Online and Onboard Evolution of Robotic Behavior Using Finite State Machines

(Extended Abstract)

Lukas König
Institute AIFB
Karlsruhe Institute of Technology
76128 Karlsruhe, Germany
lukas.koenig@kit.edu

Sanaz Mostaghim
Institute AIFB
Karlsruhe Institute of Technology
76128 Karlsruhe, Germany
sanaz.mostaghim@kit.edu

Hartmut Schmeck
Institute AIFB
Karlsruhe Institute of Technology
76128 Karlsruhe, Germany
hartmut.schmeck@kit.edu

ABSTRACT

In this paper, a control system representation based on finite state machines is utilized to build an evolutionary robotic framework where evolution is performed in a swarm of simple robots in an online and onboard manner. Experiments in simulation show that the framework is capable of robustly evolving basic benchmark behaviors like collision avoidance.

Categories and Subject Descriptors

I.2.9 [**Robotics**]: Autonomous vehicles; I.2.11 [**Distributed Artificial Intelligence**]: Intelligent agents, Multiagent systems

General Terms

Evolutionary Robotics, Algorithms

Keywords

Evolutionary Robotics, Finite State Machines, Mutation, Crossover, Collision avoidance, Onboard, Online

1. INTRODUCTION

Evolutionary robotics is a field that still attracts growing interest, since it can offer solutions to robotic tasks which are hard to implement by hand. Evolution is capable of finding control systems which outperform human solutions in terms of effectiveness in solving the task and simplicity of the controller [7]. This has been shown in swarms of robots, exploiting the emergence of collective behavior [1], as well as in single robots, by a previously performed evolution in simulation, or in other scenarios [5, 2].

In general, when performing learning algorithms, two additional properties may become important, namely learning behaviors *online* and *onboard*. *Onboard* means that agents are learning having a local view on the environment only (as defined by their sensory equipment) without information from a central control having global information. Communication between agents is also limited to some local neighborhood which can change as agents move. *Online* means that, during a run, currently evolved robotic behavior is evaluated by observing its performance on the given task.

Cite as: Online and Onboard Evolution of Robotic Behavior Using Finite State Machines, (Extended Abstract), Lukas König, Sanaz Mostaghim, Hartmut Schmeck, *Proc. of 8th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2009),* Decker, Sichman, Sierra and Castelfranchi (eds.), May, 10–15, 2009, Budapest, Hungary, pp. 1325–1326

In the framework described in this paper some new mutation and recombination operators for evolving robotic behavior based on finite state machines are designed.

2. PROPOSED APPROACH

In this paper, a new recombination operator for the framework (described in [3]) is presented which is capable of performing multi-parental child generation. Due to the onboard approach, selection cannot be defined as a population-based operator. Therefore, selection and crossover are combined. In [4], two robots produce offspring when they come spatially close to each other. Child behavior is then produced by cloning the parent with the best fitness. Since reproduction in this approach occurs unpredictably, it is difficult to control the reproduction rate and the selection pressure.

Here, the robots no longer reproduce when they happen to meet other robots, but all robots reproduce simultaneously according to some global clock; every robot mates with the one (or more) robot(s) spatially closest to itself in that moment.

The effects of the recombination operator are studied repectively in combination with the usage of a memory genome, storing a robot's best controller found so far (an elitist strategy), and without the memory genome. For both scenarios, two objective functions are studied: one for collision avoidance and one for collision avoidance with an additional task of finding a gate in the middle of the field and passing it as often as possible.

Furthermore, the mutation operator is designed in a special way to generate preferably simple controllers depending on the complexity of the task, similar to the approach by Stanley and Miikkulainen [6]. Experiments show that this requirement still works in the new scenarios. So, for collision avoidance, which is a simple behavior and can, in principle, be described in a purely reactive way, evolution finds in most cases solutions where only two or three states are involved, depending on how sophisticated the obstacle avoiding maneuver is.

3. EXPERIMENTS AND RESULTS

As the model is defined in a general way, it is applicable on different robotic platforms. It has been implemented and tested on the Jasmine IIIp robot [4] which is also simulated here. The Jasmine IIIp can process simple motoric commands like driving forwards and backwards and turning left and right. Every robot has seven infra-red sensors returning values from 0 to 255 in order to measure distances to obstacles (cf. www.swarmrobot.org).

All experiments are performed on a rectangular field. For the collision avoidance task (robots are supposed to learn to not collide with any obstacles) the field is designed to be empty, where for the gate passing task (robots are supposed to do collision avoidance and additionally pass the gate as often as possible) a wall with a gate in the middle is placed in the field (Figure 1). No additional objects are placed in the field meaning walls and robots are the only obstacles. 26 robots are placed randomly (positions and angles) in the field.

Their initial genome is set to be empty (i. e., an automaton without any states or transitions). The experiments are run for 80.000 simulation cycles; this complies to a real-world driving distance of about 320 *m* for a robot which is driving only straight forward or to an experiment time of about 15 *min*. Mutation and recombination are performed after every 100 and 200 cycles, respectively.

There are essentially three groups of typical evolved behaviors. In each of the groups, behaviors of different complexity and robustness are evolved. In runs using no gate passing as fitness function, only behaviors from the first group are evolved. However, even when using the gate passing fitness, about 90% of the runs evolved behaviors from this group.

Group 1: Collision Avoidance. Figure 1 shows a trajectory of an average robot from this group in an empty field without other robots. the X marks the starting position.

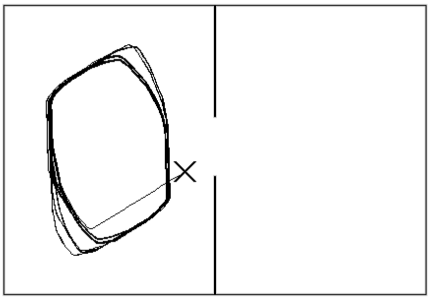

Figure 1: Trajectory of an evolved robot doing collision avoidance without passing the gate.

Group 2: Collision Avoidance and Gate Passing. Behaviors in this group are the most effective ones in gaining fitness for the entire population. Figure 2 shows the trajectory of such a robot in the field with a gate. The robot is driving close to the wall avoiding collisions until it recognizes the gate (this does not happen every time it drives past it). Then it drives through the gate, performs a small loop on the other side and drives back through the gate. Since the robot does not pass the gate every time it drives past, there is enough space for the entire population to profit from the behavior.

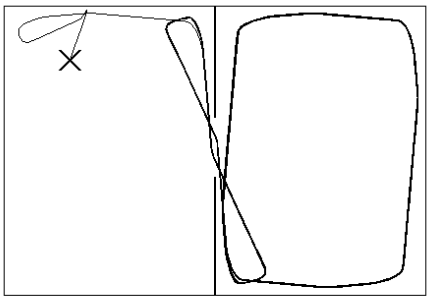

Figure 2: Trajectory of an ecolved robot passing the gate, but not blocking it for other robots.

Group 3: Constant Gate Passing. This group consists of robots which found a fitness niche exploitable for at most one or two robots at a time, gaining, however, great fitness for them. They develop a mechanism to recognize the gate and then drive constantly through it, back and forth. In some populations, the gate passing robots are interchanged, in others, always the same robot does the passing while the others cannot recognize the gate as a robot is constantly passing it. In many of these populations, collision avoidance is not evolved since the passing gained a large fitness. Figure 3 shows a trajectory of a robot performing constant gate passing.

Automata from groups 2 and 3 are typically hardly transferrable into new environments. The sensor interpretation seems to be specialized to one environment.

The experiments show that in all the scenarios, great improvements are achieved compared to experiments without the new operators. Collision Avoidance gets evolved in nearly every run (compared to only 8% in [3]), the additional task of passing the gate is evolved in about 10% of the runs. Withal, the simulation time is reduced from 2.000.000 cycles in [3] to 80.000 in the experiments described here.

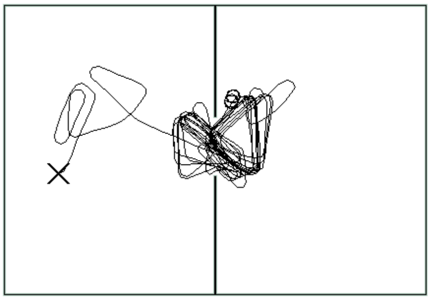

Figure 3: Trajectory of an evolved robot driving constantly through the gate.

4. REFERENCES

[1] E. Bonabeau, G. Theraulaz, and M. Dorigo. *From Natural to Artificial Systems (Santa Fe Institute Studies in the Sciences of Complexity)*. Oxford University Press, 1999.

[2] D. Floreano, P. Husbands, and S. Nolfi. *Evolutionary Robotics in Springer Handbook of Robotics*. Springer.

[3] L. Koenig and H. Schmeck. Evolving collision avoidance on autonomous robots. In M. Hinchey, A. Pagnoni, F. Rammig, and H. Schmeck, editors, *Biologically Inspired Collaborative Computing*, 2008.

[4] L. König, K. Jebens, S. Kernbach, and P. Levi. Stability of on-line and on-board evolving of adaptive collective behavior. In *Springer Tracts in Advanced Robotics*, 2008.

[5] S. Nolfi and D. Floreano. *Evolutionary Robotics. The Biology, Intelligence, and Technology of Self-Organizing Machines.* The MIT Press, Cambridge, Massachusetts, 2001.

[6] K. O. Stanley and R. Miikkulainen. Competitive coevolution through evolutionary complexification. *Journal of Artificial Intelligence Research*, 21:63–100, 2004.

[7] J. Walker, S. Garrett, and M. Wilson. Evolving controllers for real robots – a survey of the literature. *Adaptive Behavior*, 11:179–203, 2003.