From real robot swarm to evolutionary multi-robot organism

S. Kornienko, O. Kornienko, A. Nagarathinam, P. Levi

***Abstract*— Collective working allows microrobots to achieve more functionality, better performance and higher reliability on the macroscopic level. In this paper we demonstrate the on-going work in developing novel collective systems, where swarm robots work not only collectively, but are also capable of autonomous aggregation and disaggregation into a higher multi-robot organism. The main issues of such an organism, as well as its genome-based control, are discussed. We show the developed docking approach and investigate topological transformations in a prototype of self-assembling robots.**

I. Introduction

A natural example of collectively working insects and animals are very impressive from two viewpoints [1]. Firstly, despite limitation in energy, communication and perception of swarm agents, the whole swarm is able to emerge quite diverse and attain functionally-rich collective behavior. These examples cover collective activities from food foraging and labor division till nest building and collective defence [2]. Secondly, natural swarm agents develop efficient coordination techniques, allowing cooperative and competitive working in large and super-large societies [3]. Lately, technological and primarily robotic community, investigates and mimics these collective techniques in artificial collective systems by trying to improve functionality, reliability and intelligence of robotic and microrobotic systems [4].

When swarm robots work collectively, they improve their collective fitness, i.e. robots profit from collective working. This is related to e.g. collective perception and actuation, where the objects are essentially larger and heavier than each robot; such as collective energy foraging, where robots find food much quicker than doing it alone [5]; collective exploration, cleaning and many other activities, where each robot is too weak, or has too limited resources. Performing real experiments in a large swarm (100+) of microrobots Jasmine [6], we encounter a limit of collective fitness that can be achieved in a swarm. For example, when a food source is separated from a swarm by a wall, robots can never achieve this source whatever collective technique they emerge. All robots will energetically die.

Considering natural examples of collective work, we can encounter systems, which can achieve principally new collective work. For example, some fungi can aggregate into a multi-cellular symbiotic organism and perform in this way such activities that cannot be fulfilled alone or in a swarm-like way [7]. The idea is to apply this approach to swarm (or more generally to collective) robotic systems and to improve their collective performance by making robots capable for autonomous self-assembling (aggregation and disaggregation) in/from a multi-robot organism. Such an organism can be evolved from the simplest forms into more complex ones, exploring its own environment and even assimilating new environments. The swarm-based approaches, underlying self-assembling processes, differ this new trend in collective systems from known robotic research as e.g. mechanical self-assembling [8] or reconfigurable robotics [9].

In this paper we investigate a few issues of a transition between robot swarm and a single multi-robotic organism. We show the main restrictions imposed on robots capable for collective activities in swarm-like and organism-like ways. Based on the existing microrobotic platform, we developed an IR-based docking approach, providing appropriate accuracy of docking. Finally, we demonstrate the first prototype of robots and investigate topological issues of assembling into an organism.

The rest of the paper is organized in the following way. In Sec. II we describe limitations imposed on robots. Sec. III demonstrates a current development of robots and Sec. IV - first topological experiments with these robots. The Sec. V is devoted to a brief overview of the intended genome-based control of organism. Finally, in Sec. VI we conclude this work.

II. New paradigm in collective robotic systems

Collective intelligence is often associated with macroscopic capabilities of coordination among robots, collective decision making, labor division and tasks allocation in the group [10]. The main idea behind this is that robots are achieving better performance when working collectively.

The background of collective intelligence is related to the capability of swarm agents to interact jointly in one medium. Currently, there are only two different cases of such interactions. In the first case agents communicate through a digital channel, capable for semantic messages exchange. Due to information exchange, agents build different types of common knowledge [11]. This common knowledge in fact underlie collective intelligence.

The second case appears when macroscopic capabilities are defined by environmental feedback. The system builds a closed macroscopic feedback-loop, which works in a collective way as a distributed control mechanisms. In this case there is no need of complex communication, agents can interact even only kinetically.

However in nature we encounter the third case of interactions. For example, some bacteria and fungi (e.g. dictyostelium discoideum) can aggregate into a multi-cellular organism when this provides better surviving chances [7].

Authors are with the Institute of Parallel and Distributed Systems, University of Stuttgart, Universitätsstr. 38, D-70569 Stuttgart, phone: +49-711-7816373; fax: +49-711-7816250; email:Sergey.Kornienko@ipvs.uni-stuttgart.de

1-4244-1340-0/07$25.00 ©2007 IEEE

In this way, they interact not only through information exchange, they build the closest physical, chemical and mechanical interconnections. Therefore a natural step in the collective robotic systems is to investigate an aggregation from single robots into a single multi-robot organism, i.e. from robot swarm into a multi-robot organism.

Despite the similarities of both scenarios (robot swarm and multi-robot organism), such as a large number of robots, focus on collective/emergent behavior, a transition between them is a quite difficult step due to mechanical, electrical and, primarily, conceptional issues. They are related to hardware, software, behavior and organism-specific issues. In this section we consider them in relation to swarm robotics. In the table I we demonstrate a short overview of essential differences between single swarm robots and a multi-robot organism.

TABLE I

Main differences between robot swarm and multi-robot organism.

	Robot swarm	**Multi-robot organism**
Energy management	individual	shared
Communication	IR local, low speed	wired, high speed
Sensors and actuation	the same sensors and actuators in all robots	specialization in using sensors and actuators
Internal Control	individual BIOS	need of middle-ware
Collective Control	self-organization	strong coordination
Coordination	through local rules	now unknown
Behavior	generalization	strong specialization
Functionality	emerged	evolved
Adaptation	changing local rules	artificial evolution

Hardware: Energy and communication. Two primary hardware differences between robot swarm and organism are energy supply (especially energy distribution) and communication between robots. In robot swarm each robot looks for available energy resources and consumes them. In Fig. 1 we demonstrate a typical scenario with autonomous energy foraging in a swarm. Basically, all robots compete for energy resources.

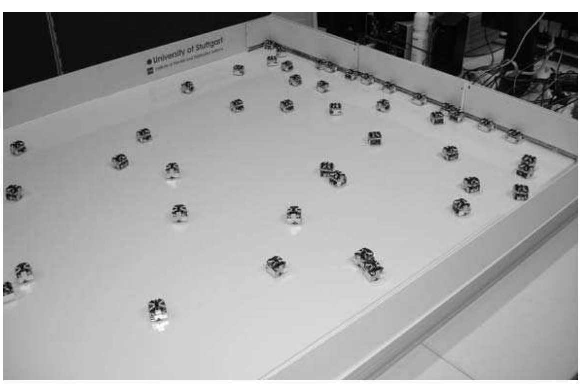


Fig. 1. *Collective energy foraging in a swarm of 50 micro-robots Jasmine. Docking station consists of two copper strips with 5V installed on the back-wall. Each robot has the Li-Po recharging chip. Robots when approaching docking station are competing.*

A multi-robot organism, see Fig. 2, possesses a common energy bus, where robots share their individual energy. Here robots cannot compete anymore for energy resources,

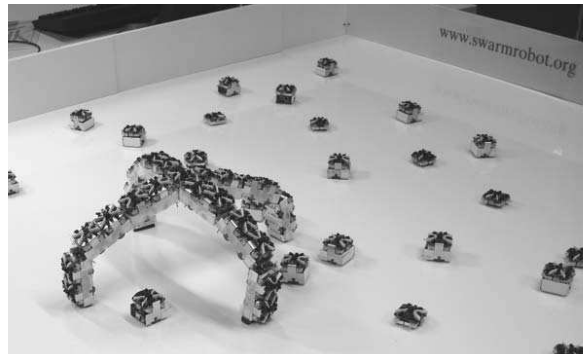


Fig. 2. *Topological model of a 3D multi-robot organism. The organism possesses a common energy bus, where all robots share their energy resources.*

other case the whole organism will not work. We can even imagine that some robots can sacrifice themselves so that the organism survives. Such energetically dead robots can serve either as a rigid construction element or they should be dissociated from the organism. We see that electrical construction and behavioral strategies for both scenarios are completely different.

The main communication in a robot swarm is performed locally through IR scheme (communication distance is about 15-20 cm.). Global message exchange is possible (with wireless ZigBee protocol), but is used exceptionally, because it does not provide information context [12]. The amount of communicating messages is quite tiny due to essential difficulties in message routing [12].

Considering communication between robots in the organism, we cannot use anymore IR or RF protocols due to interferences and relatively low transfer rate. Robots should be combined into a kind of internal communication bus (CANbus) with a high transfer rate. We should also foresee a situation 'communication through dead robots'.

Hardware: Sensors and actuation. Each robot in a swarm possesses individual sensors and locomotion. The approaches, used in many microrobotic solutions e.g. [13] [14], utilize IR-based proximity and distance sensing as well as non-holonomic differential wheeled drive with DC motors.

In the organism, all robots specialize in using only a few sensors and actuators for specific tasks. So, instead of proximity sensing, robots can use their IR sensors for making an 'organism skin'. The sensors will functionally be used in a completely different way and some of them will remain redundant. Individual robot locomotion is not necessary and is also redundant in the organism. However the robots should possess a kind of actuators, which allow working in the organism. Due to this actuation, the organisms is able to raise its own legs or generally to move. These actuators are not used in the robot swarm.

Software: Control of individual behavior. When hardware issues are quite hard, but their solution seems to be feasible, the software issues are completely open even in the conceptual part. We start with controlling individual behavior. In the Jasmine swarm each robot has its own BIOS (Basic Input Output System) providing interface to hardware, managing memory, interruption, communication and user defined activities. This BIOS is running on two/three 8MHz microcontrollers. Reaction time of the robot is about 50-100 ms in the case of Jasmine robots (without communication is about 10 ms).

In the case of an organism we encounter completely different situation. Firstly, when an individual robot has only two (or four) microcontrollers, the organism possesses about 200-400 microcontrollers. In this way it represents a large distributed system. We will encounter internal coordination and synchronization problems typical for distributed systems. Secondly, the organism should be controlled by a 'brain', where a position and functionality of this 'organism brain' is now unclear. It can be expected that such a brain is built in a distributed way, where each robot gives a part of its own computational resources for collective controlling purposes. In this case each robot should support a middle-ware, capable for a high-parallelization.

Software: Control of collective behavior. The control of collective behavior is the hardest part. The collective behavior of the robot swarm is controlled by local rules implemented in each robot. When robots are interrupting, they undergo self-organization processes, which in turn emerge a macroscopic phenomena, known as collective behavior. We often say, that collective systems produce social rules on a mezoscopic level, which finally govern collective behavior. As followed by well-know Poincaré effect [15], we are not able to predict collective behavior analytically, so that only simulation and real experiments remain for investigation of self-organized and emergent phenomena in collective robotic systems.

Comparing to the multi-robot organism, we observe another problem. Usually, emergent phenomena are quite slow and have a probabilistic character. It means that most robots in a swarm 'obey' mezoscopic social rules, however there are always a few robots which are doing something else. In organism we expect a well-coordinated behavior of all components. This concerns actuation, sensing, information transfer, and most of all, collective decision making. Currently it is not clear how to implement a transition from collective self-organized behavior in a swarm in strongly coordinated collective behavior of the organism. We suggest using a genome-based control of the organism, as described in Sec. V.

Behavior: Generalization and specialization. Swarm robots are in the most scenarios homogeneous, they have the same hardware and software. The extension boards of Jasmine robots can carry additional elements, like micro-camera, ego-positioning sensors, but the 'kern' of all robots is the same. The homogeneity of hardware and software reflects in turn in pretty homogeneous behavior of robots. Robots can play the roles of scout, messenger, 'hungry robots' and can transit from one role into another one. However comparing behavior of different roles, we can say it has the same level of homogeneity, e.g. in all roles robot move, avoid collision, getting hungry and so on.

We should remark that in some performed experiments, e.g. experiments with collective energy foraging, the robots can emerge a slight specialization of behavior. For example, some robots move mostly around the docking station, performing recharging more often than other robots. Such robots get a kind of social parasites, consuming resources, whereas other robots are working. This slight specialization has a behavioral character and does not change the functionality of robots.

Considering a multi-robot organism, we observe a strong functional specialization of all robots. This specialization depends on the position of the robot in the organism. When a robot is somewhere around a 'joint', e.g. a 'joint' of driving legs, the robot will functionally specialize in a specific driving actuation. Robots in the middle of organism specialize more in controlling and information transfer, whereas robots aside more in environmental perception. Therefore here we encounter the issues of robots functionalization in the organism.

Behavior: Adaptation and evolution. Basically all swarm robots can individually learn with some unsupervised feedback-based approaches. For example, the swarm density and geometry of arena impact most motion characteristics (e.g number of collision avoidances) and in turn, energy consumption. Robots change motion velocity to achieve a minimal energy consumption, i.e. they learn to behave in such a way which requires less energy. Moreover, robots in a swarm can also undergo collective adaptation to environmental changes. The whole swarm is capable for learning, this appears as simultaneous changing of some parameters in many robots. As an effect we observe a changed collective behavior.

Multi-robot organism has essentially more degrees of flexibility, it can change its own form and so its own functionality. It can appear different locomotion and actuation strategies. Finally, by changing scenarios-specific tools, organism can even perform different operations. The reasonable question is how the organism should learn adaptation to environment ? We do not have any answers on this question now. However we believe that evolutionary strategy, like evolutionary learning, can be very useful here.

Organism-specific issues: flexibility of software and hardware. One of obvious questions in robotic community is about flexibility of hardware and software. It is quite evident that software is much more flexible. With some tools (e.g. genetic programming) a program can even rewrite itself. The hardware is more 'hard'. The question is whether a cellular construction of multi-robot organism is a right way towards hardware flexibility ?

In nature we observe the diversity of cellular objects. In fact, almost all cells have similar construction, however they build completely different higher forms. When we consider nature more closely, we should remark, that cells themselves are from structural viewpoints much more complex than some multi-cellular live forms. When combining into organisms, cells turn off some of their functionality and so specialize in behavior. In this way a multi-robot organism, as said even by Fucuda [16], can provide a required hardware flexibility.

Organism-specific issues: shapes and functionality The last issue, related to the organism, is the control of topological shapes. Robots, when aggregating into an organism, can do it in different ways and so emerge different functionality. In Fig. 3 we demonstrate two different shapes (octopus-like and snake-like) which have different locomotion strategies. The question is how to control the building of shapes and so emergent functionality of the organism ? Evidently, that in this stage we have more questions than answers. We expect to clear these questions in a further research.

Fig. 3. *Topological models of two different multi-robot organisms of different topologies and different locomotion principles.*

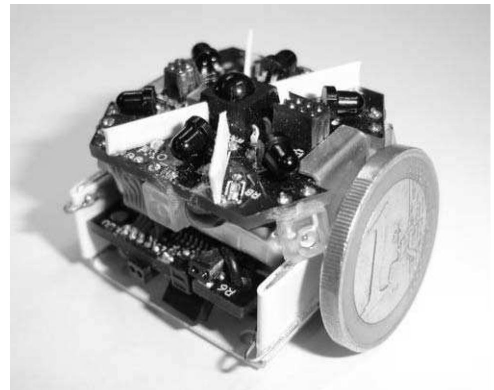


Fig. 4. *The microrobot 'Jasmine III plus'.*

III. From swarm robot Jasmine III to symbiotic Jasmine V

Jasmine III plus. For performing the swarm-experiments and testing the embodiment concept we used the microrobots Jasmine, see Fig. 4. It is a public open-hardware development at **www.swarmrobot.org**, having a goal of creating a simple and cost-effective microrobotic platform and knowledge exchange in the swarm robotics community. The micro-robot is 26×26×20mm, uses the two Atmel AVR Mega micro-controllers: Atmel Mega88 (motor control, odometry, touch, color and internal energy sensing) and Mega168 (communication, sensing, perception, remote control and user defined tasks). Both micro-controllers communicate through high-speed two-wired TWI (I2C) interface. It has generally on board 24kb flash for program code, 2kb RAM for data and 1kb nonvolatile EEPROM for saving working data.

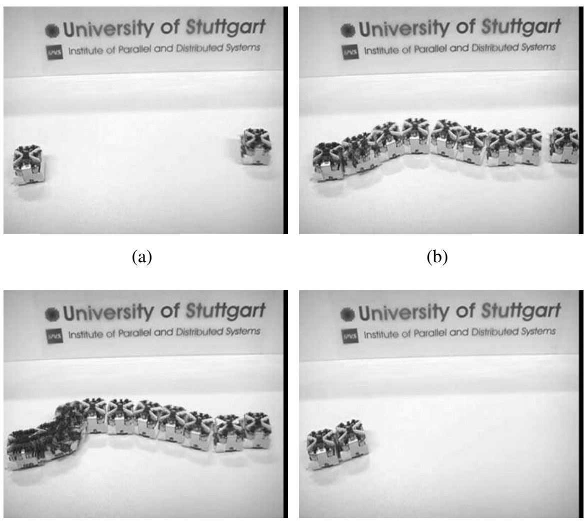
The robot has six (60° opening angle) communication channels (they are also used for proximity sensing) and one geometry-perception-channel (15° opening angle) based on separate IR receivers and transmitters. Communication area covers 360° rose-like-areal with maximal and minimal ranges of 200mm and 100mm correspondingly. The physical communication range can be decreased through a change of sub-modulation frequency. The robot has also a remote control and robot-host communication (up-link and down-link), which is isolated from all other channels (through modulation).

The robot uses two DC motors with internal gears, two differentially driven wheels on one axis with a geared motor-wheels transmission. Encoder-less odometrical system normalizes a motion of the robot (the robot is able to move straight forward and backward), and estimates the gone distance with accuracy of about 6% and a rotation angle - of 11%. Jasmine III uses 3V power supply (from 3,7V Li-Po accumulator) with internal IC-stabilization of voltage. Power consumption during motion is about 200mA, in stand - 6mA, in stand-by mode less 1 mA. The time of autonomous work is of 1-2 hours. The robot has an internal energy sensor (hungry feeling) and is also capable of autonomous docking and recharging.

The programming of the robot uses C language with open-source *gcc* compiler, there is a complete BIOS system that supports all low-level functions. Moreover, for a quick implementation of swarm behavior there is a developed jasmine-SDK system, that includes an 'operational system' and high-level functions. SDK allows a quick implementation of robotic swarm programs and supports two 'operational systems', one based on MDL (Motion Description Language) and another one based on Autonomy Cycle.

Basically, Autonomy Cycle executes continuously four steps: *reading of sensor data*, *communication*, making a *decisions* and finally *executing a plan*. The interruption service takes care about software and hardware interruptions. The plan, that a robot has to execute, represents a Petri net consisting of two parts: service part (handlers for interruptions) and user-defined-part (behavioral program for the robot). The structure of service part represents an interruption vectors system with corresponding handlers. The interruptions (like touch or low-energy) are generated by the BIOS system, users only need to write the corresponding handlers. For further details of construction and programming see www.swarmrobot.org or e.g. [14].

Jasmine V. The symbiotic Jasmine V is based on the described Jasmine IIIp and includes additional docking elements for aggregation and corresponding electronics (currently under development). The most critical operation for multi-robot organism is the self-assembling from separate robots into one organism. For this we tested possibility and accuracy of docking approach with current Jasmine IIIp robots. Robots are lead by IR light, emitted by the first robots. The second robots, by following this light, try to dock to the first robots, see Fig. 5.

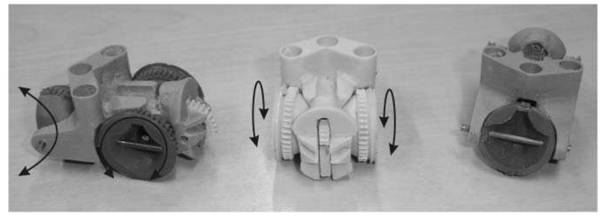


(a) (b)

(c) (d)

Fig. 5. *Test of docking approach with Jasmine IIIp.* ***(a)*** *Original state of two robots;* ***(b)*** *Difference images (each second) of the first attempt of docking;* ***(c)*** *Difference images (each second) of the second attempt of docking;* ***(d)*** *final state of robots.*

As demonstrated by many experiments, accuracy of docking is about +-5mm. Based on these experiments we developed the first prototype of the robots capable for self-assembling, shown in Fig. 6(a). Each robot has three female docking connectors and one male docking connector in the front of the robot. Two female docking connectors are placed in the wheels, so that robots have individual locomotion based on a differential drive. These docking connectors can rotate. The third female docking connector is placed behind and is capable for vertical rotation. It has a strong motor. Male docking connector has a hook-based lock mechanism. All female connectors and hook-based lock are driven by DC motors, integrated into plastic chassis. Transmission between motors and wheels is done by a worm gear, as shown in Fig. 6(b). Advantage of the worm gear is that this can fix the position of wheels, so that it can statically keep the required position (configuration of the organism) and does



(a)

(b)

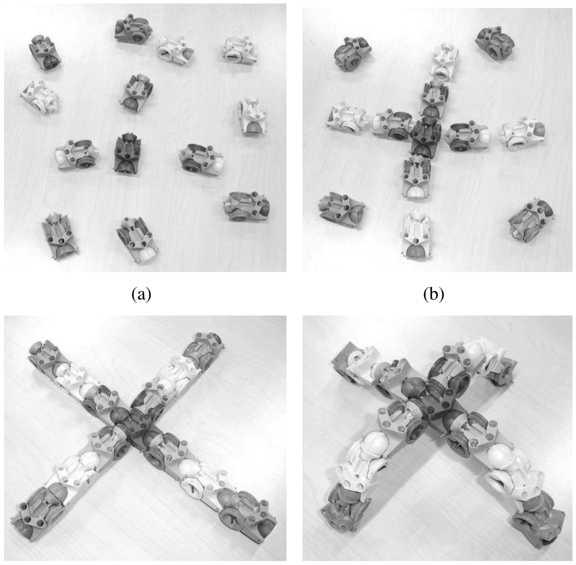
Fig. 6. ***(a)*** *The first prototype of the robot capable for self-assembling in a multi-robot organism;* ***(b)*** *The worm gears connecting integrated motors and wheels.*

not consume energy in this mode. In this way, each robot has two degrees of freedom: one vertical-plane rotation behind with strong forces and one rotation around the wheel axis.

IV. The first topological experiments

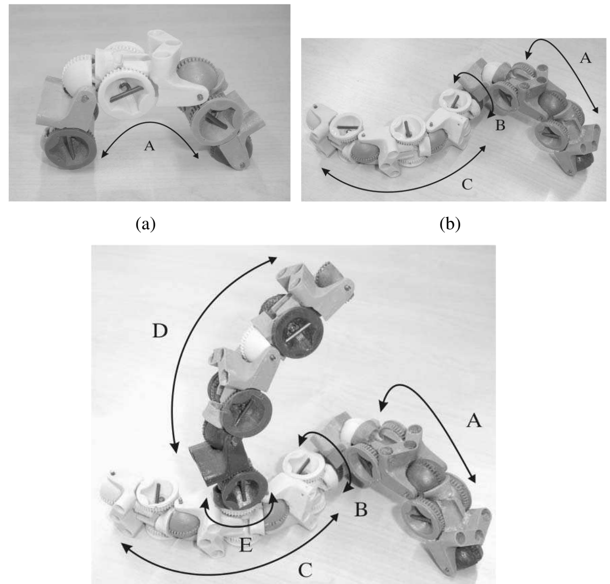
Before further development of self-assembling robots, we performed a series of topological experiments. They are related to capabilities of planar self-assembling and demonstrated 3D topologies; a relation between individual degrees of freedom (DOF) and collective DOF for the organism; locomotion principles and finally encountered open points.

1. Planar self-assembling and 3D topologies. The intended self-assembling process is performed on the surface, where the swarm robots are moving, see Fig. 7(a). One robot starts to send an attraction signal. Later on, this robot will be a seed point of the whole self-assembling. The docking approach utilized the IR-based scheme, developed and tested on Jasmine IIIp robots (shown in Fig. 5). It is expected that each robot has IR emitters and transmitters on each docking connector. At this development stage, it is not clear whether all robots will be equal (similar) or they will differ in their construction. Therefore we can assume that during the self-assembling procedure either every robot can be attracted to the organism or only specific robots will be allowed for docking, Fig. 7(b). In this state robots can either finish building the whole organism, see Fig. 7(c), or they create only some autonomous parts of the organism. When robots are docked, they start electrical and logical processes of integration in the organism. After that organism starts its own locomotion (e.g. stands up) and starts its own 'life' as an organism, see Fig. 7(d).



(a) (b) (c) (d)

Fig. 7. *Intended planar self-assembling of swarm robots into a multi-robot organism.*



(a) (b) (c)

Fig. 8. *Degrees of freedom for the organism when working in different configurations.* ***(a)*** *1 DOF when robots are assembled by using only the back connector;* ***(b)*** *3 DOF when using one rotational connector;* ***(c)*** *5 DOF when using two rotational connectors.*

2. Individual and collective degrees of freedom. We are interested whether the chosen individual degrees of freedom (DOF) allow enough collective DOF for the organism. In Fig. 8 we demonstrate a few configurations of the organism and appeared DOF. Generally, we can say that each connection to the rotational connector increase DOF by one. Using different rotational connectors, the organism can obtain different vertical, horizontal and rotational DOF.

Combining several DOF, the organism can demonstrate different locomotion principles, as shown in Fig. 9:

- Wheeled Motion. The most simple locomotion principle is using its own wheels, as shown in Fig. 7(a),(b). In this stage robots are alone (robot swarm scenario) or have small two-, three robots conciliations.

- Snake-like. This locomotion principle, shown in Fig. 9(a), is similar to a classical waves-driven locomotion. When all robots are connected only by using the back connector (1DOF), the produced 1D waves are instable. Therefore we suggest using interconnections between back- and rotational- connectors.

- Snake-legs. Combining rotational connectors with snake-like locomotion, we obtain interesting configuration, see Fig. 9(b), where robots can use short legs when encountering obstacles.

- 4 legs configuration. The shown configuration in Fig. 9(c) is quite standard 'octopus-like' locomotion used in many reconfigurable robotic systems.

- 6 legs configuration. The 6-legs configuration in Fig 9(d) represents some possible extension of the 'octopus'. Combining more different modules, organism will be capable of more advanced 8-, 10- or 12- legs locomotion even with flexible connection between legs.

3. Encountered open issues. In the first topological tests, we encounter the following problem. As already mentioned, all robots will use their available sensors, but change their functionality. For example, the IR proximity sensors from swarm scenario can be used as a 'virtual skin' in the organism scenario. However, the organism requires such sensors which are not available in a swarm robot. In Fig. 10(a) we demonstrate an example, where the organism raises its own leg. Topologically, the organism shifts the center of mass and loses equilibrium. To stabilize equilibrium, the organism has to sense the center of mass and to move so long back until equilibrium will be achieved, see Fig. 10(b). It is evident, that such sensors are not required in robot swarm scenario. Moreover, it is completely unclear now, which specific functionality is required and how to achieve this functionality in emergent/evolutionary way.

V. GENOME-BASED CONTROL OF THE ORGANISM

In this section we briefly introduce the intended genome-based control concept of the organism and the preliminary experiments with Jasmine IIIp robot. As already mentioned, the organism will be controlled through a genome framework. A genome of an artificial organism carries the total set of genes of an individual it also includes the hereditary instructions for building, running and maintaining the organism. Genes are a location of the genomic sequence and act like a unit of inheritance and hence build up a

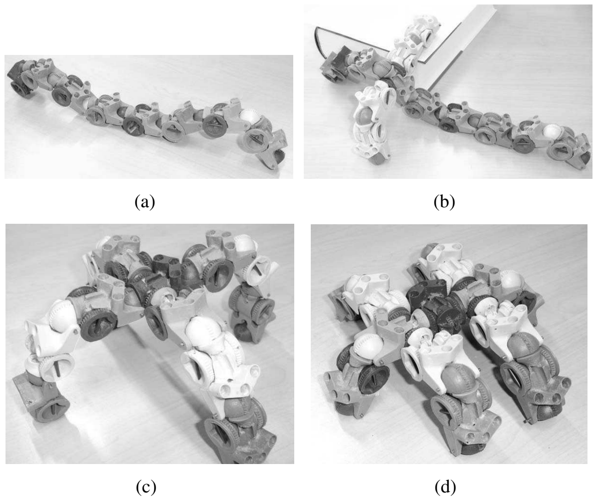


Fig. 9. *Different locomotion principles, obtained by combining several DOF.*

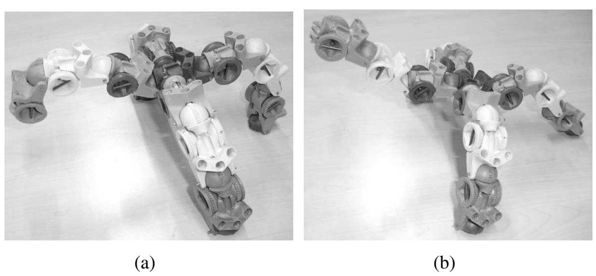


Fig. 10. *Example of extra sensors. **(a)** Organism raises its own leg, the center of mass is shifted forwards and organism loses equilibrium; **(b)** Shifting the center of mass backwards to stabilize its own spatial position.*

genome [17]. Genes are functionally complete and consist of small 'states' (by analogy to biological 'bases'). For example, the function 'move' is a gene and consists of different states that control a movement. Using these states, it is possible for genes as well as for the whole genomes to be manipulated through recombination, mutation or specific learning approaches. This feature will be exploited in the genetic software framework.

This framework extends the already mentioned control concept and has the following five stages: *Read proximity*, *Communication*, *Planning*, *Genome Interpretation* and *Execution*. *Read Proximity* is the first stage in which the values of the sensors are read. Using these values, the autonomy cycle is continued and checks for messages. If there is one, then communication is activated. The cycle continues into the *Planning* stage which performs simple planning (control) activities. As mentioned in Sec. III, the behavior of the robot is controlled through Petri nets. In the genetic framework, this Petri net is dynamically generated by the *Interpreter*. The *Interpreter* reads the sequence from the symbolic sequence, representing the encoded genetic information, and transforms it into executable Petri net(s). During *Execution* phase, token travels through this net, where the corresponding commands are finally transferred to actuators and executed. The symbolic sequence of encoded

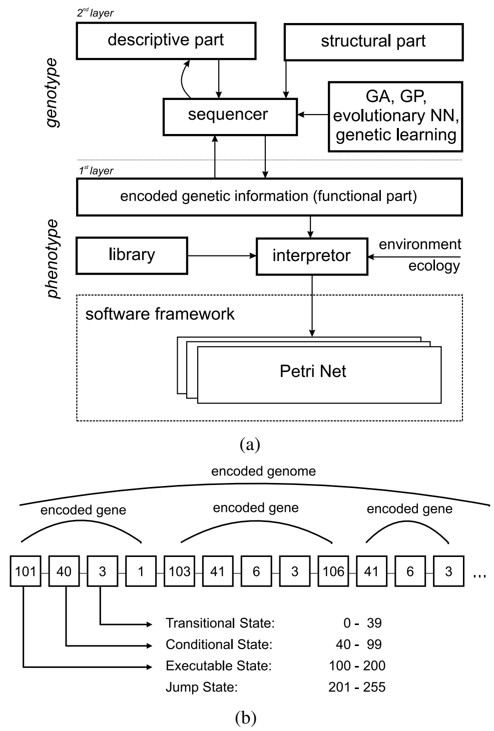


Fig. 11. ***(a)** Structure of encoding process; **(b)** Structure of encoded functional part.*

functional part can have a maximum of 2^{16} elements and consists of executable, conditional, transitional, regulative and jump states, see Fig. 11(b). Executable states determine the actions which control behavior of the robot. Conditional and regulative states decide the transitions to be made while reading the genome. The transitions point to executable states.

The sequence of the encoded genetic information is generated by the *sequencer*. *Sequencer* reads structural and descriptive parts, and generates a sequence of commands for creating a behavioral Petri net. Sequencer not only parses symbolic streams, but can also apply different evolutionary/learning operators (such as GA, GP, evolutionary neural networks, genetic learning) to the descriptive genome. The embodiment of the robots, i.e. possibilities and limitations imposed on functional behavior through a physical construction of a robot, are coded in the structural part. The combination of variable descriptive part and hard-to-modify structural part eliminates the possibility of creating unwanted functions (i.e. such functions which are impossible for a robot) in the encoded functional genome. Due to recombination or mutation in the descriptive part, variation in the genotype is implied. The descriptive and structural parts are so-called genotype of the robot.

Advantage of splitting genetic information on three levels (descriptive, structural and encoded functional) is the following. The same functionality of the robots (e.g. sense collisions) can be implemented in many different ways, it depends on the available sensors, actuators and other capabilities. For example, "sense collisions" can be done by IR-proximity sensors, by processing of camera images, by laser scanners and so on. The descriptive genome only contains abstract "sense collisions", whereas the encoded functional part contains a low-level implementation of "sense collisions". The structural part plays a role of generating rules, which transform abstract descriptions into concrete implementations on a specific robot platform. However, the encoded functional genome can also possess such modifications which are not directly contained in the genotype. For example, environmental influences, cooperation with other robots can directly influence an encoded functional genome. Since this implies a behavior of robots and involves artificial ecology, the interpretation of encoded functional genome can represent a phenotype of the robot.

Not only genotype, but also phenotype of robot can undergo evolutionary operations. For example, by using IR local communication or the wireless ZigBee modules, the encoded genome of one robots can easily be overwritten by encoded genome from other robots, see Fig. 12. This allows

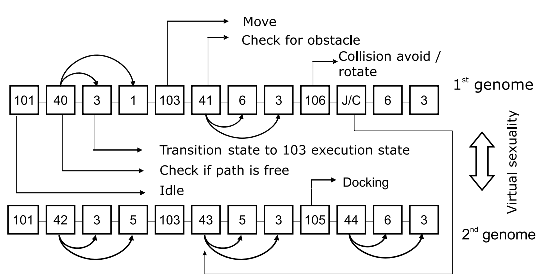


Fig. 12. *Mutation-Adding/overwriting of encoded functional genome.*

us to evolve the encoded functional genome during runtime. We refer the capability of on-line modification of encoded functional genome to the virtual robot sexuality.

On the current state of research, it is evident, that descriptive, structural and functional parts of genetic information contain different species-related and individual-related genetic information on different levels of abstraction. Any changes, caused by off-line (based on generations change: GA/GP) or on-line (based on changes during individual lifetime: evolutionary learning) approaches can be stored either in species-related or individual-related genetic memory. However it is completely unclear now, which consequences has it for the robot organism and the robot population.

VI. Conclusion

In the given paper we have shown a new paradigm in collective systems, where the robots in the swarm scenario get capable of self-assembling into a single multi-robot organism. The robots can aggregate and disaggregate autonomously in different topological forms. We demonstrated that a transition between robot swarm scenario and multi-robot organism represents a very hard problem and involves hardware, software and behavioral issues. We investigated accuracy of docking approach for the existing Jasmine IIIp robots and developed the first prototype of robots capable for working in both, swarm and organism, scenarios. For this prototype we also investigated topological issues related to the degrees of freedom and briefly demonstrated the control concept involving robots genome. Based on these experiments, we are currently developing fully functional robots. This represents the further works.

References

[1] E. Bonabeau, M. Dorigo, and G. Theraulaz. *Swarm intelligence: from natural to artificial systems*. Oxford University Press, New York, 1999.

[2] S. Camazine, J-L. Deneubourg, N.R. Franks, J. Sneyd, G. Theraulaz, and E. Bonabeau. *Self-Organization in Biological Systems*. Princeton University Press, Princeton, NJ, USA, 2003.

[3] Thomas H. Labella, Marco Dorigo, and Jean-Louis Deneubourg. Division of labor in a group of robots inspired by ants' foraging behavior. *ACM Trans. Auton. Adapt. Syst.*, 1(1):4–25, 2006.

[4] I-Swarm. *I-Swarm: Intelligent Small World Autonomous Robots for Micro-manipulation, 6th Framework Programme Project No FP6-2002-IST-1*. European Communities, 2003-2007.

[5] S. Kornienko, R. Thenius, O. Kornienko, and T. Schmickl. Re-embodiment of honeybee aggregation behavior in artificial micro-robotic system. *Adaptive Behavior (accepted for publication)*.

[6] *swarmrobot.org*.

[7] H. Haken. *Synergetics: An introduction, third edition*. Springer-Verlag, New York, 1983.

[8] A. Ishiguro and T. Maegawa. Self-assembly through the interplay between control and mechanical systems. In *Proc. of IEEE/RSJ06 Int. Conf. on Intelligent Robots and Systems*, pages 631–638, Beijing, China, 2006. IEEE.

[9] S. Murata, K. Kakomura, and H. Kurokawa. Docking experiments of a modular robot by visual feedback. In *Proc. of IEEE/RSJ06 Int. Conf. on Intelligent Robots and Systems*, pages 625–630, Beijing, China, 2006. IEEE.

[10] G. Weiss. *Multiagent systems. A modern approach to distributed artificial intelligence*. MIT Press, 1999.

[11] J.Y. Halpern and Y. Mosesi. Knowledge and common knowledge in a distributed environment. *J. of the Association for Computer Machinery*, 37(3):549–587, 1990.

[12] S. Kornienko, O. Kornienko, and P. Levi. Collective ai: context awareness via communication. In *Proc. of the IJCAI 2005, Edinburgh, UK*, 2005.

[13] A. Martinoli. *Swarm Intelligence in autonomous collective robotics: from Tools to the analysis and synthesis of distributed control strategies PhD Thesis*. Lausanne, EPFL, 1999.

[14] S. Kornienko, O. Kornienko, and P. Levi. Ir-based communication and perception in microrobotic swarms. In *Proc. of the IROS 2005, Edmonton, Canada*, 2005.

[15] S. Kornienko, O. Kornienko, and P. Levi. About nature of emergent behavior in micro-systems. In *Proc. of the Int. Conf. on Informatics in Control, Automation and Robotics (ICINCO 2004), Setubal, Portugal*, pages 33–40, 2004.

[16] T. Fukuda and T. Ueyama. *Cellular robotics and micro robotic systems*. World Scientific Publishing Co. Pte. Ltd., 1994.

[17] Bruce Alberts, Alexander Johnson, Julian Lewis, Martin Raff, Keith Roberts, and Peter Walter. *Molecular Biology of the Cell*. Garland Science, March 2002.