

# (Co)Evolution of (De)Centralized Neural Control for a Gravitationally Driven Machine <sup>\*</sup>

Steffen Wischmann, Martin Hülse, and Frank Pasemann

Fraunhofer Institute for Autonomous Intelligent Systems, Schloss Birlinghoven,  
D-53754 Sankt Augustin, Germany  
{[steffen.wischmann](mailto:steffen.wischmann@ais.fraunhofer.de), [martin.huelse](mailto:martin.huelse@ais.fraunhofer.de), [frank.pasemann](mailto:frank.pasemann@ais.fraunhofer.de)}@ais.fraunhofer.de,  
WWW home page: <http://www.ais.fraunhofer.de/INDY>

**Abstract.** Using decentralized control structures for robot control can offer a lot of advantages, such as less complexity, better fault tolerance and more flexibility. In this paper the evolution of recurrent artificial neural networks as centralized and decentralized control architectures will be demonstrated. Both designs will be analyzed concerning their structure-function relations and robustness against lesion experiments. As an application, a gravitationally driven robotic system will be introduced. Its task can be allocated to a cooperative behavior of five subsystems. A co-evolutionary strategy for generating five autonomous agents in parallel will be described.

## 1 Introduction

The *Artificial Life* (AL) approach to *Evolutionary Robotics* (ER) provides promising methods for optimizing a variety of control problems [8, 11]. This includes the optimization of structure and parameters of artificial recurrent neural networks (RNNs), morphology parameters of robots, or even co-evolution of many different populations. Within our approach to AL and ER we are using evolutionary techniques for generating RNNs controlling robot behavior [9]. We are aiming at artificial systems with so called *minimal cognition* [2, 5]. In this context we are trying to deal with minimal models of non-linear dynamical control that can offer a variety of behavioral patterns. To investigate the dynamical properties of such control structures we study relatively simple artificial systems to gain deeper insights into the essence of dynamical systems such as RNNs and robot-environment interactions.

In this paper we will present an example offering investigations of basic cooperation mechanisms among artificial agents coupled through a common body. There are many examples, where cooperation within a group of homogeneous or heterogeneous agents may have advantages over single agents in solving complex tasks [3]. One reason is the possibility of task decomposition and

---

<sup>\*</sup> In: Proceedings of the VIIIth European Conference on Artificial Life (ECAL2005), UK, Capcarrere, M. et al. (Eds.): ECAL 2005, LNAI **3630**, 179 – 188 (2005).

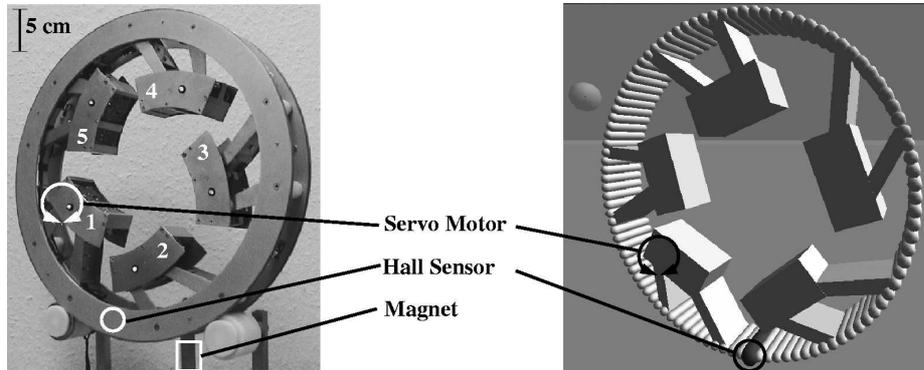


Fig. 1. The artbot *micro.eve*. Left: Hardware, Right: Simulation

task allocation. To give an example for task allocation, we use the artbot *micro.eve* (<http://www.sphericalrobots.com>). Among other intentions, Julius Popp designed *micro.eve* to provide a benchmark system for control architectures where a *simple body-consciousness emerges*. Considering *embodiment* as a fundamental aspect of creating artificial autonomous agents [4, 10], the system can be described as a set of five agents, which are able to act locally and independently. Hence the agents are connected to a common body, their local actions affect not only themselves but also the behavior of the common body and consequently also the behavior of the other agents. They have to cooperate to solve the task, which is a well known problem in collective robotics [7, 12]. For instance, in [1] an example of physically linked robots solving a common task is given. Here, the system seems much simpler. But due to this simplicity it allows detailed analysis of the underlying control structure. We will show two different kinds of control. First, we evolve one central RNN which controls all the agents. Second, we evolve a RNN for each separate agent in co-existing populations, whereas each agent can sense the action of the other agents only by sensors providing information about the common bodies' behavior. There is no explicit communication among the agents. We will analyze two resulting architectures with respect to performance, structure-function relations, and robustness.

## 2 Methods

Figure 1 illustrates the artbot *micro.eve* as well as its simulated model. The robot consists of five movable arms, which are connected to a common body. The center of mass of these arms can be actively translated by a servo motor. Through a coordinated motion of the five arms the overall center of mass of the robot can be translated in such a way, that the ring starts to rotate on two supporting rollers. Here, we defined the task of the system as rotating as fast and as harmonically as possible.

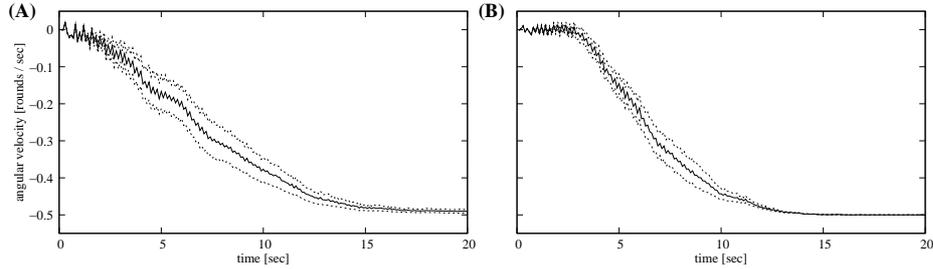
The control structures have to produce the motor signal for each arm. The sensory system consists of potentiometers in each motor, a gyroscope located inside the ring, and five hall sensors equally distributed over the ring. Each hall sensor is located between two arms, respectively. These hall sensors are binary switches emitting a peak if they pass a magnet placed at the bottom of the ring. We merged all five hall sensor to one sensory input relative to the arm index, i.e. for each arm the next hall sensor to the right has the index 1. The arm and hall sensor indices are incremented counterclockwise. The mapping  $(i, s_i)$ , where  $i$  denotes the index of the last activated hall sensor and  $s_i$  the according sensor value, is as follows:  $(0, 0.0)$ ;  $(1, 1.0)$ ;  $(2, 0.66)$ ;  $(3, 0.33)$ ;  $(4, -1.0)$ ;  $(5, -0.33)$ . Hence the output of the input neuron, which provides the hall sensor information, should be zero if no hall sensor is activated at all ( $i = 0$ ) the use of this discrete mapping was chosen instead of a monotonic function. Accordingly, within the decentralized control approach each agent has its own sight of the hall sensor. To reduce the amount of input neurons for the centralized control structures the sight of the first arm is provided as sensory input. The sensor values of the potentiometers are linearly mapped to the interval  $[-1.0, 1.0]$ . The gyroscope values are transformed to an angular velocity of the ring with a maximum at 0.5 rounds/secs. For the input of the neural network these values are also mapped to the interval  $[-1.0, 1.0]$ , where negative values indicate a counterclockwise and positive values a clockwise rotation.

For our control approach we used discrete-time artificial recurrent neural networks with standard additive neurons with sigmoidal transfer function  $\sigma = \tanh$ . For generating these controllers we used an evolutionary algorithm that allows variation of the network's structure and its parameters at the same time. A full description of the algorithm and some other applications can be found elsewhere [6]. Here, to solve the robot's task we defined the following fitness function:

$$F = |\bar{\omega}| * \left(1 - \frac{\sum_{t=0}^n |\omega(t) - \bar{\omega}|}{2n}\right) \quad (1)$$

where  $\omega$  is the angular velocity of the ring represented by the output of the mapped gyroscope sensor value. The left term  $(|\bar{\omega}|)$  rewards a high mean velocity and the right term rewards a harmonic rotation. Due to the use of  $\tanh$  as transfer function the output of the velocity sensor neuron is in a range between  $-1$  and  $+1$ . Accordingly the range of the fitness value is between  $0.0$  and  $+1.0$ . Note that we are using an implicit fitness function, i.e. no global knowledge is used. Parameters of the fitness function are solely determined by sensor information that are accessible by the agent.

To avoid dominance of specialists we evaluated every individual on 20 different randomly (uniformly distributed) generated starting conditions. For this purpose the ring angle is varied in the full range of  $2\pi$ , and the angle of each arm is varied within its complete working range. The resulting fitness value is the mean fitness of these 20 evaluation cycles. One evaluation cycle lasted 1200 evaluation steps (corresponds to 120 secs real time).

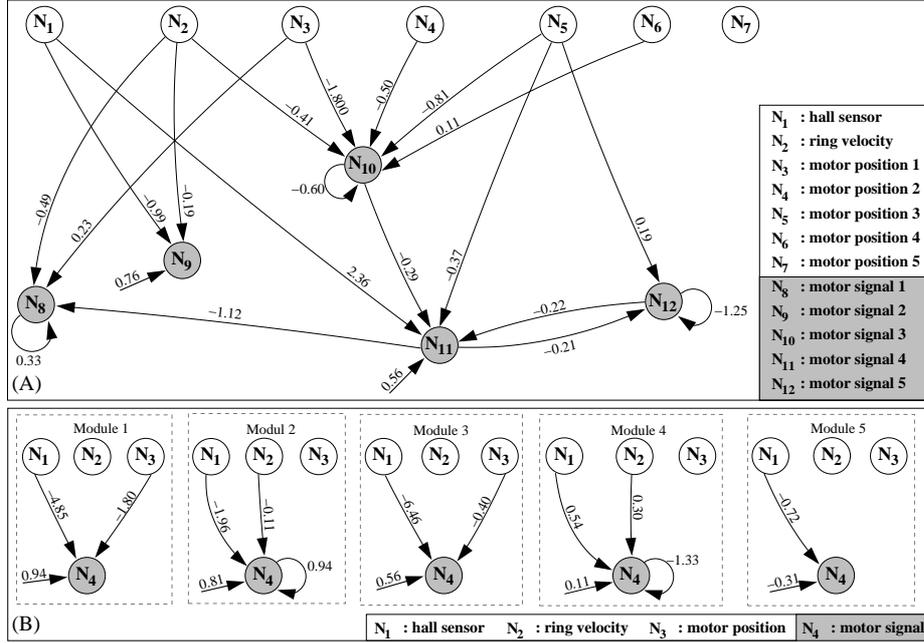


**Fig. 2.** Evolution of the angular ring velocity with time and their variance for the decentralized (A) and centralized control structures (B), starting from several initial conditions.

For generating the decentralized control architectures we applied a simple co-evolutionary strategy. Each arm is controlled by one autonomous RNN. The sensory input of the RNNs is reduced to one relative hall sensor input, as previously described, the ring velocity, and the arm's potentiometer. A single controller has one output neuron controlling the according motor signal. Every agent is evolved in a separate population. The evolutionary process for a single population is the same as for the evolution of the centralized control structure, i.e. every population has its own selection, reproduction, and variation operators. In every population the individuals are sorted according to their fitness values, starting with the highest. The offsprings get no fitness value after selection, they are appended at the end of the sorted list. For evaluation, one agent of each population is selected and applied to the arm related to its population, i.e. a group of five agents, each from a different population, is evaluated together at the same time. The selection of the group members is rank based related to the fitness value, i.e. the agents taking the first place in each population are evaluated together, than the agents on the second place and so on. In such a way the evaluation of  $i$  populations needs  $j$  evaluation cycles, where  $j$  is the number of individuals within the largest population. The fitness function (equation (1)) regards the performance of a group of agents, which have to cooperate. Therefore every agent within one group gets the same fitness value, regardless which local acting network gives the most or even the least contribution to it.

### 3 Results and Discussion

Figure 3 shows the outcome of the evolutionary process for the centralized and decentralized control approach. Both control techniques are successful in solving the given task. The mean fitness  $\overline{F}$  and its variance  $\sigma^2$  for 100 evaluation cycles with random starting conditions is 0.900 ( $\sigma^2 = 0.001$ ) for the centralized and 0.884 ( $\sigma^2 = 0.003$ ) for the decentralized control architecture. After transferring these RNNs unchanged to the real machine, the observed behavior was qualitatively the same, although the evolution of these RNNs was completely done in

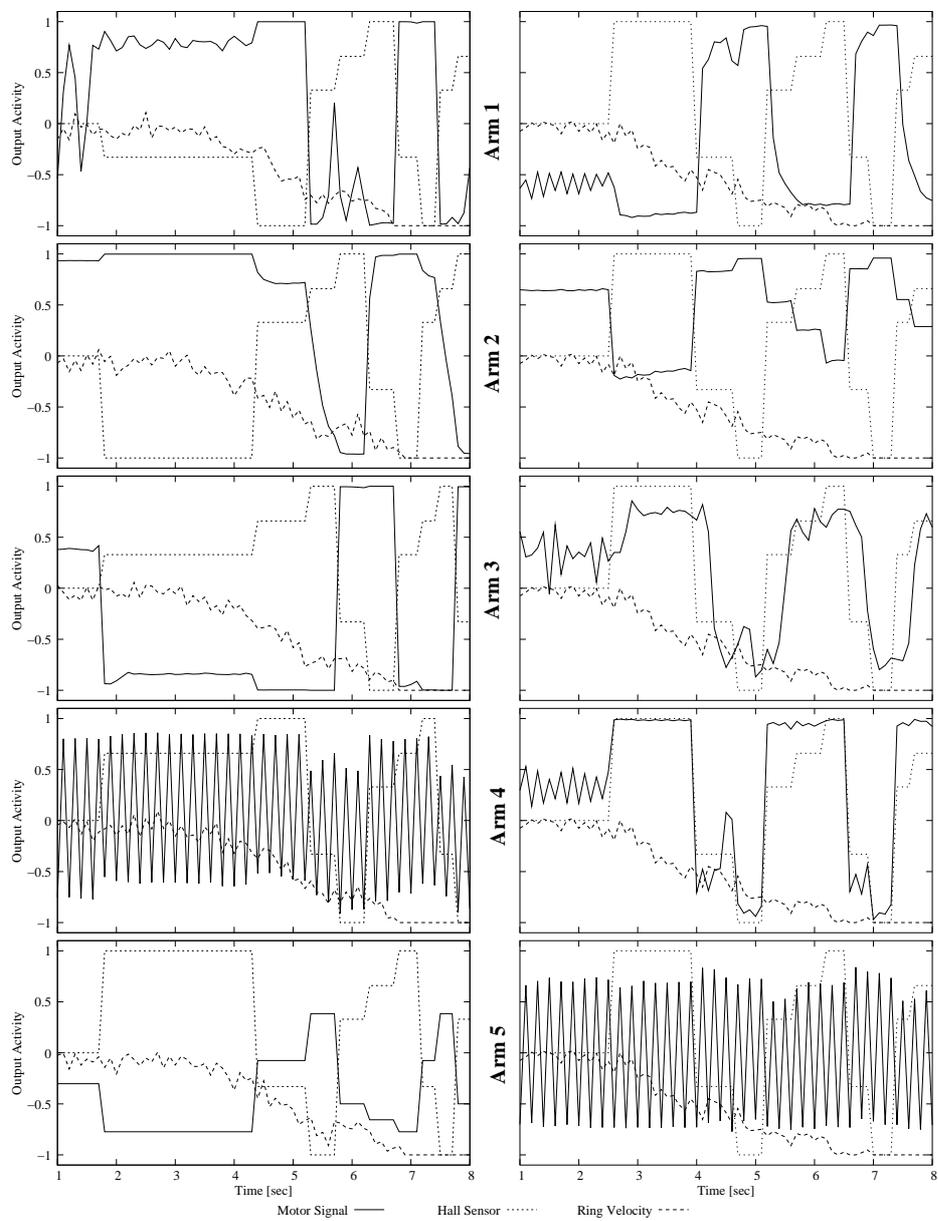


**Fig. 3.** RNNs of the centralized (A) and decentralized (B) control architectures. Input neurons are indicated by white circles and output neurons by grey circles.

simulation with a very simplified motor model, and only roughly approximated friction and noise. We observe that once the ring started to rotate from a given starting position, it harmonically rotates for the whole evaluation time, i.e. a mean fitness of 0.9 indicates that this agent can handle about 90 percent of different starting conditions. Note, that it is nearly impossible to reach a fitness value of 1.0 due to the time the agent needs to initiate a rotation. During this initialization the rotation cannot be harmonic. Therefore the first 10 seconds of an evaluation cycle did not contribute to the fitness value.

Figure 2 illustrates the evolution of the angular ring velocity with time. The average of 100 runs with random, uniformly distributed, starting conditions (ring angle, arm angles) and the variance are given for both control architectures. The centralized RNN as well as the decentralized RNN performs only counterclockwise rotation and approaches the maximum angular velocity ( $|\omega_{max}| = 0.5 \text{ rounds/sec}$ ) in about 10-15 seconds. The decentralized RNN seems to have a smoother transition to  $\omega_{max}$  whereas the centralized RNN needs about 2.5 seconds for initializing a rotation at all but than reaches  $\omega_{max}$  slightly faster.

In the following we will discuss the dynamics of the RNNs and their effect on the robot behavior. Focusing on the motor neurons, we can determine two main mechanisms controlling two behavioral states: (1) Oscillations to initialize the rotation, and (2) Strong impact of the hall sensor to maintain the rotation.



**Fig. 4.** Motor signals, hall sensor and ring velocity input of the decentralized (left) and centralized (right) control structures. The according neurons can be found in figure 3.

Looking at figure 3 and 4, we find one neuron with a period-2 oscillation in both control structures. For the centralized control this is  $N_{12}$ , and for the decentralized control  $N_4$  of module 4. The period-2 oscillation is due to a over-critical negative self-connection and persists for all the time (see figure 4). The strong impact of the hall sensor within the other modules of the decentralized control can be seen in figure 3. For the centralized RNN  $N_9$  and  $N_{11}$  are mainly controlled by the hall sensor input,  $N_8$  receives its strongest input from  $N_{11}$ , and  $N_{10}$  gets a strong input from  $N_3$  which is directly influenced by the output of  $N_8$ . In figure 4 we can see the correlation between the hall sensor input and the signal of all these motor neurons.

What does this mean for the behavior of the robot? At the beginning the hall sensor is inactive, hence the signal of the according input neuron is zero. Referring to figure 4, most of the motor neuron signals stay around an output value according to their bias terms or the input of other neurons. Therefore the overall center of mass of the ring is translated once and thus, the ring may rotate a little. Depending on the starting condition this rotation may be enough to activate a hall sensor, if this is near the magnet (see figure 1). If the hall sensor stays inactive, the rotation will stop because most of the motor signals depends on this input and the according arms will not move at all. In this case the period-2 oscillating motor neurons will move the ring very slowly until the hall sensor is activated. Then the rotation starts and is maintained as it can be seen from figure 4. For most of the time we observe two pairs of output neurons producing opposed signals. This means that two arms translating their masses to the center of the ring, and at the same time two other arms translate their masses to the ring periphery. This action maintains the ring rotation. In the decentralized control modules the two pairs get negative feedback from their hall sensory input, in contrast to the centralized control, where we find positive and negative feedback leading to the opposed movements. In the decentralized control, the opposed movement is based on the fact that each module has its own, relative, sight of the hall sensor as described in section 2, whereas the centralized control has only one sight of the hall sensor.

What is the advantage that one neuron is oscillating all the time? We can find a good reason, if we perturb the system, i.e. manually stop the ring rotation. Hence, most of the motor neurons are mainly influenced by the hall sensory input, as we saw before, most of the arms will stay at their positions when the ring stops because the hall sensor stays active at its last value. If we than release the ring, a slow movement can be observed, due to the still oscillating motor neuron, until the hall sensor changes its value, and that gives rise to a change in the other motor signals until the ring rotates harmonically again.

There is another interesting fact in the presented control structures. We saw that there are not as many interesting dynamical features as one could expect by using RNNs, which is possibly due to the simplicity of the task. On the one hand we have a period-2 oscillator and on the other hand the behavior of the others neurons seems to depend more or less directly on one sensor input. But for instance, if we look at the motor neuron of module 1 in the decentralized

**Table 1.** Mean performance and its variance of the lesion experiments.

| Lesion  | Centralized Control |            | Decentralized Control |            |
|---------|---------------------|------------|-----------------------|------------|
|         | $\bar{F}$           | $\sigma^2$ | $\bar{F}$             | $\sigma^2$ |
| none    | 0.900               | 0.001      | 0.884                 | 0.003      |
| Motor 1 | 0.005               | 0.000      | 0.740                 | 0.121      |
| Motor 2 | 0.702               | 0.120      | 0.776                 | 0.018      |
| Motor 3 | 0.739               | 0.060      | 0.671                 | 0.085      |
| Motor 4 | 0.006               | 0.000      | 0.572                 | 0.185      |
| Motor 5 | 0.579               | 0.140      | 0.753                 | 0.082      |

control architecture, we see, while the hall sensor is still inactive, the occurrence of an oscillation (see figure 4) which is not provided by the structure of the RNN. This oscillation is based on the input neuron which provides information about the position of the arm, which is controlled by the motor neuron. At the beginning the arm moves to a certain position according to the bias term, but than the motor neuron gets a strong negative feedback from the position sensor leading to an opposed movement and so on. Here we have a negative *feedback loop through the environment* that is depressed when the hall sensor becomes active due to the much stronger connection from this input neuron.

It is interesting that we can detect similar control principles in two different designs of control. But then one can ask, what is the advantage of one method over the other. First, we can see decentralization leads to less complexity in the structure of the single RNNs (figure 3). And second, due to the autonomy of the single control modules, there is no explicit communication between the decentralized control units, as we discussed it for the centralized RNN, which may lead to more robustness. To demonstrate the second point lesion experiments were done with the two introduced results of the different control designs. Both architectures were tested on 100 random, uniformly distributed, starting conditions. Each run lasted 1200 evaluation steps, i.e. 2 minutes in real time. Here, lesion means the fixation of one motor neurons output to zero value during the whole evaluation time. This could be considered as the simulation of a motor breakdown of the real hardware. Table 1 gives the mean performance, calculated with the fitness function (equation 1), and its variance for all lesion experiments performed on both control architectures.

Considering the centralized control structure, lesions of motor 1 and 4 have the strongest impact. The agent can handle no starting condition. If we look at the structure (figure 3 (A)) of the RNN, we can see that setting the output of  $N_8$  to zero leads to zero output of  $N_3$  as well, which consequently has a strong impact on  $N_{10}$ . Setting the output of  $N_{11}$  to zero will also have a strong influence on  $N_8$  due to the strong connection between these two neurons. We can see, the fixation of one output neuron also affects the dynamics of other output neurons. On the contrary, the decentralized control structures have no inter-connections between the output neurons, which is reflected in the results of the lesion experiments. In the worst case it still can handle about 57 % of the

starting conditions. The worst case is the lesion at the oscillating neuron (module 4). Interestingly, lesion of the oscillating output neuron  $N_{12}$  of the centralized control structure leads to a similar performance. That the system is still able to handle about half of the different starting conditions in this case is due to the initialization process described earlier in this section. For lesion of motor 2 and 3 at the centralized control structure, where the according motor neurons have no or only a small influence on other neurons, we observe a performance comparable to the decentralized one.

It is important to note that the described perturbations were not part of the boundary conditions during the evolutionary process. If these perturbations were included, we would expect a more robust behavior of the centralized control as well. But one should be aware of that it could be highly difficult to pre-estimate every possible kind of perturbations in real systems. In the presented example the robustness against motor breakdowns is an innate property of the decentralized control approach.

## 4 Conclusions

In this paper we demonstrated the evolution of a centralized and decentralized control architecture for the gravitationally driven artbot *micro.eve*. Both were able to successfully solve the given task. We could show that minimal structures arise out of the evolutionary process without any prior assumptions about the structure of the RNNs. By analyzing the structure-function relations, we identified similar control principles in both designs, mainly based on two mechanisms, periodic oscillations, mainly responsible for a robust initialization and robustness against perturbations, and a strong coupling of the hall sensory input, mainly responsible for maintaining the ring movement. We found that this strong coupling is also determined by inter-connections between the output neurons in the centralized RNN. Because for the decentralized control no communication was allowed, this mechanism was determined by direct connections to this sensor input. Due to this fact we could show that the autonomy of the subsystems of the decentralized control approach leads to more robustness against motor breakdowns. We saw that the action of one agent indirectly influences the action of the other agents. We think, it should be possible to handcraft a centralized control structure containing similar autonomous subsystems. But we argue that it is hard to expect such a result from an evolutionary process if no specific assumptions about the structure and parameters of RNNs are made. By identifying the main properties of the control structure one could manually transfer the properties of decentralization, such as autonomous substructures, to centralized control architectures. Another step would be to evolve homogeneous decentralized control structures, where it should be more obvious that their properties are applicable to a centralized control exhibiting more robustness to lesion experiments.

We saw that decentralization gives rise to less complexity, concerning the structure of the RNNs, and more fault tolerance in the presented example. We are not claiming that these principles could be generalized for all kinds of de-

centralized control problems. But as it is known, these issues are crucial points for many of other examples concerning decentralized control as well [13].

Even though the presented results do not directly lead to general solutions of problems in AL and ER, they provide a simple model of *minimal cognition* for an unconventional machine. Due to the few system parameters it provides a platform for studying first steps in neural control of autonomous robots, basic cooperation mechanisms, and robot-environment interactions. Furthermore, we showed how to apply the same evolutionary algorithm to the development of centralized and decentralized control architectures. We introduced a simple co-evolutionary strategy, for which the evaluation time does not increase with the number of populations. We are aware of the fact that evolution of competitive behavior may require a more complex evaluation strategy, but to solve simple cooperative tasks the presented strategy is sufficiently good.

## References

1. G. Baldassarre, D. Parisi, and S. Nolfi. Coordination and behavior integration in cooperating simulated robots. *Proc. of the 8th International Conference on the Simulation of Adaptive Behavior (SAB-2004)*, pages 385–394, 2004.
2. R. D. Beer. The dynamics of active categorical perception in an evolved model agent. *Adaptive Behavior*, 11(4):209–243, 2003.
3. Y. U. Cao, A. S. Fukunaga, and A. B. Kahng. Cooperative mobile robotics: Antecedents and directions. *Autonomous Robots*, 4:7–27, 1997.
4. A. Clark. *Being There: Putting Brain, Body, and World Together Again*. MIT Press, Cambridge, MA, USA, 1996.
5. I. Harvey, E. A. Di Paolo, R. Wood, M. Quinn, and E. Tuci. Evolutionary robotics: a new scientific tool for studying cognition. *Artificial Life*, 11:79–98, 2005.
6. M. Hülse, S. Wischmann, and F. Pasemann. Structure and function of evolved neuro-controllers for autonomous robots. *Connection Science*, 16(4):249–266, 2004.
7. A. J. Ijspeert, A. Martinoli, A. Billard, and L. M. Gambardella. Collaboration through the exploitation of local interactions in autonomous collective robotics: The stick pulling experiment. *Autonomous Robots*, 11(2):149–171, 2001.
8. S. Nolfi and D. Floreano. *Evolutionary Robotics: The Biology, Intelligence, and Technology of Self-Organizing Machines*. MIT Press, Cambridge, USA, 2000.
9. F. Pasemann, U. Steinmetz, M. Hülse, and B. Lara. Robot control and the evolution of modular neurodynamics. *Theory in Biosciences*, 120:311–326, 2001.
10. R. Pfeifer. On the role of embodiment in the emergence of cognition: Grey walter’s turtles and beyond. *Proc. of the workshop The legacy of Grey Walter, Bristol, 2002.*, 2002.
11. R. Pfeifer and C. Scheier. *Understanding Intelligence*. MIT Press, Cambridge, USA, 1999.
12. M. Quinn, L. Smith, G. Mayley, and P. Husbands. Evolving controllers for a homogeneous system of physical robots: Structured cooperation with minimal sensors. *Philosophical Transactions of the Royal Society of London, Series A: Mathematical, Physical and Engineering Sciences*, 361:2321–2344, 2003.
13. M. Resnick. *Turtles, termites, and traffic jams: explorations in massively parallel microworlds*. MIT Press, Cambridge, USA, 1994.