Adaptive Action Selection Mechanisms for Evolutionary Multimodular Robotics

Serge Kernbach[1], Thomas Schmickl[2], Heiko Hamann[2], Jürgen Stradner[2], Christopher S. F. Schwarzer[3], Florian Schlachter[1], Alan F.T. Winfield[4], Rene Matthias[5]

[1]University of Stuttgart, Germany; [2]Karl-Franzens-University Graz, Austria; [3]University of Tübingen, Germany; [4]Bristol Robotics Laboratory (BRL), UWE Bristol, UK; [5]Karlsruhe Institute of Technology, Germany

Abstract

This paper focuses on the well-known problem in behavioral robotics – "what to do next". The problem addressed here lies in the selection of one activity to be executed from multiple regulative, homeostatic and developmental processes running onboard a reconfigurable multi-robot organism. We consider adaptive hardware and software frameworks and argue the non-triviality of action selection for evolutionary robotics. The paper overviews several deliberative, evolutionary and bio-inspired approaches for such an adaptive action selection mechanism.

Introduction

Evolutionary robotics is a well-established research field, which combines several such areas as robotics, evolutionary computation, bio-inspired and developmental systems (Nolfi and Floreano, 2000). This field is characterized by multiple challenges related to platform development, onboard fitness evaluation, running time of evaluation cycles and other issues (Levi and Kernbach, 2010). Synergies between reconfigurable robotics and evolutionary computation are of special interest, because here the high developmental plasticity of the hardware platform can be exploited to realize the goal of adaptivity and reliability.

Modern reconfigurable multi-robot systems possess very high computational power and extended communication for performing evolutionary operations on-board and on-line. These hardware capabilities allow us to extend the software framework to include the whole regulative, homeostatic and evolutionary functionality for achieving long-term autonomous behavior of artificial organisms (Levi and Kernbach, 2010). In this work we focus on the issues of running multiple control processes on board the robot. These processes are created by evolutionary development, homeostasis and self-organizing control, learning, and middle- and low-level management of software and hardware. Some of these processes will have a protective role in preventing the mechatronic platform from harm during the evaluation phases. We expect that regulative and developmental processes will, in some situations, contradict each other and thus come into conflict. Multiple difficulties with action selection mechanisms are well-known in robotics (Prescott, 2008). When applied to evolutionary robotics these create problems related to, for instance, credit assignment (Whitacre et al., 2006), self-organization and fitness evaluation (Floreano and Urzelai, 2000), and robustness of behavioral and reconfiguration strategies (Andersen et al., 2009).

More generally, action selection is a fundamental problem in artificial systems targeting long-term autonomous and adaptive behavior in complex environments, especially when such a behavior is expected to be evolved (Gomez and Miikkulainen, 1997). Current thinking and experience suggests that several architectures, e.g. subsumption, reactive, insect-based or others (Brooks, 1986), need to be considered as a framework around bio-inspired and evolutionary paradigms for complex behaviorial systems.

This work is an overview paper, which introduces the problem of action selection in evolutionary modular robotics and considers a combination of behavioral, bio-inspired and evolutionary approaches for its solution. Firstly, the field of morphogenetic robotics is outlined in Sec. II, then the high complexity of the regulatory framework is underlined in Sec. III. Sec. IV reviews a number of approaches to action selection, from the literature. Secs. V and VI present several evolutionary and bio-inspired approaches, based on a combination of fixed, self-organized and evolvable controllers and hormone-based regulation. Sec. VII concludes this work.

Morphogenetic Robotics

Artificial developmental systems, in particular developmental (epigenetic) robotics (Lungarella et al., 2003), is a new and emerging field across several research areas – neuroscience; developmental psychology; biological disciplines such as embryogenetics; evolutionary biology or ecology; and engineering sciences such as mechatronics, on-chip-reconfigurable systems or cognitive robotics (Asada et al., 2009). The whole research area is devoted to ontogenetic development of an organism, i.e. from one cell to multi-cellular adult systems (Spencer et al., 2008).

A closely related field is evolutionary robotics (Nolfi and Floreano, 2000), which uses the methodology of evolutionary computation to evolve regulative structures of organisms over time. Evolutionary robotics tries to mimic biological processes of evolution (Elfwing et al., 2008), but also faces challenges of embodiment, the reality gap, adaptation or running on-line and on-board a smart microcontroller device (Baele et al., 2009).

In several aspects developmental and evolutionary methodologies differ from each other:

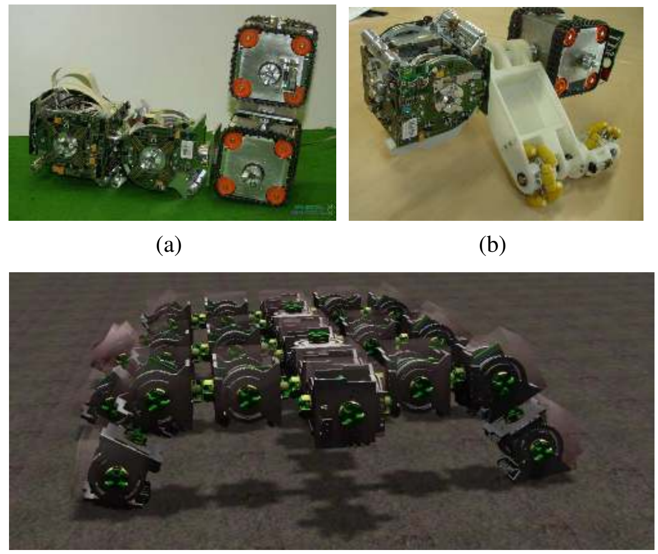
- "*... should try to endow the [developmental] system with an appropriate set of basic mechanisms for the system to develop, learn and behave in a way that appears intelligent to an external observer. As many others before us, we advocate the reliance on the principles of emergent functionality and self-organization...*" (Lungarella et al., 2003);

- "*evolutionary robotics is a new technique for the automatic creation of autonomous robots. Inspired by the Darwinian principle of selective reproduction of the fittest, it views robots as autonomous artificial organisms that develop their own skills in close interaction with the environment and without human intervention*" (Nolfi and Floreano, 2000).

Despite differences, evolutionary and developmental approaches share not only common problems, but also some ways to solve them, it seems that both are merging into one large area of self-developmental systems (Levi and Kernbach, 2010).

Both developmental and evolutionary methodologies impose a set of prerequisites on a system; one of the most important is that it should possess a high degree of *developmental plasticity*. Only then can an organism be developed or evolved. Developmental plasticity requires a specific flexible regulative, homeostatic, functional and structural organization – in this respect evolutionary/developmental systems differ from other branches of robotics. Since collective systems, due to their high flexibility and cellular-like organization, can provide such a versatile and re-configurable organization – collective robotics is a suitable subject for application of evolving and developmental approaches.

The approach used in our work is based on modularity and reconfigurability of the robot platform, as shown in Fig. 1. Individual modules possess different functionality and can dock to each other. Changing how they are connected, an aggregated multi-robot system (organism) possesses many degrees of structural and functional freedom. With a self-assembly capability, robots have control over their own structure and functionality; in this way different "self-*" features, such as self-healing, self-monitoring or self-repairing can emerge. These self-* features are related in many aspects to adaptability and evolve-ability, to emergence of behavior and to controllability of long-term developmental processes. The self-issues are investigated in manufacturing processes (Frei et al., 2008), distributed systems (Berns and Ghosh, 2009), control (Brukman and Dolev, 2008), complex information systems (Babaoglu et al., 2005) and cognitive sensor networks (Boonma and Suzuki, 2008).



(a) (b) (c)

Figure 1: **(a), (b)** Real prototypes of aggregated robots from the SYMBRION/REPLICATOR projects; **(c)** Image of the simulated multi-robot organism.

The platform, shown in Fig. 1 is a complex mechatronic system. Each module includes the main CPU, intended for behavioral tasks that require high-computational power. This CPU is a Blackfin double-core microprocessor with DSP functionality, which can run with up to 550MHz core clock and supports a μCLinux kernel. It possesses an efficient power management system and in its current version the main CPU can utilize 64Mb SDRAM. Peripheral tasks, e.g. sensor-data processing, control of brushless motors, power management and others are executed by several ARM Cortex and low-power MSP microcontrollers. Each module has an energy source with a capacity of about 35Wh. All of them are connected through Ethernet and a power sharing bus. In the next section we briefly discuss a framework of software controllers, developed for this system and introduce the problem of action selection.

Controller Framework, Middleware Architecture and the Need for Action Selection

In robotics, there are several well-known control architectures, for example subsumption/reactive architectures (Brooks, 1986), insect-based schemes (Chiel et al., 1992) or structural, synchronous/asynchronous schemes (Simmons, 1991). An overview of these and other architectures can be found in (Siciliano and Khatib, 2008). Recently, multiple bio-inspired and swarm-optimized control architectures have appeared e.g., (Kernbach et al., 2009b). In designing the general control architecture, we face several key challenges:

- **Multiple processes.** Artificial organisms execute many different processes, such as evolutionary development, homeostasis and self-organizing control, learning, middle- and low-level management of software and hardware structures. Several of these processes require simultaneous access to hardware or should be executed under real-time conditions.
- **Distributed execution.** As mentioned, the hardware provides several low-power and high-power microcontrollers and microprocessors in one robot module. Moreover, all modules communicate via a high-speed bus. Thus, the multiprocessor distributed system of an artificial organism provides essential computational resources, however their synchronization and management are a challenge.
- **Multiple fitness.** Although fitness evaluation using local sensors is mentioned in the literature, here we need to stress the problem of credit assignment related to the identification of a responsible controller, see e.g. (Whitacre et al., 2006)). Since many different controllers are simultaneously running on-board, the problem of credit assignment as well as *interference between controllers* is critical.
- **Hardware protection.** Since several controllers use the trial-and-error principle, the hardware of the robot platform should be protected from possible damage caused during the controllers' evolution.

Corresponding to the hardware architecture, the general controller framework is shown in Fig. 2. This structure follows the design principles, originating from *hybrid deliberative/reactive systems*, see e.g. (Arkin and Mackenzie, 1994). It includes rule-based control schemes, e.g. (Li et al., 2006), as well as multiple adaptive components. The advantage of the hybrid architecture is that it combines evolvability and the high adaptive potential of reactive controllers with deliberative controllers. The latter provide planning and reasoning approaches that are required for the complex activities of an artificial organism.

Meeting the challenges above raises the issue of choosing a suitable underlying middleware with an adequate architecture. As mentioned, a dual-core DSP with a μCLinux will be used as the main CPU. This approach provides much flexibility and facilitates rapid development, for instance in the use of shared standard libraries (e.g., STL, Boost and others). Although the DSP is relatively powerful computationally (given its power consumption), it nevertheless imposes some restrictions that need to be addressed.

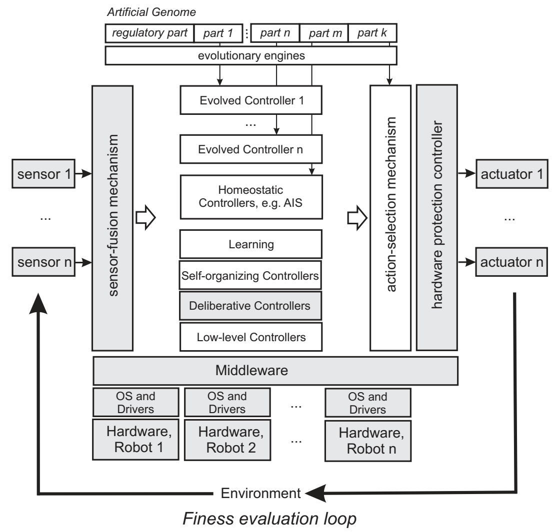


Figure 2: General controller framework. All controllers/processes are distributed in the computational system of an artificial organism, OS – operating system. Structure of controllers utilizes hybrid deliberative/reactive principle.

The most important limitation may be the fact that there is no hardware memory management unit (MMU). Due to the way the μCLinux software MMU works, we decided to design the controller framework as a set of competing applications; an approach that is quite common for UNIX environments (Tanenbaum and van Steen, 2008). For communication within the controller framework a message based middleware system has been implemented. This provides the necessary flexibility needed to implement an event-driven system without having to determine all of the timing constraints in advance (Tanenbaum and van Steen, 2008). Sockets serve as the only mechanism for inter-process communication. Although this may appear to be a disadvantage it yields some very important benefits. First, there is only one standard communications interface defined in advance, with attendant benefits in parallel development across multiple teams. Second, and with regard the robustness of the system; if, for example, a certain controller crashes, the impact of that crash is limited to a single process within the system. All the other applications remain functional and the system may even restart the crashed process later on. The same applies to the middleware itself, as it conforms to the same rules. The approach assures that connections once established are not harmed even if, for example, the addressing module itself is faulty and, therefore cancelled by the operating system (the only limitation here will be the creation of new connections as this is impossible without addressing modules). For connections to other robot modules via

Ethernet the same socket mechanism is used, as for standard Ethernet communications. With this framework we are able to create several controllers which use, for example, evolutionary engines with a structure encoded in an artificial genome. It is assumed that there are also a few task-specific controllers placed hierarchically above other controllers. These task-specific controllers are in charge of the macroscopic control of an artificial organism. They may, for instance, use deliberative architectures with different planning approaches, e.g. see (Weiss, 1999).

Finally, a hardware protection controller closes the fitness evaluation loop for the evolvable part of the controllers (Kernbach et al., 2009a). This controller has a reactive character and monitors activities between the action selection mechanism and actuators as well as exceptional events from the middleware. It prevents actions that might immediately lead to damage to the platform (e.g., by mechanical collisions).

The action selection mechanism is one of the most complex elements of the general controller framework. This mechanism reflects a common problem of intelligent systems, i.e. "what to do next", (Bratman, 1987). This problem is especially challenging in evolutionary robotics for several reasons. Firstly, the fitness evaluation loop will include a combination of different controllers, so it may be difficult to find a unique correlation between a specific evolved controller and its own fitness value. Secondly, several controllers on different levels will be simultaneously evolved, so that some co-evolutionary effects may appear. Among other problems, we should also mention the multiple co-dependencies between fixed, self-organized and evolving controllers.

Action Selection Mechanism

Formally, action selection is defined as follows: "given an agent with a repertoire of available actions ... the task is to decide what action (or action sequence) to perform in order for that agent to best achieve its goals" (Prescott, 2008). Within the context of the projects general controller framework shown in Fig. 2, the role of the action selection mechanism is to determine which controller(s) are driving the actuators at any given time. At one level the action selection mechanism can be thought of as a switch, selecting which of the controllers is connected to the actuators; however a simple switch would fail to provide for, firstly, smooth motor transitions from one controller to another and, secondly, the fact that in this hybrid deliberative/reactive architecture some controllers will need to be prioritized for short time periods (e.g., for obstacle avoidance) whereas others need periods of control over longer time spans (perhaps subsuming low-level reactive elements) to achieve high level goals. In practice, therefore, the action selection mechanism will need to combine some or all of the following elements:

- prioritization of low-level reactive controllers so that they are given control with very low latency;
- vector summation or smoothing between some controller outputs in order to achieve jerk free motor transitions on controller switching, and
- a time multiplexing scheme to ensure that different controllers are granted control with a frequency and for time periods appropriate to achieving their goals.

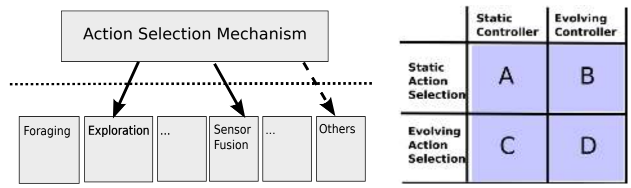
Action selection mechanisms have been the subject of research in both artificial and natural systems for some years, see for instance (Maes, 1990; Hexmoor et al., 1997; Prescott et al., 2007). However, in a recent review Bryson suggests that no widely accepted general-purpose architecture for action selection yet exists (Bryson, 2007). Relevant to the present work is a review of compromise strategies for action selection (Crabbe, 2007). A compromise strategy is one in which instead of selecting a single controller, the action selection mechanism combines several controller outputs in such a way as to achieve a compromise between their (otherwise conflicting) goals; (Crabbe, 2007) suggests that a compromise strategy is more beneficial for high-level than low-level goals.

It is important to note that the action selection mechanism embeds and encodes design rules which will critically influence the overall behavior of the robot. In order to arbitrate between, possibly conflicting, controller goals the action selection mechanism will certainly need to access internal state data for the robot (i.e. from the homeostatic controllers), and may need to access external sensor data. Furthermore, given that those action selection design rules and their parameters may be difficult to determine at design time, we are likely to require an evolutionary approach; hence the connection between the genome structure/evolutionary engine and the action selection mechanism shown in Fig. 2. We may, for instance, evolve the weights which determine the relative priority of controllers as in (González et al., 2006), or co-evolve both controllers and action selection parameters (González, 2007).

Evolution and Action Selection

The action selection mechanism can be seen as a two-tiered architecture of the robot controller (Fig. 3(a)). On the lower tier are activities like elementary actions (e.g. turn right), behavior routines (e.g. random walk) or sub-controllers (e.g. sensor fusion). The upper tier is the action selection mechanism, that controls which activities are running at the moment.

The adaptiveness of the entire robot control can be increased by applying evolutionary approaches to the different tiers of the architecture (Fig. 3(b)): **(A)** Neither the controller nor the action selection module adapts. **(B)** The action selection is static and the activities evolve. **(C)** The action selection mechanism evolves and the activities are



(a) (b)

Figure 3: **(a)** Two-tier architecture with action selection and activities. **(b)** Evolution at the different tiers of the architecture.

static. **(D)** Both action selection mechanism and activities evolve.

One concept for approach **(B)** is a static planning system where a plan to achieve a goal is formulated as a series of activities described by fitness functions. At each step of the plan, the actual controller for the corresponding activity is evolved by online evolutionary algorithms using the fitness function. In this way, the overarching plan does not adapt but the execution of the individual steps evolves. Examples for activities of such a plan can be “Sense Energy Source” or “Robot Aggregation”.

An extreme example for approach **(C)** is a large monolithic evolving neural network as the action selection mechanism. The activities are direct sensor and actuator actions, like reading sensor values and setting motor velocities. An increase in complexity of the activities allows a reduction in the action selection mechanism. For example, instead of direct commands, activities can be small controllers such as collision avoidance or sensor fusion. With very complex activities that control complete behaviors, like foraging, resting or exploring for example, the action selection mechanism can degenerate into a simple priority management system that checks for which “needs” are the most urgent. While a complex neural network can be difficult to evolve efficiently, a priority system can be evolved easily by parameter evolution of the weights or thresholds of different needs and motivations.

Approach **(D)** offers the most flexibility and adaptiveness of the controller architecture. This could possibly be a simple combination of **(B)** and **(C)**. It is conceivable that the action selection adapts to a changing environment by changing priorities of preferences for subordinated activities. In case no matching controller is available for an operation, the action selection can define new fitness functions and evolve new activities to suit the current needs, or evolve existing activities for extended tasks.

In the next section a hormone based controller for approach **(D)** is presented.

Biologically Inspired Mechanism

Artificial Hormone Control

Within the scope of the SYMBRION/REPLICATOR projects, we follow a bio-inspired approach of decentralized coordination of action selection which is distributed across the robot modules: On the one hand, all robot modules, that form the organism, act as autonomous units which have a repertoire of behavioral programs available (actions/controllers). A localized action selection mechanism is needed, which decides within each single unit which action has to be selected. On the other hand, the whole organism has to decide “as a whole”, which action it will perform based on its current status, on its past experience, on its current goals, and on the current set of sensor information. To achieve this difficult task, we developed the Artificial Homeostatic Hormone System (AHHS) which mimics the spread of cellular signals (chemicals, hormones) within multicellular (metazoan) organisms (Schmickl and Crailsheim, 2009; Stradner et al., 2009). This set of controllers, often called “hormone controllers” allow cells (robot modules) to specialize within the robot organisms and to reflect specific physiological states by a simple physiological model that mimics excretion, dilution, diffusion, (chemical) interaction, and degradation of hormones. Within the robotic organism, gradients of hormones emerge over time, reflecting not only the modules’ positions in the organism but also important status information, such as the current energy level. In a hierarchical approach, the globally influenced hormone status within a robot module can help to select an optimal local controller. In turn, the execution of local controllers can significantly alter the hormone system, thus, via diffusion to neighboring modules, alter the behavior of controllers in nearby modules. This way, the AHHS controller allows not only decentralized action selection, but also inter-modular communication between different sub-controllers, hardware abstraction, and sensor integration. See Fig. 4 for a graphical representation of the AHHS design as described above. The concept of AHHS is related to gene regulatory networks (Bongard, 2002). However, here each edge has its own activation threshold and redundant edges with different activations between two hormones are allowed.

Action selection is not only about choosing the right action but also about how selected actions integrate to low-level motor commands in a robot (see Öztürk, 2009). The AHHS allows multiple hormones to affect the actuators of the robots in parallel by integrating various chemical stimulations (see Fig.5 for a schematic of this process).

In the following we present results of a simplified scenario to demonstrate the principles of action selection in a hormone controller. We restrict ourselves to a single robot module and we use the AHHS directly to control the robot without having sub-controllers as described in the general concept above. However, the robot’s body is virtually separated into two compartments (c.f. Fig. 5a) between which

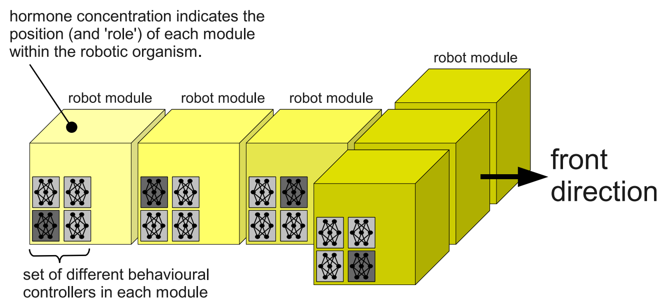


Figure 4: Schematic representation of decentralized action selection that is provided in various 'body parts' of the organism by the AHHS robot control.

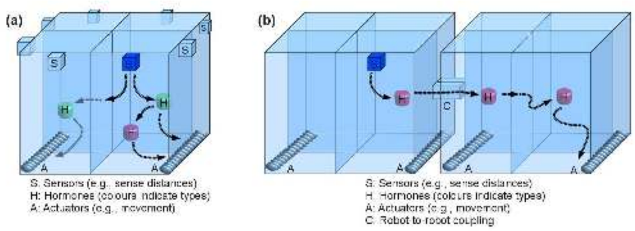
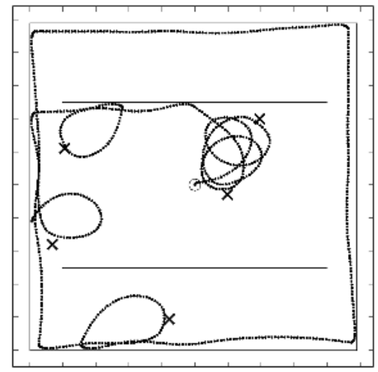
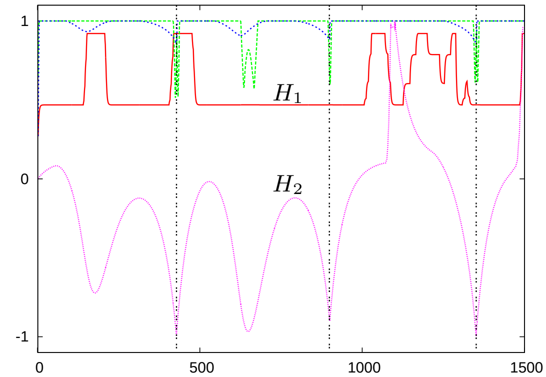


Figure 5: In the AHHS, actuators are influenced by various hormone states in parallel, this way allowing signal integration to produce "mixed" or blended actions.



(a) circle is initial pos., crosses show sequence of emitters



(b) The three vertical lines indicate the time at which emitters were reached; note local minima of H_2 at $t = 175$ and $t = 647$ showing the misses in approaching the emitter.

Figure 6: Robot's trajectory using AHHS controller and the dynamics of five hormones responsible for action selection.

hormones diffuse. Each compartment is associated with one half of the robot. The left compartment contains the left wheel and all proximity sensors of the left half (similar for the right half).

The task of the hormone controller is to control a robot module in a 2-D arena, to catch light emitters, and to explore the arena. Thus, basically two actions are needed to succeed in this task: exploration/wall avoidance and a gradient ascent behavior. The arena consists of surrounding and additional walls in the upper and lower third (see Fig. 6(a)). In addition, it includes one randomly positioned emitter. Both, the walls and the emitter, are perceived by the robot, if they are within range of the sensors (range of light sensors about 50% and range of proximity sensors about 10% of the arena width). The intensity of the sensor signal depends on the distance to the walls and the emitter, respectively. If the robot reaches the emitter (distance < robot diameter) the emitter is erased and reappears at a random position.

The fitness function, that is applied in the artificial evolution, rewards the successful locating of the light emitter, but also – at smaller scale – the exploration of the arena. Thus, the robot has to switch between the action of exploration, if no emitter is detected (i.e., it is too far away to have any significant impact on the sensors), and the gradient ascent, if the emitter is detected. The trajectory of the best individual of the 1000th generation is plotted in Fig. 6(a).

The evolved hormone reaction network of the best evolved controller is complex. We restrict ourselves to a description of the most prominent features. In the hormone network we identify two major hormone interactions that represent the actions: exploration/wall avoidance and gradient ascent. Without any significant input the robot drives in wide right turns forming spirals. If it approaches a wall it avoids collisions because of two controller rules. First, the production of hormone H_1 (see Fig 6(b)) is triggered by the proximity sensor that points 45 degrees to the right (the closer the wall the higher the hormone production). Second, another rule controls the right wheel depending on hormone H_1. With increasing value of this hormone the wheel is accelerated resulting in a turn to the left. Hence, a wall following behavior emerges during which the robot keeps the wall to the right. A question concerning action selection is when to stop the wall following action and continuing the gradient ascent in order to reach the light emitter. This is controlled by hormone H_2. Its value is reduced with increasing input of the left light sensor (bright light results in low H_2). A second rule controls the left wheel which is decelerated mainly for values of $H_2 \in [-0.2, -0.6]$. This slowdown of the left wheel results in a left turn. Hence, the robot interrupts its wall following behavior and turns towards the light (which is always to the left as the robot follows the wall counterclockwise). Hence, we have identified the relevant trigger (hormone H_2) for the action selection mechanism in this hormone network. Obviously, this is a simplified application of AHHS and in future applications we will aim for much more complex tasks of multi-modular robotics.

Adapting Hormone Control

The hormone controllers mentioned in the previous section are subject to evolutionary adaptation. A data structure called "genome" contains rule descriptions and other parameters, which describe some physicochemical properties of the simulated hormones (production rates, decay rates, diffusion rates). In addition, these data describe how one hormone can influence the dynamics of the concentrations of other hormones. The genome is modified by a process of artificial evolution, which allows the embedded action selection to adapt over time to a given body shape or to changes in the environmental conditions. In our evolutionary approach, the fitness of the system reflects multiple levels of adaptation: The whole organism level (e.g., efficiency of shapes and gait patterns) but also on the individual module level (e.g., energetic efficiency of singular modules within the organism).

Conclusion

In this paper we have briefly presented hardware and software frameworks for a reconfigurable multi-robot system. The mechatronic platform provides a high hardware plasticity in terms of structural reconfiguration, changeable locomotion and actuation, and sharing and distribution of power and information. Because of the complexity of regulative, homeostatic and evolutionary mechanisms there are multiple processes that require simultaneous access to actuators. Based on preliminary experiments these processes are expected to display contradictory characteristics. For example, the homeostatic system can require minimization of energy consumption, whereas the evolutionary system may require more energy for performing evaluation runs.

The problem of action selection considered in this paper is highly non-trivial in this context. It is not only related to the classical problem of action selection, well-known in robotics, but also has new aspects related to fitness estimation, credit assignment, evolving of multiple controllers and other issues. The problem of action selection requires a complex deliberative framework and specific controller architectures.

In this paper we have considered a hybrid controller framework, which has reactive and deliberative components. The evolutionary part, which consists of genome, evolutionary engines and evolvable controllers, represents in fact only a small part of the whole framework. It seems that evolving all regulatory structures of real robots from scratch is not feasible because of technology limitations, very specific sensor-actor systems and complexity. Furthermore, it is not fully clear whether this is a general property or is related only to technological artefacts.

Beside the hybrid framework, this paper has proposed evolutionary and bio-inspired solutions to the problem of action selection. The evolutionary approach combines fixed, self-organized and evolvable controllers; moreover the action selection mechanism can also be integrated into the evolutionary loop. The bio-inspired approach is guided by the hormone systems and based on the distribution of hormonal intensity (and between different hormones) in different compartments of a robot, and across robots in a multi-robot organism.

Acknowledgement

The "SYMBRION" project is funded by the European Commission within the work programme "Future and Emergent Technologies Proactive" under the grant agreement no. 216342. The "REPLICATOR" project is funded within the work programme "Cognitive Systems, Interaction, Robotics" under the grant agreement no. 216240. The authors also wish to acknowledge the contribution of all members of both projects.

References

Andersen, T., Newman, R., and Otter, T. (2009). Shape homeostasis in virtual embryos. *Artificial life*, 15(2):161–183.

Arkin, R. and Mackenzie, D. (1994). Planning to behave: A hybrid deliberative/reactive robot control architecture for mobile manipulation. In *Proc. of the Fifth International Symposium on Robotics and Manufacturing*, volume 5, pages 5–12, Maui, Hawaii.

Asada, M., Hosoda, K., Kuniyoshi, Y., Ishiguro, H., Inui, T., Yoshikawa, Y., Ogino, M., and Yoshida, C. (2009). Cognitive developmental robotics: A survey. *IEEE Transactions on Autonomous Mental Development*, 1(1):12–34.

Babaoglu, O., Jelasity, M., Montresor, A., Fetzer, C., Leonardi, S., van Moorsel, A., and van Steen, M. (2005). *Self-star Properties in Complex Information Systems: Conceptual and Practical Foundations (Lecture Notes in Computer Science)*. Springer-Verlag New York, Inc.

Baele, G., Bredeche, N., Haasdijk, E., Maere, S., Michiels, N., Van de Peer, Y., Schmickl, T., Schwarzer, C., and Thenius, R. (2009). Open-ended on-board evolutionary robotics for robot swarms. In Tyrrell, A., editor, *Proc. of the IEEE Congress on Evolutionary Computation (IEEE CEC-09)*, Trondheim, Norway. IEEE Press.

Berns, A. and Ghosh, S. (2009). Dissecting self-* properties. *Self-Adaptive and Self-Organizing Systems, International Conference on*, 0:10–19.

Bongard, J. (2002). Evolving modular genetic regulatory networks. In *Proceedings of the 2002 Congress on Evolutionary Computation (CEC'02)*, pages 17–21.

Boonma, P. and Suzuki, J. (2008). Exploring self-star properties in cognitive sensor networking. In *Proc. of the International Symposium on Performance Evaluation of Computer and Telecommunication Systems (SPECTS-2008)*, pages 36–43.

Bratman, M. E. (1987). *Intention, Plans, and Practical Reason*. Harvard University Press, Cambridge, MA.

Brooks, R. A. (1986). A robust layered control system for a mobile robot. *Journal of Robotics and Automation*, 2(1):14–23.

Brukman, O. and Dolev, S. (2008). Self-* programming: Runtime parallel control search for reflection box. *Self-Adaptive and Self-Organizing Systems, International Conference on*, 0:481–482.

Bryson, J. (2007). Special issue (edited by J.J. Bryson): Mechanisms of action selection. *Adaptive behavior*, 15(1).

Chiel, H., Beer, R., Quinn, R., and Espenschied, K. (1992). Robustness of a distributed neural network controller for locomotion in a hexapod robot. *IEEE Transactions on Robotics and Automation*, 8(3):293–303.

Crabbe, F. L. (2007). Compromise strategies for action selection. *Philosophical Transactions of the Royal Society. B.*, 362:1559–1571.

Elfwing, S., Uchibe, E., Doya, K., and Christensen, H. (2008). Biologically inspired embodied evolution of survival. In Michalewicz, Z. and Reynolds, R. G., editors, *Proc. of the 2008 IEEE Congress on Evolutionary Computation IEEE Congress on Evolutionary Computation*, volume 3, pages 2210–2216, Hong Kong. IEEE Press.

Floreano, D. and Urzelai, J. (2000). Evolutionary robots with on-line self-organization and behavioral fitness. *Neural Networks*, 13(4-5):431–443.

Frei, R., Serugendo, G. D. M., and Barata, J. (2008). Designing self-organization for evolvable assembly systems. In *Proc. of the Second IEEE International Conference on Self-Adaptive and Self-Organizing Systems (SASO-2008)*, pages 97–106.

Gomez, F. and Miikkulainen, R. (1997). Incremental evolution of complex general behavior. *Adaptive Behavior*, 5:317–342.

González, F. (2007). The coevolution of robot behavior and central action selection. In *Nature Inspired Problem-Solving Methods in Knowledge Engineering*, volume 4528 of *Lecture Notes in Computer Science*, pages 439–448. Springer-Verlag.

González, F., Reyes, J., and Ríos-Figueroa, H. (2006). Integration of evolution with a robot action selection model. In Gelbukh, A. and García, C. R., editors, *Proc. of the 5th Mexican International Conference on Artificial Intelligence (MICAI-06)*, volume 4293 of *Lecture Notes in Computer Science*, pages 1160–1170, Apizaco, Mexico. Springer-Verlag.

Hexmoor, H., Horswill, I., and (Eds.), D. K. (1997). Special issue: Software architectures for hardware agents. *Journal of Experimental and Theoretical Artificial Intelligence*, 9(2/3).

Kernbach, S., Meister, E., Scholz, O., Humza, R., Liedke, J., Ricotti, L., Jemai, J., Havlik, J., and Liu, W. (2009a). Evolutionary robotics: The next-generation-platform for on-line and on-board artificial evolution. In Tyrrell, A., editor, *Proc. of the IEEE Congress on Evolutionary Computation (IEEE CEC-2009)*, Trondheim, Norway. IEEE Press.

Kernbach, S., Thenius, R., Kernbach, O., and Schmickl, T. (2009b). Re-embodiment of honeybee aggregation behavior in artificial micro-robotic system. *Adaptive Behavior*, 17(3):237–259.

Levi, P. and Kernbach, S., editors (2010). *Symbiotic Multi-Robot Organisms: Reliability, Adaptability, Evolution*. Springer-Verlag.

Li, G., Lin, K.-C., and Xia, Z. (2006). Rule-based control of collaborative robots. In *Proc. of the 16th International Conference on Artificial Reality and Telexistence*, pages 68–72, Los Alamitos, CA, USA. IEEE Computer Society.

Lungarella, M., Metta, G., Pfeifer, R., and Sandini, G. (2003). Developmental robotics: a survey. *Connect. Sci.*, 15(4):151–190.

Maes, P. (1990). How to do the right thing. *Connect. Sci. J. Spec. Issue Hybrid Systems*, 1:291–323.

Nolfi, S. and Floreano, D. (2000). *Evolutionary Robotics: The Biology, Intelligence, and Technology of Self-Organizing Machines*. The MIT Press, Cambridge, MA. / London.

Öztürk, P. (2009). Levels and types of action selection: the action selection soup. *Adaptive Behavior*, 17(6):537–554.

Prescott, T. (2008). Action selection. *Scholarpedia*, 3(2):2705.

Prescott, T., Bryson, J., and Seth(Eds), A. (2007). Theme issue on models of natural action selection. *Philosophical Transactions of the Royal Society. B.*, 362(1485).

Schmickl, T. and Crailsheim, K. (2009). Modelling a hormone-based robot controller. In *MATHMOD 2009 - 6th Vienna International Conference on Mathematical Modelling*.

Siciliano, B. and Khatib, O., editors (2008). *Springer Handbook of Robotics*. Springer-Verlag.

Simmons, R. (1991). Coordinating planning, perception, and action for mobile robots. *SIGART Bull.*, 2(4):156–159.

Spencer, J., Thomas, M., and McClelland, J. (2008). *Toward a Unified Theory of Development: Connectionism and Dynamic Systems Theory Re-Considered*. Oxford University Press.

Stradner, J., Hamann, H., Schmickl, T., and Crailsheim, K. (2009). Analysis and implementation of an artificial homeostatic hormone system: A first case study in robotic hardware. In *Proc. of the 2009 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS'09)*. IEEE Press.

Tanenbaum, A. and van Steen, M. (2008). *Verteilte Systeme - Prinzipien und Paradigmen*. Pearson Studium, München, Germany, 2nd edition.

Weiss, G. (1999). *Multiagent systems. A modern approach to distributed artificial intelligence*. MIT Press.

Whitacre, J., Pham, T., and Sarker, R. (2006). Credit assignment in adaptive evolutionary algorithms. In *Proc. of the 8th annual conference on Genetic and evolutionary computation (GECCO-06)*, New York, NY, USA. ACM.