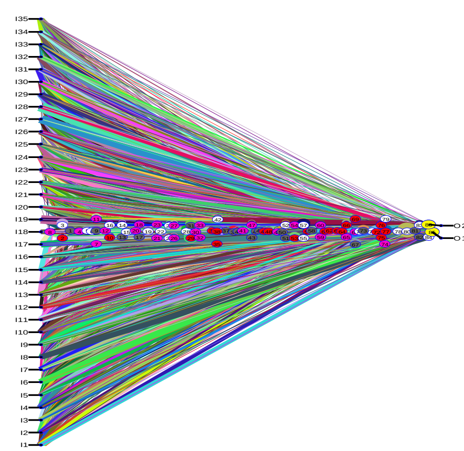


Figure 4 An example robot neural network controller

Formally, the genome for a network **C** can be specified by the two dimensional matrix of real numbers

$$\mathbf{C} = [\mathbf{W} : \mathbf{N}'] \qquad (1)$$

where **N'** is a matrix of scalars extracted from the formatted structure **N**. Mutation of a network can be formalized by the compound relation

$$\mathbf{C}' = M_s(M_c(M_w(\mathbf{C}))) \qquad (2)$$

where **C** is the chromosome of the parent network and **C'** is the resulting mutated offspring network chromosome. M_w M_c and M_s are genetic operators that mutate the weights, the connections, and the neuron structure of the network respectively. Any or all of the different types of mutation can occur during propagation.

3. EXPERIMENTS INVOLVING THE EvBot II ROBOTS

Evolved neural controllers were used in two sets of experiments in the generic test-bed. First, competition of evolved controllers against a knowledge-based controller was used to measure the absolute fitness of evolving populations. Figure 5 shows the progression of acquisition of game playing ability over the course of evolution of population 2. The figure show competitions performed with the best current individual taken every 100 generations starting with generation 150 and ending with generation 550. Each generation was tested in the three test worlds, and in each case 240 games were played against the knowledge-based controller (for a total of 15 tournaments).

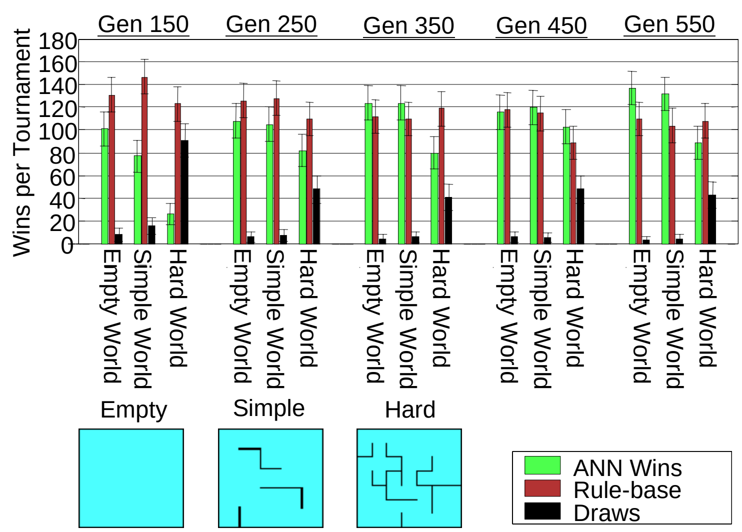


Figure 5. Performance of population 2 in competition against the knowledge-based controller at several generation points during the course of evolution.

The second set of experiments conducted dealt with the passive sonar sensors and direction control. No evolutionary learning was used in these experiments, here the peak magnitude of the polar diagram determined the direction control of the robot. That is, the robot was attracted to a sound source, i.e., the sound of a helicopter. The third set of experiments used both evolutionary learning and the passive sonar. These experiments emulated the directional control of a UAV homing in on a target.

4. CONCLUSIONS

In evolutionary controller learning the sequence of grouped tournament results shows a steady increase in game playing ability. This is especially prominent in the most difficult test world. At generation 150, the best individual in the evolving population was only able to win 26 games out of 250 while the Knowledge-based controller won 123 games. At generation 450, the evolving controllers were able to play competitively with the Knowledge-based controller, wining 103 games to the Knowledge-based's 89. This steady progression of fitness is not directly apparent in the raw training fitness data from the evolution of population 2. These results indicate that evolved behaviors were achieving absolute improvement over time. The direction control experiments using passive sonar and passive sonar along with evolutionary learning produced exceptional results. The overall impression is that the EvBot II robot platform is a generic test-bed for conducting experiments into, evolutionary robotics, cooperating robot colonies, UAV emulation, and in distributed sensors organization and communication.

REFERENCES

[1] F. Gomez, R. Miikkulainen, "Incremental Evolution of Complex General Behavior", *Adaptive Behavior*, Vol. 5, pp. 317-342, 1997.

[2] 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.

[3] J. Kodjabachian and J.-A. Meyer, "Evolution and Development of Neural Networks Controlling Locomotion, Gradient-following, and Obstacle Avoidance in Artificial Insects", *IEEE Transaction on Neural Networks*, Vol. 9, No.5, September 1998.

[4] D. Filliat, J. Kodjabachian, and J. A. Meyer, "Incremental Evolution of Neural Controllers for Navigation in a 6 Legged Robot", Sugisaka and Tanaka, editors, *Proc. of the Fourth International Symposium on Artificial Life and Robotics*, Oita Univ. Press, 1999.

[5] N. Jakobi, "Running Across the Reality Gap: Octopod Locomotion Evolved in a Minimal Simulation", *Proceedings of the First European Workshop on Evolutionary Robotics: EvoRobot'98*, 1998.

[6] D. Floreano and S. Nolfi, "Adaptive Behavior in Competing Co-evolving Species", Mantra Technical Report, LAMI, Swiss Federal Institute of Technology, Lausanne, 1997. Also submitted to ECAL97.

[7] D. Floreano and 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.

[8] N. Jakobi, P. Husbands, and I. Harvey, "Noise and the Reality Gap: The Use of Simulation in Evolutionary Robotics", F. Moran, A. Moreno, J. Merelo, and P. Chacon, editors, *Advances in Artificial Life:* Proc. 3rd European Conference on Artificial Life, pages 704--720. Springer-Verlag, Lecture Notes in Artificial Intelligence 929, 1995.

[9] 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, April 2002.

[10] I. Harvey, P. Husbands, D. Cliff, A. Thompson and N. Jakobi, "Evolutionary Robotics: the Sussex Approach", *Robotics and Autonomous Systems*, Vol. 20, Nos. 2-4, pp. 205-224, 1997.

[11] Andrew L. Nelson, Edward Grant, Gregory J. Barlow, and Thomas C. Henderson. "A colony of robots using vision sensing and evolved neural controllers." Proceedings of the IEEE Conference on Intelligent Robots and Systems. Las Vegas, NV. October 2003.