MIT Libraries | DSpace@MIT

MIT Open Access Articles

Multirobot inspection of industrial machinery

Massachusetts Institute of Technology

*From Distributed Coverage
Algorithms to Experiments
with Miniature Robotic Swarms*

BY NIKOLAUS CORRELL
AND ALCHERIO MARTINOLI

Multirobot Inspection of Industrial Machinery

Inspection of aircraft and power generation machinery using a swarm of miniature robots is a promising application both from an intellectual and a commercial perspective. Our research is motivated by a case study concerned with the inspection of a jet turbine engine by a swarm of miniature robots. This article summarizes our efforts that include multirobot path planning, modeling of self-organized robotic systems, and the implementation of proof-of-concept experiments with real miniature robots. Although other research tackles challenges that arise from moving within three-dimensional (3-D) structured environments at the level of the individual robotic node, the emphasis of our work is on explicitly incorporating the potential limitations of the individual robotic platform in terms of sensor and actuator noise into the modeling and design process of collaborative inspection systems. We highlight difficulties and further challenges on the (lengthy) path toward truly autonomous parallel robotic inspection of complex engineered structures.

For certain tasks, multirobot systems are a promising alternative to a single robot solution because they offer a higher

level of robustness due to redundancy and the potential for individual simplicity. Also, the possibility of conducting work in parallel potentially allows for a faster task execution, e.g., in a coverage or an exploration task. This property is even more striking when size constraints on the robotic platform do not allow inspection of an environment with a single robot in acceptable time. In addition to the locomotion constraints that are specific to the environment, such a scenario poses numerous design challenges such as limited interrobot communication, determining the position or relative range and bearing [1], and the design of efficient and robust algorithms for coordination of a robot team. Benefits and challenges of miniature multirobot coverage are well illustrated by the automatic inspection of (jet) turbines (Figure 1), which is a promising commercial application [2]. To minimize failures, jet turbine engines have to be inspected at regular intervals for evidence of internal distress such as cracking or erosion. This is usually performed visually by using borescopes as well as ultrasound (US) and eddy current sensors [3], which is a time-consuming and cost-intensive process, particularly if it involves dismantling the turbine. One possible solution for speeding up and automating the inspection process is to rely on a swarm of autonomous, miniature robots that could be
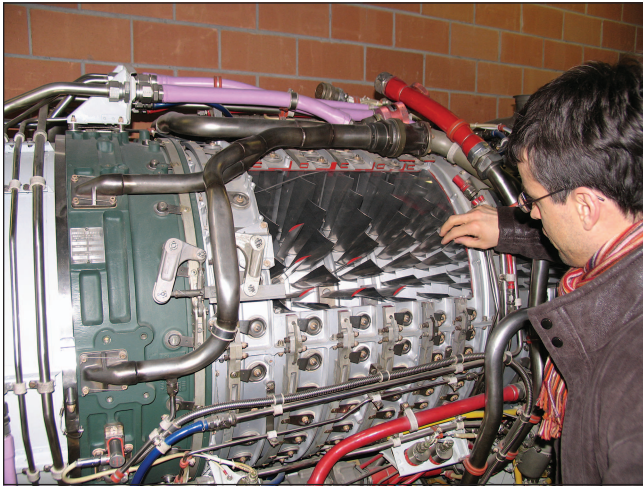
Figure 1. *The compressor section of a jet turbine. The internal dimensions are within the same order of magnitude as those of the miniature robotic systems used in this article.*

released into the turbine while still attached to the wing [4]. With the immediate prospect to reduce the downtime during regular inspection intervals, the final goal of such an approach is a distributed control architecture that allows for a shift from a schedule-based maintenance system to a condition-based system, which is based on smart sensors and actuators [5]. Here, the deployment of mobile sensors rather than the installation of permanent sensors [6] is a compromise between the increased system cost and the benefits arising with an in situ inspection [3].

Although this idea is intellectually appealing and could pave the way for other similar applications in the inspection of potentially complex, engineered or natural structures, it involves a series of technical challenges that drastically limit possible designs of robotic sensors and can loosely be classified into three engineering thrusts: miniaturization of sensors and actuators, control of distributed hybrid systems, and sensor fusion for providing information to a human operator or an expert system. The distributed system can be considered hybrid in the sense as that the individual robotic platform is

Figure 2. *A simplified mock-up of a jet turbine being inspected by a swarm of miniature robots showcased during the Swiss-wide Festival Science-et-Cité in Spring 2005. (Photo courtesy Alain Herzog.)*

controlled by a series of reactive continuous control laws, which are switched by some logic function or algorithm. All three thrusts are dominated by strong constraints on available energy, sensing, actuation, and computation, which render certain control approaches—particularly those that require rich sensor information for performing extensive reasoning on the individual robotic node—unfeasible. Rather, a distributed system of unreliable or less controllable robotic nodes requires the analysis of algorithms from a probabilistic perspective. Finally, commands by human users that address the properties on the swarm level need to be synthesized into control inputs to the individual robots.

The focus of our work [7] is on algorithms for coordinating a robot swarm for the coverage [8] of relevant parts of the turbine's interior, where individual units are subject to the extreme miniaturization constraints on the individual platform, rather than developing specific solutions for locomotion or inspection for an individual robot in such an environment (see e.g., [9] or [10], and [11], respectively, and references therein). We undertake experimentation with real hardware (Figure 2), which serves both as a validation and motivation for our algorithms, where emphasis is on the robustness with respect to the sensor and actuator noise of minimalist platforms in use.

In the following sections, we first summarize the design challenges imposed by our case study and then describe our experimental setup and hardware that we developed. Finally, we compare results from both probabilistic and deterministic control strategies.

Design Challenges

The turbine inspection scenario imposes a series of constraints that drastically influence the possible design choices for the robotic platform and potential coordination algorithms:

- ◆ Miniaturization can be considered as the toughest constraint. Miniaturization significantly limits the choice of potential actuators, sensors, and available energy. In particular, the available volume for energy storage on a miniature platform limits the overall movement autonomy, computational power, and communication.
- ◆ Energy limitations might be overcome by providing the robots with tethers [2], which would also be useful for easily removing broken or stuck robots from the turbine. Tethers, however, have the disadvantage of requiring stronger actuators, as the robot has not only to self-locomote but also to pull the potentially entangled tether that might quickly outweigh the robotic platform, particularly if it is to be robust enough for the manual removal of the robots. In a distributed system, entangling of tether cables is even more likely and imposes additional constraints on path-planning algorithms.
- ◆ Because of the shielded and narrow structure of the turbine that might act as a Faraday cage, communication is limited to short range. For the same reason, closed-loop control of the system by an outside supervisor (agent) is essentially unfeasible.
- ◆ Reliable locomotion in a highly structured, upside-down environment poses tremendous mechanical challenges.

Algorithms and analysis presented in this article tackle miniaturization, energy limitations, and limited range communication experimentally, although we are not exploring other locomotion principles than wheeled differential-drive robots.

In addition to the physical constraints, the inspection task also presents various algorithmic challenges:

◆ Potentially redundant sensory information provided by the robot swarm needs to be fused and annotated with the location within the turbine where it was recorded.
◆ The (3-D) data recorded within the environment needs to be analyzed, e.g., for detecting flaws, potentially using an expert system.
◆ Appropriate control commands need to be synthesized and sent to the robot swarm to achieve a desired collective behavior, e.g., for more closely inspecting a certain region of the structure.

A Miniature Platform for Autonomous Inspection

Our robotic inspection nodes (Figure 3) are based on the Alice miniature robot [12], developed by Gilles Caprari at the Autonomous System Laboratory when still affiliated with EPFL. The Alice has a cubic shape of approximately 2 cm side length and is operated by a PIC 16F877 microprocessor (4 MHz, 384 B of RAM, 8 kB ROM). Driven by two watch (stepper) motors in a differential-drive configuration, it can travel with a top speed of 4 cm/s. It is endowed with four IR modules, which can serve as very crude proximity sensors (up to 3 cm) and local communication devices (up to 6 cm in range), providing a simple communication channel at around 500 b/s, which can also be used for crude interrobot local positioning. Its energetic autonomy with a 40-mAh (at 4.5 V) NiMH rechargeable battery ranges from 10 min to 10 h, depending on the actuators and sensors used (see Table 1 for the detailed energy consumption of selected components). The reason for the extreme differences in autonomy is not the actual cumulative power consumption but rather the maximal possible drain that the battery is supporting. In practice, significant voltage drops are already observed for drains of more than 0.5 C (1 C corresponds to the nominal capacity), which makes the simultaneous operation of the camera and the radio modules (described later) impossible.

To improve the computational and communication capabilities for ad hoc networking among the robotic swarm and to eventually transmit the recorded data to a base station, we developed an extension board, providing a Texas Instruments (TI) MSP430 microprocessor (2 kB RAM, 60 kB ROM), a TI CC2420 radio (ZigBee ready), and 4-MB flash memory. The module can be conveniently programmed in TinyOS (http://www.tinyos.net), which provides a growing number of ready-to-use libraries for different purposes and allows easy integration with a wide range of compatible static sensor networks.

For inspection and localization, we designed a camera module endowed with a PixelPlus Po3030k VGA miniature camera that is downsampled to 30×30 pixels in red, green, and blue (RGB) color. Using a PIC40F4620 with 4 kB RAM at 32 MHz for image acquisition and processing, the Alice is able to take pictures at a rate of around 2 Hz (Figure 4), as well as uniquely identify color markers in the environment
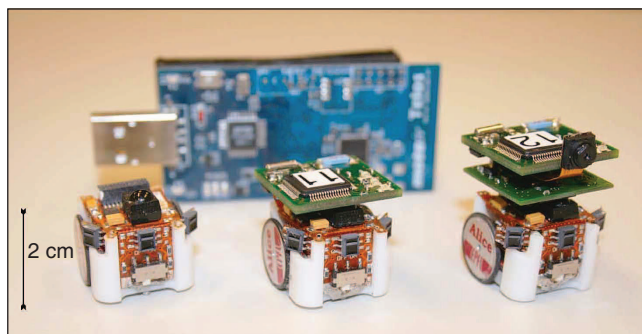
Figure 3. *The miniature robot Alice (2 cm × 2 cm × 2 cm) endowed with extension modules providing ad hoc networking (middle) and imaging capabilities (right). A moteiv Telos mote, which inspired the design of the communication module, is shown in the background.*

(Figure 5). The Alice and the extension modules communicate via an I^2C two-wire bus (a block diagram is shown in Figure 6). With the two extension modules mounted, the inspection robot fits well into a parallelepiped of 2 cm × 2 cm × 3 cm.

Experimental Setup

We simplify the real 3-D environment by unrolling the axis-symmetric geometry of the turbine into a flat representation with the blades as vertical extrusions. Blades are made from aluminum and aligned in a 5×5 pattern on a 60 cm × 65 cm large arena (Figure 2) made of steel. The blades are fixed by self-adhesive magnetic tape. The fact that the arena is entirely made from metal leads to significant communication loss due to electromagnetic absorption, particularly when a robot's antenna is incidentally in direct contact with a blade.

For algorithms that require localization, the upper part of the blades is equipped with a unique color marker that consists of three colored bars (Figure 5). Saturation or depletion of any of the three color channels (red, green, and blue) is used to encode 3 b per color. Using the middle bar as references (all channels at 50%) allows us to encode 64 different codes, of which we are using 25 for identifying each blade. Experiments showed 95% accuracy (average of 100 experiments) for correctly identifying a blade.

## Distributed Coordination Schemes for Multirobot Inspection

In our experiments, we are not concerned with the detection or mapping of flaws but rather with the individual and group motion

Table 1. Energy consumption for selected individual subsystems of the inspection platform.

Individual Subsystem	Energy Consumption (mW)
Alice, motors off	4.5
Alice, full-speed drive	15
Radio module active	60
Radio module sleep	<1
Camera module active	60
Camera module sleep	15
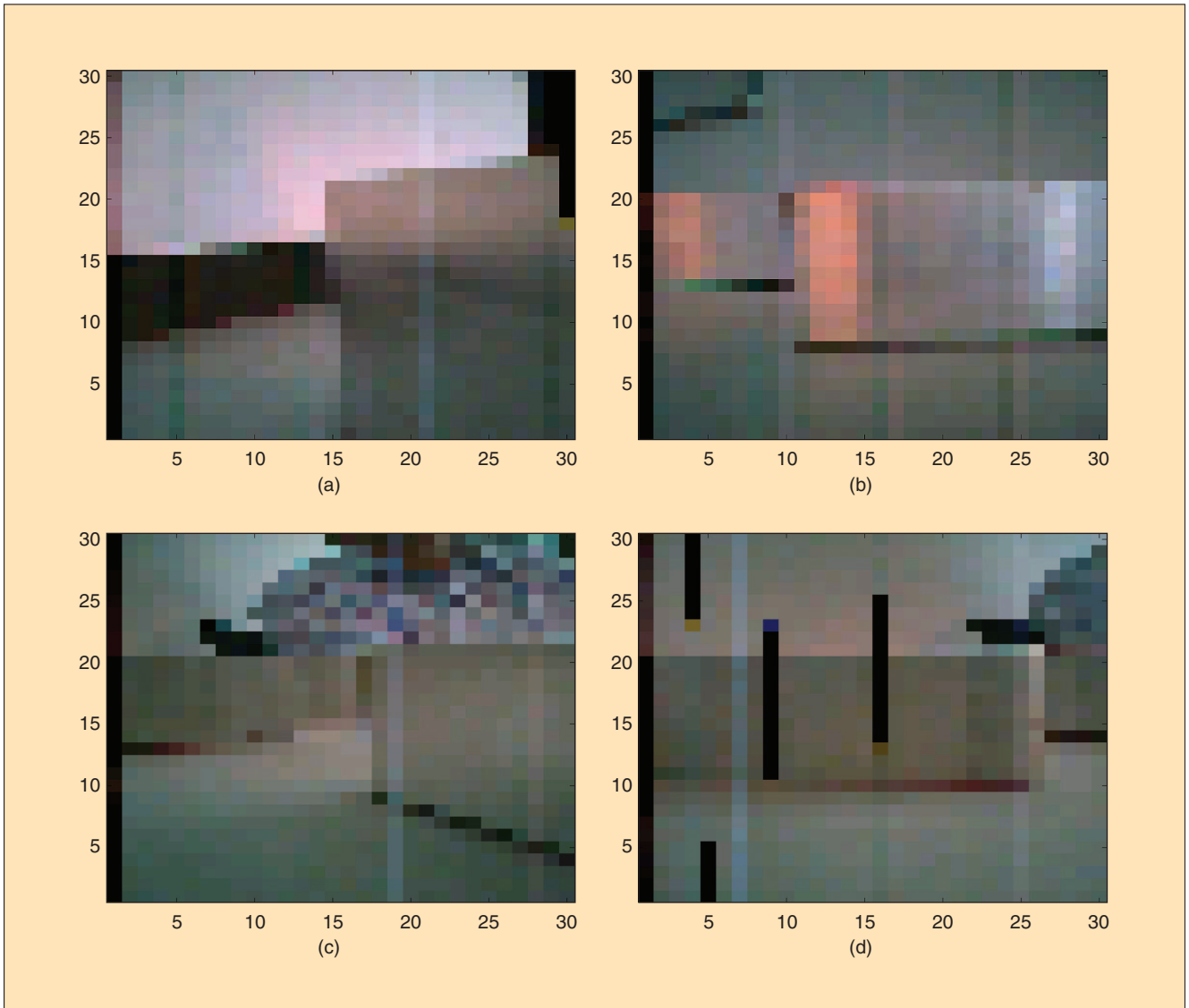
Figure 4. Pictures (30 × 30 pixels) taken by the on-board camera and transmitted over the radio with 72 packets of 25 B. Vertical black stripes indicate packet loss. (a) The arena boundary (painted in black) can be seen. (c and d) The experimenter's upper part of the body is visible in the background.
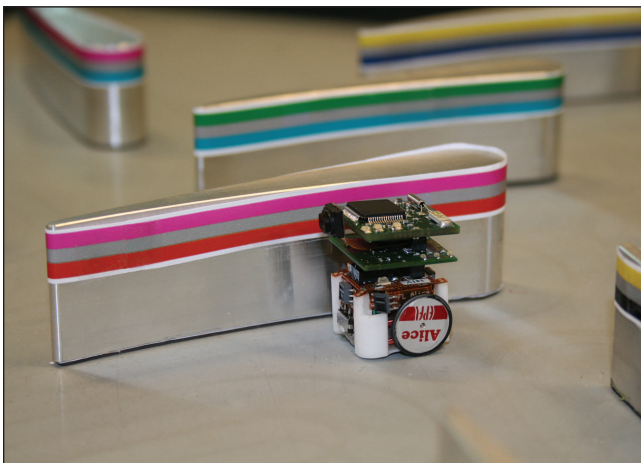
Figure 5. The fully equipped Alice in an environment with colored markers. The two-color code (the middle bar serves as reference) can be recognized with 95% accuracy.

given the constraints of the turbine scenario. For the sake of simplicity, we therefore assume that circumnavigating a blade in its totality is a good emulation of a scanning-for-flaws maneuver.

We consider various algorithms, which can be classified among the control paradigm used, as well as based on their requirements for the individual robotic platform. On the one hand, we consider a fully reactive approach that has minimal requirements on the robotic platform (low bandwidth, local communication, no localization). Local infrared communication is then used for increasing the dispersion of the robots in the environments. In this scenario, the radio and camera can potentially be used for inspection but require offline processing for mapping sensory and image data to the location where they were recorded. On the other hand, we consider deliberative approaches that require the ability of creating a topological map, as well as a sufficient bandwidth for sharing maps among the robots, which requires some sort of localization. An additional benefit of localization is the potentially easy mapping of sensory data onto the arena.

Reactive Inspection using Local Communication

The motivation for a fully reactive approach is the potential for its implementation on extremely minimalist robotic plat–forms. The basic idea is to eventually cover the environment by moving from blade to blade reactively. Local communica–tion is used for enhancing dispersion in the environment. We will first describe the robot behavior and then present a method–ology for modeling and predicting cover–age performance.

Robot Behavior

The necessary behaviors for circumnavi–gating all blades and avoiding collisions can be divided as follows: search, avoid other robots, avoid a wall, and circumnavigate a blade. We implemented the following sequence of behaviors: upon encountering a blade, which can be distinguished from a wall by their color, a robot starts circumnavigating its boundary until a time–out expires (10 s in our experiments), and it arrives at its tip. The combination of a time-out with a physical event (arriving at the tip) ensures that blades are circumnavigated with the least amount of redundancy and that the influence of wheel–slip and other disturbances (which count toward the inspection time) are limited. Robots perform another sweep along one side of the blade with a probability of 50%, as leaving a blade at its tip will induce a drift of the robots through the environment and thus lead to a lower probability of inspection for some blades than others. This robot controller can be summarized by the finite state machine (FSM) diagram of Figure 7.
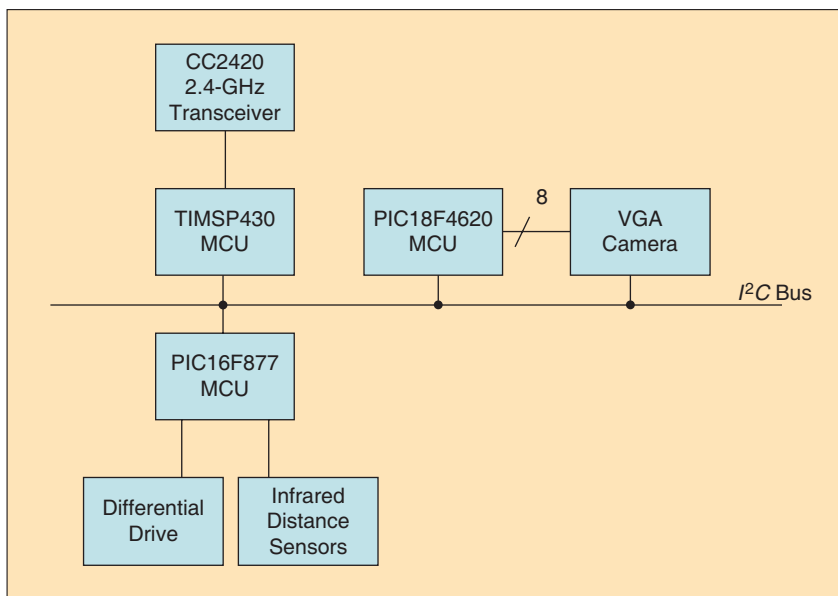
Figure 6. Block Diagram of the inspection platform measuring around 2 cm × 2 cm × 3 cm, endowed with two watch motors for differential drive, a 2.4-GHz ZigBee-compliant wireless radio, a VGA camera, and three microcontrollers connected by an I^2C two-wire bus.
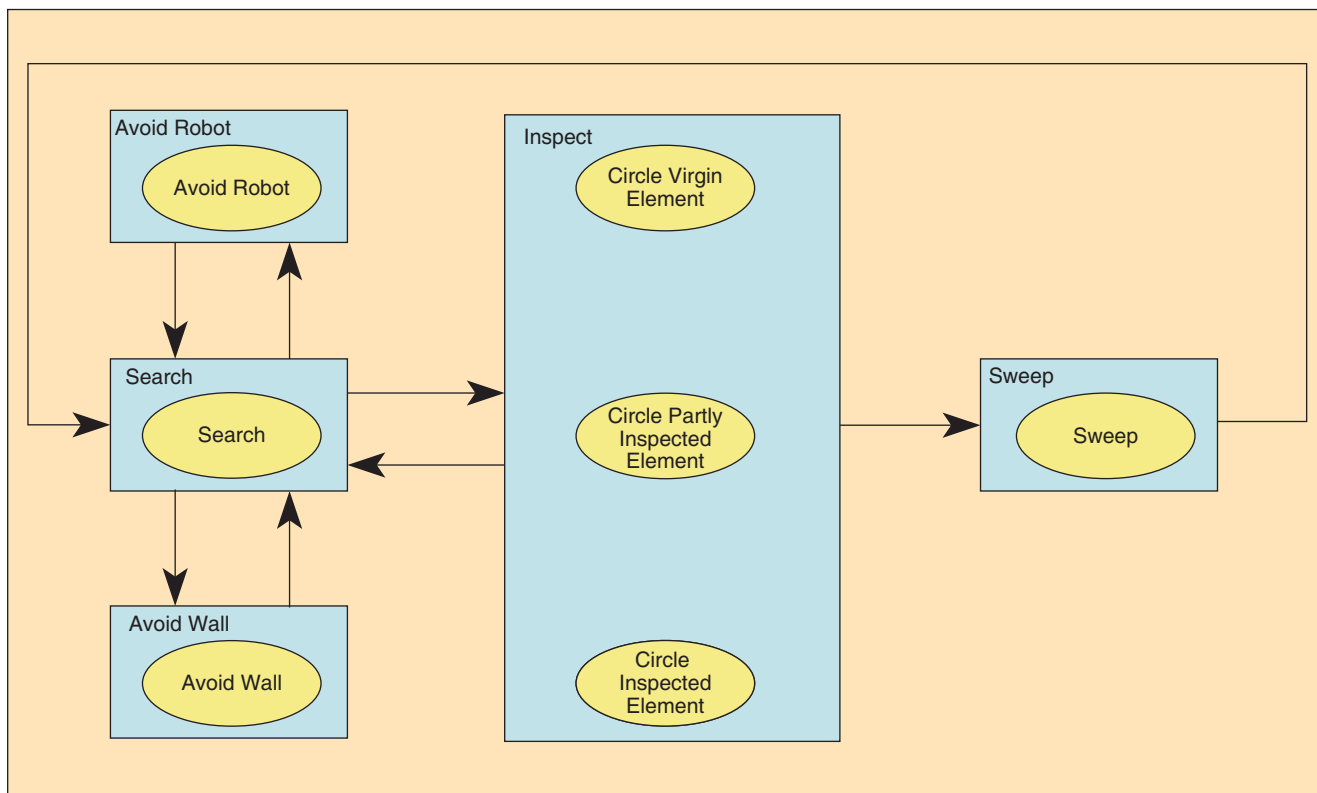
Figure 7. FSM and PFSM. The FSM (squares) is of lower granularity than the PFSM (ellipses) and does not consider the state of an element (virgin, partly inspected, or inspected), as this information is not known to an individual robot.
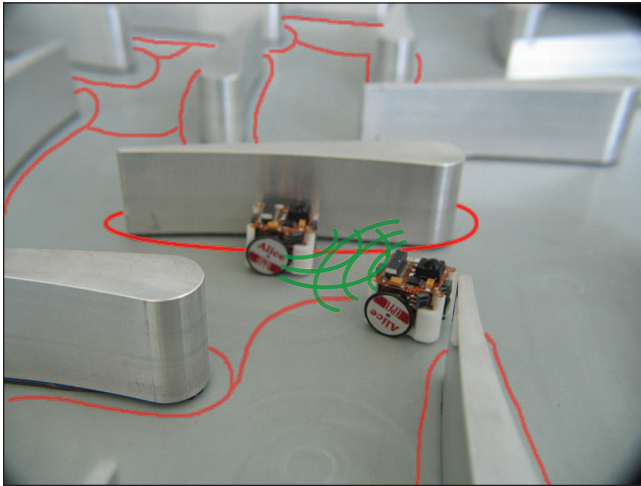
Figure 8. *Using the self-organized, reactive controller, robots are reactively moving through the environment and inspect blades for a fixed amount of time. Blades are then left as soon as the tip is reached. Robots and blades are differentiated using on-board infrared sensors.*

Robots can communicate locally by modulating the signal sent on the infrared emitter-receiver pairs. This is used to communicate a robot state to other robots, and it is exploited by the following additional behaviors, which aim at reducing redundant coverage. For instance, a meeting between two robots during the circumnavigation of the same blade will prompt one of the robots to abandon the inspection. In case of a front–to-front encounter, the robot with the blade to its left-hand side will abandon the inspection, whereas in case of a back–to-front encounter, the robot that detects the other robot by its front sensors will abandon the inspection. The behavior of the robots and sample trajectories are illustrated in Figure 8.

Probabilistic Modeling

Because of the high amount of noise that is intrinsic to miniature robotic platforms and fully reactive coordination, deterministic models are unsuitable for modeling the collective dynamics of the system described earlier. Rather, we abstract the FSM of an individual robot to a Probabilistic FSM (PFSM)

that captures the dynamics of our system at a sufficient level of detail [13], [14].

If we assume a uniform distribution of robots and objects in the environment, the probability to inspect an uncovered blade is proportional to the total number of uncovered blades. Working with time-discrete models, given the number $[M_v(k)]$ of uncovered blades at time step k, and the probability to encounter one blade as p_e, the probability for encountering a virgin blade at time step k is given by $p_e M_v(k)$. In a PFSM for an individual robot, $p_e M_v(k)$ is then the probability to switch from searching to inspection of a virgin element at time step k. Notice that covering of a virgin or inspected element corresponds to the same state in the FSM but is captured by distinct states in the probabilistic model (Figure 7) because this information is crucial for the system performance considered but not known by individual robots (and therefore not exploited at the controller level). The other state transitions follow similar reasoning, which calculates the probability of an event by combining the encountering probability of an object (or the intersection of two objects) with the number of such objects at a given time. In the model, we approximate the real probability distribution of leaving a given state with its mean and assume constant probabilities over the experiment as model parameters. The inverse of the average time spent in a state then yields the constant probability for leaving that state. Encountering probabilities and state durations necessary for modeling the inspection case study are summarized in Table 2. One can then simulate such a system for an arbitrary number of robots and thus keep track of the number of robots in various relevant states.

The described formalism also allows us to summarize the average state transitions by a set of difference equations. For instance, the number of robots $N_v(k)$ inspecting a virgin blade are given by the following equation:

$$N_v(k + 1) = N_v(k) + p_e M_v(k) N_s(k) - \frac{1}{T_e} N_v(k), \qquad (1)$$

where T_e is the average time needed for inspection. In words, the number of robots inspecting a virgin blade is increased by the number of searching robots that encounter a virgin blade. Robots leave N_v at an average rate of $1/T_e$, which corresponds to an average time of T_e spent in this state. The equations for

---

Table 2. State variables keeping track of the number of robots in a particular state, as well as the coverage state.

State Variable	Description	Parameter	Description
N_s	Number of robots searching	p_e, p_w, p_R	Probability to detect a blade, a wall, or any other robot during one time-step of the model
N_{ar}, N_{aw}	Number of robots avoiding another robot or a wall		
N_v, N_p, N_i	Number of robots inspecting a virgin blade, a partly inspected blade, or an inspected blade		
N_b	Robots acting as a beacon, when sweeping back along a blade's contour	T_e, T_{ar}, T_{aw}, T_b	Average time to inspect a blade, avoid a robot or a wall, and to sweep back along a blade's contour
M_v, M_p, M_i	Number of virgin, partly inspected, and inspected blades	α	Coupling among robots ($\alpha = 0$) corresponds to no communication

the other states are constructed similarly and allow us to calculate coverage progress using the following difference equation for the number of virgin blades:

$$M_\nu(k+1) = M_\nu(k) - p_e N_\nu(k - T_e). \qquad (2)$$

Note that all parameters of this macroscopic representation of the swarm dynamics are parameters that have a direct relation with the physical characteristics of the individual team member. For instance, the encountering probability for a blade p_e is proportional to the size of the blade, a robot's sensor range, and its speed, whereas the time needed for inspection T_e is a function of the blades' circumference and the time-out chosen on the robotic platform. This property allows us to use the macroscopic model for optimizing the swarm with respect to a certain metric (here, time to complete coverage) and thus for model-based synthesis of individual robot controllers.

Figure 9 compares the prediction for the number of inspected blades $N_i(k)$, given by the total number of blades (25) minus the number of virgin blades $N_\nu(k)$ at time k, for 100 real-robot experiments with swarms of 10, 20, and 30 robots. For each experiment, the robots were randomly distributed in the environment and tracked by an overhead camera using the open-source software Swistrack (http://swistrack.sourceforge.net) [15]. The experiment was considered terminated, when the boundaries of each blade in the environment have been covered at least once. The model parameters have then been calculated based on the experimental data using a system identification process [7], [16].

### Noncollaborative Deliberative Distributed Coverage

By creating a topological map with blades as nodes and navigable routes between them as edges, robots can calculate noncollaborative, complete coverage paths online. Coverage is achieved by the exploration of a spanning tree constructed online using a depth-first search algorithm. Robots travel along the spanning tree by executing a series of reactive behaviors that allow them to navigate from one blade to any other blade in its four neighborhood. Although this approach is theoretically complete, even with limited sensor and actuator noise, robots are usually unable to accurately navigate from blade to blade, which causes the algorithm to deteriorate to probabilistic completeness. We implemented this algorithm on a team of ten Alice robots that executed the algorithm described earlier in parallel (without explicit collaboration). Upon navigation error (if positively detected by a robot), robots restarted a spanning tree and eventually completed coverage. Over ten real-robot experiments, coverage was achieved within 788 ± 375 s as opposed to 303 ± 112 s (mean ± SD) using the self-

Limited computation, communication, and available energy arising when downsizing a robotic platform seem to be pertinent challenges.

organized, reactive approach. This counterintuitive result (a reactive approach outperforms a deliberative algorithm) can be explained mainly by the fact that the necessary reactive navigation schemes that underlie the deliberative algorithm for moving from blade to blade are very time-consuming when compared with the reactive movements in the self-organized approach. In fact, one can show that the deliberative approach always outperform a reactive algorithm if the blade-to-blade navigation time is the same and noise is low enough so that a robot covers more than one blade before failing.

Collaborative Deliberative Distributed Coverage

Coverage time to completion and also redundancy can be drastically reduced by sharing information about task progress. Upon the reception of coverage progress of other robots, a robot can take this information into account for determining the next blade to which it will move by calculating the Dijkstra's shortest path to the next unexplored node. Modeling the environment as a graph with blades as nodes and edges as navigable routes between them allows us to formally investigate the key properties of our algorithms. Sensor noise, e.g., on the vision-based localization mechanism, and actuator noise, e.g., due to wheel-slip, can instead be accommodated by simulating multiple instances of the graph model. When calibrated and validated using data from real robotic experiments (ranging from simple tests for the localization subsystem to a limited
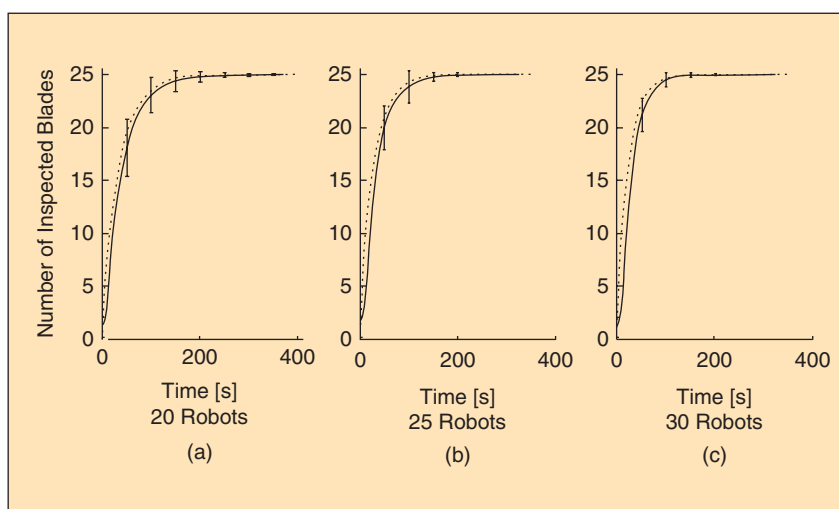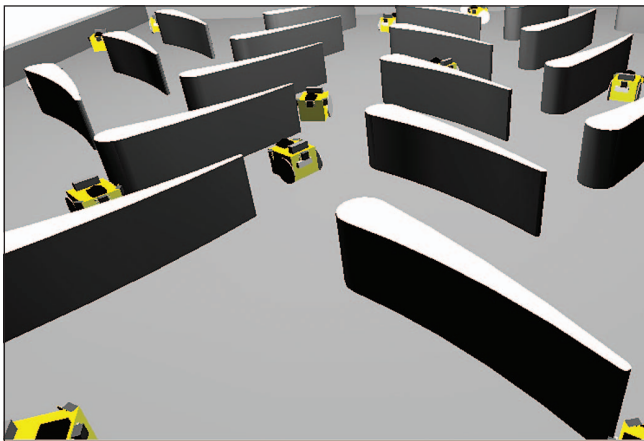
Figure 9. *Prediction of the macroscopic model (dotted line) and coverage progress of a swarm of (a) 20, (b) 25, and (c) 30 robots (100 real robot experiments per swarm size) using the self-organized, reactive controller (full line). Error bars show the standard deviation of the experiments.*
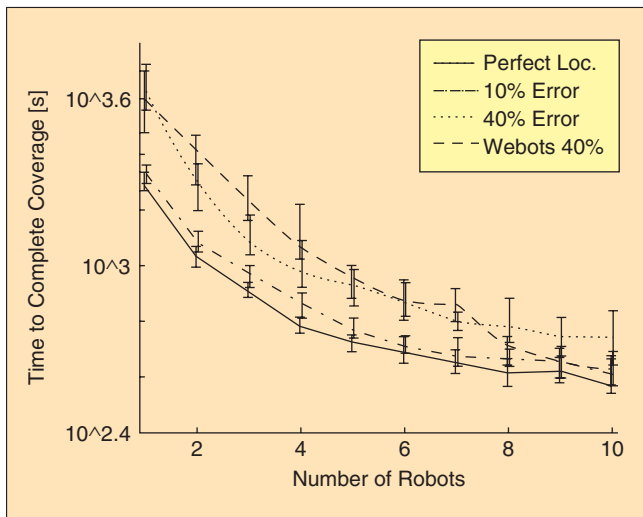
number of experiments with the full system), and realistic simulation (Figure 10), such abstract models allow us to explore a wide range of system parameters and collect statistical evidence of their dynamics. For instance, using the microscopic graph model and Webots (http://www.cyberbotics.com) simulations (100 experiments for each team size and parameter set), we can show that the collaborative algorithm gracefully degrades under the influence of erroneous localization (Figure 11) and limited/erroneous communication (Figure 12) to a randomized or non-collaborative version of the deliberative algorithm, respectively.

Finally, assuming sufficient computational power and communication bandwidth, robots can also arbitrate coverage tasks among them. For achieving a near-optimal solution, however, the environment needs to be known beforehand. We implemented such an algorithm that uses a market-based algorithm for trading coverage tasks among the robots using an external host computer for computation of shortest paths and corresponding bids. As cost function serves the length of the shortest path over all coverage tasks allocated to one robot, which is an instance of the traveling salesman problem. To take into account robot failures (ranging from wheel-slip to total loss), the coverage tasks are reallocated recurrently. Real robot results for teams of five robots for the reactive approach and the three deliberative approaches (noncollaborative, collaborative, market-based) are compared in Figure 13.

Discussion

Self-organized and reactive algorithms have been shown to be very competitive on a platform with limited capabilities and

Figure 10. Realistic simulation of the inspection scenario using the embodied simulator Webots from Cyberbotics, Ltd.

Figure 12. Median time to complete coverage using the collaborative, reactive algorithm when the communication range is limited (microscopic discrete event system simulation, 100 experiments per configuration).
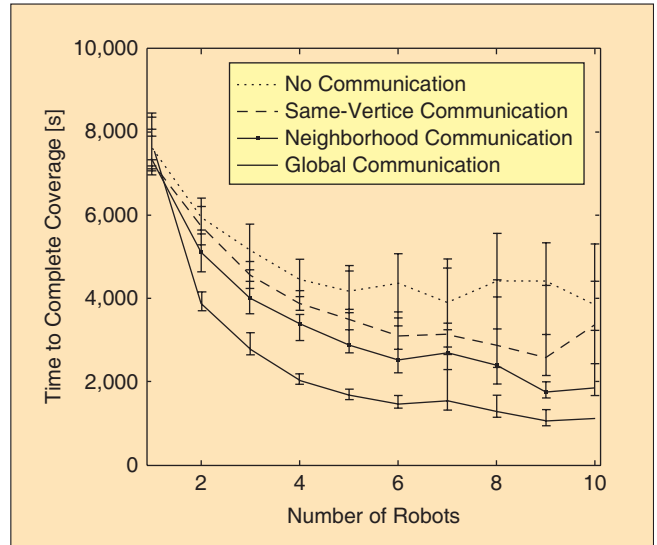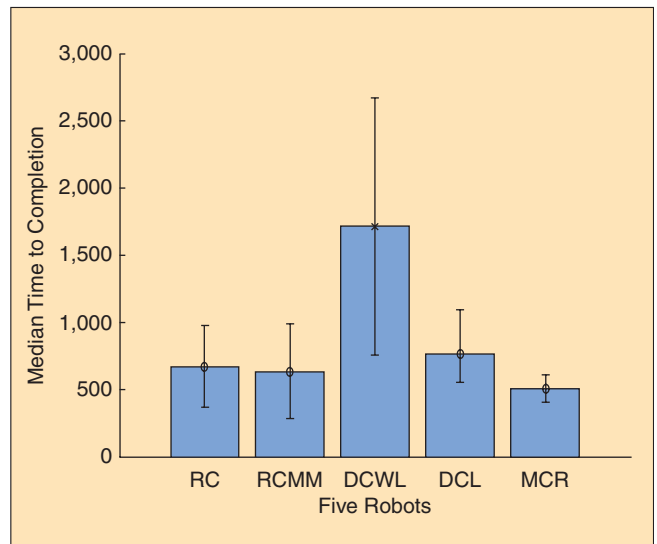

Figure 11. Median coverage time and 95% confidence interval for global communication and different localization errors for microscopic discrete event system simulation (100 experiments each) and the collaborative, deliberative algorithm. Results from realistic simulation (100 experiments per team size in Webots) are superimposed.

Figure 13. Experimental results with five miniature robots for the reactive algorithms without (RC) and with (RCMM) collaboration, as well as for the deliberative non-collaborative (DCWL), collaborative (DCL), and market-based algorithm (MCR).

might allow for even further downscaling of the robotic platform due to the minimal requirements on the robotic unit. However, reactive solutions seem to be best suited for regular environments. For instance, in our experiments, all blades have the same size and a single time-out parameter is sufficient. In a real turbine, however, the size of each blade changes as a function of its stage, and an optimal algorithm would require the calibration of additional time-outs—given that a robot could estimate the stage it is currently processing. This information in turn, will enable more deliberative approaches, which might then become favorable over fully reactive solutions for performance reasons.

Indeed, localization appears to remain a major challenge to 1) associate collected sensory information with the location where it was recorded and 2) enhance the performance by allowing robots to communicate using a common frame of reference. Using markers, either optical- or radio-based, e.g., radiofrequency identification (RFID) tags, is an accepted policy but limited to man–made environments. Optical markers scale badly, in particular when on-board processing is limited. Possible solutions are relative coding schemes or relative range and bearing systems, which are however difficult to obtain on miniature robotic platforms. Centralized beacons are an alternative that combine radio and US emissions [17]. In the turbine inspection scenario, these could be mounted on holes placed in regular intervals along the turbine that were originally foreseen for borescope inspection. However, the narrow, highly structured environment within the turbine will make time–of–flight measurements of US signals difficult due to unpredictable reflections and echoes.

From a safety and quality assurance perspective, provably complete deliberative approaches seem to be preferable to reactive approaches. However, deliberative algorithms have shown to be strongly affected by sensor and actuator noise, which causes them to deteriorate to probabilistic approaches. Also, the possibility of physically getting stuck, which will potentially require dismantling the turbine at the very end, is independent from the chosen control paradigm. For coping with these issues, rethinking of current approaches for algorithmic design is necessary, and new methods for modeling unreliable systems have to be developed. A similar transition has been already undergone in the simultaneous localization and mapping (SLAM) community, where uncertainty is explicitly taken into account for algorithmic design. In miniature multirobot systems and swarm robotics, only few modeling approaches that reflect the probabilistic nature of the system have been developed. Such models are however necessary in order for self-organized or reactive approaches to become a viable alternative for engineering-dependable (i.e. predictable) miniature multirobot systems (see [18]).

Although the limitation of our experiments to differential drive robots seems reasonable as the miniature robotic platform used in this article has been readily endowed with drives made out of fibrillar adhesives [19], allowing them to climb up a wall, and also magnetic wheels are being used on slightly larger platforms [9]–[11], we believe that the regular structure of the turbine environment is more suited to locomotion by a customized truss-climbing mechanism, which would also ease localization by node-counting. We note that the energy consumption and navigation accuracy of the chosen locomotion method might vary drastically and thus strongly influences the remaining degrees of freedom for designing the whole system.

Conclusion and Outlook

This work systematically explores algorithms for the distributed boundary coverage problem on a turbine inspection case study with respect to varying amounts of planning and coordination. The presented approaches range from minimalist reactive schemes to highly coordinated, deliberative algorithms. It turns out that minimalist approaches yield comparably good performance (in terms of time to completion) when the amount of sensor and actuator noise in the system is high or when the available resources are limited, which has been illustrated in particular with respect to localization. As soon as additional resources and capabilities become available to the platform, we also show that their use is beneficial, even if the information they provide is unreliable. In this case, the additional benefit of employing more advanced hardware and algorithms becomes marginal when compared with its cost.

Limited computation, communication, and available energy arising when downsizing a robotic platform seem to be pertinent challenges; improvements in technology will then lead to applications of the lessons learned in this work on even smaller domains. The commercial potential of such approaches is, however, not yet clear, as only few applications and real–world use cases for subminiature inspection systems are imaginable given the technological barriers still to be overcome. In our work so far, we were particularly concerned neither with human–swarm interfaces nor with expert systems that extract meaningful information from the sensory information collected by the robot team. Although seemingly independent from the multirobot coordination problem, it is likely that potential expert systems will need to control the collective behavior of the swarm, e.g, for guiding it toward points of particular interest. In this case, synthesis methodologies are necessary for generating the necessary individual behavior. Finally, for moving toward real applications, currently available sensor technology for inspection (e.g., US, eddy current, optical) needs to be evaluated for its potential to be used in situ and integrated into miniature robotic platforms.

Acknowledgments

Keywords

Swarm robotics, turbine inspection, self-organization, distributed coverage, networked robotic systems.

References

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

[2] E. Wong and J. S. Litt, "Autonomous multi-agent robotics for inspection and repair of propulsion systems," in *Proc. AIAA 1st Intelligent Systems Technical Conf.*, Chicago, IL, 2004, AIAA 2004-6364.

[3] *Acceptable Methods, Techniques, and Practices—Aircraft Inspection and Repair*, Federal Aviation Administration AC 43.13-1B, 1998.

[4] J. S. Litt, E. Wong, M. Krasowski, and L. Greer, "Cooperative multi-agent mobile sensor platforms for jet engine inspection—Concept and implementation," in *Proc. IEEE Int. Conf. Integration of Knowledge Intensive Multi-Agent Systems*, Cambridge, MA, 2003, pp. 716–721.

[5] S. Garg, "Controls and health management technologies for intelligent aerospace propulsion systems," in *Proc. 42nd Aerospace Sciences Meeting and Exhibit*, Reno, NV, 2004, AIAA 2004-949.

[6] G. W. Hunter, "Morphing, self-repairing engines: A vision for the intelligent engine of the future," in *Proc. AIAA/ICAS Int. Air and Space Symp.*, 2003.

[7] N. Correll, "Coordination schemes for distributed boundary coverage with a swarm of miniature robots: Synthesis, analysis and experimental validation." [Online]. Ph.D. thesis, Ecole Polytechnique Fédérale, Lausanne, Switzerland, 2007. Available: http://infoscience.epfl.ch/record/110263

[8] H. Choset, "Coverage for robotics—A survey of recent results," *Ann. Math. Artif. Intell.*, vol. 31, pp. 113–126, 2001.

[9] F. Tâche, W. Fischer, R. Siegwart, R. Moser, and F. Mondada, "Adapted magnetic wheel unit for robots inspecting narrow pipes and complex shaped structures," in *Proc. 2007 IEEE/ASME Int. Conf. Advanced Intelligent Mechatronics*, Zurich, Switzerland, Sept. 2007, pp. 1–6.

[10] W. Fischer, F. Tâche, and R. Siegwart, "Magnetic wall climbing robot for thin surfaces with specific obstacles," in *Proc. Int. Symp. Field and Service Robotics (FSR)*, Chamonix, Switzerland, July 2007, to be published.

[11] M. Friedrich, W. Galbraith, and G. Hayward, "Autonomous mobile robots for ultrasonic NDE," in *Proc. IEEE Ultrasonics Symp.*, Oct. 2006, pp. 902–905.

[12] G. Caprari and R. Siegwart, "Mobile micro-robots ready to use: Alice," in *Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems*, Alberta, Canada, Oct. 2005, p. 32953300.

[13] A. Martinoli, K. Easton, and W. Agassounon, "Modeling swarm robotic systems: a case study in collaborative distributed manipulation," *Int. J. Robot. Res.*, vol. 23, no. 4, pp. 415–436, 2004.

[14] K. Lerman, C. V. Jones, A. Galstyan, and M. J. Matarić, "Analysis of dynamic task allocation in multi-robot systems," *Int. J. Robot. Res.*, vol. 25, no. 4, pp. 225–242, 2006.

[15] N. Correll, G. Sempo, Y. Lopez de Meneses, J. Halloy, J.-L. Deneubourg, and A. Martinoli, "SwisTrack: A tracking tool for multi-unit robotic and biological systems," in *IEEE/RSJ Int. Conf. Intelligent Robots and Systems (IROS)*, 2006, pp. 2185–2191.

[16] N. Correll and A. Martinoli, "System identification of self-organizing robotic swarms," in *Proc. 8th Int. Symp. Distributed Autonomous Robotic Systems (DARS)* (Distributed Autonomous Robotic Systems VII). Springer-Verlag, pp. 31–40, 2006.

[17] N. Priyantha, "The cricket indoor location system," Ph.D. thesis, Massachusetts Institute of Technology, Cambridge, MA, June 2005.

[18] A. F. T. Winfield and J. Nembrini, "Safety in numbers: Fault tolerance in robot swarms," *Int. J. Model. Ident. Control*, vol. 1, no. 1, pp. 30–37, 2006.

[19] M. Greuter, G. Shah, G. Caprari, F. Tâche, R. Siegwart, and M. Sitti, "Toward micro wall-climbing robots using biomimetic fibrillar adhesives," in *Proc. 3rd Int. Symp. Autonomous Minirobots for Research and Edutainment*, Awara-Spa, Fukui, Japan, 2005, pp. 39–46.

Nikolaus Correll received his diploma in electrical engineering from Swiss Federal Institute of Technology in Zurich (ETHZ), and Ph.D. degree in computer science from the Swiss Federal Institute of Technology in Lausanne (EPFL). Since November 2007, Nikolaus is a postdoctoral associate at the Computer Science and Artificial Intelligence Laboratory, Massachusetts Institute of Technology, Cambridge. During 2003–2007, he pursued graduate studies with the Swarm-Intelligent Systems Group, EPFL. During his master's studies, he spent a term at the Department of Automatic Control, at Lund Institute of Technology, Lund, Sweden, and his master's thesis was about collaborative coverage at the Collective Robotics Group, California Institute of Technology, Pasadena, where he continued to work as research assistant. His research interests are the analysis and design of large-scale distributed robotic systems, mixed animal–robot societies, and tracking and monitoring of collective systems. Correll serves as reviewer for several international robotics journals and proceedings. He is the recipient of the Best Paper Award at the 9th International Symposium on Distributed Autonomous Robotic Systems in 2006 (with Alcherio Martinoli) and an IFRR Student Travel–Fellowship Award at the 10th International Symposium on Experimental Robotics in 2006 (with Alcherio Martinoli and Samuel Rutishauser). He is a Member of the IEEE.

Alcherio Martinoli holds a diploma in electrical engineering from the Swiss Federal Institute of Technology in Zurich (ETHZ), and a Ph.D. degree in computer science from the Swiss Federal Institute of Technology in Lausanne (EPFL). Since 1 May 2008, he is an associate professor at the School of Architecture, Civil and Environmental Engineering and the head of the Distributed Intelligent Systems and Algorithms Laboratory. Prior to this, he served as Swiss National Science Foundation Assistant Professor at the School of Computer and Communication Sciences and head of the Swarm-Intelligent Systems Group. Before joining EPFL, he carried out research activities at the Institute of Biomedical Engineering of the ETHZ, the Institute of Industrial Automation of the Spanish Research Council in Madrid, Spain, and at the California Institute of Technology, Pasadena. His research interests focus on techniques to design, control, model, and optimize distributed intelligent systems, including swarms of robots, sensor and actuator networks, intelligent vehicles, and micro- and nanosystems. Martinoli has published more than 70 articles in his area of expertise. He is currently a member of the editorial board of the new journal on Swarm Intelligence published by Springer-Verlag and reviewer for several major international journals and conferences in his area of expertise. He has been general cochair for IEEE SIS 2005, program cochair for ANTS 2006, steering committee member for ROBOCOMM since 2007, and associate editor for IEEE ICRA from 2007 to 2009. He received the 2006 Best Teacher Award for computer and communication sciences from the EPFL General Student Association and the Best Paper Award at DARS 2006 (with Nikolaus Correll). He is a Member of the IEEE.

Address for Correspondence: Nikolaus Correll, Computer Science and Artificial Intelligence Laboratory, Massachusetts Institute of Technology Cambridge, MA 02139, USA. E-mail: nikolaus@csail.mit.edu.