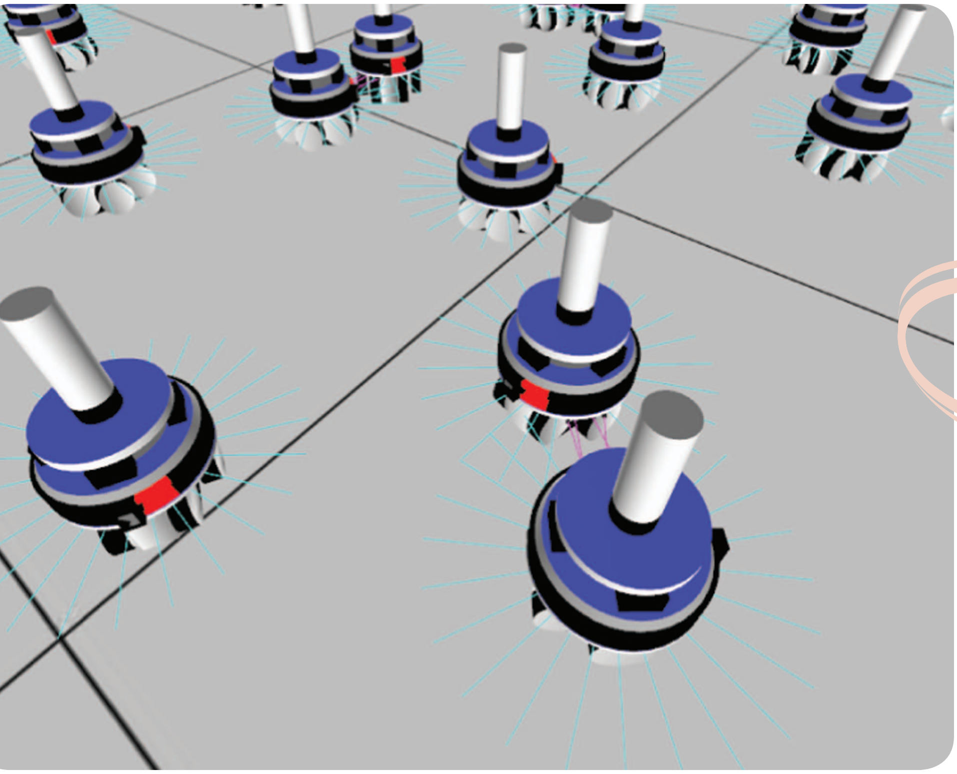
Swarmanoid

A Novel Concept for the Study of Heterogeneous Robotic Swarms



By Marco Dorigo, Dario Floreano, Luca Maria Gambardella, Francesco Mondada, Stefano Nolfi, Tarek Baaboura, Mauro Birattari, Michael Bonani, Manuele Brambilla, Arne Brutschy, Daniel Burnier, Alexandre Campo, Anders Lyhne Christensen, Antal Decugnière, Gianni Di Caro, Frederick Ducatelle, Eliseo Ferrante, Alexander Förster, Jerome Guzzi, Valentin Longchamp, Stéphane Magnenat, Javier Martinez Gonzales, Nithin Mathews, Marco Montes de Oca, Rehan O'Grady, Carlo Pinciroli, Giovanni Pini, Philippe Rétornaz, James Roberts, Valerio Sperati, Timothy Stirling, Alessandro Stranieri, Thomas Stützle, Vito Trianni, Elio Tuci, Ali Emre Turgut, and Florian Vaussard

Digital Object Identifier 10.1109/MRA.2013.2252996
Date of publication: 13 September 2013

Swarm robotics systems are characterized by decentralized control, limited communication between robots, use of local information, and emergence of global behavior. Such systems have shown their potential for flexibility and robustness [1]–[3]. However, existing swarm robotics systems are by and large still limited to displaying simple proof-of-concept behaviors under laboratory conditions. It is our contention that one of the factors holding back swarm robotics research is the almost universal insistence on homogeneous system components. We believe that swarm robotics designers must embrace heterogeneity if they ever want swarm robotics systems to approach the complexity required of real-world systems.

To date, swarm robotics systems have almost exclusively comprised physically and behaviorally undifferentiated agents. This design decision has its roots in ethological models of self-organizing natural systems. These models serve as inspiration for swarm robotics system designers, but are often highly abstract simplifications of natural systems and, to date, have largely assumed homogeneous agents. Selected dynamics of the systems under study are shown to emerge from the interactions of identical system components, ignoring the heterogeneities (physical, spatial, functional, and informational) that one can find in almost any natural system.

The field of swarm robotics currently lacks methods and tools with which to study and leverage the heterogeneity that is present in natural systems. To remedy this deficiency, we propose swarmanoid, an innovative swarm robotics system composed of three different robot types with complementary skills: foot-bots are small autonomous robots specialized in moving on both even and uneven terrains, capable of self-assembling and of transporting objects or other robots; hand-bots are autonomous robots capable of climbing some vertical surfaces and manipulating small objects; and eye-bots are autonomous flying robots that can attach to an indoor ceiling, capable of analyzing the environment from a privileged position to

1070-9932/13/$31.00©2013IEEE

collectively gather information inaccessible to foot-bots and hand-bots (see Figure 1).

Swarmanoid exploits the heterogeneity and complementarity of its constituent robot types to carry out complex tasks in large, three-dimensional (3-D), man-made environments. Humanoid robots are usually assumed to be the most efficient robot type for man-made environments. One of the goals of the swarmanoid project was to refute this assumption. The term *swarmanoid* is, in fact, a compound of *swarm* and *humanoid*. The system has no centralized control and relies on continued local and nonlocal interactions to produce collective self-organized behavior. The swarmanoid architecture provides properties that are difficult or impossible to achieve with a more conventional robotic system. Swarmanoid shares the strengths of existing swarm systems. Robots of a particular type are directly interchangeable, providing robustness to failures and external disturbances. However, swarmanoid's heterogeneous nature gives it a flexibility that previous swarm systems cannot match. Different sensing and actuating modalities of its heterogeneous components can be combined to cope with a wide range of conditions and tasks. The swarmanoid even features dynamic self-reconfigurability: groups of robots can get together on a by-need basis to locally form ad hoc coalitions or integrated structures (by connecting to each other) that can perform more complex tasks. Thanks to the heterogeneity of the robots in the swarm, these coalitions can flexibly integrate a variety of skills.

To the best of our knowledge, the swarmanoid represents the first attempt to study the integrated design, development, and control of a heterogeneous swarm robotics system. In the following sections, we first discuss the issues and challenges intrinsic to heterogeneous swarm robotics systems. We then give an overview of the swarmanoid system. Finally, we describe the experimental scenario devised to demonstrate the capabilities of the swarmanoid.

Heterogeneous Robotic Swarms: Issues and Challenges

Heterogeneous robotic swarms are characterized by the morphological and/or behavioral diversity of their constituent robots. In a heterogeneous swarm robotics system, the need for physical and behavioral integration among the different hardware platforms results in a considerable amount of extra complexity for the design and implementation of each different type of constituent robotic agent. This integration complexity must be dealt with both in the hardware design and at the level of behavioral control.

Robots within a heterogeneous swarm must be able to cooperate. At the hardware level, this imposes the minimum requirement that the various robot types have common communication devices and the sensory capabilities to recognize one another's presence. Even this basic design requirement is not trivial to realize. Robot communication devices are often tailored to a particular robot morphology and functionality. Flying robots, for example, need communication devices that are light and power-efficient, whereas for ground-based robots, higher performance devices that are heavier and consume more power may be appropriate. The challenge is thus to ensure that functionally similar devices with very different design criteria can seamlessly interface with one another.

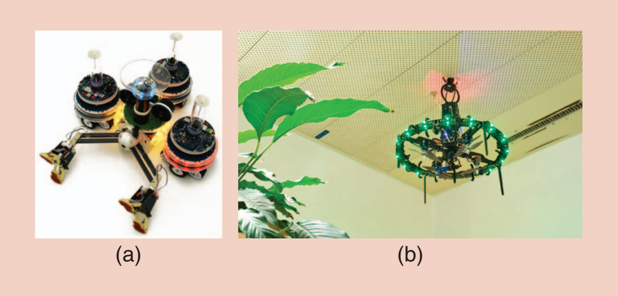


Figure 1. The swarmanoid robots. (a) Three foot-bots are assembled around a hand-bot and are ready for collective transport. The hand-bot has no autonomous mobility on the ground and must be carried by foot-bots to the location where it can climb and grasp objects of interest. (b) An eye-bot attached to the ceiling has a bird's-eye view of the environment and can thus retrieve relevant information about the environment and communicate it to robots on the ground.

Swarm robotics systems also favor less direct interaction modalities. Stigmergic interactions, for example, are mediated by the environment [4] and have proven effective in swarm systems. In a heterogeneous swarm, the difficulty is to ensure that the manipulation and sensing mechanisms of morphologically and functionally different robots are coherent enough to enable stigmergy. In fact, any form of nonsymbolic communication [e.g., visual communication using light-emitting diodes (LEDs) and a camera] requires a design effort to ensure a sufficient level of sensor and actuator integration between robot types.

Physical cooperation is often considered necessary in a swarm system to allow the swarm to overcome the physical limitations of single agents. An interesting possibility for physical interaction—often observed in biological systems—is self-assembly, i.e., the ability of different individuals to connect to one another thus forming larger composite entities. In robotics, this form of interaction can enable complex forms of cooperation. The implementation of self-assembly in homogeneous swarm robotics systems has already proven challenging [5]. Designing and implementing self-assembly-capable hardware in a heterogeneous system is significantly more complex. Different robot types in a heterogeneous swarm each have their own intended functionality, which imposes constraints on the morphology of each robot type and on their required sensors and actuators. Self-assembly is implicitly challenging among specialized heterogeneous robots, as it demands a certain level of morphological, sensory, and actuator compatibility among robot types.

Behavioral control is a challenge for any swarm robotics system. Individual control rules must be found that result in the desired collective behavior. The complexity of the design task resides in the indirect relationship between the robot's proximal level (i.e., the level of the individual controller that deals with the sensors, actuators, and communication devices) and the swarm's distal level (i.e., the overall organization, which refers to the dynamics and self-organizing properties of a complex heterogeneous robotic system).

In heterogeneous robotic swarms, the challenge is much harder. The would-be designer of a heterogeneous swarm system can potentially leverage the specialization of different robots to increase system efficiency. Different robot types may also have complementary sensory and actuation capabilities that allow them to work together in many different configurations such that the whole is more than the sum of the parts. But this additional potential comes at the cost of much higher complexity that must be managed by the system designer.

To enable cooperation in a heterogeneous robotic swarm, the choice of the communication modality is crucial. Communication is an essential aspect of any distributed robotic system and can take many different forms, ranging from indirect stigmergic interactions to networked structured communication.

In a heterogeneous swarm, it is necessary to consider both intra- and intergroup coordination. To enable intragroup coordination, it is necessary to develop interaction modalities within homogeneous groups. To enable intergroup coordination between groups composed of different robot types, the common communication system must enable useful information to be conveyed between robots that have radically different perceptions of the environment. This challenge opens a host of new and interesting problems, such as defining shared attention mechanisms within and between groups, or exploiting intragroup coordination and communication as behavioral templates for the development of intergroup coordination strategies.

To support the development of robot behaviors for swarms of robots, simulation is a fundamental tool. Real-world experimentation in swarm robotics is often impractical because of the necessity of testing behaviors with large numbers of robots. Simulation of heterogeneous swarms poses further challenges, as the different robot types may have different simulation requirements. A simulation tool for heterogeneous robots must, therefore, simultaneously offer scalability for increasing number of robots and flexibility to support highly diverse robots designs.

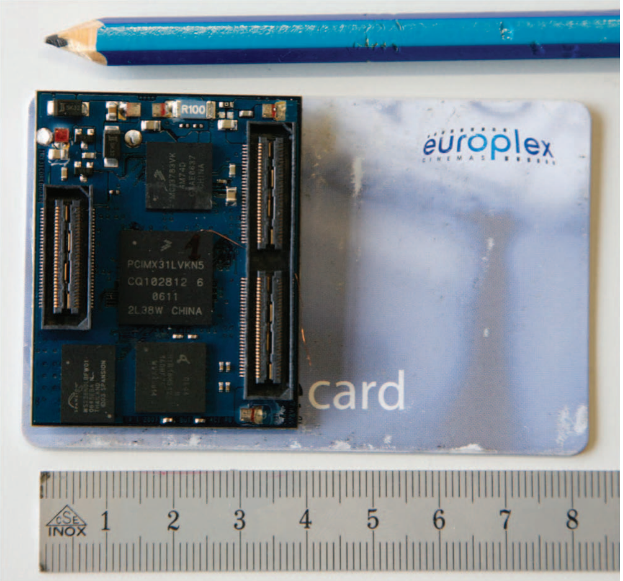


Figure 2. The half-credit-card-size i.MX31 main processor board.

Swarmanoid Technologies

Research on the swarmanoid has been guided by the issues broached in the previous section. As discussed, the various constituent robot types must be able to interact, either physically or through communication. We tackled the interaction problem from the outset by designing a set of common technologies to provide a uniform hardware architecture. In this section, we first describe these common technologies and then detail the hardware design of the three robotic platforms. Finally, we present the dedicated simulator that we developed.

Common Technologies

All robots have a multiprocessor architecture, consisting of a main processor that takes care of CPU-intensive tasks, such as vision and higher-level control, and several microcontrollers that take care of real-time sensor reading and actuator control. This design choice represents a clear architectural shift away from the classic single-microcontroller robot to a distributed, intrinsically modular design. The resulting ability to design and test components in isolation increases component quality and allows for parallel development of different components.

We designed and developed a common main processor board for all the robot types. The board is based on a 533-MHz i.MX31 ARM 11 processor and features 128 MB of RAM, 64 MB of Flash, a USB 2.0 host controller, and an energy and I/O companion chip (see Figure 2). The microcontrollers are based on the DsPIC 33, as it provides good computational power, includes fixed-point and DSP instructions, and has low power consumption.

To provide access to the different devices of the robot, we developed a low-level software architecture, called ASEBA [6], that abstracts the peculiar features of the different robot modules and offers an easy-to-use tool for robotic experimentation. ASEBA is an event-based architecture consisting of a network of processing units that communicate using asynchronous messages called events. The usual read/write transactions from the main processor to the microcontrollers are replaced by events sent from any node to any other node on the common communication bus. All nodes send events and react to incoming events. In our robots, the typical network is formed by the main processor board and the various microcontrollers, which communicate through a controller area network (CAN) bus. The microcontrollers correspond to the different sensors and motor devices that are implemented on the robot. The robots' behaviors are based on the data provided by these sensor and motor devices. These data can be either processed locally by the microcontroller or communicated through an asynchronous event. Asynchronous events are implemented as messages that have an identifier and payload data. By exchanging events and processing data both locally and remotely, complex control structures can be implemented. The network of processing units can be extended through TCP-IP to any remote host. For development and debugging, for example, an integrated development environment running on a desktop computer can be incorporated into the control structure [6].

Another essential feature of the swarmanoid is communication between different robotic platforms. We have designed and implemented a common sensing and communication system for all robot types that is based on a combination of infrared (IR) and radio communication. This system provides relative localization and structured communication signals. The system, referred to as the range and bearing communication system, was inspired by similar devices developed by Pugh et al. [7] and Gutierrez et al. [8]. These previous devices, however, were severely limited in both range and precision. We therefore decided to design a new integrated device. Our new device allows relative localization (from 10 cm up to 5 m for the foot-bots and hand-bots and up to 12 m for the eye-bots), data communication at a relatively high rate, and full 3-D operation, all interference-free. Our system uses a combination of new techniques to optimize the way a range measurement is attained and how data are transmitted. To obtain a measurement with an increased dynamic range, we use a four-stage cascaded amplifier. Each of the four stages is designed to output a voltage corresponding to a complementary region of the maximum range. To optimize the speed of range measurement, we removed the data from the IR signal and instead transmit it over a 2.4-GHz transceiver. The transceiver also manages synchronization of different devices through a simple turn-taking algorithm [9].

Foot-Bot

The foot-bot (Figure 3) is an autonomous robot that is conceptually modular at all levels: mechanics, electronics, and software. Mechanical modularity is achieved by stacking modules on top of one another, following well-defined specifications. The modularity of the electronics is achieved by partitioning the required functionality of each module to make them as independent as possible. Each module is provided with its own local processing power, thus enabling the distributed architecture based on ASEBA. The different modules share battery power, some common control signals (e.g., power enable or reset), and the communication buses (CAN and I2C).

The foot-bot is 28-cm high and has a diameter of 13 cm. It is powered by a 3.7-V, 10-Ah lithium-polymer battery contained in the base module, which also houses a mechanism allowing battery hot-swapping. This capability is provided by a super-capacitor, which maintains the power supply of the robot for 10 s during battery exchange. The foot-bot has differential drive motion control and is composed of two 2-W motors, each powering a rubber track and a wheel (this combination is collectively referred to as a *treel*). The maximum speed of the foot-bot is 30 cm/s. The base of the foot-bot includes IR sensors, some acting as virtual bumpers and others as ground detectors. These sensors have a range of some centimeters and are distributed around the robot on the main printed circuit board (PCB): 24 are outward-facing for obstacle detection, and eight are downward-facing for ground detection. Additionally, four ground sensors are placed under the lowest part of the robot, between the treels. The base of the foot-bot also contains a radio frequency identification reader and writer with an antenna situated on the bottom of the robot, close to the ground. To allow for proprioceptive orientation measurement in all terrain conditions, the foot-bot base includes three-axis accelerometers and gyroscopes. All functionality contained in the base module is managed by three local dsPIC microcontrollers.

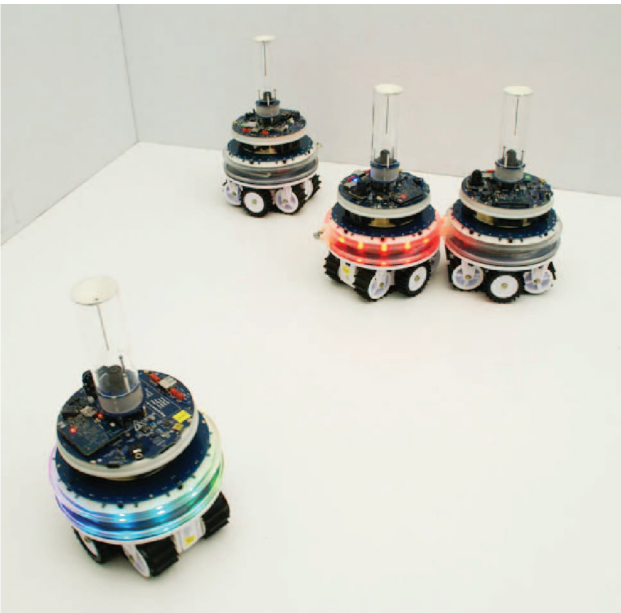


Figure 3. The foot-bot robotic platform. The foot-bot has a differential drive system that uses a combination of tracks and wheels to provide mobility on rough terrain. Two of the foot-bots in this figure have illuminated their LED communication ring. These red–green–blue (RGB) colored signals are detectable by the onboard cameras of other foot-bots.

The gripper module is stacked above the base module and enables self-assembly either among foot-bots or between foot-bots and hand-bots. Self-assembly is achieved through a docking ring and a gripping mechanism with complementary shapes. The shape of the docking ring physically guides the gripper into place, thus providing passive vertical alignment. The entire gripper module can be rotated around the foot-bot, thus providing active horizontal positioning. A two-dimensional (2-D) force sensor allows the foot-bot to measure the effort applied on the docking ring. This traction sensor is placed between the main structure of the foot-bot body and the docking ring. Additionally, the module contains red–green–blue (RGB) LEDs enclosed inside the docking ring, which can be used for color-based communications with other foot-bots and hand-bots. The range and bearing module contains the eponymous sensing and communication device common to all the robots of the swarmanoid. It is very simple mechanically but has complex analog electronics. The distance scanner module is based on four IR distance sensors mounted on a rotating platform. We coupled two sensors of different ranges (40–300 mm and 200–1,500 mm) to cover both short and long distances. The platform rotates continuously to make 360° scans. To minimize the wear and maximize the life of the scanner, the fixed part transfers energy by induction to the rotating part, and the rotating and fixed parts of the module exchange data using IR light. Finally, the upper module includes the cameras, an LED beacon, the i.MX31 ARM 11 processor, and its peripherals, including a Wi-Fi board and a flash card reader. Two cameras are available: a top/front camera and an omnidirectional camera.

The foot-bot design solves many issues that we experienced in our previous research on the s-bot—the precursor robot of the foot-bot that was developed in the swarm-bots project [10]–[12]. The foot-bot is a much more stable platform. Its slightly increased size (in comparison with the s-bot) and modular design allowed us to develop stronger and higher-quality components. The energy autonomy of the foot-bot has been improved thanks to new battery technology and to the hot-swap mechanism, thus removing battery life as a constraint on the duration of experiments. The novel modular design ensures flexibility of the system, which can be extended simply by adding new components. For instance, new sensor modules can be easily plugged in without any redesign. In summary, the foot-bot is an excellent tool for swarm robotics experimentation, as it features enhanced autonomy, short- and long-range perception, robot–robot and robot–environment interaction, self-assembling abilities, and a rich set of devices for sensing and communication. These features are not currently found in any other collective robot platform (see [13]–[20]).

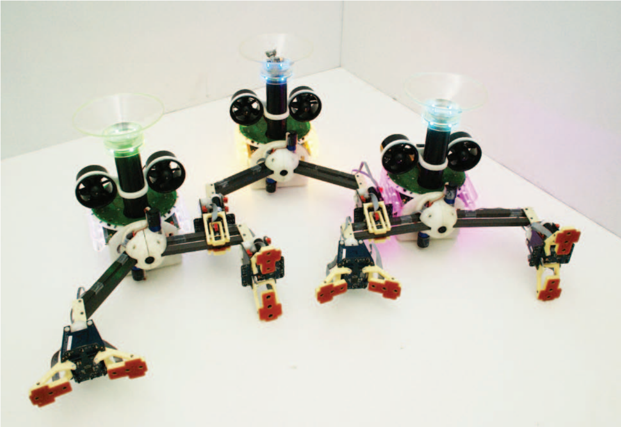


Figure 4. Three hand-bots assembled together. The hand-bot is an autonomous robot capable of climbing vertical structures and manipulating objects. The grasping hands provide basic manipulation abilities. They also enable hand-bots to physically connect to one another, thus forming larger composite robotic entities.

Hand-Bot

The hand-bot has no autonomous mobility on the ground, but it is able to climb vertical structures, grasp small objects such as books or files, and bring such objects to the ground. For the swarmanoid to transport an object, the hand-bot can grasp the object while itself being transported by the foot-bots. The hand-bot can thus interact physically with other robots of the swarmanoid.

There are many existing climbing robots, which rely on various techniques to implement the climbing mechanism. For a recently published overview of existing climbing systems, see [21]. In designing the hand-bot, we considered magnetic attachment systems, grasping hands, suction pads, dry adhesion mechanisms, and mechanisms based on some external aid, such as a rope or a pole. Our design combines grasping hands with a climbing-assistance device that uses a rope launcher and a magnetic attachment system. The grasping hands give the robot precise control of its vertical climbing path, whereas the climbing-assistance device provides most of the force required for vertical movement. The rope can be launched from the hand-bot to attach to the desired position on the ceiling. For multiple launches, the hand-bot can actively detach and retrieve the rope before recharging the system in preparation for the next launch. The grasping hands can also manipulate objects (see Figure 4). The hand-bot is 29-cm high, 41-cm wide in its widest configuration (with its arms fully retracted) and 47-cm long in its longest configuration (with its arms fully extended).

The rope launcher and the magnetic system modules are the most challenging parts of the robot design because of the complexity of the modules and the robustness required by their operation. The attachment system includes the magnet for attaching to ferromagnetic ceilings, a motor to switch the magnetic field and cancel the attachment force, a processor controlling the system, an IR receiver to get commands from the hand-bot, and super-capacitors to store energy. The whole system requires 1.4 mA for standby power supply and can survive powered on for 35 min. When switched on, the magnet can provide a vertical force of 140 N [22]. The upper part of the launcher contains RGB LEDs that can be used for signaling between robots. Two fan propellers attached to the launcher provide the hand-bot with orientation and limited position control while suspended on the rope.

The main body of the hand-bot protects the launcher mechanisms and hosts a number of devices. In the front section, a high-resolution camera looks forward toward the area accessible by the grasping hands. The battery—identical to that of the foot-bot—is housed within the main body, as is the range and bearing system and the docking ring. The hand-bot's range and bearing system and docking ring are identical in functionality to those of the foot-bot but have been modified to fit the shape of the hand-bot. Around the main body, the docking ring allows connections from foot-bots. The ring contains 12 RGB LEDs for visual signaling. Finally, the hand-bot features two identical arms, which provide climbing and manipulation abilities. The arms are parallelogram-based structures that ensure that the two grippers are aligned with the body. The two arms are mounted symmetrically on the central rotating system—the head—and provide one independent and one coupled degree of freedom to each gripper, a total of three degrees of freedom. Each grasping hand contains an embedded low-resolution color VGA camera and 12 distance sensors, which can be used together to locate and grasp objects in the environment. The gripper was designed to support the weight of the robot when the arms are in a vertical position. This implies a high grasping force of 25 N. The gripper can also rotate with a load of 2 N (e.g., the weight of a book). Furthermore, the gripper is able to grasp the arms of other hand-bots, thus enabling the formation of a physically connected structure, as shown in Figure 4. By collectively launching their attachment systems to the ceiling, assembled hand-bots can climb and control their position in 3-D space (for more details, see [23]).

In summary, the hand-bot is a compact robot dedicated to climbing and manipulation scenarios. At the electronic level, the robot has an architecture identical to the foot-bot and shares most of the basic components. It is similarly modular and also supports the ASEBA architecture. Many components are shared with the foot-bot and eye-bot, such as the i.MX31 processor board, the motor controllers, the range and bearing system, and the battery.

Eye-Bot

The eye-bot is an autonomous flying robot designed to operate in indoor environments (see Figure 5). The eye-bots work in synergy with the rest of the swarmanoid: they provide an aerial view to detect objects of interest and the actions of other robot types. The size of an eye-bot has been optimized to obtain a platform small enough to fly in a large room without interfering with other platforms and capable of flying in narrow corridors to explore the environment. Innovative methods have been employed to dramatically increase mission endurance. For example, the eye-bot features a ceiling attachment system that enables an energy-saving operation mode in which the eye-bot can power down its flight systems while continuing to scan the environment and communicate with the rest of the swarmanoid.

The eye-bot was designed around an advanced quadrotor structure, which allowed us to reduce the size of the robot without sacrificing payload capability or flight endurance. Recent advances have permitted the stable control of small hover-capable robots like quadrotors [24]. However, although altitude stability is feasible, hovering robots usually suffer from drift. Platform drift is an unavoidable result of imbalances in the rotor blades, differing air-flow over the airframe, and turbulence from down-wash or external forces such as wind. This drift is commonly compensated for with absolute positioning. In outdoor systems, absolute positioning usually relies on a global positioning system (GPS). Absolute positioning indoors has been implemented using color vision cameras [25] or IR 3-D motion tracking cameras, e.g., the Vicon system [26]. Such tracking systems provide high-accuracy measurements of position and altitude at fast refresh rates (1–5 mm at 200 Hz), allowing the control of a small aircraft in highly dynamic maneuvers such as multiflip trajectories [26]. However, this approach requires an environment with preinstalled sensory infrastructure. This kind of preinstallation is not feasible in many real-world scenarios.

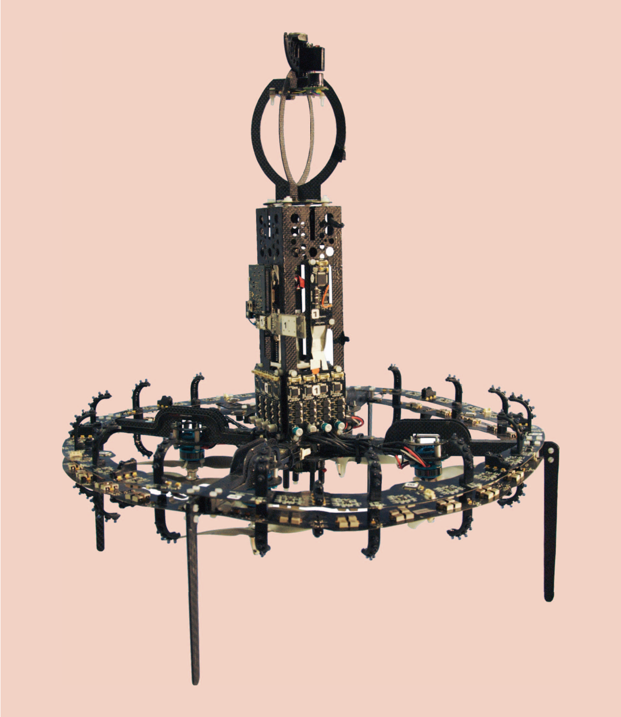


Figure 5. The eyebot platform. The eyebot is an autonomous flying robot suited to collective exploration of indoor environments. Eye-bots stay in contact with each other using the range and bearing communication device. The ceiling attachment device allows eye-bots to extend their energy autonomy by powering down their flight systems.

Common approaches to autonomous flight with onboard sensors exploit either laser scanners or visual processing [27], [28]. Laser scanners are heavy and computationally expensive, while vision-based approaches are highly dependent on the available ambient illumination, which may be insufficient or unpredictable in many situations. Similar problems affect optic-flow approaches, which require significant environment texture and contrast [29]. In summary, previous approaches have many limitations and only function in certain environments.

In contrast, the eye-bots are collectively capable of autonomous flight without any of these limitations. Flying eye-bots can maneuver using sensory information from other static eye-bots, communicated over the onboard range and bearing communication system. By having at least one eye-bot attached to the ceiling that provides a static reference point, it is possible to control the unknown egomotions and the platform drift. A cooperating network of eye-bots attached to the ceiling [30] thus enables indoor navigation while avoiding the use of absolute positioning systems, such as GPS, 3-D tracking cameras (which require preinstallation), illumination-dependent visual processing, or computationally expensive laser scanning.

The eye-bot uses a quadrotor-like propulsion configuration but with a 4 × 2 coaxial rotor system. Each rotor system consists of a coaxial counter-rotating brushless motor (Himax Outrunner HC2805-1430), which provides 500 g thrust at 9 V (750 g at 12 V). This gives a total platform thrust of at least 2,000 g, sufficient to lift the payload for the advanced sensory-motor systems. The main body has a carbon fiber structure and houses the batteries and the main PCBs, such as the flight computer and the i.MX31 ARM 11 processor. Attached to the bottom of the body structure is the propulsion system, which consists of four carbon fiber arms that support the motors, the rotary systems, and the range and bearing module. On top of the eye-bot resides the ceiling attachment mechanism. Finally, the eye-bot has four carbon fiber legs for support. These legs also protect the rotors and the delicate pan-tilt camera system. In total, the carbon fiber structure weighs only 270 g. The outer diameter is 50 cm, and the total height, including the legs and ceiling attachment, is 54 cm.

As mentioned above, the eye-bot is reliant on the range and bearing communication device. This communication system allows an eye-bot to communicate with other eye-bots, to coordinate movements in 3-D, and to facilitate controlled flight without platform drift. The system is fully compatible with the similar devices developed for the foot-bot and the hand-bot and permits bidirectional communication between the different robotic platforms. The system mounted on the eye-bots provides the range and bearing of robots within 12 m as well as low-bandwidth local communication.

Inter-robot communication can also take place via color-based visual signals. An array of RGB LEDs around the perimeter of the eye-bot can be illuminated in different color patterns. To view the color LED rings of other robots and to detect target objects of interest, the eye-bots are equipped with a high-resolution color CMOS camera mounted on a two-axis pan-tilt mechanism. This allows the eye-bot to have high-resolution imaging in the volume of space beneath the eye-bot. The same pan-tilt mechanism additionally holds a 5-mW Class IIIA laser pointer. This laser can be pointed in any direction beneath the eye-bot.

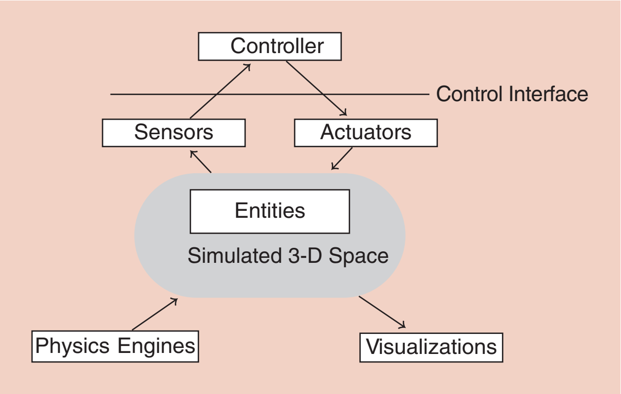


Figure 6. The architecture of the ARGoS simulator.

Simulation

ARGoS is a new simulator that we designed and implemented in-house to simulate the swarmanoid robots and to enable fast prototyping and testing of robot controllers. ARGoS is unique in that it offers high scalability without sacrificing flexibility.

In traditional simulator designs, such as those of Webots [31], USARSim [32], and Gazebo [33], accuracy is the main driver and is achieved at the cost of limited scalability. Simulators designed for scalability, such as Stage [34], are focused on very specific application scenarios, thus lacking flexibility. To achieve both scalability and flexibility, in the design of ARGoS we made a number of innovative choices.

ARGoS' architecture is depicted in Figure 6. Its core, the simulated space, contains all the data about the current state of the simulation. Such data are organized into sets of entities of different types. Each entity type stores a certain aspect of the simulation. For instance, positional entities contain the position and orientation of each object in the space. Entities are also organized into hierarchies. For example, the embodied entity is an extension of the positional entity that includes a bounding box. Robots are represented as composable entities, i.e., entities that can contain other entities. Each individual robot feature is stored into dedicated entity types. For instance, each robot possesses an embodied entity and a controllable entity that stores a pointer to that robot's sensors, actuators, and control code.

Our organization of data in the simulated space provides both scalability and flexibility. Scalability is achieved by organizing entities into type-specific indexes, optimized for speed. For instance, all positional entities are organized into space hashes, a simple, state-of-the-art technique to store and retrieve spatial data. Flexibility is ensured because entities are implemented as modules. In addition to the entities offered natively by ARGoS, the user can add custom modules, thus enriching ARGoS' capabilities with novel features.

Analogously, the code accessing the simulated space is organized into several modules. Each individual module can be overridden by the user whenever necessary, thus ensuring a high level of flexibility. The modules are implemented as plug-ins that are loaded at runtime.

Controllers are modules that contain control code developed by the user. Controllers interact with a robot's devices through an application programming interface (API) called the common interface. The common interface API is an abstraction layer that can make underlying calls to either a simulated or a real-world robot. In this way, controllers can be seamlessly ported from simulation to reality and back, making behavior development and its experimental validation more efficient.

Sensors and actuators are modules that implement the common interface API. Sensors read from the simulated space and actuators write to it. The optimized entity indexes ensure fast data access. For each sensor/actuator type, multiple implementations are possible, corresponding to models that differ in computational cost, accuracy, and realism. In addition, sensors and actuators are tightly coupled with robot component entities. For instance, the foot-bot wheel actuator writes into the wheel-equipped entity component of the foot-bot. Such coupling greatly enhances code reuse. New robots can be inserted by combining existing entities, and the sensors/actuators depending on them work without modification.

Visualizations read the simulated space to output a representation of it. Currently, ARGoS offers three types of visualization: 1) an interactive graphical user interface based on Qt and OpenGL, 2) a high-quality offline 3-D renderer based on POV-Ray, and 3) a textual renderer designed to interact with data analysis and plotting software such as Matlab and GNUPlot. Figure 7 shows some of the visualization possibilities of ARGoS.

One of the most distinctive features of ARGoS is that the simulated space and the physics engine are separate concepts. The link between them is the embodied entity, which is stored in the simulated space and updated by a physics engine. In ARGoS, multiple physics engines can be used simultaneously. In practice, this is obtained by assigning sets of embodied entities to different physics engines. The assignment can be done in two complementary ways: 1) manually, by binding directly an entity to an engine, or 2) automatically, by assigning a portion of space to the physics engine, so that every entity entering that portion is updated by the corresponding engine. Physics engines are another type of module. Currently, three physics engines are available: 1) a 3-D dynamics engine based on the open dynamics engine (ODE) library, 2) a 2-D dynamics engine based on the Chipmunk library, and 3) a custom-made 2-D kinematic engine.

To further enhance scalability, the architecture of ARGoS is multithreaded. The simulation loop is designed in such a way that race conditions are avoided and CPU usage is optimized. The parallelization of the calculations of sensors/actuators and of the physics engines provides high levels of scalability. The results reported in [35] show that ARGoS can simulate 10,000 simple robots 40% faster than real time. ARGoS has been released as open source software (http://iridia.ulb.ac.be/argos/) and currently runs on Linux and Mac OS X.

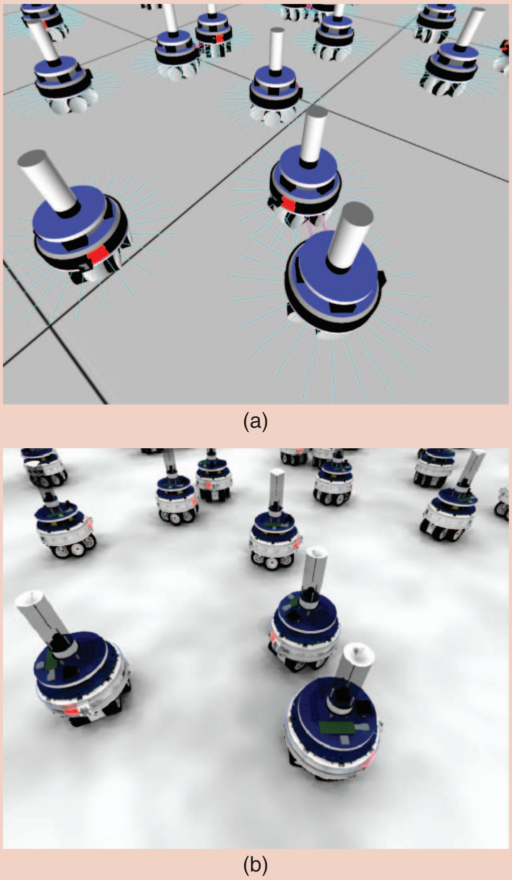
Swarmanoid in Action

Search and Retrieval: Behavioral Control

To demonstrate the potential of the swarmanoid concept, we developed an integrated search and retrieval behavior. The search-and-retrieval behavior is designed to allow the swarmanoid to retrieve objects in a complex 3-D environment. Objects are placed on one or more shelves in a human habitable space (such as an office building). The swarmanoid robots are assumed to start from a single deployment area. The swarmanoid must first find the shelves containing relevant objects and then transport the objects from the shelves back to the deployment area.

The swarmanoid search-and-retrieval behavior that we developed is shown in Figure 8. Eye-bots collectively explore the environment and search for the target location (Eye-Bot Swarm Search). They gradually build a wireless network that spans the environment by sequentially connecting to the ceiling. Each new flying eye-bot that joins the search is guided to the edge of the network by the eye-bots already in place. Having reached the edge of the network, the searching eye-bot continues flying, thus exploring new terrain. The eye-bot will, however, stop flying and attach to the ceiling when at the limit of its communication range with the rest of the network. The network remains connected using the range and bearing communication system [30].

To free up potentially scarce eye-bot resources, foot-bots incrementally form a complementary wireless network on the ground that follows the eye-bot network topology, but extends only in the most promising search directions identified by the eye-bots (Foot-Bot Chain). The eye-bot network and the foot-bot network can pass range and bearing messages between each other, and thus together act as an integrated heterogeneous exploration and communication network. As the slower foot-bot network catches up with the eye-bot network, eye-bots are freed up for further exploration. Thus the


(a)

(b)

Figure 7. Screen-shots from different visualizations: (a) Qt-OpenGL and (b) POV-Ray.

eye-bots provide a fast and systematic exploration of the environment, whereas foot-bots provide longer-term storage of exploration information on the ground. Whenever an exploring eye-bot finds a shelf containing objects, it communicates knowledge of its discovery back to the deployment area through the heterogeneous network of eye-bots and foot-bots.

The swarmanoid now needs hand-bots at the shelf location to retrieve the objects. In the deployment area, foot-bots thus assemble to hand-bots (Foot-Hand-Bot Self-Assemble) and start collectively transporting them to the shelf [36]. We refer to the composite entity formed by the foot-bots assembled to a hand-bot as a foot-hand-bot [see Figure 1(a)]. Guided by the foot-bot chain, the foot-hand-bots can navigate through the environment, following the shortest path from the deployment area to the shelf (Foot-Hand-Bot Navigate). When the foot-hand-bot arrives at a shelf location, the eye-bot that found the shelf conveys information about the 3-D location of a particular object on the shelf to the foot-hand-bot. The information tells the foot-hand-bot's constituent hand-bot where it should climb and to what height. Based on this information, the foot-hand-bot aligns itself vertically with the object it will retrieve. Then the foot-bots disassemble from the hand-bot and act as markers to subsequent foot-hand-bots letting them know not to approach the shelf at that location (Foot-Hand-Bot Align and Disassemble). The hand-bot climbs the structure, grasps the object, and descends from the shelf (Hand-Bot Retrieve Object). The foot-hand-bot then reassembles (Foot-Hand-Bot Self-Assemble) and follows the foot-bot chain back to the deployment area (Foot-Hand-Bot Navigate). Once the object has been successfully returned to the deployment area, the foot-hand-bot disassembles, and its constituent robots wait until they are needed for further swarmanoid tasks (Foot-Hand-Bot Disassemble and Reset).

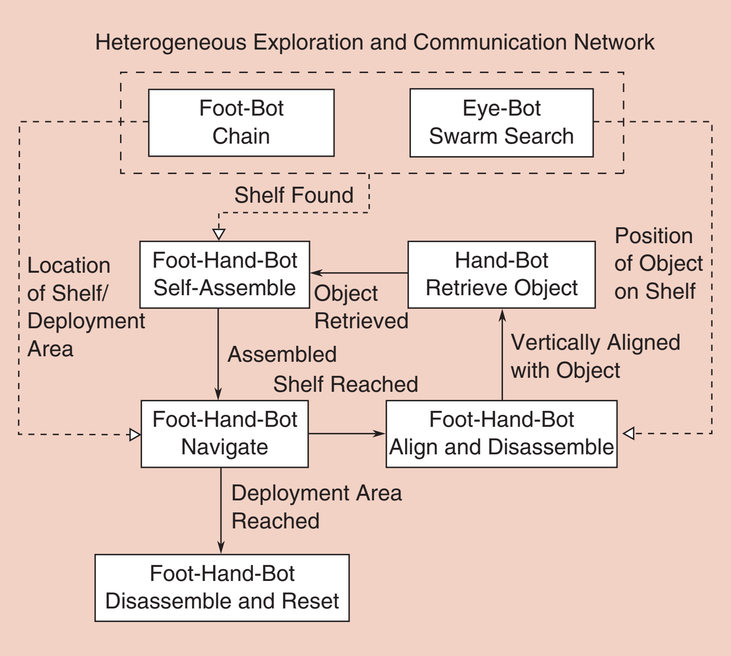

Heterogeneous Exploration and Communication Network
Foot-Bot Chain
Eye-Bot Swarm Search
Shelf Found
Location of Shelf/ Deployment Area
Foot-Hand-Bot Self-Assemble
Object Retrieved
Hand-Bot Retrieve Object
Position of Object on Shelf
Assembled
Vertically Aligned with Object
Shelf Reached
Foot-Hand-Bot Navigate
Foot-Hand-Bot Align and Disassemble
Deployment Area Reached
Foot-Hand-Bot Disassemble and Reset

Figure 8. Search-and-retrieval behavioral components. The dotted lines represent information transfer. The solid lines represent handing over of control to another behavioral component. The swarmanoid must initially be launched by executing the heterogeneous exploration and communication network behavioral components, while each execution of the foot-hand-bot disassemble and reset behavior marks the successful return of a single object to the deployment area.

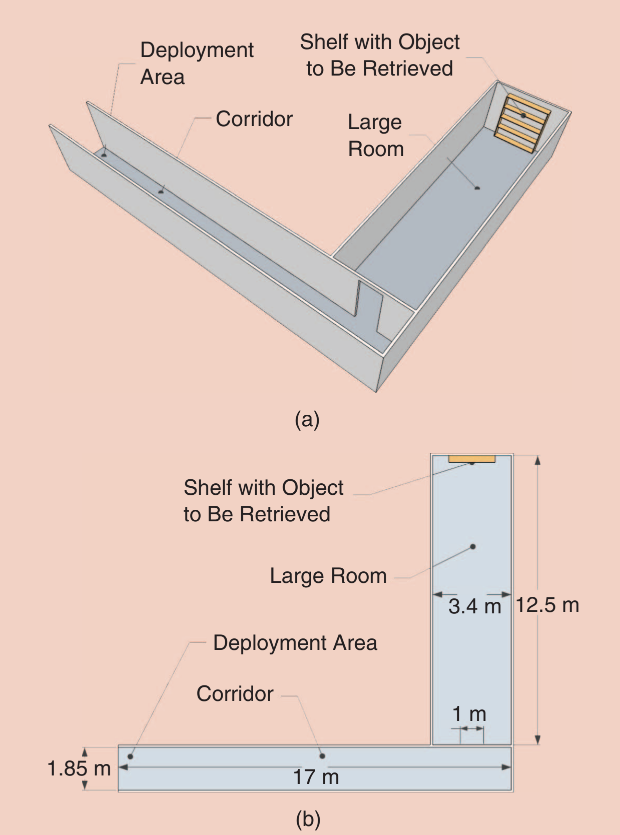


Figure 9. A drawing of the test arena for the swarmanoid demonstration. (a) The parallel projection. (b) The floorplan.

Search and Retrieval: A Real-World Demonstration

We demonstrated our integrated search-and-retrieval behavior in a real-world scenario. Our experiment involved a real-world instantiation of the generic search-and-retrieval task in an environment containing a single shelf and book. The arena we used can be seen in Figure 9. We successfully demonstrated that a swarmanoid with no a priori knowledge of the environment was able to find the shelf and retrieve the book. This scenario integrated various swarmanoid abilities, ranging from task allocation to collective search, from self-assembly to cooperative transport, from object retrieval to cooperative navigation in complex environments.

Figure 10 shows a snapshot from a video of a successful experiment. The video shown in Figure 10 is available in the supplementary electronic material on IEEE *Xplore* as well as at http://www.swarmanoid.org/swarmanoid-full-experiment-movie. A video addressed to the general public, edited together from different experiments, won the Best Video Award at the AAAI 2011 Video Competition and can be viewed at http://www.aaaivideos.org/2011/swarmanoid_the_movie.

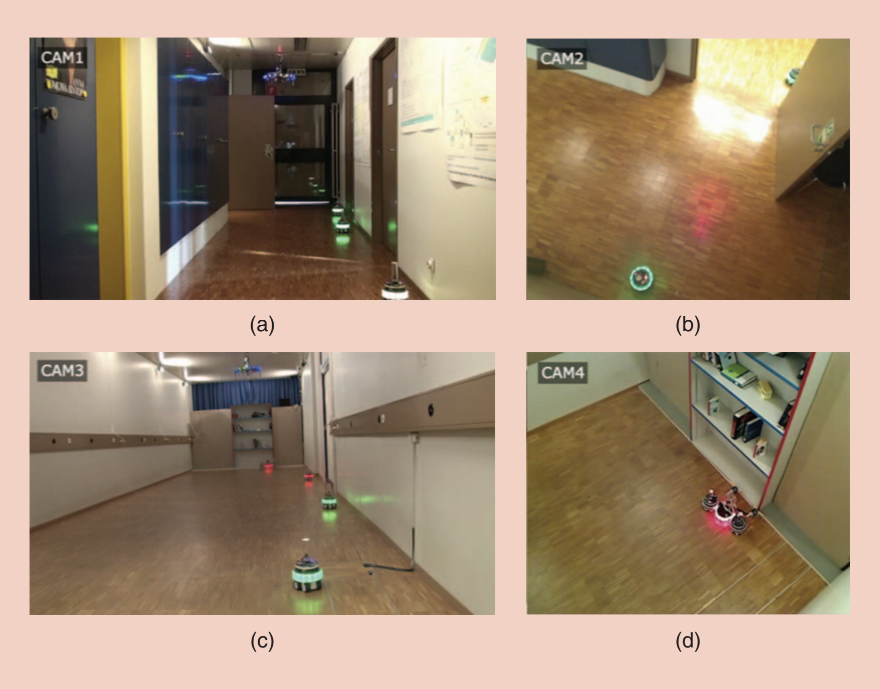


Figure 10. Snapshots from the video demonstrating the swarmanoid involved in the object retrieval experimental scenario. The video was shot from four different cameras simultaneously, giving full coverage of a single experiment. It is available in the supplementary electronic material on IEEE *Xplore* and at http://www.swarmanoid.org/swarmanoid-full-experiment-movie. (a) View from the deployment area toward the doorway. (b) View of the doorway. (c) View from the doorway toward the shelf. (d) View of the shelf.

Further Experimentation

Our success in realizing the experimental search-and-rescue scenario demonstrates the viability of the swarmanoid concept and gives a concrete example of how heterogeneous swarms can solve complex problems. The abilities of the swarmanoid are not, however, limited to the scenario we presented above. The swarmanoid can, in principle, carry out a wide range of tasks that require the parallel operation of the three robot types and that leverages the robots´ different abilities.

Within the swarmanoid framework, we have conducted many experiments, both in simulation and with real robots. The development of control algorithms for the swarmanoid followed multiple research lines. Behavior-based approaches have been employed for tasks such as recruitment, collective transport, collective exploration, etc. [37], [38]. Evolutionary robotics techniques have been used to synthesize efficient neural network controllers for behavioral synchronization and for path formation between two target areas [39], [40]. These studies demonstrate the potential of heterogeneous robotic swarms and point to a new way of tackling complex application scenarios in the real world.

Summary and Conclusions

Advancements of the state of the art in swarm robotics can be pursued by relying on heterogeneous swarm systems composed of a large number of robots presenting behavioral and/or physical heterogeneities. To this end, it is necessary to develop tools and methodologies that enable the use of such heterogeneous systems. We identified relevant issues and challenges, in particular highlighting the difficulty of delivering the tightly integrated robotic hardware necessary to enable physical and behavioral interaction between different robot types.

We presented the swarmanoid as a new robotic concept in heterogeneous swarm robotics. The hardware and software of the swarmanoid robots leveraged common technologies to ensure seamless integration of the different platforms. The resulting compatibility of different robot types enabled us to explore different coordination mechanisms and strategies in a heterogeneous swarm. The experimental scenario we defined demonstrates the suitability of the swarmanoid robotic concept for tackling complex problems in 3-D human-made environments. Future work will use the swarmanoid robotic platforms to develop a rigorous methodological approach for the design of behaviors for swarm robotics systems, especially focusing on hierarchical, heterogeneous control, and communication.

Acknowledgments

This work was supported by the Swarmanoid project, funded by the Future and Emerging Technologies program (IST-FET) of the European Commission (IST-022888) and by the European Research Council (ERC) through the ERC Advanced Grant E-SWARM: Engineering Swarm Intelligence Systems (contract 246939).

The information provided is the sole responsibility of the authors and does not reflect the European Commission’s opinion. The European Commission is not responsible for any use that might be made of data appearing in this publication.

Marco Dorigo, Mauro Birattari, Manuele Brambilla, Alexandre Campo, Rehan O’Grady, Thomas Stützle, and Vito Trianni acknowledge support from the Belgian F.R.S.-FNRS.

References

[1] E. Bonabeau, M. Dorigo, and G. Theraulaz, *Swarm Intelligence: From Natural to Artificial Systems*. New York: Oxford Univ. Press, 1999.

[2] M. Dorigo and E. Şahin, "Swarm robotics—special issue editorial," *Auton. Robot.*, vol. 17, nos. 2–3, pp. 111–113, 2004.

[3] V. Trianni, "Evolutionary swarm robotics," in *Evolving Self-Organising Behaviours in Groups of Autonomous Robots*, (Series Studies in Computational Intelligence). Berlin, Germany: Springer-Verlag, 2008, vol. 108.

[4] P. Grassé, "La reconstruction du nid et les coordinations interindividuelles chez Bellicositermes natalensis et Cubitermes sp. La théorie de la stigmergie: Essai d'interprétation du comportement des termites constructeurs," *Insectes Sociaux*, vol. 6, pp. 41–81, Mar. 1959.

[5] R. Groß and M. Dorigo, "Self-assembly at the macroscopic scale," *Proc. IEEE*, vol. 96, no. 9, pp. 1490–1508, Sept. 2008.

[6] S. Magnenat, P. Rétornaz, M. Bonani, V. Longchamp, and F. Mondada, "ASEBA: A modular architecture for event-based control of complex robots," *IEEE/ASME Trans. Mechatron.*, vol. 16, no. 2, pp. 321–329, 2011.

[7] J. Pugh and A. Martinoli, "Relative localization and communication module for small-scale multi-robot systems," in *Proc. IEEE Int. Conf. Robotics Automation*, Los Alamitos, CA, 2006, pp. 188–193.

[8] Á. Gutiérrez, A. Campo, M. Dorigo, D. A. Martín, L. Magdalena, and F. M.-H. Maciá, "An open localisation and local commnunication embodied sensor," *Sensors*, vol. 8, no. 11, pp. 7545–7563, 2008.

[9] J. Roberts, T. Stirling, J.-C. Zufferey, and D. Floreano, "2.5D infrared range and bearing system for collective robotics," in *Proc. IEEE/RSJ Int. Conf. Intelligent Robots Systems*, Los Alamitos, CA, 2009, pp. 3659–3664.

[10] F. Mondada, G. Pettinaro, A. Guignard, I. Kwee, D. Floreano, J.-L. Deneubourg, S. Nolfi, L. Gambardella, and M. Dorigo, "SWARM-BOT: A new distributed robotic concept," *Auton. Robots*, vol. 17, no. 2–3, pp. 193–221, 2004.

[11] M. Dorigo, "SWARM-BOT: An experiment in swarm robotics," in *Proc. IEEE Swarm Intelligence Symp.*, 2005, pp. 192–200.

[12] M. Dorigo, E. Tuci, V. Trianni, R. Groß, S. Nouyan, C. Ampatzis, T. Labella, R. O'Grady, M. Bonani, and F. Mondada, "SWARM-BOT: Design and implementation of colonies of self-assembling robots," in *Computational Intelligence: Principles and Practice*, G. Yen and D. Fogel, Eds. New York: IEEE Computational Intelligence Society, 2006, ch. 6, pp. 103–135.

[13] S. Kernbach, E. Meister, F. Schlachter, K. Jebens, M. Szymanski, J. Liedke, D. Laneri, L. Winkler, T. Schmickl, R. Thenius, P. Corradi, and L. Ricotti, "Symbiotic robot organisms: REPLICATOR and SYMBRION projects," in *Proc. Performance Metrics Intelligent Systems Workshop*, New York, 2008, pp. 62–69.

[14] K. Konolige, D. Fox, C. Ortiz, A. Agno, M. Eriksen, B. Limketkai, J. Ko, B. Morisset, D. Schulz, B. Stewart, and R. Vincent, "Centibots: Very large scale distributed robotic teams," in *Experimental Robotics IX*, (Series Springer Tracts in Advanced Robotics). M. H. Ang and O. Khatib, Eds. Berlin, Germany: Springer Verlag, 2006, vol. 21, pp. 131–140.

[15] A. Turgut, F. Gokce, H. Celikkanat, L. Bayindir, and E. Şahin, "Kobot: A mobile robot designed specifically for swarm robotics research," Dept. Comput. Eng., Middle East Technical Univ., Ankara, Turkey, Tech. Rep. METU-CENG-TR-2008-01, 2008.

[16] R. Rusu, G. Lazea, R. Robotin, and C. Marcu, "Towards open architectures for mobile robots: ZeeRO," in *Proc. 2006 IEEE Int. Conf. Automation, Quality Testing, Robotics,* Los Alamitos, CA, 2006, pp. 260–265.

[17] V. Zykov, P. William, N. Lassabe, and H. Lipson, "Molecubes extended: Diversifying capabilities of open-source modular robotics," in *Proc. IROS-2008 Self-Reconfigurable Robotics Workshop*, Los Alamitos, CA, 2008.

[18] P. Rybski, A. Larson, H. Veeraraghavan, M. Anderson, and M. Gini, "Performance evaluation of a multi-robot search & retrieval system: Experiences with MinDART," *J. Intell. Robot. Syst.*, vol. 52, no. 3, pp. 363–387, 2008.

[19] K. Yoo, Y. Lee, H. Choi, B. Yoo, and D. Kim, "Swarm robotics: Self assembly, physical configuration, and its control," in *Proc. SICE-ICASE, Int. Joint Conf.,* Los Alamitos, CA, 2006, pp. 4276–4279.

[20] H. Zhang, W. Wang, Z. Deng, G. Zong, and J. Zhang, "A novel reconfigurable robot for urban search and rescue," *Int. J. Adv. Robot. Syst.*, vol. 3, no. 4, pp. 359–366, 2006.

[21] B. Chu, K. Jung, C.-S. Han, and D. Hong, "A survey of climbing robots: Locomotion and adhesion," *Int. J. Precis. Eng. Manuf.*, vol. 11, no. 4, pp. 633–647, 2010.

[22] F. Rochat, P. Schoeneich, M. Bonani, S. Magnenat, F. Mondada, H. Bleuler, and H. Christoph, "Design of magnetic switchable device and applications in climbing robot," in *Emerging Trends in Mobile Robotics*, H. Fujimoto, M. O. Tokhi, H. Mochiyama, and G. S. Virk, Eds. Singapore: World Scientific, 2010, pp. 375–382.

[23] M. Bonani, P. Rétornaz, S. Magnenat, H. Bleuler, and F. Mondada, "Physical interactions in swarm robotics: the hand-bot case study," in *Proc. 10th Int. Symp. Distributed Autonomous Robotic Systems*, Berlin, Germany, 2010.

[24] E. Altug, J. Ostrowski, and C. Taylor, "Control of a quadrotor helicopter using dual camera visual feedback," *Int. J. Robot. Res.*, vol. 24, no. 5, pp. 329–341, 2005.

[25] G. Hoffmann, S. Waslander, and C. Tomlin, "Quadrotor helicopter trajectory tracking control," in *Proc. AIAA Guidance, Navigation, Control Conf.*, Reston, VA, 2008, pp. 1–14.

[26] S. Lupashin, A. Schöllig, M. Sherback, and R. D'Andrea, "A simple learning strategy for high-speed quadrocopter multi-flips," in *Proc. IEEE Int. Conf. Robotics Automation*, Los Alamitos, CA, 2010, pp. 1642–1648.

[27] F. Kendoul, K. Nonami, I. Fantoni, and R. Lozano, "An adaptive vision-based autopilot for mini flying machines guidance, navigation and control," *Auton. Robot.*, vol. 27, no. 3, pp. 165–188, 2009.

[28] N. Guenard, T. Hamel, and R. Mahony, "A practical visual servo control for a unmanned aerial vehicle," *IEEE Trans. Robot.,* vol. 24, no. 2, pp. 331–341, 2008.

[29] A. Beyeler, J.-C. Zufferey, and D. Floreano, "3D vision based navigation for indoor microflyers," in *Proc. IEEE Int. Conf. Robotics Automation*, Los Alamitos, CA, 2007, pp. 1336–1341.

[30] T. Stirling, S. Wischmann, and D. Floreano, "Energy-efficient indoor search by swarms of simulated flying robots without global information," *Swarm Intell.*, vol. 4, no. 2, pp. 117–143, 2010.

[31] O. Michel, "Cyberbotics Ltd.—Webots: Professional mobile robot simulation," *Int. J. Adv. Robot. Syst.*, vol. 1, no. 1, pp. 39–42, Mar. 2004.

[32] S. Carpin, M. Lewis, J. Wang, S. Balakirsky, and C. Scrapper, "USARSim: A robot simulator for research and education," in *Proc. IEEE Conf. Robotics Automation*, Piscataway, NJ, 2007, pp. 1400–1405.

[33] N. Koenig and A. Howard, "Design and use paradigms for Gazebo, an open-source multi-robot simulator," in *Proc. IEEE/RSJ Int. Conf. Intelligent Robots Systems*, Piscataway, NJ, 2004, pp. 2149–2154.

[34] R. Vaughan, "Massively multi-robot simulation in Stage," *Swarm Intell.*, vol. 2, no. 2, pp. 189–208, 2008.

[35] C. Pinciroli, V. Trianni, R. O'Grady, G. Pini, A. Brutschy, M. Brambilla, N. Mathews, E. Ferrante, G. A. Di Caro, F. Ducatelle, M. Birattari, L. M. Gambardella, and M. Dorigo, "ARGoS: A modular, parallel, multi-engine simulator for multi-robot systems," in *Swarm Intell.,* vol. 6, no. 4, pp. 271–295, 2012.

[36] E. Ferrante, M. Brambilla, M. Birattari, and M. Dorigo, "Socially mediated negotiation for obstacle avoidance in collective transport," in *Proc. 10th Int. Symp. Distributed Autonomous Robotic Systems,* Berlin, 2010.

[37] C. Pinciroli, R. O'Grady, A. Christensen, and M. Dorigo, "Self-organised recruitment in a heterogeneous swarm," in *Proc. Int. Conf. Advanced Robotics*, Los Alamitos, CA, 2009, pp. 1–8.

[38] F. Ducatelle, G. Di Caro, C. Pinciroli, and L. Gambardella, "Self-organized cooperation between robotic swarms," *Swarm Intell.*, vol. 5, no. 2, pp. 73–96, 2011.

[39] V. Trianni and S. Nolfi, "Self-organising sync in a robotic swarm: A dynamical system view," *IEEE Trans. Evol. Comput.,* vol. 13, no. 4, pp. 722–741, 2009.

[40] V. Sperati, V. Trianni, and S. Nolfi, "Self-organised path formation in a swarm of robots," *Swarm Intell.*, vol. 5, no. 2, pp. 97–119, 2011.

Marco Dorigo, IRIDIA, Université Libre de Bruxelles, Brussels, Belgium. E-mail: mdorigo@ulb.ac.be.

Dario Floreano, LIS, École Polytechnique Federal, Lausanne, Switzerland. E-mail: dario.floreano@epfl.ch.

Luca Maria Gambardella, IDSIA, USI-SUPSI. E-mail: luca@idsia.ch.

Francesco Mondada, LSRO, École Polytechnique Fédérale de Lausanne, Switzerland. E-mail: francesco.mondada@epfl.ch.

Stefano Nolfi, ISTC, Consiglio Nazionale delle Ricerche, Italy. E-mail: stefano.nolfi@istc.cnr.it.

Tarek Baaboura, LSRO, École Polytechnique Fédérale de Lausanne, Switzerland. E-mail: djmrhyde@hotmail.com.

Mauro Birattari, IRIDIA, Université Libre de Bruxelles, Brussels, Belgium. E-mail: mbiro@ulb.ac.be.

Michael Bonani, LSRO, École Polytechnique Fédérale de Lausanne, Switzerland. E-mail: michael.bonani@epfl.ch.

Manuele Brambilla, IRIDIA, Université Libre de Bruxelles, Brussels, Belgium. E-mail: manu.brambi@gmail.com.

Arne Brutschy, IRIDIA, Université Libre de Bruxelles, Brussels, Belgium. E-mail: abrutschy@ulb.ac.be.

Daniel Burnier, LSRO, École Polytechnique Fédérale de Lausanne, Switzerland. E-mail: daniel.burnier@epfl.ch.

Alexandre Campo, IRIDIA, Université Libre de Bruxelles, Brussels, Belgium. E-mail: alexandre.campo@ulb.ac.be.

Anders Lyhne Christensen, IRIDIA, Université Libre de Bruxelles, Brussels, Belgium, and Instituto de Telecomunicações and Instituto Universitário de Lisboa (ISCTE-IUL), Lisbon, Portugal. E-mail: anderslyhne@gmail.com.

Antal Decugnière, IRIDIA, Université Libre de Bruxelles, Brussels, Belgium. E-mail: antal.decugniere@gmail.com.

Gianni Di Caro, IDSIA, USI-SUPSI, Lugano, Switzerland. E-mail: gianni@idsia.ch.

Frederick Ducatelle, IDSIA, USI-SUPSI, Lugano, Switzerland. E-mail: frederick@idsia.ch.

Eliseo Ferrante, IRIDIA, Université Libre de Bruxelles, Brussels, Belgium. E-mail: eliseofe@gmail.com.

Alexander Förster, IDSIA, USI-SUPSI, Lugano, Switzerland. E-mail: alexander@idsia.ch.

Jerome Guzzi, IDSIA, USI-SUPSI, Lugano, Switzerland. E-mail: jerome@idsia.ch.

Valentin Longchamp, LSRO, École Polytechnique Fédérale de Lausanne, Switzerland. E-mail: valentin.longchamp@gmail.com.

Stéphane Magnenat, LSRO, École Polytechnique Fédérale de Lausanne, Switzerland. E-mail: stephane.magnenat@mavt.ethz.ch.

Javier Martinez Gonzales, IRIDIA, Université Libre de Bruxelles, Brussels, Belgium. E-mail: xsmarty@gmail.com.

Nithin Mathews, IRIDIA, Université Libre de Bruxelles, Brussels, Belgium. E-mail: nmathews@ulb.ac.be.

Marco Montes de Oca, IRIDIA, Université Libre de Bruxelles, Brussels, Belgium. E-mail: mmontes@iridia.ulb.ac.be.

Rehan O'Grady, IRIDIA, Université Libre de Bruxelles, Brussels, Belgium. E-mail: rehanog@gmail.com.

Carlo Pinciroli, IRIDIA, Université Libre de Bruxelles, Brussels, Belgium. E-mail: cpinciro@ulb.ac.be.

Giovanni Pini, IRIDIA, Université Libre de Bruxelles, Brussels, Belgium. E-mail: gpini@ulb.ac.be.

Philippe Rétornaz, LSRO, École Polytechnique Fédérale de Lausanne, Switzerland. E-mail: philippe.retornaz@epfl.ch.

James Roberts, LIS, École Polytechnique Fédérale de Lausanne, Switzerland. E-mail: James.Roberts@sensefly.com.

Valerio Sperati, ISTC, Consiglio Nazionale delle Ricerche, Milan, Italy. E-mail: valerio.sperati@istc.cnr.it.

Timothy Stirling, LIS, École Polytechnique Fédérale de Lausanne, Switzerland. E-mail: timothy.stirling@gmail.com.

Alessandro Stranieri, IRIDIA, Université Libre de Bruxelles, Brussels, Belgium. E-mail: alessandro.stranieri@ulb.ac.be.

Thomas Stützle, IRIDIA, Université Libre de Bruxelles, Brussels, Belgium. E-mail: stuetzle@ulb.ac.be.

Vito Trianni, IRIDIA, Université Libre de Bruxelles, Brussels, Belgium, and ISTC, Consiglio Nazionale delle Ricerche, Italy. E-mail: vtrianni@ulb.ac.be.

Elio Tuci, ISTC, Consiglio Nazionale delle Ricerche, Italy. E-mail: elio.tuci@istc.cnr.it.

Ali Emre Turgut, THK University, Ankara, Turkey. E-mail: ali.turgut@gmail.com.

Florian Vaussard, LSRO, École Polytechnique Fédérale de Lausanne, Switzerland. E-mail: florian.vaussard@epfl.ch.