A.L. Nelson, E. Grant, “Developmental analysis in evolutionary robotics,” in *Proceedings of the 2006 IEEE SMC Mountain Workshop on Adaptive and Learning Systems (SMCals’06)*, July 2006, pp. 201-206.

Developmental Analysis in Evolutionary Robotics

A. L. Nelson
Androtics, LLC
PO Box 44065, Tucson, AZ 85733, USA
E-mail: alnelson@ieee.org
Phone: 520-822-6921

E. Grant
Dept ECE, NC State University
Raleigh, NC 27695, USA
E-mail: egrant@ncsu.edu
Phone: 919-513-0787, Fax: 919-515-5523

Abstract—**This paper presents a developmental analysis of robot controllers created using evolutionary robotics (ER) methods. ER uses artificial evolution to automatically design and synthesize intelligent robot controllers. An aggregate fitness function that injects relatively little *a priori* task knowledge into the evolving controllers was used. We analyze the course of development of robot controllers evolving to perform a competitive goal-locating task. To sample the course of evolution, controllers were taken from progressively more advanced generations, and were then tested in a novel environment. Developments and changes in the controllers’ abilities and competencies were identified and correlated with overall controller fitness. As evolution progressed, it was found that robots evolved more complex high-level behaviors that were not explicitly selected for by the fitness function.**

Index Terms—Evolutionary robotics, developmental robotics, evolutionary neural networks, swarm robotics.

I. Introduction

Evolutionary robotics (ER) is an area of autonomous robot control research. The main objective of ER is to develop automatic methods for synthesizing intelligent autonomous robot controllers. These automatic methods should not require extensive *a priori* knowledge of the particular control tasks for which the robot controllers are intended. The goal then is to achieve a form of embodied machine learning that can create new behaviors for robots that go beyond the mere optimization of *a priori* known control strategies.

In order for the full potential of ER to be achieved, methods of controller fitness evaluation must be developed that allow for the evolution of novel solutions not previously envisioned by human researchers. The less information a fitness function contains about features of a given solution to a given control task, the more freedom the evolutionary process will have in evolving a novel solution. The most unbiased fitness functions use information derived only from high-level success or failure to complete a given overall task. These are known as *aggregate fitness functions*. Unfortunately, initial populations of controllers often have no detectable ability to complete complex tasks, so pure success-failure aggregate fitness functions produce no selection pressure and evolution cannot commence (the *bootstrap problem*).

To address this issue we use a multimodal fitness function that initially uses some minimal task-related features early in evolution (a bootstrap mode), and then switches to pure aggregate success-failure selection when any individual within the population is capable of completing the overall task to some degree.

In this paper we present a case study of how evolving controllers acquire specific abilities and behaviors during evolution.

Robot controllers were evolved to perform a competitive searching task. During the course of evolution, individuals were selected and saved from the population for later developmental analysis. Controllers were selected from the 1st generation, the 50th, and then every 100 generations thereafter up to 650 generations. This provided us with a set of individual controllers that represents the gradual development of abilities over the course of evolution in the population as a whole.

The set of robot controllers spanning the course of evolution were tested in a novel environment that had not been used during evolution with constant initial conditions. The behaviors of the robots from each test generation were observed, analyzed and correlated with fitness function modes.

The majority of the testing was done in simulated environments. However, the research platform used here has been extensively validated in real robots. Further, the resulting robot controllers from the final generation of the evolved population investigated in this work were tested in real robots operating autonomously and asynchronously in a real environment.

The paper is organized as follows: the remainder of the Introduction Section presents a review of related research. Section II presents an overview of the methodology used here, the robot systems, and the evolutionary neural network controller. Section III presents results analyzing the development of behaviors in the controller population over the course of evolution. Section IV offers closing remarks.

A. Related Work

The field of ER has been reviewed in several publications [1][2][3]. Much of the previous research focused on evolving controllers for simple tasks such as phototaxis [4][5], or object avoidance [6][7]. The most complex robot controllers evolved using ER might include three or four coordinated fundamental sub-behaviors [8][9][10][11]. The fitness functions used to evolve these more complex controllers were fairly intricate, and relatively selective for an *a priori* known or pre-defined solution.

Locomotion in combination with obstacle avoidance in legged robots has been reported in several ER studies [12][13][6][14]. Filliat *et al*. [12] evolved locomotion and object avoidance controllers for a hexapod robot using networks of threshold neurons. Controllers were evolved in simulation and transferred to real robots for testing. Jakobi *et al.* [13] described the use of minimal simulation to evolve controllers for an eight-legged robot with sixteen leg actuators.

Kodjabachian *et al.* [6] described incremental evolution of walking, object avoidance and chemotaxis in a simulated six-legged insectoid robot. Hornby *et al.* [14] described the evolution of ball chasing using an 18-DOF quadruped robot.

Object pushing behaviors were evolved in [15][16]. This task required that differential drive robots push small cylinders toward a light source. In [17], Lee *et al.* investigated a similar box-pushing behavior using Genetic Programming (GP).

Several examples of competition in the form of co-evolution of competing species have been reported in the literature. Cliff and Miller investigated the co-evolution of competing populations of predator and prey robots [18][19]. Similar works have been reported in [2][20][21][22].

Evolution of controllers using competition within a single population (*intra-population competition*) is investigated in [23] and further analyzed in the current work.

The most complex tasks addressed in the literature involve some form of sequential action. Nolfi [8] reported on the evolution of a garbage collection behavior in which a robot must pick up pegs in an arena and deposit them outside the arena. Ziemke [24] studied the evolution of robot controllers for a task in which a robot must collide with objects ("collect" them) in one zone and avoid them in another. In [9] Floreano *et al.* reported on the evolution of a behavior in which robots move to a light and then back to a home zone. Another example of evolving controllers for a relatively complex task is reported in Tuci *et al.* [10]. Robot controllers evolved to produce lifetime learning in order to predict the location of a goal object based on the position of a light source.

Flocking behaviors have also been investigated. Ashiru described the evolution of a simple robot flocking behavior in [11]. A robot coordination task in which two robots evolve to move while maintaining mutual proximity is reported by Quinn in [25]. Baldassarre *et al.* [26] evolved homogeneous controllers for a task in which four robots must move together in a small group toward a light or sound source. In [27] aggregation of small robots into a larger structure is investigated and makes use of a relatively complex hand-formulated fitness function.

The development of methods for general fitness selection during evolution of controllers is crucial to the future of ER. This view is reflected to a degree in the literature [28][7] and as early as the mid 1960's it was pointed out that creating a method of fitness selection capable of selecting for complex novel behavior was likely to be difficult [29].

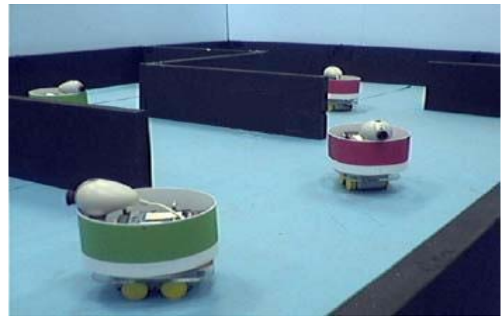

Fig. 1. The robot maze environment containing several robots.

II. Methodology

ER applies population-based artificial evolution to evolve autonomous robot controllers. The process of controller evolution consists of repeating cycles of controller fitness evaluation and selection that are roughly analogous to a generation in natural evolution. During each cycle, or *generation*, individual controllers taken from a population of controllers perform a task or engage in an evaluation period. This involves transferring each controller into a robot (either real or simulated) and allowing the robot to interact with its environment (which may include other robots) for a period of time. Following this, each controller's performance is evaluated based on a fitness function (objective function). The fitness function is at the heart of any evolutionary computing application. It is responsible for determining which solutions (controllers in the case of ER) within a population are better at solving the particular problem at hand. In the final step of every cycle, a genetic algorithm (GA) is applied. The GA uses information generated by the fitness selection function to select and propagate the fittest individuals in the current population to the next generation. During propagation, controllers are altered slightly using stochastic genetic operators such as mutation and crossover to produce offspring that make up the next generation of controllers. This process is repeated for many generations to train populations of robot controllers to perform a given task.

A. The Task

Neural network-based robot controllers were evolved to play a robot version of the competitive team game *Capture the Flag*. In this game, there are two teams of mobile robots and two stationary goal objects. Robots on the first team and one of the goals are of one color (red). The other team members and their goal are another color (green). In the game, robots of each team must try to approach the other team's goal object while protecting their own goal. The robot which first comes in contact with its opponent's goal wins the game for its team. The game is played in maze worlds of varying configurations.

B. The Robots

A multirobot system composed of two teams of robots was used for this research [30] (Fig. 1). Each robot is approximately 6 in. in diameter, and uses skid steering differential drive systems. The robots use x86 class processors running the Linux operating system and MATLAB. Robots support video data acquisition and rely solely on video for all sensing of their environment [31]. Each robot is fully autonomous and capable of performing all computing, control and data management on board.

C. The Evolutionary Neural Networks

The evolvable controller structures used in this research belong to a class of generalized network architectures. These networks contain feed forward and feedback connections, mixed types of neurons, and variable time-delayed internal connections. Neuron activation function types include sigmoid, linear, step-threshold, and Gaussian radial basis functions. Other research involving complex networks can be found in [2][13].

The connectivity and weighting relationships are contained in a variable-size two-dimensional matrix **W**. Information specifying neuron types are stored in a vector structure **N**, with one formatted field per neuron. Current and past network inputs and neuron functional levels (outputs) are stored in an ordered matrix, **I**. The network input-output relationship is given by:

$$\mathbf{I}(t+1) = \mathit{Network}(\mathbf{I}(t), \mathbf{N}, \mathbf{W}) \tag{1}$$

and

$$\mathbf{o} \subset \mathbf{i}(t+1) \tag{2}$$

where **o** is a vector of values from specified output neurons and is a subset of $\mathbf{i}(t+1)$, the first row of the new $\mathbf{I}(t+1)$. During each sensor-motor update cycle of the robot controller the functional *Network* in (1) calculates the activations of each of the neurons specified in **N** (in order), and places the resulting values in successive elements of **I** until the output neurons are updated.

An example of a controller network is shown in Fig. 2. The network uses 150 inputs to accommodate range sensor information and produces two drive-wheel commands that control the robot's differential-steering wheel motors.

D. Fitness Function (Objective Function)

Often, ER experiments use complicated task-specific fitness functions to determine controller fitness and selection during evolution. Unfortunately, the use of such fitness functions can restrict the evolutionary search space and drive the evolving populations of robot controllers toward *a priori* known solutions. To avoid this we apply a form of aggregate fitness evaluation that injects as little *a priori* knowledge into the evolving controllers as possible.

In this work, robot controller fitness was based on robot performance in tournaments of games. Each tournament involved all of the controllers in the current population.

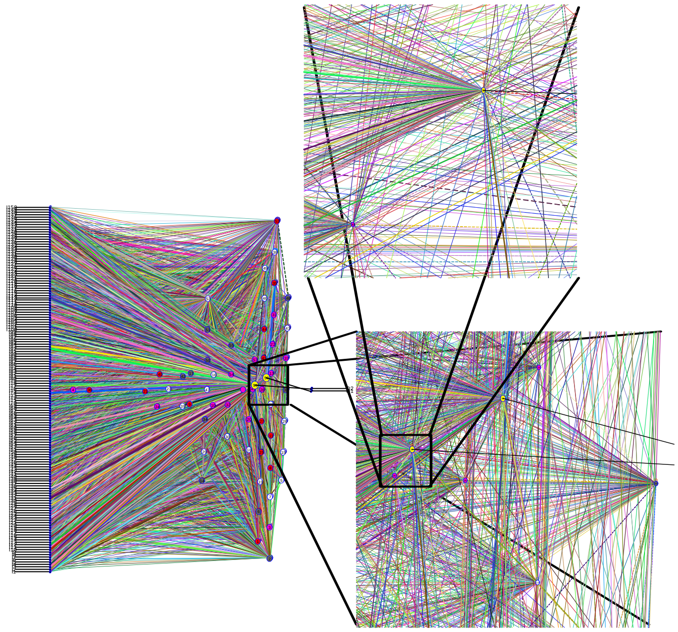

Fig. 2. An evolved controller network shown in several magnifications. This particular network is one of the best performing networks evolved using this evolutionary robotics system.

Robots on a given team used homogeneous (but separate) controllers, so each game represented a competition between two individuals from the evolving population. Controllers in the population competed directly against one another, hence the performance of one controller could have a direct effect on the performance of another.

A fitness function with two mutually exclusive modes was used. The function has an initial mode that accommodates sub-minimally competent nascent populations and a second mode that selects for aggregate fitness based only on overall success or failure to complete the task (winning or losing games).

Fitness $F(p)$ of an individual p in population **P** ($p \in \mathbf{P}$) is given by:

$$F(p) = F_{\text{mode\_1}}(p) \oplus F_{\text{mode\_2}}(p) \tag{3}$$

where $F_{\text{mode\_1}}$ is the initial minimal-competence mode and $F_{\text{mode\_2}}$ is the purely success/failure based mode. In (3) $\oplus$ indicates dependent exclusive-or: if $F_{\text{mode\_2}}$ is non-zero for *any* member of the current population, it is used exclusively to evaluate *all* controllers in the current population. Otherwise fitness is calculated using $F_{\text{mode\_1}}$.

The first mode of the fitness function selects for the ability to travel a given distance D through an environment:

$$F_{\text{mode\_1}} = F_{dist} - s \tag{4}$$

where F_{dist} calculates a penalty proportional to the difference between distance d traveled by the best robot on a team, and the maximum required distance D, and s is a penalty constant applied in the case that all the robots on a team stop moving. F_{dist} is given by

$$F_{dist} = \begin{cases} -\alpha * (D - d) & \text{if } d < D \\ 0 & \text{otherwise} \end{cases} \tag{5}$$

where D is defined as half the length of the training environment's greatest dimension and α is a constant of proportionality.

The minimal competence mode selects for the minimal condition that at least one member of a team of robots should be able to travel halfway through its environment without getting stuck. There is no information encoded into this mode for how this might be accomplished, and once the initial (minimal-competence) mode is satisfied, it provides no further selective pressure.

The second fitness function mode $F_{\text{mode\_2}}$ calculates fitness based only on success or failure of the controllers to complete the overall task, i.e. winning games. Each controller in the current population plays two games per tournament (or generation). Fitness scores are based on the number of wins achieved. The possible win/lose outcomes of these games incur different levels of fitness. If both games are won, the controller receives a score of 3. If one game is won and one is played to a draw, a score of 1 is given, and if one game is won and the other is lost a score of 0.5 is given. Recall that if no controller in the population manages to win a game in a given tournament, the initial minimal competence mode $F_{\text{mode\_1}}$ dominates and all robots receive fitness ratings according to (4).

E. Evolutionary Conditions

During each generation, the individuals from the fittest 50% of the population were selected and retained, and their offspring replaced the least fit 50% of the population.

Populations were kept at a constant size of 40 networks. Network weight mutation rates were set at 25%, and weight mutation magnitudes were selected from a uniform distribution on the interval [-1, 1]. These and other algorithm

Table I. Parameter settings used during evolution.

Parameter	Setting
Population size	40
Sensor inputs	150
Initial network size	60 neurons
Chance of adding or removing a neuron (during mutation)	70%
Weight initialization range	[-1 1], uniform dist.
Weight mutation magnitude	[-1 1], uniform dist.
Weight mutation rate	25%
Initial feed forward connectivity	60%
Initial feedback connectivity	20%
Chance of adding or removing a connection (during mutation)	70%
Elitism level (per generation)	Single best from previous generation
Population replacement rate	50%
Generations (per evolution)	650

III. RESULTS

A. Developmental Analysis of Evolving Controllers

A set of experiments was performed to investigate the development of controller abilities during evolution. Controllers taken from different generations spanning the course of evolution were tested and compared. Fig. 3 shows a series of eight games, each involving controllers taken from advancing generations of the population.

In each game, the best controller from the generation being tested was used to control the competing robots. The first game (Fig. 3 panel (a)) used an original progenitor controller from the initial un-evolved population. In the second game (panel (b)), the best controller from generation 50 was used. In the subsequent panels (c) to (h), 100 generations progressively separated the "ages" of the populations. The same initial positions for robots and goals were used in each game.

Robots show increasing levels of performance over the course of evolution. The un-evolved controllers in panel (a) collide with walls almost immediately. At the 50th generation, one of the robots is able to make progress through the environment before running into a wall. Generations 150, 250, and 350 show increasing levels of wall avoidance and navigation but robots are unable to locate the goals and cannot win games. At the 250th and 350th generations (panels (d) and (e)), robots are able to travel indefinitely without getting stuck but none of them are able find their opponent's goal. It is during the 450th, 550th, and 650th generations that controllers have evolved to be able to win games. The final three games of Fig. 3 all terminate with wins. In the last three panels, robots appeared to be executing a "left-hand mouse rule" search strategy in conjunction with object avoidance.

Table II summarizes the discussion of acquisition or development of behaviors over the course of evolution. The

Table II. Qualitative acquisition of behaviors over the course of evolution. Solid dots indicate that a behavior is observed in that generation. Open dots indicate that the behavior has been superseded by another.

Generation	Rudimentary wall avoidance	Simple navigation (always turn left)	Avoid own goal	Complex navigation (left and right turns)	Left-hand mouse search strategy
0					
50	●				
150	●	●			
250	○	●	●		
350	○	○	●	●	
450	○	○	●	●	●
550	○	○	●	●	●
650	○	○	●	●	●

A.L. Nelson, E. Grant, "Developmental analysis in evolutionary robotics," in *Proceedings of the 2006 IEEE SMC Mountain Workshop on Adaptive and Learning Systems (SMCals'06)*, July 2006, pp. 201-206.

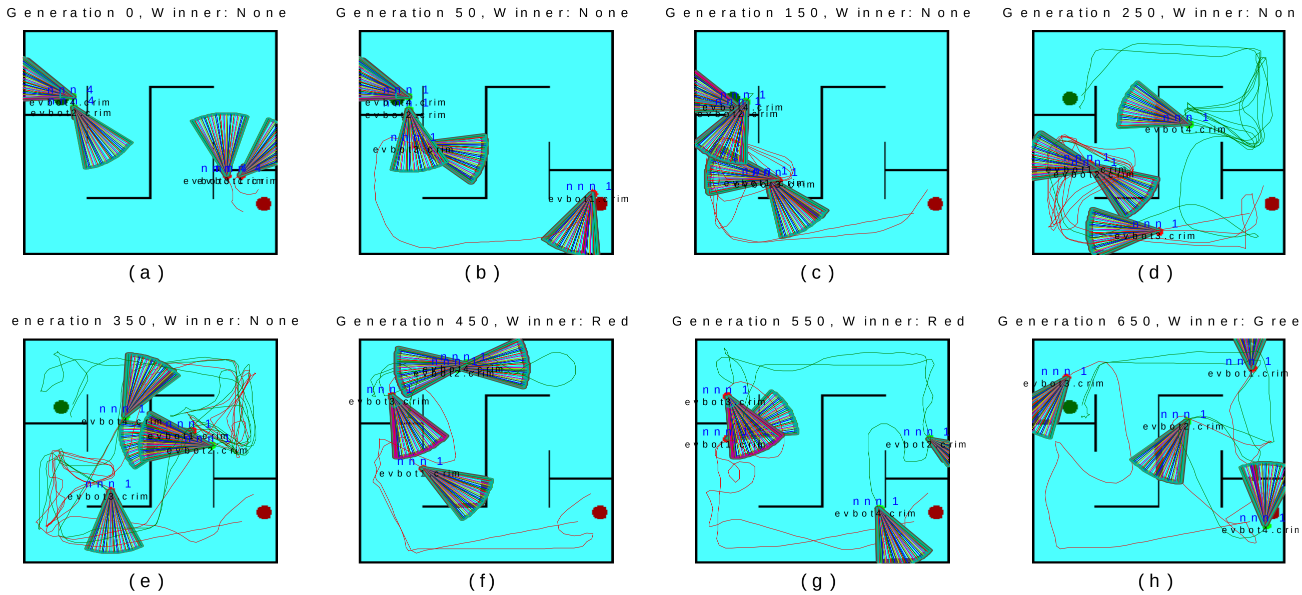

Fig. 3. Sequence of games played with controllers from sequential generations of the evolving population.

identification of a particular behavior represents a qualitative human assessment based on observation of robots during game sequences. In the later generations, exact behavior is very difficult to predict, and behaviors observed from the distal (exterior to the robot-controller system) are not necessarily reducible to discrete behaviors at the proximal level (from the point of view of the robot controller). Fundamental behaviors or skills at the proximal level are likely to be inextricably coupled with one another, and with sensor inputs [32].

B. *Generalization of Evolved Controllers*

Fig. 4 shows an example game played between controllers taken from the 450th generation of the population. The game was conducted in a simulated maze environment that was novel to the evolved controllers (i.e. not seen during evolutionary training). In Fig. 4, the smaller dots with the fan-like graphics are the robots. The fan-like graphics represent sensor data and are not physical objects. The paths taken by the robots during the games are indicated by the irregular curves. There are two robots on each team to make a total of four robots in each game. The larger dots represent the stationary goal objects. The heavy black line segments represent walls. The figure demonstrates that evolved controllers generalize to novel environments. This maze world is many times larger than any world seen by the controllers during evolution and includes novel structures, such as cross-shaped cul-de-sacs, long corridors, and large open spaces.

Robots being controlled by the best neural networks from the 450th generation very rarely collide with objects. A collision can result in the immobilization of the robot. Two of the robots (one from each team) did eventually become stuck during the game shown in Fig. 4. One robot on the green team collided with an object and became permanently immobilized near the 400th time step. Similarly, one robot from the red team became permanently immobilized after the 700th time step. The other two robots continued to travel about the environment for the duration of the game (about 1300 time steps).

As a final note, robots using controllers from the population have evolved limited abilities to extricate themselves from collisions (being stuck). These evolved controllers are not purely reactive. Controllers were observed to remain immobile for many time steps (30 or more) and then to back up and spin around. Even so, most of the observed evolved behaviors were reactive. A close examination of controller outputs revealed that actuator commands stabilize relatively quickly, but do not reach a constant steady state. In a simple experiment, a controller from the population was repeatedly fed identical sensor inputs. Outputs did not reach exact steady states even after 30 time steps (data not shown).

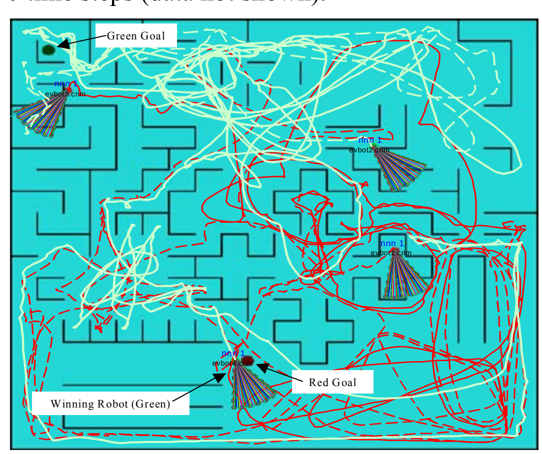

Fig. 4. Evolved controllers operating in a complex world. Small circles with indicate are the robots (fan-like graphics represent sensor data), and larger circles are the goals. The paths taken by the robots during the simulation are indicated by dotted and solid lines.

IV. Conclusion

In this paper the course of behavior acquisition in robots using neural network-based controllers was investigated.

Further work will include investigations into a broader range of environmental and algorithmic conditions and for robots using a greater range of actuators and sensors. In addition, it would be desirable to investigate alternatives to using a bootstrap mode to evolve controllers for complex tasks.

References

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

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

[3] L.A. Meeden, D. Kumar, "Trends in evolutionary robotics," in Soft Computing for Intelligent Robotic Systems, L.C. Jain, T. Fukuda, Eds., Physica-Verlag, New York, NY, 1998, pp. 215-233.

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

[5] 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, Apr. 2002.

[6] J. Kodjabachian, 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, pp. 796-812, Sept. 1998.

[7] 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.

[8] S. Nolfi, "Evolving non-trivial behaviors on real robots," Robotics and Autonomous Systems, vol. 22, no. 3-4 pp. 187-198, 1997.

[9] D. Floreano, J. Urzelai, "Evolutionary robots with on-line self-organization and behavioral fitness," Neural Networks, vol. 13, no. 4-5, pp. 431-443, June 2000.

[10] E. Tuci, M. Quinn, I. Harvey, "Evolving fixed-weight networks for learning robots," in Proceedings of the 2002 Congress on Evolutionary Computing, Honolulu, HI, vol. 2, 2002, pp 1970-1975.

[11] I. Ashiru C.A.Czarnecki, "Evolving communicating controllers for multiple mobile robot systems," in Proceedings of the 1998 IEEE International Conference on Robotics and Automation, vol. 4, 1998, pp. 3498-3503.

[12] D. Filliat, J. Kodjabachian, J.A. Meyer, "Incremental evolution of neural controllers for navigation in a 6 1egged robot," in Sugisaka and Tanaka, Eds., Proc. Fourth International Symposium on Artificial Life and Robotics, Oita Univ. Press, 1999.

[13] N. Jakobi, "Running across the reality gap: octopod locomotion evolved in a minimal simulation," in P. Husbands, J.A. Meyer, Eds., Evolutionary Robotics: First European Workshop, EvoRobot98, Springer-Verlag, 1998, pp. 39-58.

[14] G.S. Hornby, S. Takamura, J. Yokono, O. Hanagata, M. Fujita, J. Pollack, "Evolution of controllers from a high-level simulator to a high dof robot," Evolvable Systems: From Biology to Hardware; Proceedings of the Third International Conference (ICES 2000), J. Miller, Ed., Lecture Notes in Computer Science, vol. 1801, Springer, 2000, pp. 80-89.

[15] A. Ishiguro, S. Tokura, T. Kondo, Y. Uchikawa, "Reduction of the gap between simulated and real environments in evolutionary robotics: a dynamically-rearranging neural network approach," in The 1999 IEEE International Conference on Systems, Man, and Cybernetics: Conference Proceedings, Vol. 3, 1999, pp. 239-244.

[16] K. Kawai, A. Ishiguro, P. Eggenberger, "Incremental evolution of neurocontrollers with a diffusion-reaction mechanism of neuromodulators," in International Conference on Intelligent Robots and Systems 2001 Proceedings, vol. 4, Maui, HI, Oct. 29- Nov. 3, 2001, IEEE/RSJ, pp. 2384-2391.

[17] W. Lee, J. Hallam, H. Lund, "Applying genetic programming to evolve behavior primitives and arbitrators for mobile robots," in Proceedings of the 1997 IEEE International Conference on Evolutionary Computation, 1997, pp. 495-499.

[18] D. Cliff and G. F. Miller, "Tracking the Red Queen: measurements of adaptive progress in co-evolutionary simulations," in F. Moran, A. Moreno, J. J. Merelo, P. Cachon Eds., Advances in Artificial Life: Proceedings of the Third European Conference on Artificial Life (ECAL95). Lecture Notes in Artificial Intelligence 929, Springer-Verlag, 1995, pp.200-218.

[19] D. Cliff, G. F. Miller "Co-evolution of pursuit and evasion ii: simulation methods and results," in P. Maes, M. Mataric, J.-A. Meyer, J. Pollack, W. Wilson Eds., From Animals to Animats 4: Proceedings of the Fourth International Conference on Simulation of Adaptive Behavior (SAB96). MIT Press Bradford Books, pp.506-515, 1996.

[20] S. Nolfi, D. Floreano, "Co-evolving predator and prey robots: Do 'arms races' arise in artificial evolution?" Artificial Life, vol. 4, no. 4, pp. 311-335, 1998.

[21] G. Buason, T. Ziemke, "Competitive co-evolution of predator and prey sensory-motor systems," in Applications of Evolutionary Computing – Proceedings of the second European workshop on Evolutionary Robotics, G. Raidls et al. eds., Springer-Verlag, Berlin, Heidelberg, 2003, pp. 605-615..

[22] D. Floreano, S. Nolfi, F. Mondada, "Competitive co-evolutionary robotics: from theory to practice," in R. Pfeifer, Ed., From Animals to Animats 5. Proceedings of the Fifth International Conference on Simulation of Adaptive Behavior (SAB'1998), Cambridge, MA, MIT Press, 1998.

[23] A.L. Nelson, E. Grant, T.C. Henderson, "Evolution of neural controllers for competitive game playing with teams of mobile robots," Journal of Robotics and Autonomous Systems, vol. 46, no. 3, pp. 135-150, Mar 2004.

[24] T. Ziemke, "Remembering how to behave: recurrent neural networks for adaptive robot behavior," in Medsker and Jain Eds., Recurrent Neural Networks: Design and Applications, Boca Raton, CRC Press, 1999.

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

[26] G. Baldassarre, S. Nolfi, D. Parisi, "Evolving mobile robots able to display collective behaviours," in C. K. Hemelrijk, E. Bonabeau, Eds., Proceedings of the International Workshop on Self-Organisation and Evolution of Social Behaviour, Monte Verità, Ascona, Switzerland, Sept. 8-13, 2002, pp. 11-22.

[27] M. Dorigo, V. Trianni, E. Sahin, T. Labella, R. Grossy, G. Baldassarre, S. Nolfi, J-l. Deneubourg, F. Mondada, D. Floreano, L. Gambardella, "Evolving self-organizing behaviors for a swarm-bot," Autonomous Robots, vol. 17, no. 23, pp. 223-245, Sept. 2004.

[28] D. Floreano, J. Urzelai, "Evolutionary robots: the next generation," in The 7th International Symposium on Evolutionary Robotics (ER2000): From Intelligent Robots to Artificial Life, Gomi T. Ed., pp. 231-266, AAI Books, 2000.

[29] L. Fogel, A. Owens, M. Walsh, Artificial Intelligence Through Simulated Evolution, John Wiley & Sons, Inc, New York, 1966.

[30] J. Galeotti, S. Rhody, A.L. 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), San Diego, CA, Nov. 7-9, 2002, pp. 86-91.

[31] A.L. Nelson, E. Grant, G.J. Barlow, T.C. Henderson, "A colony of robots using vision sensing and evolved neural controllers," Proceedings of the 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS'03), Las Vegas, NV, vol. 3, Oct. 27-31, 2003, pp. 2273-2278.

[32] S. Nolfi, "Evolutionary robotics: exploiting the full power of self-organization," Connection Science, vol. 10, pp. 167-183, 1998.