N. Correll and A. Martinoli. Towards Multi-Robot Inspection of Industrial Machinery - From Distributed Coverage Algorithms to Experiments with Miniature Robotic Swarms. IEEE Robotics & Automation Magazine. 16(1):103-112.

TOWARDS MULTI-ROBOT INSPECTION OF INDUSTRIAL MACHINERY

FROM DISTRIBUTED COVERAGE ALGORITHMS TO EXPERIMENTS WITH MINIATURE ROBOTIC SWARMS

Nikolaus Correll[1]
Computer Science and Artificial Intelligence Laboratory, Massachusetts Institute of Technology, Cambridge, MA 02139, USA

Alcherio Martinoli[1]
Distributed Intelligent Systems Laboratory, École Polytechnique Fédérale Lausanne
CH-1015 Lausanne, Switzerland

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 multi-robot path planning, modeling of self-organized robotic systems, and implementation of proof-of-concept experiments with real miniature robots. While other research tackles challenges that arise from moving within 3D 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 towards truly autonomous parallel robotic inspection of complex engineered structures.

Keywords: Swarm Robotics, Turbine Inspection, Self-Organization, Distributed Coverage, Networked Robotic Systems

INTRODUCTION

For certain tasks multi-robot systems are a promising alternative to a single robot solution, as 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 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. Besides locomotion constraints that are specific to the environment, such a scenario poses numerous design challenges such as limited inter-robot communication, determining position or relative range and bearing [3], and design of efficient and robust algorithms for coordination of a robot team. Benefits and challenges of miniature multi-robot coverage are well-illustrated by the automatic inspection of (jet) turbines (Figure 1), which is a promising commercial application [4]. In order 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

[1] This work has been carried out while both authors were with the Swarm-Intelligent Systems Group, EPFL, Switzerland

usually performed visually using borescopes as well as using ultra-sound and eddy current sensors [5], a process which is time-consuming and cost-intensive, in particular 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 which could be released into the turbine while still attached to the wing [10]. With the immediate prospect to reduce down-time 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 based on smart sensors and actuators [11]. Here the deployment of mobile sensors rather than installation of permanent sensors [12] is a compromise between increased system cost and the benefits arising with an in-situ inspection [5].

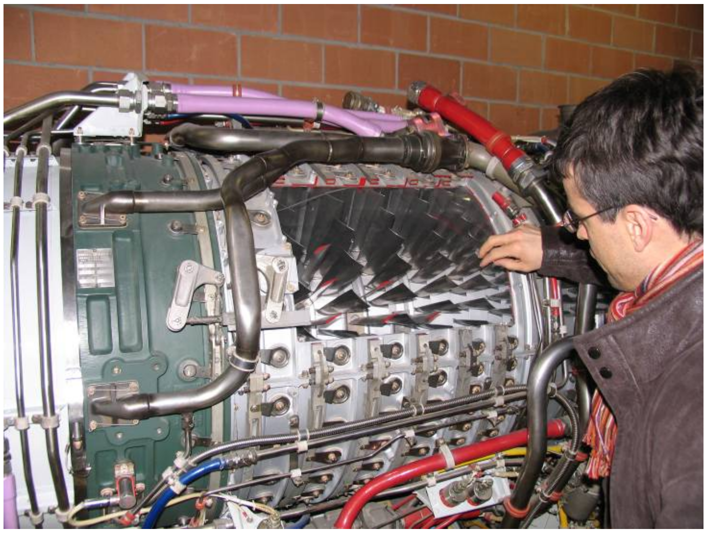

Figure 1: The compressor section of a jet turbine. The internal dimensions are within the same order of magnitude as those of the miniature robotic systems used in this paper.

While this idea is intellectually appealing and could pave the way for other similar applications in inspection of, potentially complex, engineered or natural structures, it involves a series of technical challenges that dramatically 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 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 renders certain control approaches – in particular 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 analysis of algorithms from a probabilistic perspective. Finally, commands by human users that address properties on the swarm level need to be synthesized into control inputs to the individual robots.

The focus of our work [1] is on algorithms for coordinating a robot swarm for coverage [6] 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 for instance [7] or [8], and [9], respectively, and references therein). We undertake experimentation with real hardware (Figure 2), which serves both as validation and motivation for our algorithms, where emphasis is on robustness with respect to sensor and actuator noise of minimalist platforms in use.
In the following sections we will first summarize the design challenges imposed by our case study and then describe our experimental setup and hardware that we developed. Finally we will compare results from both probabilistic and deterministic control strategies.

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

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 [4], which would be also 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, in particular if it is to be robust enough for 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.

- Due to 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 paper tackle miniaturization, energy limitations, and limited range communication experimentally, although we are not exploring other locomotion principles than wheeled differential-drive robots.
Besides 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 (three-dimensional) 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 in order to achieve a desired collective behavior, for instance for inspecting more closely a certain region of the structure.

A MINIATURE PLATFORM FOR AUTONOMOUS INSPECTION

Our robotic inspection nodes (Figure 3) base on the Alice miniature robot [13], developed by Gilles Caprari at the Autonomous System Laboratory, EPFL. The Alice has a cubic shape of approximately 2cm side length, and is operated by a PIC 16F877 Microprocessor (4MHz, 384 byte of RAM, 8kB ROM). Driven by two watch (stepper) motors in a differential-drive configuration, it can travel with a top-speed of 4cm/s. It is endowed with 4 IR modules which can serve as very crude proximity sensors (up to 3cm) and local communication devices (up to 6 cm in range), providing a simple communication channel at around 500bps, which can also be used for crude inter-robot local positioning. Its energetic autonomy with a 40mAh (at 4.5V) NiMH rechargeable battery ranges from 10min to 10h, depending on the actuators and sensors used (refer to Table 1 for detailed energy consumption of selected components). The reason for the extreme differences in autonomy is not the actual cumulative power consumption but rather due to the maximal possible drain that the battery is supporting. In practice, significant voltage drops are already observed for drains of more than 0.5C (1C corresponds to the nominal capacity), which makes simultaneous operation of camera and radio module, which are described below, impossible.
To improve computational and communication capabilities for ad-hoc networking among the robotic swarm and to eventually transmit recorded data to a base station, we developed an extension board, providing a Texas Instruments (TI) MSP430 microprocessor (2kB RAM, 60kB ROM), a TI CC2420 radio (ZigBee™ ready), and 4MByte Flash-Memory. Conveniently, the module can be programmed in TinyOS, 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 down-sampled to 30x30 pixels in RGB color.

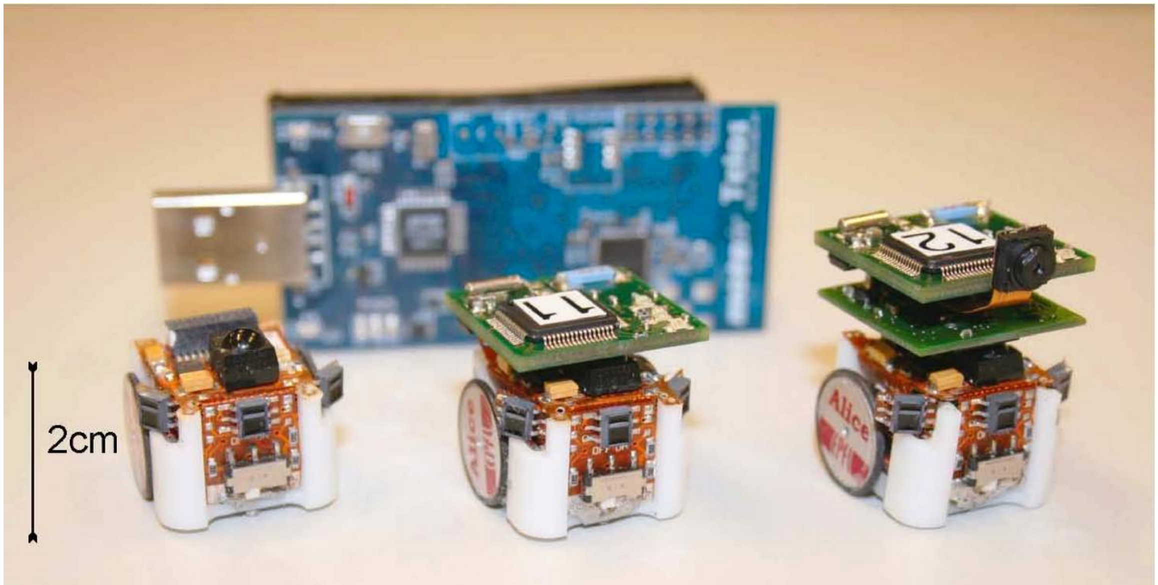

Figure 3: The miniature robot Alice (2cm x 2cm x 2cm) endowed with extension modules providing ad-hoc networking (middle) and imaging capabilities (right). A moteiv Telos mote, which inspired the design of the communication module that is shown in the background.

Using a PIC40F4620 with 4kB RAM at 32MHz for image acquisition and processing, the Alice is able to take pictures at a rate of around 2Hz (Figure 4), as well as uniquely identify color markers in the environment (Figure 5). The Alice and the extension modules communicate via an I2C two-wire bus (a block-diagram is shown in **Error! Reference source not found.**). With the two extension modules mounted, the inspection robot fits well into a volume of 2cm x 2cm x 3cm.

Individual Subsystem	***Energy Consumption***
Alice, motors off	4.5mW
Alice, full-speed drive	15mW
Radio module active	60mW
Radio module sleep	<1mW
Camera module active	60mW
Camera module sleep	15mW

Table 1: Energy consumption for selected individual sub-systems of the inspection platform.

Experimental Setup

We simplify the real 3D 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 5x5 pattern on a 60cm x 65cm large arena (Figure 2) made from 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; in particular when a robot's antenna is incidentally in direct contact with a blade.

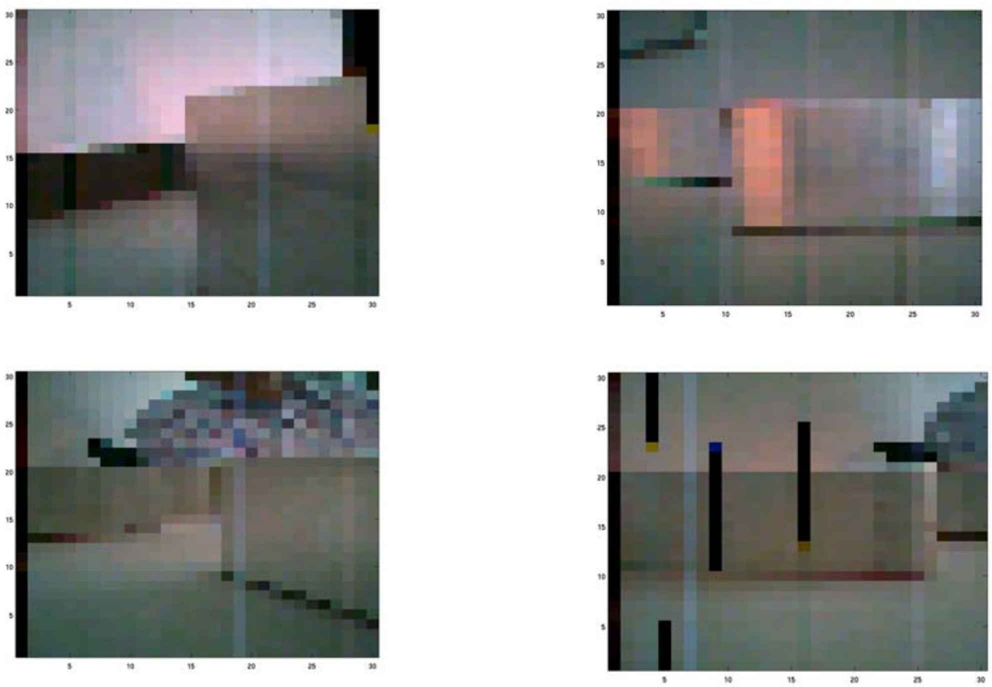

Figure 4: Pictures (30x30 pixels) taken by the on-board camera and transmitted over the radio with 72 packets of 25 bytes. Vertical black stripes indicate packet loss. The arena boundary (painted in black) can be seen in the top left picture; in the bottom row the experimenter's upper part of the body is visible in the background.

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 3 color channels (red, green, and blue) is used to encode 3 bits 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 over 100 experiments) for correctly identifying a blade.

Distributed Coordination Schemes for Multi-robot Inspection

In our experiments we are not concerned with detection or mapping of flaws, but rather with the individual and group motion 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 on their requirements on 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 dispersion of the robots in the environments. In this scenario, radio and camera can potentially be used for inspection, but require off-line 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 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.

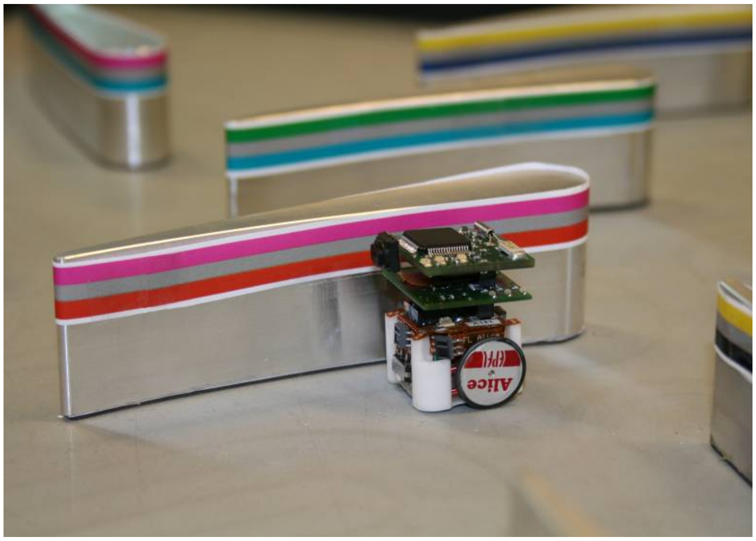

Figure 5: The fully equipped Alice in an environment with colored markers. The two-color code (the middle bar serves as reference) can be recognized with 95% accuracy.

Reactive Inspection using Local Communication

The motivation for a fully reactive approach is the potential for its implementation on extremely minimalist robotic platforms. The basic idea is to *eventually* cover the environment by moving from blade to blade reactively. Local communication is used for enhancing dispersion in the environment. We will first describe the robot behavior, and then present a methodology for modeling and predicting coverage performance.

Robot Behavior

The necessary behaviors for circumnavigating all blades and avoiding collisions can be decomposed 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 (10s 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 towards 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 is summarized by the Finite State Machine (FSM) diagram depicted in Figure 7.

Robots can communicate locally by modulating the signal send 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 inspection. In case of a front-to-front encounter, the robot with the blade to its left hand side will abandon inspection, whereas in case of a back-to-front encounter the robot that detects the other robot by its front sensors will abandon inspection. The behavior of the robots and sample trajectories are illustrated in Figure 8.

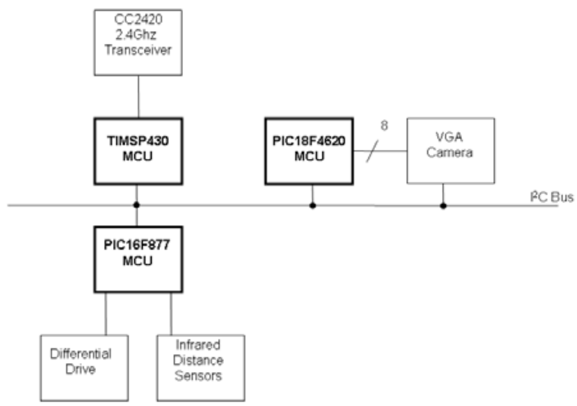

Figure 6: Block-Diagram of the inspection platform measuring around 2cm x 2cm x 2cm, endowed with 2 watch motors for differential drive, a 2.4GHz ZigBee™- compliant wireless radio, a VGA camera, and three microcontrollers connected by an I^2C two-wire bus.

Probabilistic Modeling

Due to 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 above. 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 [14], [15].

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. Given the number $M_v(k)$ of uncovered blades at time k, and the probability to encounter *one* blade as p_e, the probability for encountering a virgin blade at time 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 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 (see Figure 7). 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 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.

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. α=0 corresponds to no communication.

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

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

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

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

$$(2) \qquad M_v(k+1) = M_v(k) - p_e N_v(k - T_e)$$

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 number of virgin blades $N_v(k)$ according to (2) and the total number of blades (25 blades), for 100 real-robot experiments with swarms of 10, 20, and 30 robots. For each experiment, robots were randomly distributed in the environment and tracked by an overhead camera using the open-source software *Swistrack*[2] [19]. 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 [1].

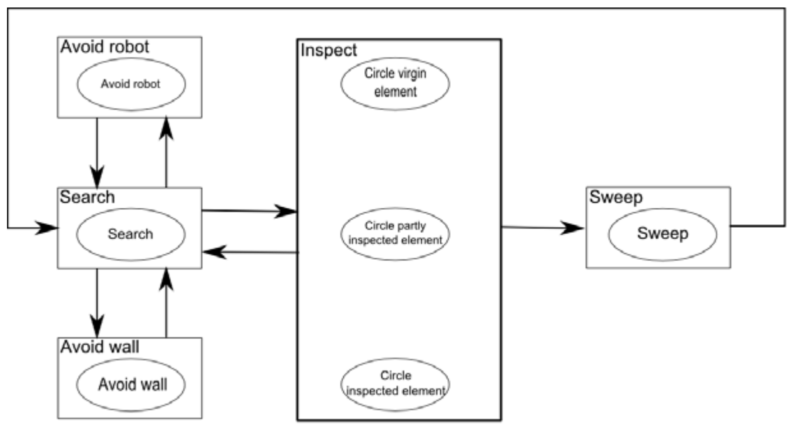

Figure 7: Finite State Machine (FSM) and Probabilistic Finite State Machine (PFSM). The FSM (squares) is of lower granularity than the PFSM (ellipses) and does not consider the state of an element (virgin, partly inspected, or inspected) as this information is not known to an individual robot.

Non-Collaborative Deliberative Distributed Coverage

By creating a topological map with blades as nodes and navigable routes between them as edges, robots can calculate non-collaborative, complete coverage paths on-line. Coverage is achieved by 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 4-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 10 Alice robots that executed the algorithm described above in parallel (without explicit collaboration). Upon navigation error (if positively detected by a robot) robots restarted a spanning tree and eventually completed coverage. Over 10 real-robot, experiments coverage was achieved within 788±375s as opposed to 303±112s (mean +/- std.) using the self-organized, reactive approach. This counter-intuitive 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 compared with the reactive movements in the self-organized approach. In fact, one can show that the

[2] http://swistrack.sourceforge.net

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.

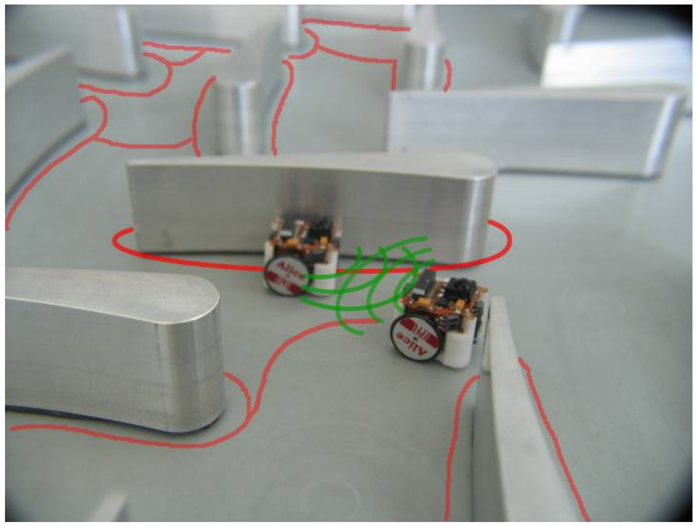

Figure 8: Using the self-organized, reactive controller, robots are reactively moving through the environment and inspect blades for a fixed amount of time. Blades are then left as soon as the tip is reached. Robots and blades are differentiated using their infra-red sensors.

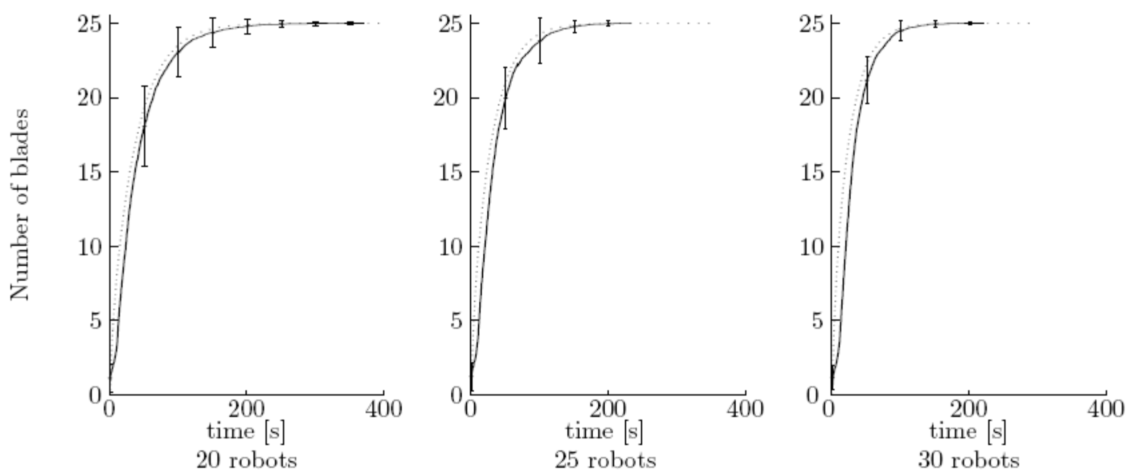

Figure 9: Prediction of the macroscopic model (dotted line) and coverage progress of a swarm of 20, 25, and 30 robots (100 real robot experiments per swarm size) using the self-organized, reactive controller (full line). Error bars show the standard deviation of the experiments.

Collaborative Deliberative Distributed Coverage

Coverage time to completion but also redundancy can be drastically reduced by sharing information about task progress. Upon 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 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 sub-system to a limited number of experiments with the full system [2]), 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*[3] 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.

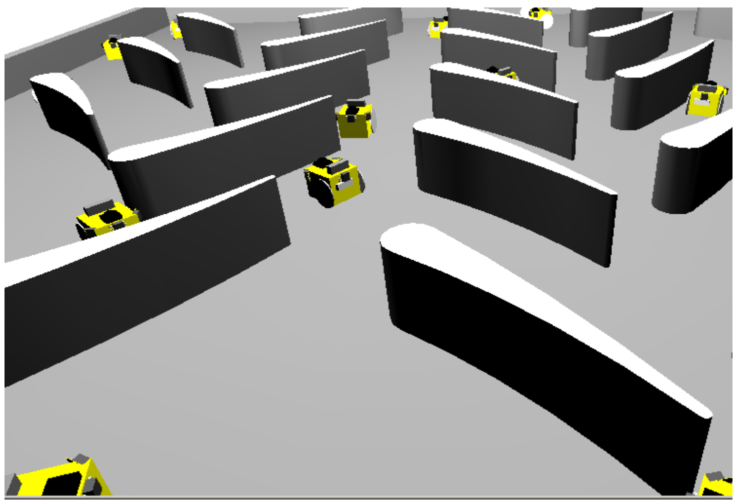

Figure 10: Realistic simulation of the inspection scenario using the embodied simulator *Webots*™ from Cyberbotics, Ltd.

Finally, assuming sufficient computational power and communication bandwidth, robots can also arbitrate coverage tasks among them. For achieving an 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. 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. In order to take into account robot failures (ranging from wheel-slip to total loss), the coverage tasks are re-allocated recurrently. Real robot results for teams of 5 robots for the reactive approach and the three deliberative approaches (non-collaborative, collaborative, near-optimal) are compared in Figure 13.

DISCUSSION

Self-organized/reactive algorithms have been shown to be very competitive on a platform with limited capabilities and might allow for even further downscaling of the robotic

[3] http://www.cyberbotics.com

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.

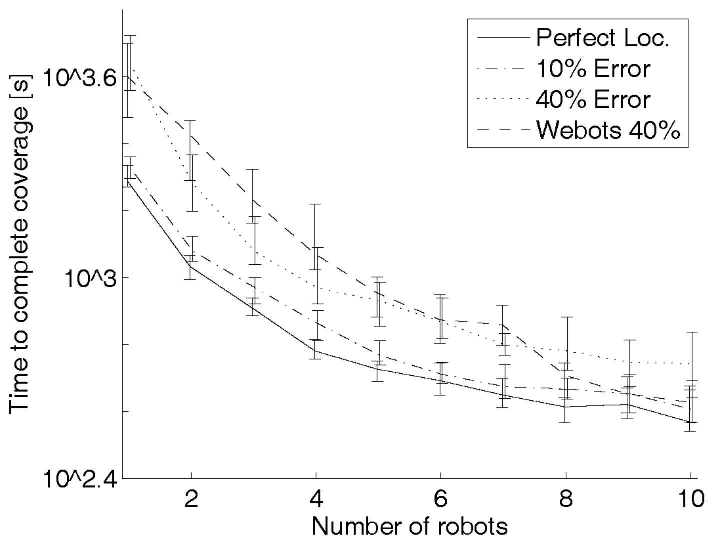

Figure 11: Median coverage time and 95% confidence interval for global communication and different localization errors for microscopic discrete event simulation (100 experiments each) and the collaborative, deliberative algorithm. Results from realistic simulation (100 experiments per team size in Webots)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

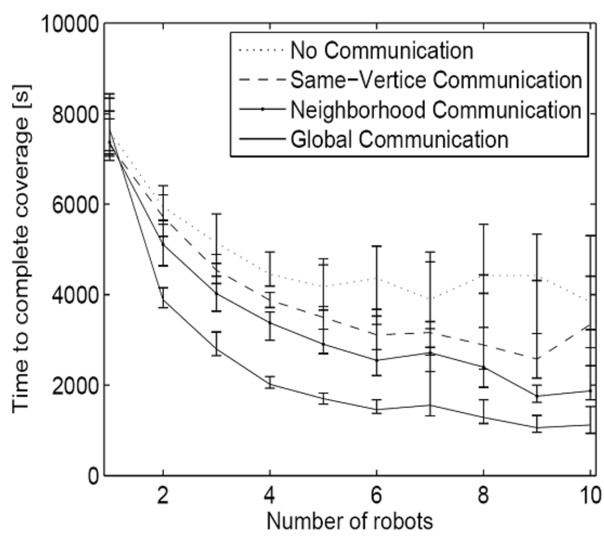

Figure 12: Median time to complete coverage using the collaborative, reactive algorithm when the communication range is limited (microscopic discrete event simulation, 100 experiments per configuration).

Indeed, localization appears to remain a major challenge in order to (a) associate collected sensory information with the location where it was recorded, and (b) enhance performance by allowing robots to communicate using a common frame of reference. Using markers, either optical- or radio-based, e.g., radio frequency identification (RFID) tags, is an accepted policy but limited to man-made environments. Optimal 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. An alternative are centralized beacons that combine radio and Ultra-Sound (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.

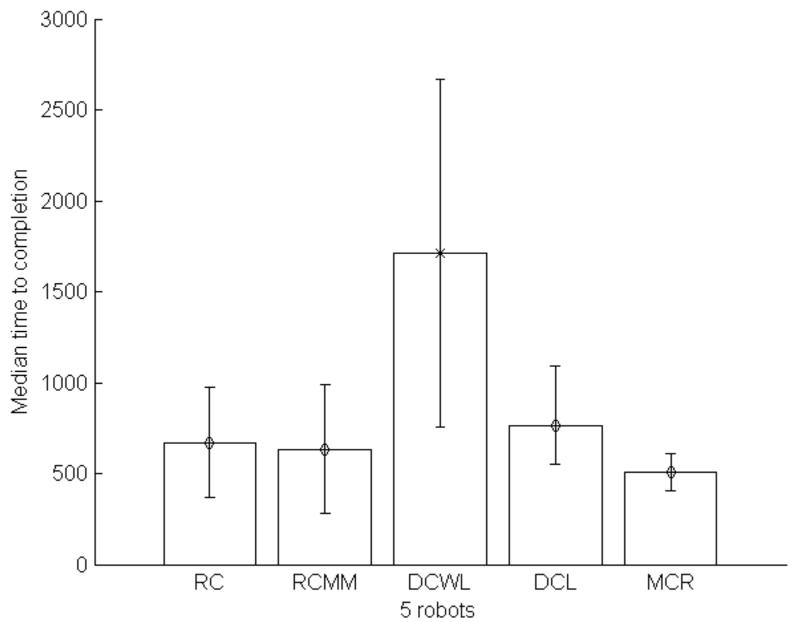

Figure 13: Experimental results with 5 miniature robots for the reactive algorithms without (RC) and with (RCMM) collaboration, as well as for the deliberative non-collaborative (DCWL), collaborative (DCL), and near-optimal algorithm (MCR).

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, re-thinking 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 multi-robot 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/reactive approaches to become a viable alternative for engineering-dependable (i.e. predictable) miniature multi-robot systems (see also the work of [18]).

While the limitation of our experiments to differential drive robots seems reasonable as the miniature robotic platform used in this paper has been readily endowed with drives made out of fibrillar adhesives [16], allowing them to climb up a wall, and also magnetic wheels are being used on slightly larger platforms [7][8][9], 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 to its cost.

Limited computation, communication, and available energy arising when down-sizing 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 sub-miniature inspection systems are imaginable given the technological barriers still to be overcome. In our work so far, we were neither particularly concerned 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 multi-robot coordination problem, it is likely that potential expert systems will need to control the collective behavior of the swarm, for instance for guiding it towards points of particular interest. In this case, synthesis methodologies are necessary for generating the necessary individual behavior. Finally, for moving towards real applications, currently available sensor technology for inspection (e.g., ultrasound, eddy current, optical) needs to be evaluated for its potential to be used in-situ and integrated into miniature robotic platforms.

ACKNOWLEDGEMENTS

This work has been initiated while the authors were at the California Institute of Technology, primarily supported by the NASA Glenn Center and in part by the Caltech Center for Neuromorphic Systems Engineering under the US NSF Cooperative Agreement ERC-9402726. Nikolaus Correll is currently sponsored by a Swiss postdoctoral NSF grant (contract Nr. PBEL2-118737) and Alcherio Martinoli by a Swiss

NSF professorship (contract Nr. PP002-116913). This work has been carried by both authors at EPFL with the sponsorship of a Swiss NSF grant (contract Nr. PP002-68647/1). The authors would like to thank Xavier Raemy for his support with hardware development, and Samuel Rutishauser and Patrick Amstutz for collaborating on the deliberative control approaches.

REFERENCES

[1] Correll, N. Coordination schemes for distributed boundary coverage with a swarm of miniature robots: Synthesis, analysis and experimental validation, Ph.D. thesis #3919, Ecole Polytechnique Fédérale Lausanne (2007).

[2] N. Correll and A. Martinoli. System Identification of Self-Organizing Robotic Swarms. In The 8th Int. Symp. on Distributed Autonomous Robotic Systems (DARS), Minneapolis, USA. Distributed Autonomous Robotic Systems VII, pages 31-40. Springer Verlag, 2006.

[3] Pugh, J., and A. Martinoli. Relative Localization and Communication Module for Small-Scale Multi-Robot Systems. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 188 - 193, Orlando, FL, USA, April, 2006.

[4] Wong, E. and J.S. Litt. Autonomous Multi-Agent Robotics for Inspection and Repair of Propulsion Systems. In *Proceedings of the AIAA 1st Intelligent Systems Technical Conference*. 2004. Chicago, IL, USA.

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

[6] H. Choset, Coverage for robotics—a survey of recent results, Annals of Mathematics and Artificial Intelligence 31 (2001) 113–126.

[7] Tâche, F., W. Fischer, R. Siegwart, R. Moser, and F. Mondada. Adapted Magnetic Wheel Unit for Robots Inspecting Narrow Pipes and Complex Shaped Structures. In *Proceedings of the* 2007 IEEE/ASME International Conference on Advanced Intelligent Mechatronics, Zurich, Switzerland, September, 2007. To appear.

[8] Fischer, W., F. Tâche, and R. Siegwart. Magnetic Wall Climbing Robot for Thin Surfaces with Specific Obstacles. In *Proceedings of the Inernational Symposium on Field and Service Robotics (FSR)*, Chamonix, Switzerland, July, 2007. To Appear.

[9] Friedrich, M., W. Galbraith and G. Hayward. Autonomous mobile robots for ultrasonic NDE. In *Proceedings of the IEEE Ultrasonics Symposium*, pages 902-905, October, 2006.

[10] Litt, J.S., E. Wong, M. Krasowski, and L. Greer. Cooperative Multi-Agent Mobile Sensor Platforms for Jet Engine Inspection - Concept and Implementation. In *Proceedings of the IEEE International Conference on Integration of Knowledge Intensive Multi-Agent Systems*. 2003. Cambridge, MA, USA.

[11] Garg, S. Controls and Health Managment Technologies for Intelligent Aerospace Propulsion Systems. In *Proceedings of the 42nd Aerospace Sciences Meeting and Exhibit*. 2004. Reno, NV, USA.

[12] Hunter, G.W. Morphing, Self-Repairing Engines: A Vision for the Intelligent Engine of the Future. In *Proceedings of the AIAA/ICAS International Air & Space Symposium*. 2003.

[13] Caprari, G. and R. Siegwart. Mobile Micro-Robots Ready to Use: Alice. In *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems*, Alberta, Canada, October, 2005.

[14] Martinoli, A., K. Easton, and W. Agassounon, Modeling Swarm Robotic Systems: A Case Study in Collaborative Distributed Manipulation. *International Journal of Robotics Research*, 2004. **23**(4): p. 415-436.

[15] Kristina Lerman, Chris V. Jones, Aram Galstyan and Maja J. Matari´c, “Analysis of Dynamic Task Allocation in Multi-Robot Systems.” *Int. J. of Robotics Research*, **25**(4): pp. 225-242, 2006.

[16] Greuter, M., Shah, G., Caprari, G., Tâche, F., Siegwart, R. and Sitti, M. Toward Micro Wall-Climbing Robots Using Biomimetic Fibrillar Adhesives. In *Proceedings of the 3rd International Symposium on Autonomous Minirobots for Research and Edutainment*, Awara-Spa, Fukui, Japan, 2005.

[17] Priyantha, N. The Cricket Indoor Location System *PhD Thesis*, Massachusetts Institute of Technology, June 2005.

[18] Winfield AFT and Nembrini J, Safety in Numbers: Fault Tolerance in Robot Swarms', *Int. J. Modelling Identification and Control*, **1** (1), 30-37, 2006.

[19] 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 International Conference on Intelligent Robots and Systems (IROS)*, pages 2185-2191, 2006

AUTHOR BIO

Nikolaus Correll (M’05) earned a Diploma in Electrical engineering from Swiss Federal Institute of Technology in Zurich (ETHZ), and a Ph.D. in Computer Science from the Swiss Federal Institute of Technology in Lausanne (EPFL).

Since November 2007, Nikolaus is a post-doctoral fellow at the Computer Science and Artificial Intelligence Laboratory, Massachusetts Institute of Technology, Cambridge, USA. From autumn 2003 to autumn 2007, he pursued graduate studies with the Swarm-Intelligent Systems Group, EPFL. During his master’s studies, he spent a term with the Department of Automatic Control, at Lund Institute of Technology, Lund, Sweden. He wrote his master’s thesis about collaborative coverage at the Collective Robotics Group, California Institute of Technology, Pasadena, USA, 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.

Dr. Correll serves as reviewer for several international robotics journals and proceedings and is technical program co-chair for NANO-NETS 2008. He is the recipient of the “Best Paper Award” at the 9th Int. Symposium on Distributed Autonomous Robotic Systems in 2006 (with Alcherio Martinoli) and an “IFRR Student Travel-Fellowship Award” at the 10^{th} Int. Symposium on Experimental Robotics in 2006 (with Alcherio Martinoli and Samuel Rutishauser).

Alcherio Martinoli (M’99) has a Diploma in Electrical Engineering from the Swiss Federal Institute of Technology in Zurich (ETHZ), and a Ph.D. in Computer Science from the Swiss Federal Institute of Technology in Lausanne (EPFL).

Since May 1, 2008 he is an Associate Professor at the School of Architecture, Civil and Environmental Engineering and the head of the Distributed Intelligent Systems Laboratory. Prior to this appointment, 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, at the Institute of Industrial Automation of the Spanish Research Council in Madrid, Spain, and at the California Institute of Technology, Pasadena, U.S.A., where he is maintaining a part-time Visiting Associate position in Mechanical Engineering. His research interests focus on techniques to design, control, model, and optimize distributed, real-time, embedded systems, including swarms of robots, sensor and actuator networks, intelligent vehicles, and micro- and nano-systems.

Prof. Martinoli has published more than 70 papers in his area of expertise. He is currently associate editor for 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 co-chair for IEEE SIS 2005, program co-chair for ANTS 2006, steering committee member for ROBOCOMM 2007, associate editor for IEEE ICRA 2007 and 2008, and is steering committee member for ROBOCOMM 2009. He received from the EPFL General Student Association the 2006 Best Teacher Award for Computer and Communication Sciences and the Best Paper Award at DARS 2006 (with Nikolaus Correll).