The MarXbot, a Miniature Mobile Robot Opening new Perspectives for the Collective-robotic Research

9 authors, including:

Michael Bonani
École Polytechnique Fédérale de Lausanne
41 PUBLICATIONS   **2,076** CITATIONS

SEE PROFILE

Philippe Retornaz
École Polytechnique Fédérale de Lausanne
33 PUBLICATIONS   **1,013** CITATIONS

SEE PROFILE

Daniel Burnier
École Polytechnique Fédérale de Lausanne
10 PUBLICATIONS   **774** CITATIONS

SEE PROFILE

Florian Vaussard
École Polytechnique Fédérale de Lausanne
13 PUBLICATIONS   **619** CITATIONS

SEE PROFILE

Some of the authors of this publication are also working on these related projects:

Project   thymio education robot View project

Project   AZIMUT-3 View project

The MarXbot, a Miniature Mobile Robot Opening new Perspectives for the Collective-robotic Research

Michael Bonani, Valentin Longchamp, Stéphane Magnenat, Philippe Rétornaz, Daniel Burnier, Gilles Roulet, Florian Vaussard, Hannes Bleuler, and Francesco Mondada

Abstract— Collective and swarm robotics explores scenarios involving many robots running at the same time. A good platform for collective-robotic experiments should provide certain features among others: it should have a large battery life, it should be able to perceive its peers, and it should be capable of interacting with them. This paper presents the marXbot, a miniature mobile robot that addresses these needs. The marXbot uses differential-drive treels to provide rough-terrain mobility. The marXbot allows continuous experiments thanks to a sophisticated energy management and a *hotswap* battery exchange mechanism. The marXbot can self-assemble with peers using a compliant attachment mechanism. The marXbot provides high-quality vision, using two cameras directly interfaced with an ARM processor. Compared to the related work, the marXbot has better energy management, vision, and interaction capabilities. By allowing complex tasks in large environments for long durations, the marXbot opens new perspectives for the collective-robotic research.

I. INTRODUCTION

Collective and swarm robotics explores scenarios involving many robots running at the same time. This sets constraints to the robotic platforms. To be usable in laboratories, these robots must be small and affordable. Moreover, to collaborate within a group, a robot must provide several specific features:

First, it must have a large battery life, and, if possible, it must be able to recharge itself. Indeed, it is not easy to manage a group of robots and ensure that all robots start at the same time. Thus if the first robot is mostly out of battery when the last is finally ready, the experimenter is in trouble. The likelihood of the situation grows according to the number of robots involved in the experiment.

Second, a robot must be able to perceive its peers. There are multiple ways to do so, but because of its range and of the amount of information it provides, vision is a candidate of choice. Moreover, vision is a passive sensor which avoids interferences. Typically, in a collective setup, omnidirectional vision is interesting because it directly gives information about the surrounding of the robot. In this context, it is desirable that the robot is both able to see near, for example to do visual servoing to interact with its peers, and to see far, to locate distant robots or targets.

All authors are with EPFL-LSRO, Station 9, 1015 Lausanne, Switzerland - http://mobots.epfl.ch. Please send correspondence to francesco.mondada@epfl.ch.

Finally, a robot should be capable of interacting with its peers or with the environment. Recently, in the field of swarm robotics, several works have emphasized the ability to physically assemble with peers, see for instance [1]. When doing so, it is interesting that the assembling process is as compliant as possible, to simplify the control and to improve reliability.

This paper presents the marXbot, a miniature mobile robot that we have developed. We have built the marXbot upon our experience with the S-bot [2], and improved several aspects that are critical for enabling certain collective experiments, as shown by [3]. In this paper, we show the design of these aspects and give experimental results that validate the improvements, compared to the related work and to the S-bot.

II. RELATED WORK

TABLE I shows a selection of miniature robots that have been used recently for collective experiments. We have decided to highlight the capabilities of autonomy (battery life and mobility), long range perception, and interaction (robot-robot and robot-environment).

We see that the robots use various battery technologies. These range from simple AA batteries to advanced Lithium-based batteries. Most robots target an autonomy of 2 to 4 h. None of the surveyed robots have a way to automatically exchange their battery, nor do they provide facilities to self-recharge. Robots that evolve on flat surfaces have a differential-drive with wheels. However, robots that tackle more realistic environments usually have tracks.

At the level of the perception, most robots use vision, either using directional or omnidirectional cameras. However, not all the robots embed the computational power to process the images onboard. Some thus must transmit their images remotely, or use additional hardware to support vision. Some robots use additional sensors for long-range perception, such as ultrasound or triangulation-based infrared.

About half of the reviewed robots have physical interaction capabilities, either a gripper to interact with the environment or a attachment device to self-assemble, or both.

Despite being 6 years old, with respect to autonomy, perception, and interaction, the S-bot is still close to the state of the art. In the marXbot, we have pushed these aspects further to provide a better platform for collective experiments.

III. OVERVIEW OF THE MARXBOT

The marXbot is a modular miniature mobile robot (see Fig. 1). A base module provides rough-terrain mobility thanks

robot/author	size	battery	mobility	perception	interaction	comm.	processing	ref.
Jasmine	2.6×2.6×2.6 cm	LiPo, 2 h autonomy	wheels	none	none	radio	none	[4]
AmigoBot	33×28×15 cm	Pb, 26 Wh, 2 h autonomy	wheels	ultrasound, opt. vision	none	opt. radio	ad hoc	[5]
Kobot	∅12×7 cm	LiPo, 7 Wh, 10 h autonomy	wheels	opt. omnicam	none	Xbee	opt. PXA255	[6]
Zeero	∅≈25 cm	4×AA, 9 Wh	wheels	pan-tilt CMUcam2, ultrasound, IR	none	Bluetooth	PXA255	[7]
FlockBots	∅18 cm	NiMH, 16 Wh, 2 h autonomy	wheels	pan-tilt CMUCam2, IR	simple gripper	Wi-Fi	PXA255	[8]
Molecubes	66×66×66 cm	16 Wh 1 h autonomy	opt. wheels	opt. vision	assembling, gripper	opt. Bluetooth	opt. ARM 11	[9]
Mindart	29×24×37 cm	NiCad, 20 Wh	tracks	beacon & vision	gripper	none	Scenix SX	[10]
Yoo, K.H. et al.	n.a.	n.a.	tracks	vision	self-assembling	RF	off-board	[11]
JL-1	35×25×15 cm	4 h autonomy	tracks	vision	self-assembling	Wi-Fi	PXA255	[12]
S-bot	∅12×15 cm	LiIon, 10 Wh, 2 h autonomy	treels	omnicam	gripper, self-assembling	Wi-Fi	PXA255	[2]

TABLE I: A selection of robots that have been used recently for collective experiments. The *perception* column lists the long range (> 20 cm) sensing capabilities of the robot, which excludes proximity sensors and bumpers. The *processing* column lists the vision-capable processing unit of the robot (> 100 MIPS), which excludes microcontrollers.

to treels, a combination of tracks and wheels already present on the S-bot [2]. This base also provides energy to the rest of the robot thanks to its hot-swappable battery (Sec. V). In addition, this module also contains the basic bricks for obstacle avoidance and odometry, as it embeds proximity sensors, a 3-axis gyroscope, and a 3D accelerometer. An attachment module provides self-assembling capabilities with peer marXbots [13] (Sec. VI). This module allows the docking of the other robots and can feel the force they apply. A range and bearing module provides a new capability compared to most of the related works. This module allows the robot to compute a rough estimate of the direction and the distance of the neighboring robots. This innovation enables many interesting swarm behaviors [14]. A distance scanner module allows the robot to build a 2D map of its environment [15]. Finally, a main computer module provides a complete Linux-based operating system to the robot. This module thus enables advanced cognitive capabilities [16]. The main computer also drives two cameras, one looking front and one oriented towards an omnidirectional hyperbolic mirror (Sec. VII).

IV. DISTRIBUTED CONTROL

Compact miniature mobile robots embed many sensors and actuators. To limit the wiring, these hardware devices are managed by a network of microcontrollers distributed throughout the robot. Typically, robots from TABLE I, such as the S-bot, employ microcontrollers connected through an I^2C bus [2]. Often, an ARM-based main computer running Linux polls the microcontrollers at regular intervals to read the sensor values and to set the actuator commands. These read and write operations all have to transit through the I^2C bus. These robots thus suffer from bus overloading and excessive latency to external stimuli, which limits their performances.

To solve these problems, on the marXbot, we have replaced the I^2C bus with a CAN bus [17]. Moreover, as microcontrollers we have used dsPICs, which are faster and have a larger memory than usual microcontrollers. Based on this hardware, we have developed an event-based low-level control architecture, called ASEBA [18]. Using the asynchronous communication capabilities of the CAN bus, the microcontrollers now communicate through events. The event emission policy and the microcontrollers' behaviors are controlled by code running inside a virtual machines on the dsPICs. The virtual machine allows safe execution and rapid code update. Running user-defined code close to the hardware provides a fast reactivity to environmental stimuli, which improves the robot's performances [18]. Moreover, this allows to exploit the peripheral processing power to filter raw data and thus to offload the main computer. The ARM-based main computer is also connected to the CAN bus and communicate with the microcontrollers through events. This enables the integration of a large number of peripherals inside the robot; in its nominal configuration, the marXbot has 5 modules for a total of 10 microcontrollers, managing 5 motors, 41 LEDs and 80 sensors.

V. BATTERY AND ENERGY MANAGEMENT

The study of swarm and collective behaviors in reality requires many robots running at the same time. Existing robots do not fulfill optimally this requirement. For instance, the S-bot integrated a Li-Ion battery in its chassis, providing energy for 1 to 2 hours, depending on the task. This battery took 2 hours to recharge, during which the robot had to stay idle. Moreover, as the battery was molded inside the chassis, it was impossible to replace, which lead to disparities of running time with the aging of the robots. In overall, the availability of a group of S-bots for an experiment was less than 40 %.
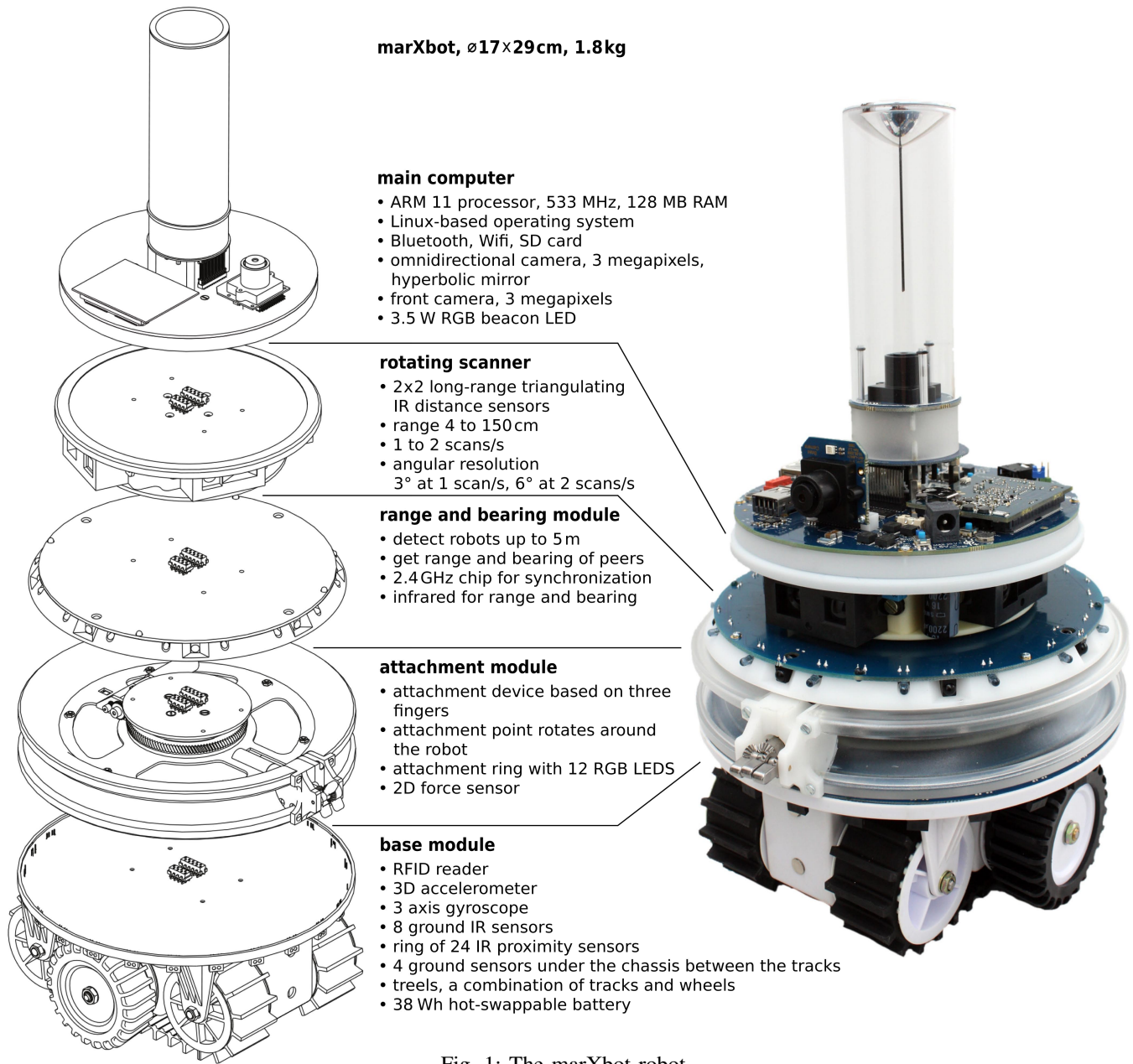
marXbot, ⌀17×29cm, 1.8kg

main computer
- ARM 11 processor, 533 MHz, 128 MB RAM
- Linux-based operating system
- Bluetooth, Wifi, SD card
- omnidirectional camera, 3 megapixels, hyperbolic mirror
- front camera, 3 megapixels
- 3.5 W RGB beacon LED

rotating scanner
- 2x2 long-range triangulating IR distance sensors
- range 4 to 150 cm
- 1 to 2 scans/s
- angular resolution 3° at 1 scan/s, 6° at 2 scans/s

range and bearing module
- detect robots up to 5 m
- get range and bearing of peers
- 2.4 GHz chip for synchronization
- infrared for range and bearing

attachment module
- attachment device based on three fingers
- attachment point rotates around the robot
- attachment ring with 12 RGB LEDS
- 2D force sensor

base module
- RFID reader
- 3D accelerometer
- 3 axis gyroscope
- 8 ground IR sensors
- ring of 24 IR proximity sensors
- 4 ground sensors under the chassis between the tracks
- treels, a combination of tracks and wheels
- 38 Wh hot-swappable battery

Fig. 1: The marXbot robot.

In the marXbot, we have risen the availability of the robot compared to the state of the art by allowing the robot to replace its battery by itself. This resulted in the concept of *hot swapping* the battery. To maintain the robot's vital functions during a swap, the base embeds two large capacitors (10 F). These charge from the battery when it is present, and when it is removed, a specific electronics swaps the power supply of critical systems to the capacitors. The critical systems are mainly the main computer and its wireless-network device. The capacitors provide energy for more than 15 s, which is long enough to swap the battery. During this period, the other systems such as the motors are switched off.

We designed the mechanics of the base module around the hot-swapping capability. In particular, an external device can
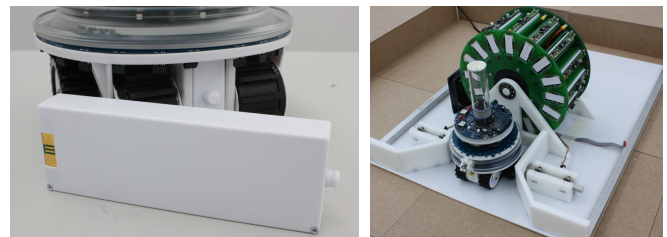

Fig. 2: The battery (left) and the exchange and recharge station (right).

easily exchange the battery by drawing it on the horizontal

Module	Power Cons.
main computer with Wifi	2.2 W
base, standing still	1 W
rotating scanner	2.2 W
attachment, still	0.8 W
range and bearing	0.9 W
Sum of all modules	7.1 W

TABLE II: The constant power consumption of the different modules.

axis. We have built and successfully validated[1] such an automatic battery replacement station (Fig. 2, right). To provide protection and monitoring, we have developed our own battery casing and electronics (Fig. 2, left). The marXbot can at any time read its current power consumption and the battery's voltage and load. The battery has a capacity of 38 Wh (10.6 Ah at 3.6 V) which allows the marXbot to run for 4 h with all its sensors and motors in average usage, and for 7 h when using the same features as a robot such as the S-bot. This exceeds the capacities and the autonomies seen in the related work.

The good battery life of the marXbot is due to its advanced power management. We use an I2C bus to manage the power on the different modules of the marXbot. To save energy, the robot can put its unneeded modules into deep sleep with a special command, in which they consume only a few milliamperes. When the modules are needed again, they can quickly be wake up, using the I2C bus, to perform the desired task. We have measured the basic power consumption of every module (TABLE II). We also have modelled the power usage of the motors, the beacon LED and the infrared sensors according their respective parameters (speed, light intensity, and sampling frequency). Fig. 3 shows the power model of the two locomotion motors of the base module. These models allow the robot to estimate how much energy it will use for a series of actions. This is useful to estimate when the robot should go to exchange its battery during long experiments.

We have taken actual power consumption measures during a SLAM experiment [15], in which the robot randomly moves and uses its rotating scanner to build a map of its environment. This experiment utilizes the base module, the rotating scanner, and the main computer. According to our power models, the expected power consumption should be 7.5 W and we have measured an average power consumption of 7.72 W (Fig. 4).

VI. ATTACHMENT DEVICE

As shown by the *Swarmbots* project, the ability of a mobile robot to assemble with its peers is essential for many interesting swarm behaviors [1]. However, as demonstrated by [13], there are critical mechatronic features to successful achieve self-assembling. The assembling process must be compliant. In particular, the more of the robot perimeter is connectable the better. Moreover, the attachment device

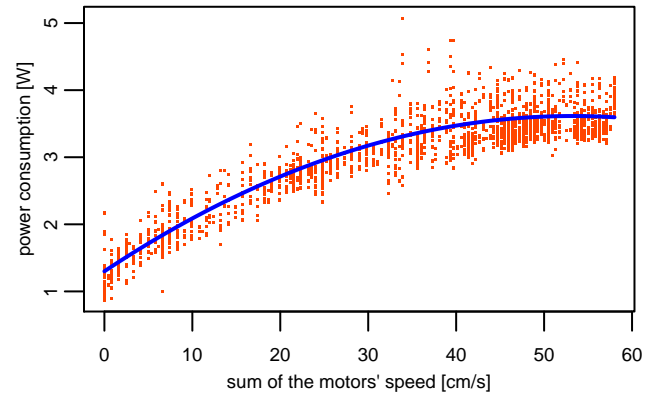[1]See video http://www.youtube.com/watch?v=zEeLjeJslH4

Fig. 3: Power consumption measurements (red) and the power model (blue) of the marXbot locomotion motors. The model is a second-order polynomial.
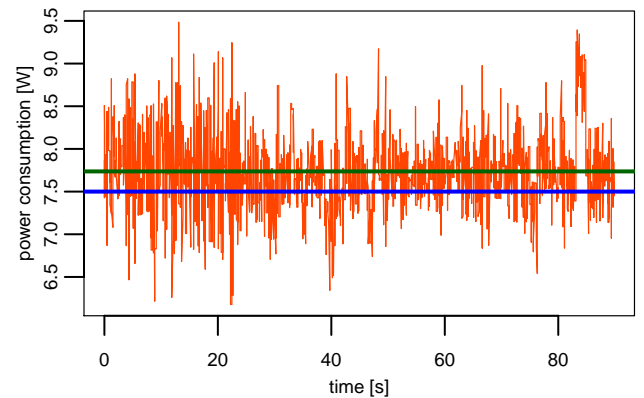

Fig. 4: Power consumption measurements of the whole marXbot in a SLAM experiment [15] (red), the mean of the measurements (green), and the expected power consumption according to our model (blue).

should attach successfully even if it is not fully correctly positioned. When operating in rough terrain, the robots should be able to attach even if they are not perfectly aligned or vertically oriented. As a robot cannot be certain to attach successfully at the first try, this robot must be able to detect attachment failures and to retry until it succeeds.

The current state of the art in self-assembling is held by the S-bot [13]. However, its attachment system has several limitations. The marXbot does not have these limitations. In the S-bot, the attachment is not possible on the whole perimeter. In addition to the front of the robot, where the gripper obviously prevents any attachment, the S-bot has dead
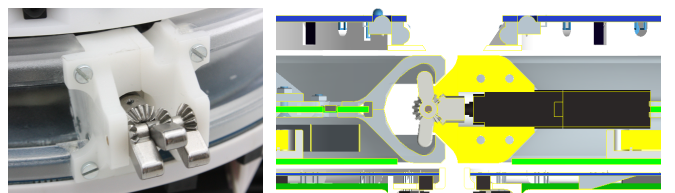

Fig. 5: Attachment device and mechanism.

points where attachment is impossible due to mechanical constraints. The marXbot does not have any dead point, so excepted on the front the attachment is possible on the whole perimeter, which corresponds to 320°. The S-bot uses a gripper, which must close at a precise distance of the peer robot to successfully connect. This restrictive requirement does not allow the S-bot to achieve a attachment rate of 100% in rough terrain. On the marXbot, the attachment is done with fingers that expand inside a circular opening (Fig. 5, left). That way, the connecting robot just needs to go forward until it touches the other robot, and then expand its fingers to create a tight grasp (Fig. 5, right).

In Fig. 6 we show an exploded view of the attachment device. At the mechanical level, this module is one of the most complex of the marXbot, and for this reason we detail its internals in the rest of this paragraphe. We refer to the different parts from Fig. 6. A is the printed circuit board (PCB) controlling the force sensors. On top of it, a support piece B provides structural integrity and attachment for the other parts. This parts attaches to the base of the marXbot at its bottom and to another PCB L at its top. Part B also has a hole in its center, to allow the connector K which contains the power and data busses. Attached to B through four thunder sheets, a part C provides attachment for the rotation system. One strain gauge is glued on each thunder sheet to allow the robot to measure the force applied by the other robot on the horizontal plane. The rotating system is composed of a fixed part with a gear (I and J) and a rotating ring (E to H). A flex cable D provides the connection between the fix part and the rotating ring. It allows rotation from +360° to -220°. The rotating ring is mainly composed of two plastic parts F enclosing a PCB H. This PCB contains a microcontroller that drives the rotation and the connection-system motors as well as 12 RGB LEDs. The rotation motor G has a triple multi-start worm gear, to make the rotation mechanism reversible. Thus once connected the rotation turret automatically rotates to follow the movement of the robot. The connection mechanism E is based on a conical gear driving three fingers (see Fig. 5). On the top of the module, a PCB L provides electrical and mechanical connections for another module. These connections are the same for all modules, which provides physical modularity.

To validate the attachment device of the marXbot, we have reproduced the self-assembling experiments of [13]. In these experiments, a moving marXbot looks for the beacon LED of a fixed robot using its front camera. To do so, it rotates clockwise until it sees the other robot; then approaches it, correcting its trajectory with a P controller using the camera as input. When the moving robot is close to the fixed one, its slows down and attaches to it. As in [13], the robot uses its proximity sensors to decide when to grasp. To detect whether the attachment was successful, the moving robot displaces back slightly and tries to rotate its ring. If the rotation fails, the attachment is good. If the rotation succeeds, then the attachment is bad, and the robot retries to attach. It does so by maneuvering back some centimeters such that it will attach to a slightly different position. We use the same controller
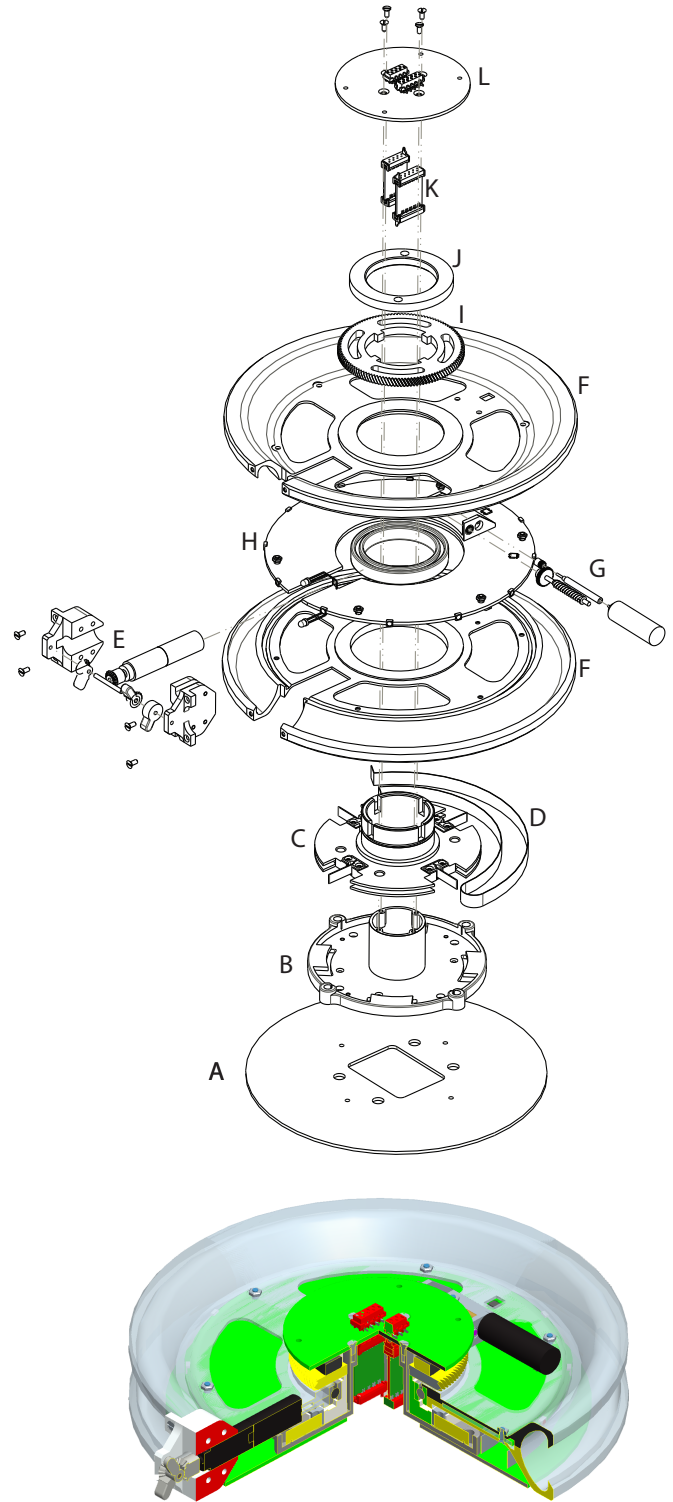

Fig. 6: Mechanical structure of the marXbot attachment module. On top, an exploded view and at the bottom, a 3/4 view.
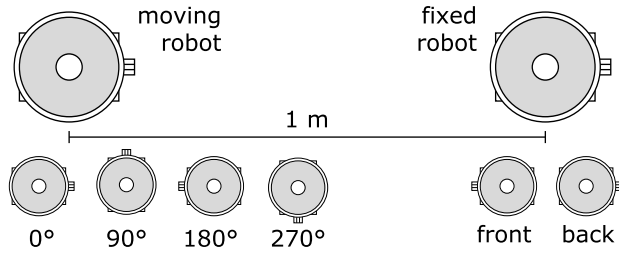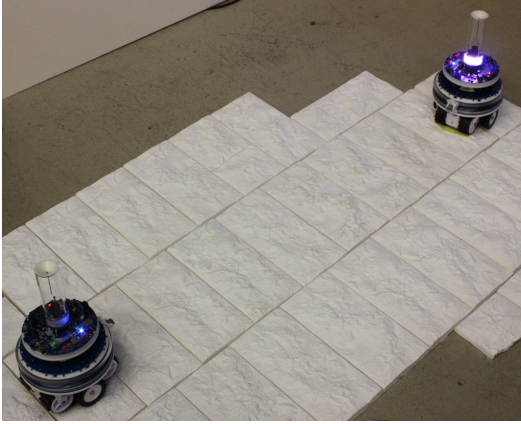
Fig. 7: Setup for attachment experiments.

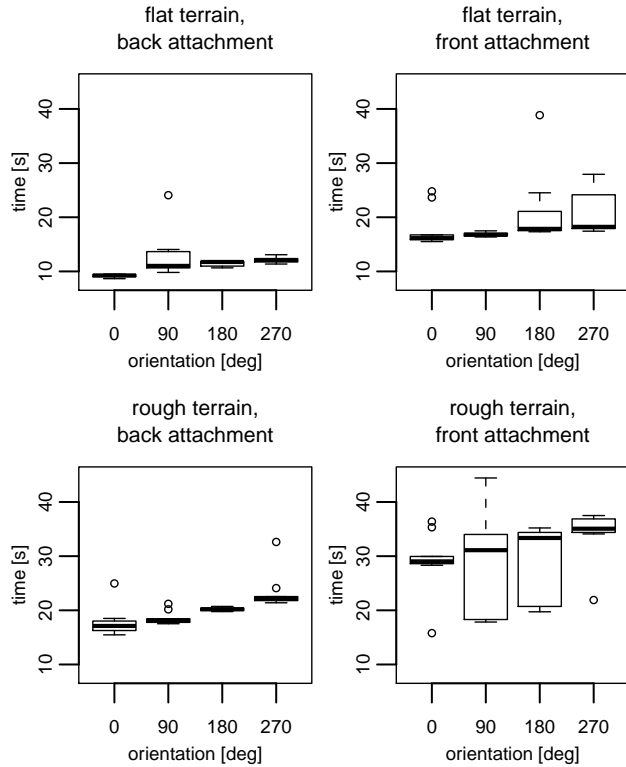Fig. 8: Photo of the rough terrain setup.

Fig. 9: Time to attach to a peer marXbot for a given starting orientation. Results on different terrain types and peer's orientation.

for flat and rough terrains, but in the latter all the speeds are divided by a factor of two.

We tested the combination of two types of terrain (flat and rough Fig. 8), four different starting orientations for the moving robot (0°, 90°, 180°, 270°), and two different orientations of the fixed robot (back and front). For each combination, we conducted 10 runs with two robots at an initial distance of 1 m (Fig. 7). The marXbot successfully attaches in all situations, including rough terrain, where the S-bot sometimes failed to attach. In the experiments from [13, Table III], when approaching a robot from the back from a distance of 50 cm, the S-bot took in average 28.7 s to attach in the flat terrain, and 71.3 s in the rough one. The marXbot is much faster, when approaching from a distance of 1 m, the marXbot only takes in average 11.4 s to attach in the flat terrain (with an average of 0.025 retry per trial), and 19.9 s in the rough one (0.025 retry). In the most difficult condition, when approaching the fixed robot from its front, the marXbot takes 20.0 s in the flat terrain (1.37 retry), and 30.8 s in the rough one (0.85 retry). Fig. 9 shows the detailed measurements for different starting orientations.

VII. MAIN COMPUTER AND VISION

In mobile robotics, vision is becoming a standard sensing modality as it provides a lot of information. For the S-bot, the omnidirectional camera has been a key feature for many experiments [19]. One of the usual limitations of robotic vision systems is the camera itself and its interface with the processor. On the S-bot the camera is connected through USB and acquires images in JPEG format. The processor of the main computer must decompress these images prior to use them, which consumes an important part of its processing power. The processor of the marXbot, the i.MX31, has an *Image Processing Unit* which is able to manage CMOS camera sensors and acquire images through a dedicated *Camera Sensor Interface*. The image data are transferred into the system memory using the Direct Memory Access (DMA) unit present in the *Image Processing Unit*. This allows the robot to acquire raw images and directly process them without dedicating a part of its processing power to the acquisition. On the *Camera Sensor Interface*, we have multiplexed the signals of two CMOS camera sensors. The first sensor is pointing towards an hyperbolic mirror shaped to our needs and is an omnidirectional camera. Modern manufacturing technologies allow us to produce this mirror at low cost. The second sensor has standard optics and can be mounted in front of the robot or on the top of it (pointing up) according to the needs. The omnidirectional camera and the standard camera can be used together but the i.MX31 cannot acquire images from both at the same time due to hardware limitations.

To validate the vision system, we have acquired with a marXbot similar images to the ones acquired by the S-bot. We compared the distances at which the marXbot can detect other robots with the results obtained with the S-bot [19, Fig 2.b]. The setup consists of a central marXbot that uses its omnidirectional camera to look for the surroundings marXbots. These surrounding marXbots have their attachment-ring LEDs
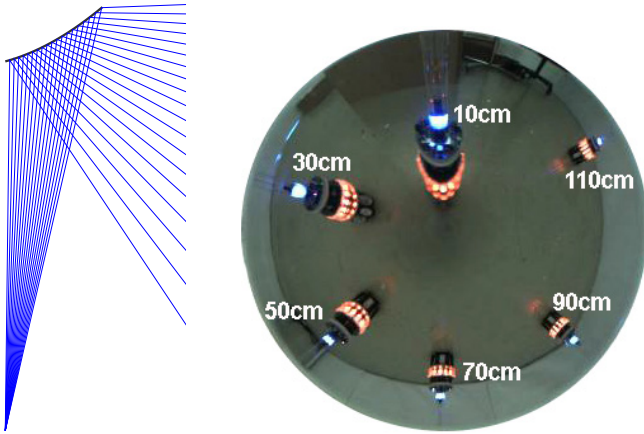
Fig. 10: Omnidirectional vision. Left: cut of the mirror with rays. Right: image acquired with surrounding robots; notice that the marXbot does not see itself. The attachment-ring LEDs are in red, the beacon LED is in blue.

as well as their beacon emitting light so that they can be tracked from the image by extracting the corresponding color blobs. Fig. 10 shows the image acquired by the central marXbot in the experimental setup. When detecting using the LED rings, the marXbot is able detect the robots up to 110 cm while the S-bot was only able to do it until 70 cm. When the beacon is used, this distance can be extended up to 150 cm.

VIII. CONCLUSION

Based on the experience of the S-bot, we have built a new platform, the marXbot. The marXbot allows continuous experiments thanks to a sophisticated energy management and a *hotswap* battery exchange mechanism. As the marXbot has an updated design, is faster, and has more and better sensors than the S-bot, it performs basic operations such as self-assembling much more efficiently. By allowing complex tasks in large environments for long durations, the marXbot opens new perspectives for the collective-robotic research.

REFERENCES

[1] E. Tuci, R. Groß, V. Trianni, F. Mondada, M. Bonani, and M. Dorigo, "Cooperation through self-assembly in multi-robot," *ACM Transactions on Autonomous and Adaptive Systems*, vol. 1, no. 2, pp. 115–150, 2006.

[2] F. Mondada, G. C. Pettinaro, A. Guignard, I. Kwee, D. Floreano, J.-L. Deneubourg, S. Nolfi, L. Gambardella, and M. Dorigo, "SWARM-BOT: a New Distributed Robotic Concept," *Autonomous Robots, special Issue on Swarm Robotics*, vol. 17, no. 2-3, pp. 193–221, 2004.

[3] F. Ducatelle, A. Förster, G. D. Caro, and L. Gambardella, "New task allocation methods for robotic swarms," in *9th IEEE/RAS Conference on Autonomous Robot Systems and Competitions*, May 2009.

[4] 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," pp. 62–69, 2008.

[5] 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* (M. H. Ang and O. Khatib, eds.), vol. 21 of *Springer Tracts in Advanced Robotics*, ch. 13, pp. 131–140, Springer, 2006.

[6] A. Turgut, F. Gokce, H. Celikkanat, L. Bayindir, and E. Sahin, "Kobot: A mobile robot designed specifically for swarm robotics research," tech. rep., Technical Report METU-CENG-TR-2007-05, Dept. of Computer Eng., Middle East Tech. Univ., Ankara, Turkey, 2007.

[7] R. Rusu, G. Lazea, R. Robotin, and C. Marcu, "Towards Open Architectures for Mobile Robots: ZeeRO," in *Proceedings of the Automation, Quality and Testing, and Robotics International Conference (AQTR 2006)*, pp. 260–265, IEEE Press, 2006.

[8] ECLab Evolutionary Computation Laboratory, George Mason University, "Flockbots." http://cs.gmu.edu/ eclab/projects/robots/flockbots.

[9] V. Zykov, P. William, N. Lassabe, and H. Lipson, "Molecubes Extended: Diversifying Capabilities of Open-Source Modular Robotics," in *IROS-2008 Self-Reconfigurable Robotics Workshop, accepted*, 2008.

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

[11] K. Yoo, Y. Lee, H. Choi, B. Yoo, and D. Kim, "Swarm Robotics: Self Assembly, Physical Configuration, and Its Control," in *SICE-ICASE, 2006. International Joint Conference*, pp. 4276–4279, 2006.

[12] H. Zhang, W. Wang, Z. Deng, G. Zong, and J. Zhang, "A novel reconfigurable robot for urban search and rescue," *International Journal of Advanced Robotic Systems*, vol. 3, no. 4, pp. 359–366, 2006.

[13] R. Groß, M. Bonani, F. Mondada, and M. Dorigo, "Autonomous Self-assembly in Swarm-Bots," *IEEE Transactions on Robotics*, vol. 22, no. 6, pp. 1115–1130, 2006.

[14] G. Di Caro, F. Ducatelle, and L. Gambardella, *Wireless Communications for Distributed Navigation in Robot Swarms*, pp. 21–30. Lecture Notes in Computer Science, Springer, 2009.

[15] S. Magnenat, V. Longchamp, M. Bonani, P. Rétornaz, P. Germano, Bleuler, Hannes,, and F. Mondada, "Affordable SLAM through the Co-Design of Hardware and Methodology," in *Proceedings of the 2010 IEEE International Conference on Robotics and Automation*, IEEE Press, 2010. To appear.

[16] S. Magnenat, P. Schoeneich, F. Rochat, P. Rétornaz, M. Bonani, V. Longchamp, M. Voelkle, T. Barras, D. Burnier, P. Noirat, T. Baaboura, F. Vaussard, and F. Mondada, "Autonomous Construction by a Mobile Robot in Unknown Environments with Scarce Resources." Presented at CogSys 2010, ETH Zurich, Switzerland, 2010.

[17] I. Standards, *Road Vehicles Interchange of Digital Information - Controller Area Network - ISO 11898*. International Organization for Standardization, 1993.

[18] S. Magnenat, P. Rétornaz, M. Bonani, V. Longchamp, and F. Mondada, "ASEBA: A Modular Architecture for Event-Based Control of Complex Robots," *IEEE/ASME Transactions on Mechatronics*, 2010. To appear.

[19] S. Nouyan, R. Groß, M. Bonani, F. Mondada, and M. Dorigo, "Teamwork in Self-Organized Robot Colonies," *IEEE Transactions on Evolutionary Computation*, vol. 13, no. 4, pp. 695–711, 2009.