Algorithmic requirements for swarm intelligence in differently coupled collective systems

Jürgen Stradner a, Ronald Thenius a, Payam Zahadat a, Heiko Hamann a,b, Karl Crailsheim a, Thomas Schmickl a,⇑

a Artificial Life Laboratory at the Department of Zoology, Karl-Franzens University Graz, Universitätsplatz 2, A-8010 Graz, Austria
b Department of Computer Science, University of Paderborn, Zukunftsmelde 1, 33102 Paderborn, Germany

ARTICLE INFO

Article history:
Available online 7 March 2013

ABSTRACT

Swarm systems are based on intermediate connectivity between individuals and dynamic neighborhoods. In natural swarms self-organizing principles bring their agents to that favorable level of connectivity. They serve as interesting sources of inspiration for control algorithms in swarm robotics on the one hand, and in modular robotics on the other hand. In this paper we demonstrate and compare a set of bio-inspired algorithms that are used to control the collective behavior of swarms and modular systems: BEECLUST, AHHS (hormone controllers), FGRN (fractal genetic regulatory networks), and VE (virtual embryo-genesis). We demonstrate how such bio-inspired control paradigms bring their host systems to a level of intermediate connectivity, what delivers sufficient robustness to these systems for collective decentralized control. In parallel, these algorithms allow sufficient volatility of shared information within these systems to help preventing local optima and deadlock situations, this way keeping those systems flexible and adaptive in dynamic non-deterministic environments.

1. Introduction

In collective adaptive systems, the degree of coupling between group members do strongly vary in nature and the artificial world. This range starts with systems showing a low degree of coupling among individuals, which is the case, for example, whenever individuals in a group move randomly and independent from other group members [36]. Examples of such systems are gases or robots programmed to do only collision avoidance. Given that in natural systems communication and interaction is usually local, there is a high volatility in the amount of information that can be exchanged by explicit communication and other interactions. A higher degree of coupling can be observed in systems in which coordinated movements like flocks are performed [47]. In such systems partners stay in local neighborhood for a longer period of time thus communication allows these local partners to adapt their behavior (and maybe also their inner state) to each other. Thus, such coordinated motion patterns can be seen as a median degree of coupling. On the other extreme, there are also very rigidly coupled systems where members of the group stay together for a long time and communication/interaction is restricted to local neighbors [70]. Examples for such systems are crystals, solid bodies or cells of an organism, which are strongly physically coupled. This distinction made above, also holds for collectives of artificial agents, virtual or embodied in robots, which can be arranged in different levels of coupling. Their movement can be random, coordinated as in a flock, or, in the case of modular/cellular robotics, the motion of the robotic sub-units can be tightly connected.

In this article we divide collective systems in three separated levels of coupling (Fig. 1). These levels are described...
in the following and the pros and cons of the degree of coupling are depicted.

1.1. Dynamic random coupling

If movement is purely randomized there is a problem with volatility of information that gets lost within this class of systems and random motion introduces unfavorable amounts of noise into the collective information processing. In consequence, we claim that a swarm of agents that exhibits mainly random motion has to perform a behavior (a swarm algorithm) which compensates for this low degree of coupling. One well-known example to compensate for such easy losses of information is 'stigmergy' [22] where information is dropped in the environment and stays stable for an amount of time even though many different agents pass through this limited area frequently. Stigmergy is often found in (eu-) social animals like ants [4], honeybees [14,13], termites [22], wasps [35] and cockroaches [1]. Besides these natural examples, the algorithmic concept of stigmergy was used in mathematical optimization [19] and also in coordination of robotic swarms [5]. However, also in stigmergic systems, information has to pass away over time. In robotics, such an approach has been used in several variants. Virtual pheromones projected from the top of the robotic arena [62,21], 'virtual pheromones' exchanged by direct robot-to-robot communication [44,45], 'virtual pheromone trail' left by agents in the environment, for example by excitation of glow paint [10,39] and 'virtual nectar' exchanged by robots in the trophallaxis-algorithm [49], which was inspired by the frequent exchange of liquid food in honeybee workers [16,17]. In this article we discuss aggregation-based robotic mechanisms like the BEECLUST algorithm [56,36] as one exemplary case how random motion has to 'crystallize' under specific local circumstances to allow collective computation and/or collective decision making.

1.2. Dynamic correlated coupling

Systems we categorize into this level have a desired ratio of randomness to constancy that allows for a high degree of adaptivity and flexibility. Neighboring relations are dynamic but not uncorrelated as in random motion. Acquired knowledge about neighbors are valid for long time intervals compared to the frequency of necessary actions. In addition, such behaviors are self-sustaining. For example, in collective motion the knowledge about the distance to the next neighbor has value because this neighbor will stay a neighbor for a relevant period of time and additionally this information helps in coordinating motion which even increases the persistence of neighborhood relations.

1.3. Static coupling

This class refers to all physically coupled systems that consist of modules which are 'docked' to each other. Such systems are not confronted with the problem of random motion inducing noise because the modules stay together for a long time. The problem here is the lack of flexibility which comes to light in multiple issues. A simple scenario in which this problem arises is a shape, which is not suited for the given task. Thus, a physically coupled system might be too rigid and possibly lacks flexibility. Examples for systems with static couplings are natural organisms [41], cellular robotics [6] or modular robotics [40,58,26]. The latter example deals with a bio-inspired approach for modular robotic systems which pushes to get rid of the strong rigidity of classical robotics. Also the algorithms executed on autonomous cells/modules have to overcome the rigidity that is imposed by the physical coupling and have to introduce some long-distance interaction between modules which are not directly connected neighbors. Several approaches have been performed in this domain, ranging from central pattern generators (CPGs) [32,31] to hormone-inspired control paradigms [58,52,24]. In this article, we discuss several of these static coupling approaches, how they allow collective computation and globally dynamic patterns of growth and of collective motion.

The different levels of coupling (dynamic random coupling, dynamic correlated coupling, and static coupling) influence issues like energy and information inside a collective (Fig. 1). If there is a strong coupling which comes
with stable spatial relationship a certainty of information transfer is given. Also the number of direct neighbors is stable and hence a constant in the environment of an individual. On the other side of the spectrum of coupling strength in a collective system there is the problem of too much flexibility. As denoted by arrows in Fig. 1, there is no strict separation between the three levels of coupling but rather smooth transitions.

In the following sections we will present approaches and algorithms including their application experiments to swarm systems which reduce the drawbacks of being in dynamic random coupling mode or in static coupling mode. The paper is structured in reference to Fig. 1. The next section deals with an algorithm called BEECLUST which brings the system from the left-hand side to the middle of Fig. 1. Section 3 introduces two algorithms (Sections 3.1 (AHHS) and 3.2 (FGRN)) which perform in systems with static couplings with the aim of pushing the system to more flexibility.

Additionally, in Section 4 we will exploit a novel approach which acts on all coupling-levels simultaneously, Virtual Embryogenesis (VE), and investigate the evolvability of the different coupling-levels. The VE approach mimics biological processes of growth based on concepts reaching from early works on decentralized, self-organized groups of agents [71] to models of growth processes of embryos and their interactions with evolutionary processes (EvoDevo) [43,42]. Also the combined body-controller evolution described in [59,57,34] is within focus of the VE approach.

We will conclude the paper with a summary of the algorithms described and a critical discussion on what is still missing to completely understand the impact of the degree of coupling in swarms.

## 2. Algorithms for systems with dynamic random coupling

Systems that are based on random motion, such as loosely coupled members in swarms, are characterized by the high degree of information volatility as mentioned above. Therefore, the accessibility and reliability of information has to be increased in order to generate any form of coordinated behavior in such a system. Two types of information occur: information that is associated with spatial features (e.g., landmarks, light conditions, temperature) and information that is defined by relations between agents (e.g., team mates, couples of sender and receiver, neighbors). When relations between agents change fast and persistently, it is better to rely on spatial features. As discussed above, that is the swarm’s option of applying the stigmergy approach. A second option is to overcome the high dynamics in agent-agent relations. That is achieved by making sure that neighboring agents are likely to stay within their mutual neighborhoods. The analogy to that behavior from physics would be the cooling process of a gas or fluid which is the reduction of particle speeds according to kinetic theory and consequently an extension of neighboring relations in time. Here we discuss the example of BEECLUST, which is an algorithm for cue-based swarm aggregation and achieves coordinated motion by implementing the second option (slowing down the neighborhood changes). However, it applies also the first option (leveraging spatial features) to a certain extent.

### 2.1. BEECLUST

The BEECLUST algorithm (see Fig. 2) is based on observations of young honeybees [64], was analyzed in many models [30,51,55,25,23,2], and was even implemented in a swarm of robots [56]. This algorithm allows a swarm to aggregate based on cues (e.g., temperature, light, sound) although individual agents do not perform a greedy gradient ascent. In addition, a BEECLUST-controlled swarm is able to break symmetries [25].

Say the agents are initially randomly distributed in space with initially randomly distributed headings. Then this swarm would be fully uncoordinated and an assumed graph representing the agents by its nodes and the agents’ neighboring relations by edges would show high dynamics in its adjacency matrix. However, once two agents approach each other close enough to allow for mutual perception then they will stop. Staying stopped means they conserve their neighborhood relation. The mutual perception can be viewed as a cue-based process under the assumption that perceiving another agent is the perception of a cue. An additional cue, the spatial feature of this considered spot (e.g., temperature), is leveraged because the waiting time is proportional to the measurement of this local feature. Other agents might run into these two robots by chance forming a cluster of aggregated agents. Once the cluster has reached a certain size (defined by the number of stopped agents) there is a break-even between the expected number of incoming agents and the expected number of outgoing agents. Then the cluster is permanently expanding assuming a constant density of agents in the surroundings. The change in neighborhoods within such a cluster is obviously little because agents that are fully surrounded by other agents cannot escape. Hence, a BEECLUST-controlled swarm self-organizes a cooling process in its neighborhood relations which allows for high

1.) Each agent moves straight until it perceives an obstacle $O$ within sensor range.

2.) If $O$ is a wall the agent turns away and continues with step 1.

3.) If $O$ is another agent the agent measures the local value.

The higher this value the longer the agent stays still.

After this waiting period, the agent turns away from the other agent and continues with step 1.

Fig. 2. The BEECLUST algorithm.
accessibility and reliability of neighbor-based information. This information is, in turn, used as a cue in the aggregation process itself.

The graphs in Fig. 3 document the decreasing volatility of the neighborhood relations over time. The edges shown in Fig. 3 were obtained by triangulation and were introduced to visualize the cooling process. These edges are virtual while the algorithm actually relies on neighborhoods defined by the agents’ sensor range which is here less than 2% of the arena side length. Moving agents have fast changing neighborhoods and clustered agents have static neighborhoods except for agents passing at less than the sensor range. As the number of clustered agents increases the degree of connectivity increases and the volatility of neighborhood relations decreases. Hence, the BEECLUST-controlled swarm starts in a state that is to be placed in the left third of Fig. 1 and moves toward the right hand side.

3. Algorithms for systems with static couplings

The strongest degree of coupling in a swarm is reached by physically connected group members in a modular system (cf. Fig. 1, right). Problems that arise concerning information and energy transfer between the modules, here, are that it is too robust and tight. E.g., no inner localization can be performed because there are no differences between the single modules inside. Important features for the behavior of the natural or artificial organism like symmetry breaking cannot be establish. Another prominent problem is being not flexible concerning configurations depending on the task. The field of modular reconfigurable robotics tries to keep the balance between the flexibility of the shape and reliability of information transfer.

3.1. Artificial homeostatic hormone system (AHHS)

In this section we describe an algorithm based on reaction–diffusion systems which is able to build a gradient with which, e.g., the self-localization inside the organism can be established. It is able to perform shape–independently in a reconfigurable robot system. Thus, we do not claim that this algorithm is able to bring a system being on the right of Fig. 1 toward the left hand side by itself but it is able to cope with modular reconfigurable systems which get a rigid system more flexible on their own.

The original idea of the algorithm presented here was introduced by Schmickl and Crailsheim [50] and in more detail by Schmickl et al. [52]. It is called artificial homeostatic hormone system (AHHS). For an application of AHHS as a controller performing on robot hardware see [60,54]. Schmickl et al. [53] report perspectives of the AHHS approach. An improved version of the AHHS is reported in [24,28], which is applied in this work. An application of this approach to modular robotics is reported in [27].

An AHHS is the implementation of a dynamical system consisting of several state variables (artificial hormones) and a system of ordinary differential equations (ODE) that govern their dynamics. The key feature of AHHS is how the parameters of these ODE are encoded and determined. Concentrations of the artificial hormones are allowed to increase independently but are subject to a certain decay as well, which decreases the hormone concentration independently. Rules manipulate hormone concentrations and influence the hormone dynamics based on sensory input and based on the concentrations of other hormones. The actuators are controlled by rules that translate hormone concentrations to actuator control values.

Here, AHHSs are used as control devices on the modules of a modular robotic organism which implements a statical coupled system. The hormones diffuse between connected modules. This influence of the hormone concentrations in neighboring modules makes the spatial topology of the modular robotic system relevant for the functionality of the AHHSs in the organism. The topology could be a virtual compartment structure within the module in allusion to compartmentalizations in biological cells [65] or an aggregate of several modules, as it the case here.

We implemented homogeneous parametrized AHHSs on each module of a virtual modular robotic organism. The organism consists of seven modules. In the experiments described here only one compartment per module is implemented. Diffusion implies an implicit communication between the modules. The functionality of the control-
ler in each module is altered by the diffusion process which is in turn influenced by the topology of the organism. Here, we used shapes of T, X, H and I as four different topologies.

As inputs, we used four different functions in the experiment. These functions defined the sensor input of the first module that we call head module. We distinguished no input, squared-pulse, linear-increase and sinus input. These input functions are derived from experiments with autonomous, mobile robot systems where the input is a distance sensor pointing to the front of the organisms. In this case, having no input means the situation when the organism is going straight without being confronted with an obstacle. An input of a squared-pulse function is given when the organism moves sideways and thereby it is moving past an obstacle. The linear-increase input arises when the organism moves towards an obstacle. The sinus input happens with a caterpillar-like movement when the distance of the head facing the ground varies over time.

For the experiments AHHS is initialized 30 times with random parametrization. Two hormones and 30 rules are used. The hormone concentration is altered by random initialized decay and base production rates and by the sensor input. This sensor input is only presented at the first module, but the information can be transported by diffusion through all seven modules as explained above.

To analyze the performance of the AHHS when applied in different robotic body shapes we conducted 480 runs. All four input functions were applied to each of the four shapes with 30 random parametrized AHHSs. Fig. 4 shows the evolution of the dynamics of values and concentrations for one shape in each column. The upper row presents the sensor value (S0) which was applied to the head module (mod0). The two middle rows present the concentrations of hormone zero (H0) in head module (mod0) and in the tail module (mod6). The bottom row are the actuator values taken from the last module. The course of the dynamics are shown for 3000 steps in these four exemplary runs. Each with one of the shapes, schematically presented above the dynamics, and one of the input functions (for all 480 runs see Supplemental material A).

Fig. 4. Dynamics of the sensor value (first row), hormone concentrations of the hormone 0 in module 0 and module 6 (second and third row, respectively) and the actuator value (last row) over 3000 time steps. Four runs with randomly initialized AHHSs are shown exemplary with the used shaped presented schematically above the dynamic plots (for all 480 runs see Supplemental material A). Modules 0 and 6 are marked red in the schematic shape to stress the positions of sensor input (module 0) and actuator’s readout position (module 6).
also to a chaotic behavior of the actuator value (cf., Fig. 4, first column, last graph).

An X shaped organism being confronted with a squared-pulsed input signal is shown in Fig. 4, second column. In this AHHS a two-point equilibrium is established by the hormone concentration and actuator value. The value of the equilibria is altered due to the squared pulses. The information of the sensor in the head module is transported to the actuator value in tail module.

The example chosen with an H shaped organism is with a linear input function (Fig. 4, third column). In the head module the input is transformed to a maximum function. At the other end of the organism – in the tail module – the information of the sensor value is reduced to a step function of the H0 concentration. This function is mapped on the actuator value.

Similar transformation of information is taking place when observing the example with a snake-like (I) topology of the robot organism and a sinus input function (Fig. 4, last column). The dynamic of the sinusoidal input is still expressed by the hormone concentration in the head module, but lost until it reaches the tail module.

Even with the AHHS being randomly parameterized first favorable information processing tasks can be observed in the results of these experiments (i.e., mapping a maximum function to a step function). With the possibility of using optimization techniques for tweaking the parameters it becomes clear that AHHS is a valuable tool for performing on a statically coupled robotic organism with reconfiguration abilities. After the presentation and investigation of the first approach for modular systems in static coupling (right third in Fig. 1) an alternative approach for this coupling level is presented in the following.

3.2. Fractal gene regulatory network (FGRN)

In this section another algorithm for controlling a statically coupled robotic system is presented. This is the implementation of fractal gene regulatory network (FGRN) which is represented as a motion controller of an I shaped (snake-like) modular robot in a physics-based simulator. In contrast to AHHS, which was used in the previous section, diffusion is not implemented in this system and there is no explicit communication between the modules. However, the system has to deal with the fact of no initial difference between the modules and also needs a mechanism of distance-interaction between the modules. In order to meet these requirements, local sensors are used on every module. By using the local sensors, an implicit communication is achieved through environmental feedback.

FGRN which is introduced in Bentley [9] is a variant of computational gene regulatory networks (GRNs) (such as [48,3,20,11]). FGRN is inspired by biological cells [38]. It has been investigated for robustness and efficiency in Bentley [8] and successfully implemented as a single controller unit for different tasks such as controlling conventional robots [7], producing patterns [9], motion planning [76], and pole-balancing [37]. It has also been implemented as distributed controllers of modular robots [75,74,77].

An FGRN system can be implemented as several units that run independent from each other but all contain an identical encoding. The units are capable of interacting with their local environment. Other units can also be considered as a part of the environment and the behavior of an FGRN unit can be implicitly influenced by the behavior of its neighbors.

An FGRN unit consists of encoding substances – called genes – and interaction substances – called ‘fractal proteins’. Genes belong to different types and make proteins with various functionalities. A schematic representation of the interactions inside an FGRN unit is summarized in Fig. 6 (see [9] for a detailed description).

A fractal protein consists of two parts: shape and concentration level. Shapes are encoded in the genes by three real values \( (x, y, z) \) which represent the center and size of a window on a fractal set (see Fig. 5). Concentration levels represent the current amount of proteins as real values. At every time-step, shapes of the existing fractal proteins define a system of ordinary differential equations (ODE) that indicate the current dynamics of the system. The concentration levels perform as the state-variables. Any change in a protein concentration level that leads to a decrease to zero or raise from zero means that the protein is deleted or added to the system in the current time-step which in turn leads to a different ODE system for the next time-step.

In order to use an FGRN system as controllers of a modular robot, every module contains an FGRN unit. All the units contain an identical copy of the genes but run independently from each other. In every step during runtime of each module, an interaction cycle is executed. Input values provided by local sensors of the modules are normalized and specify the concentration levels of input proteins, the output values are generated by the units and are scaled into appropriate range to be used as control signals of the actuators of the modules.

In the current experiment, FGRNs are generated as controllers of a modular robot with snake morphology consisting of three homogeneous modules. The task is to locomote the snake as fast as possible. Every module uses an actuator which is a central hinge and contains an FGRN unit as its local controller. There is no explicit communication between the modules and consequently between the FGRN units. But every module contains proximity sensors in their front and rear faces. Since the modules are tightly coupled in a modular robot, behavior of a module can influence the inputs received by the proximity sensors of the neighboring modules and the evolved behavior of the whole system relies on this coupling (Fig. 7 shows a sketch and a screenshot of the moving robot).

The experiment is performed in Symbricator3D [73]. Symbricator3D is a simulator designed for the projects [63,46] and uses the design of the prototype described in [29]. It is based on the game engine Delta-3D and uses the Open Dynamics Engine for the simulation of dynamics.

A variant of a Genetic Algorithms (see [9] for details) is used to optimize a population of 30 randomly generated FGRNs as distributed controller for the robot.

In order to have a look in the internal dynamics, an arbitrary controller is chosen and the dynamics of the concen-
tration levels of proteins inside the three modules and their generated output values are presented in Fig. 8. The internal dynamics of each module is based on the two input proteins, and the regulatory protein. Note that all the modules contain identical FGRN genome and any variation in the dynamics and behavior of the FGRN units initiates from different inputs received from the sensors.

As represented in Fig. 8, in the front and middle modules regulatory protein level stays at zero permanently, therefore the output is solely produced based on the input values. In the rear module, the regulatory protein is produced frequently whenever the input from the rear sensor gets a large value which is the case in the beginning of the process. When the regulatory protein gets a positive value, the output stays static with a small positive value and it is not influenced by any of the input values. When the regulatory protein vanishes, the output only reacts to the inputs. The input value of 'covered' proximity sensors (rear sensor of the front module, both sensors of the middle module and front sensor of the rear module) is small due to the placing of the sensors. By moving a hinge of a module, the distances which are sensed by the sensors of the neighboring modules change by a small value. Even though these changes are small they still may have an influence on the generated output. This way the behavior of each module may influence the behavior of the others.

The two approaches (AHHS and FGRN) presented in this section, provide means of localization and distance-communication in order to help the modules to identify their role in relation to the whole organism and compensate for the lack of flexibility of possible interactions between modules in a statically coupled configuration.

Next we consider how to control the transitions from dynamic random coupling to dynamic correlated coupling and to static coupling.

4. Algorithms for systems to control of transitions between coupling levels

In some applications both extremes mentioned in Section 1 exist in parallel: mobile agents move around randomly, and under given conditions, aggregate to a static linked system. One example of such a switch between the extremes is the self-organized building process of a statically coupled multi-robotic systems out of individual mobile robotic units with limited sensory abilities. Solitaire robotic units with a limited sensor-range drive around in the environment (being in the state of dynamic
random coupling), searching for a to-be-built robotic organism. As soon as they find one, they try to find the right place to dock on the robotic organism (indicated by a docking signal on the surface of the robotic organism). While doing this the robotic unit has to avoid all kinds of obstacles, be it other robotic units attracted by the robotic organism, be it already finished parts of the robotic organism itself. If the given goal is to increase the building efficiency (e.g., regarding speed), it would be advantageous to organize the mobile units to move in groups in the environment, i.e., with a dynamic correlated coupling. This is to decrease the occurrence of jamming effects and the number of collisions. Thus, it leads to a higher supply of robotic modules to the areas where the building process of the multi-robotic organism (representing the level of static coupling) takes place.

The method presented here automatically leads to a balance between the two extremes of coupling levels. It is the bio-inspired approach of Virtual Embryogenesis (VE) \cite{70,67,18,66}. The VE process, in detail explained in \cite{70,68}, is a model of processes observable during the evolution of the developmental process controlling the growth of biological embryos. In nature the function of a cell in an embryo is determined by its position within the growing body. A cell draws the information about its position from self-organized processes, which include the interaction of diffusing substances, called morphogens (for more details see \cite{15,41,33}) with the genome of the cells of the embryo. Morphogens are emitted and detected by the cells of the embryo. The reaction of a cell (e.g., growth, specialization, duplication, emission of other morphogens) to a given concentration is determined by the genome of the cell. VE

**Fig. 6.** Interactions in an FGRN unit in every cycle.

**Fig. 7.** Screenshot of the robot in move and a sketch of the robot and its sensors. Sensors are represented as black dots in the sketch.
models these processes and allows the evolution of self-organized building processes, which are able to react dynamically to noise in the environment or even to damage taken by the growing virtual embryo during the growth process [69]. Please note, that the VE-system develops a reactive building process resulting in a morphological structure, and not a morphological structure alone.

4.1. Virtual Embryogenesis (VE)

To investigate, how a system can switch from one extreme coupling state (mentioned in Section 1) into another, we developed a simulated experimental setup to observe the evolution of a behavioral program. This program allows individual autonomous robots to behave in a dynamical random and dynamical correlated coupling level of a collective, and in parallel build a multi-robotic organism with static coupling consisting of these autonomous robots.

All experiments shown in this paper are based on simulation experiments using the simulation environment Netlogo 4.1.2 [72]. The models presented in [70,69] were adapted to the problem of organizing assembly processes in modular robotics [46,63]. The cells of the embryo are represented by virtual individual robotic modules (see Fig. 9), which are able to move in a simulated environment.

![Fig. 8.](image1.png)

Fig. 8. Inputs and regulatory concentrations, and output value for the three modules in the first 100 consecutive time-steps. Red and magenta lines represent input protein concentrations related to rear and front sensors respectively. Blue lines represent regulatory concentrations, and black lines represent generated output values. Notice, the range of the y-axis for the middle module is altered.

![Fig. 9.](image2.png)

Fig. 9. Scheme of a robot implemented into the VE simulation environment. The sensory inputs are interpreted as morphogen-levels by the genome, what results in the production of proteins, that influence the actuators (e.g., the motors if the genome steers a single robotic unit, the ‘docking signals’ if the genome controls the building process of the organism). In contrast to the basic VE system (described in [70] only the VE-genome is used to steer the individual robot. The other components (e.g., the diffusion process of morphogens with the organism) are of course used during the building process of the robotic organism (not shown in this figure for depicting reasons).
We adapted the VE system in a way, that the former cells of the virtual embryo are now models of docked robots inside a statically coupled robotic organism (comparable to the experiments described [69]): To increase the size of the robotic organism the docked robots have to turn on a 'docking signal' (see Fig. 9 and Fig. 13(a)). This signal can be seen by autonomous robots (described in detail below), which then have the possibility to react to this docking signal, approach the robot emitting the signal and dock to it. This way the growth process of biological organisms (as described in [70]) is emulated in this system. Due to the fact, that robots can only produce these 'docking signals' at the outer borders of the robotic organism, the growth of the organism takes place only at these boarders. A 'pushing around' of robots (as described in [70]) within the robotic organism is not possible, due to the fact, that during the building process of the robotic organism the robots are already physically coupled to each other.

As mentioned above we implemented agents, which are able to move around in the environment in a Braithen-berg-vehicle manner [12]. These agents are called 'movers' and represent individual robotic units with dynamic coupling. If a mover crashes into any obstacle, be it another mover or the growing robotic organism (without the intend to dock there), it is removed from the experiment. The movers are controlled by the same genetic code as the robots that build the robotic organism. This way the genome has to solve two problems: On the one hand, the genome has to deliver a behavioral program that allows an individual robot to operate efficiently together with other robots in a given environment. The tasks that are to fulfill are: ‘do not crash into obstacles’, and ‘find the right place at the growing robotic organism and dock there’. On the other hand, the same genome has to solve the problem of building a robotic organism within the given constraints of the fitness function, as described in [67]. The movers have two distance sensors for obstacles of any kind, placed in the front of the robots (±45° relative heading, see Fig. 9), with an opening angle of 45°, and one single central sensor, heading to the front, able to detect a docking light (opening angle 10°, see Fig. 9). The sensor values are fed as morphogen levels into the VE system, the actuators (left and right motor) are controlled by special protein levels (comparable to the proteins described in [70], Table 1). The motor is set to 'driving' by default, the produced proteins were able to modulate the speed of the wheels to control the driving direction and speed of the single robot. The authors are aware of the fact, that only some parts of the VE system (genome, proteins) are adapted for the control of individual robots, and other parts, which are essential for an EvoDevo system (e.g., the diffusion of morphogens within an organism), are removed for the purpose of controlling a moving robot. Please note that these essential parts are of course used during the organization of the growth process.

As soon as a mover has docked to the robotic organism, it becomes part of the robotic organisms and looses its ability to move around in the environment. The robotic organism itself is not able to move.

To investigate how good the adapted VE system is able to evolve genomes for body formation and individual robot control in parallel, we performed two experiments:

- One experiment with a fitness-function that required a fast growing process. This way we wanted to put pressure on the genome to develop well reacting movers. As a target-pattern, we choose a square (see Fig. 11(a)). Please note, that the target pattern is bigger than the number of robots available, to allow all kinds of shapes to develop within this experiment, as long as they develop fast.

- One experiment with a fitness-function that required a precise growing process. With these experiments we wanted to investigate, how the adapted VE system handles the problem of the evolution of a complex building procedure and the evolution of a controller for the movers. As target shape we used a 'H', as depicted in Fig. 11(b).

<table>
<thead>
<tr>
<th>Number of</th>
<th>‘H’-shape</th>
<th>Square-shape</th>
</tr>
</thead>
<tbody>
<tr>
<td>Experiments</td>
<td>6</td>
<td>6</td>
</tr>
<tr>
<td>Reactive to environment</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>Uphill walkers</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>Swarms</td>
<td>1</td>
<td>2</td>
</tr>
</tbody>
</table>

Fig. 10. 3D-Screenshot of the adapted VE simulation environment. All simulated robots have the same genome, determining both: the behavior of the moving robotic modules and of the modules within the robotic organism. This way the genome controls the behavior as well as the growth of the robotic swarm, as well as the building process of the robotic organism. Blue boxes indicate robotic organism modules (immobile), green areas indicate 'docking signals', brown boxes indicate 'movers', brown lines indicate trajectories of 'movers' (only shown for depicting reasons). Please note that movers can not detect the shown trajectories.
All experiments were performed for 100 generations, each with 50 individual genomes per generation. Each individual genome was tested for 80 time-steps. All simulated robots (‘movers’ and modules in the robotic organism) within one experiment were run by the same genome. Each evolutionary run was repeated 6 times. Default-speed of individual robots was set to 0.5 patches/time-step. World size was set to $31 \times 31$ patches. All other settings (except the changes of the system mentioned above) are identical to [69]. Please note, that all experiments were finished after 100 generations, even if no optimal solution was found. With the given settings, optimized for the scientific questions raised above, the VE system did not develop even near optimal results. The experiments are designed to investigate the ability of the VE system to control individual agents, not to develop perfect body shapes. The optimization of the presented process for given applications is topic of ongoing investigations.

The evaluation of different behaviors was done manually with a post-evaluation process, in which the best evolved genomes of each generation were tested in a scenario, which consisted only of one-module robotic organisms and 2 movers. The reaction of the movers regarding objects in the arena and docking signals in the sensor range was classified in the four categories as described below and it was recorded.

The analysis of the genomes evolved in the described experiments showed that four types of behavioral classes of the movers had evolved, differing in the ability of the movers to react to the environment:

- ‘No reaction’ (see Fig. 12(a)). The movers moved straight through the arena, without any reaction to any kind of sensory input.
- ‘Uphill-walkers’ (see Fig. 12(b)). The movers reacted to docking-lights in the sensor-range by approaching and docking to the robotic organisms.
- ‘Reactive robots’ (see Fig. 12(c)). The movers reacted to obstacles in the environment by changing its direction. Docking lights were ignored.
- ‘Swarm robots’ (see Fig. 12(d)). The movers reacted not only to docking-lights in the sensor-range, but also to other robots in sensor range by following them. The result was that groups of robots were cruising through the environment (see Fig. 13(b)).

These four behavioral classes led to different behavioral patterns on group level which can be assigned to the coupling levels: Dynamic random coupling is observed in those groups, in which individual robots did not interact with each other, or reacted to each other in an uncoordinated manner showed an evenly distribution of group members within the environment. Those groups, that consisted of ‘movers’ which followed each other, showed a simple flocking behavior (depicted in Fig. 13) and thus, belong to the level of dynamic correlated coupling.

To investigate, if the adapted VE process is able to react to different conditions during the evolutionary process, two different fitness functions were used (described above). It showed that during evolution the number of developed reactive patterns increased with the necessity of a faster growth process. The frequencies of the observed “following behaviors” are depicted in Table 1.

In this section we have shown that a system that masters the transition from dynamic random coupling to dynamic correlated coupling and also to static coupling can be synthesized by artificial evolution.

5. Conclusion and discussion

In this article we reported bio-inspired swarm algorithms that deal with the problem of high volatility in information due to high dynamics in neighborhood relations and algorithms that focus on introducing more flexibility to statically coupled systems, that is, driving swarm systems away from the two extremes of coupling towards the moderate solution of dynamic correlated couplings. With the bio-inspiration of our approaches we aim for general principles although we often refer to robotic agents in this work. We propose that the moderate level of coupling is to be favored because the two extremes (left and right hand side of Fig. 1) come with several drawbacks as discussed above.

With the BEECLUST algorithm we investigated an algorithm which implements a transition from dynamic random coupling to dynamic correlated coupling which allows for stable information transfers within limited time periods. In this paper BEECLUST was investigated as a mere clustering process in a homogeneous environment. In other settings, for example as reported in [25], BEECLUST is run in an inhomogeneous environment and interpreted as a collective decision process. The above principle of generating dynamic correlated couplings is a prerequisite for collective divisions. Only if there is a reliable neighborhood a cluster is able to converge on a locally defined consensus. However, at the same time these neighborhoods should not be too static and have at least some exchange on a global level with other regions and clusters to stay adaptive to a dynamic environment.
The presented results of AHHS were focused on investigations of several topologies of statically coupled systems. In addition, AHHS is designed to cope with scaling topologies and even dynamic transitions between topologies [26]. Due to the fact that AHHS is also used in single agents with a virtual inner structure [61,54] its functionality can be related to that of the described VE-system. Even though the transient from random moving agents to a statically coupled collective system is possible with AHHS, it was originally not intended to operate that way. It was designed to perform in a modular, but physically coupled system. The reported results, here, show that both internal communication and internal states can be implemented by this approach. Hence, the algorithm is capable to perform in reconfigurable system, is robust to dynamical changes, and implements a considerable flexibility.

In addition, the FGRN was also used to control a physics-based simulated coupled system. In the implemented system, there was no explicit communication between the units but all the units used their local sensors. Since the dynamics of the system is restricted due to the physical coupling of the units, the information provided by each sensor reflects to some extent the status of the other units and is considered an implicit communication between the units. This implicit communication via environmental feedback increases flexibility and it can be extendable to more complex configurations.

The VE-system allows an evolutionary approach to the problem of balancing the two extreme sides of level of coupling in mobile robotic units while building a multi-robotic organism. Punishing the system for collisions by removing mobile robotic units (which are the substrate for the building process of the robotic organism) yields an effective evolution of VE-systems. These VE-systems generate behavioral programs (coded into the genome of the VE), which are able to organize the swarm phase and also a

![Fig. 12. Screenshots from the VE-Simulation environment showing different behaviors of 'movers' evolved within the VE-system. All shown screen-shots were produced during the post-evaluation experiments, in which only one robotic organism and two 'movers' were used. For explanation of shown objects see Fig. 10.](image-url)
reactive building process for a robotic organism. It showed that the evolving genomes react to different levels of evolutionary pressure by development of different levels of reactive controllers for robot motion. Whether the parallel evolution of two behaviors within one genome is limited compared to a modular approach, is subject to ongoing research. For the future we plan, besides optimizing the VE, to adapt the described process to real-world robotic hardware (first results regarding this topic are presented in [18]). We plan to investigate in detail, how the ‘self-healing’ abilities of the VE (described in [69]) are influenced by the ability to coordinate growth-processes on the organism level and swarm phase in parallel. We will use the presented model also to investigate the interactions of genes relevant for ‘mover’ control with those relevant for body formation.

Subject of ongoing research is the integration of the FGRN-system, the AHHS approach, and the VE process in a modular software approach with hand-overs between controllers at run-time to produce a robust, self-organizing software system with maximal flexibility operating on all levels of coupling. Finally, the research based on the levels-of-coupling perspective will be pursued in both design of swarm algorithms for artificial systems and as a guiding principle for models of natural swarms.

Acknowledgements

This work is supported by: Austrian Science Fund (FWF): P19478-B16 and P23943-N13 (REBODIMENT); EU-IST-FET project ‘SYMBRION’, No. 216342; EU-ICT project ‘REPLICATOR’, No. 216240; EU-IST-FET project ‘SYMBRION, No. 216342; Austrian Science Fund (FWF): P15961-B06 and P19478-B16; EU-ICT ‘CoCoRo’, No. 270382; EU-ICT ‘ASSISIq’, No. 601074.

Appendix A. Supplementary data

Supplementary data associated with this article can be found, in the online version, at doi:10.1016/j.chaos.2013.01.011.

References

Camazine S, Deneubourg J-L, Franks NR, Sneyd J, Theraulaz G, Karsai I, Penzes Z. Optimality of cell arrangements and rules of
Jin Y, Schramm L, Sendhoff B. A gene regulatory model for the
Hereford JM. Analysis of BEECLUST swarm algorithm. In: Proc. of the