

# Exploration of an Incremental Suite of Microscopic Models for Acoustic Event Monitoring Using a Robotic Sensor Network

Christopher M. Cianci, Jim Pugh, and Alcherio Martinoli

**Abstract**—Simulation is frequently used in the study of multi-agent systems. Unfortunately, in many cases, it is not necessarily clear how faithfully the details of the simulated model represent the behavior of the physical system. Often, the effects of the environment in which the system is to be placed are even neglected entirely. Taking into account the *entire* system (including interactions with the target environment), establishing a clear hierarchy among *multiple* levels of modeling not only enhances the fidelity of the individual models, but also emphasizes the tradeoffs inherent in each. Understanding and leveraging the full spectrum of models allows the use of fast, high-level models for exploration in the parameter space, the results of which can be verified on more precise low-level models. Here, we demonstrate the generation of a family of models for a robotic wireless sensor network engaged in an acoustic detection task. Quantitative correspondence is shown between modeling levels and with the physical system.

## I. INTRODUCTION

By the canonical definition, a sensor network is a system consisting of “many spatially distributed low-cost sensing nodes that collaborate with each other but operate autonomously, with information being routed to whichever node can best use the information.” [1] Particularly as recent focus has shifted heavily toward *wireless* sensor networks and their potential to bring “spatially distributed collaboration” closer to “low-cost” ([2], [3]), the question of how to efficiently design and manage control of such networks is of ever increasing importance.

Many of the most common sensor network applications to date have been based upon the sampling of continuously available parameters (such as temperature, humidity, or other environmental factors, as in [4], [5]), which has allowed them to take advantage of extremely low duty cycles in the interest of extending network lifetime. However, in situations where the phenomena of interest are spatially and/or temporally unpredictable, the problem becomes slightly more complicated and the response of the network to environmental changes necessitates increased dynamism in behavior.

While there have been several attempts at modeling sensor networks for data filtering [6], [7], data prediction [8], network classification [9], [10], and system performance [11], all such work that we are aware of tends to focus on a single level of modeling for a very specific aspect of the system (typically either sensing or networking). Here we propose a slightly different approach: applying a statistical *multi-level* modeling methodology which allows us to capture

the dynamics of the entire system together, at multiple levels of abstraction. This type of analysis has become commonplace in swarm-robotic systems [12], and should be equally applicable to sensor networks—which can also easily be considered multi-agent systems, just with slightly different constraints and capabilities.

Sensor networks produce spatio-temporal monitoring data which can be considered both dense, compared with traditional monitoring and measuring techniques, and sparse, compared with the information gathered and used for the control of many multi-robot systems. A robotic sensor network has the potential to capitalize on benefits from the extremes of both systems: power management and explicit communication from the network; self deployment, reconfiguration, and collection afforded by the ability of the robots to self-locomote. Not all of these capabilities will be exploited here, but we hope to leverage these possibilities in the future.

An element frequently encountered in many distributed systems is the concept of consensus [13], which is applied in mobile networks of autonomous agents [14], in stationary wireless sensor networks [15], and as bacterial quorum in natural systems [16]. Here we use a simple fully distributed consensus algorithm as a case study.

The remainder of the article will be organized as follows: Section II explains the basic structure and motivation behind multi-level modeling, and in Section III we present a simple concrete case study (and a physical implementation of the same) as an example system which can be modeled. In Section IV, a brief overview of the different microscopic models and their construction and calibration is given (module-based, continuous spatial, discrete spatial, and discrete non-spatial). Obtained results are presented in Section V and compared in Section VI, along with concluding remarks and future directions.

## II. THE MULTI-LEVEL MODELING HIERARCHY

Performing systematic experiments directly on the target hardware system can be cumbersome, costly, time-consuming, or even impossible for logistical reasons such as safety or availability—particularly in multi-agent systems. However, by demonstrating correspondence with higher abstraction layer representations of the system, we can gather and analyze additional information which may eventually be applied back to the design and control of the target system. In this context, simulation can therefore be a very useful tool for bridging the gap between theory and experiment. It is not intended to be a substitute for real experiments, but rather

The authors are members of the Swarm-Intelligent Systems Group at the École Polytechnique Fédérale de Lausanne, CH-1015 Lausanne, Switzerland. {chris.cianci, jim.pugh, alcherio.martinoli}@epfl.ch

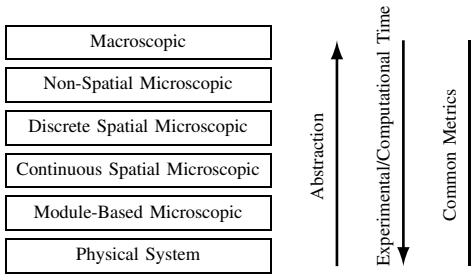


Fig. 1. An example of a hierarchical suite of models. Higher level models are simpler and faster, while lower levels are more complex, precise, and accurate in their representation of the target physical system. Performance metrics are shared between all levels, allowing us to exploit the advantages of any and all the levels where appropriate, while results which are directly and quantitatively comparable. Note that macroscopic models (e.g., rate equations) are not yet treated in this paper.

a supplement, allowing additional flexibility and diversity in the tests performed.

The real system naturally forms the basis of the hierarchy, representing the ground truth against which the subsequent modeling levels will be evaluated. Once the hierarchy has been established, the faster high-level models can be used for broad exploration of potential modifications to the algorithm and its parameters, and results of interest can then be verified on the more accurate low-level models. A brief overview of the types of models that will be presented in Section IV is shown in Fig. 1.

### III. CASE STUDY: DETECTING ACOUSTIC EVENTS

In general, the problem of resource allocation is not limited to power management within the network, but extends also to the treatment of data and interrupts destined for the operator, outside the network. Various monitoring and detection applications naturally have a wide range of requirements regarding false positives and false negatives, and the relative severity of either occurrence.

For the purposes of illustration, we will consider a scenario in which false positives are particularly undesirable, as they may trigger the invocation of a costly (or otherwise resource-intensive) procedure. Such an environment places particular emphasis on measurement confidence, and we have constructed a simplistic collective decision algorithm accordingly, exploiting the multi-level modeling framework to carry out further systematic exploration of its behavior and perform some analysis.

Acoustic event detection has been selected as an example of a domain where the measurement target is unpredictable in space and time. Our treatment here will use acoustic events as an illustration, but may be straightforwardly applied to any modality which is localized in space and time.

In this system, we will attempt to increase measurement confidence by requiring a consensus among an arbitrary integer  $C$  different nodes that a significant event has occurred, despite the presence of an additional, lower intensity, “undesirable” source (proportional to the intensity  $I_e$  of the desirable events:  $\frac{I_u}{I_e} = \{0.5, 0.75, 0.95\}$ ).

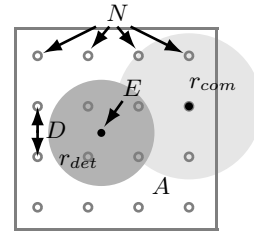


Fig. 2. Illustration of basic characteristics and parameters of the experimental setup. Sixteen nodes were spaced approximately 50 cm apart on a grid in a 1.5 by 1.5 meter area. (The  $r_{det}$  and  $r_{com}$  are shown here as Heaviside approximations based on  $P_{det}(r)$  and  $P_{com}(r)$  (see Fig. 6 for details).

#### A. Performance Metric

In order to have a quantitative method of reporting system performance, we define a metric function  $M$  in terms of the number of desirable events successfully detected and the number of undesirable events successfully ignored, in relation to the actual number presented:

$$M(\alpha, \beta) = \alpha \frac{E_{det}}{E_{tot}} + \beta \left( 1 - \frac{E_{fp}}{\max(E_{fp}, E_{tot})} \right) \quad (1)$$

where  $E_{det}$  is the number of events reported,  $E_{tot}$  the total number presented, and  $E_{fp}$  the number of false positives reported. The coefficients  $\alpha$  and  $\beta$  may be balanced according to the severity one wishes to associate with either term, so long as they sum to one for normalization.

#### B. Experimental Setup

The physical aspects of the system and its environment can be described by a set of six parameters (as illustrated in Fig. 2):

$A$	area of interest
$N$	number of available nodes
$D(N)$	distribution of available nodes
$E$	set of events occurring in the environment
$P_{det}(r, I_e)$	probability of detecting an event of intensity $I_e$ at a distance $r$
$P_{com}(r, I_t)$	probability of message reception with intensity $I_t$ at a distance $r$

Of these,  $A$ ,  $N$ , and  $D$  may be directly and arbitrarily selected by the experimenters (though the precise realization of  $D(N)$  may be perturbed by noisy factors beyond their control). Characteristics of the desired target events  $E$  are specifiable, but the actual events (and their locations, times, etc...) are by definition unpredictable. The two probabilities  $P_{det}$  and  $P_{com}$ , while indirectly controllable, are often inherent qualities of the physical agent being used, and should be determined empirically for the chosen hardware.

#### C. Hardware Platform

The experiments described here were performed on a fleet of e-pucks<sup>1</sup> (a miniature robotic platform recently developed at the École Polytechnique Fédérale de Lausanne, shown in Fig. 3-Left) [18]. The standard e-puck has a trinaural microphone array on-board, which was used in conjunction

<sup>1</sup><http://www.e-puck.org>

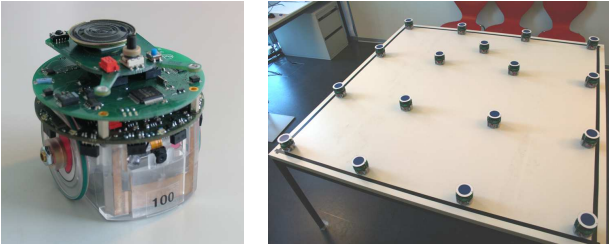


Fig. 3. LEFT: The e-puck, a small-scale experimental robotic platform. Shown here with the radio communication board stacked between the basic module and the jumper board, allowing the implementation of sensor networks and other networked robotic systems. RIGHT: The experimental setup described in Section III-B. The 16 nodes arranged in a grid configuration remain stationary (fixed), while the event source wanders randomly around the arena emitting short acoustic events at 1 second intervals. Markers are placed on top of the robots for tracking with an overhead camera.

with a simple digital filter to detect acoustic pulses at approximately 3.6 kHz. It is also equipped with a small speaker, allowing it to emit sound. Additionally, the e-pucks have been fitted with a custom extension turret for short-range radio communication using the subset of the 802.15.4 and ZigBee protocols present in TinyOS (an embedded operating system commonly used in wireless sensor networks) [17], and are therefore fully interoperable with the commercially available MicaZ [2] and Telos [3] platforms. The transmission power of the communication module is software-controllable, and passes through a custom attenuation circuit yielding effective maximum ranges between approximately 10cm and 5m. More details about the radio turret used can be found in [18].

For the present study, we construct the default physical system by distributing  $N = 16$  agents over a regular grid ( $D = 50\text{cm}$  spacing) in a rectangular arena ( $A = 1.5 \times 1.5$  meters). Each robot is equipped with a communication device (radio) and an acoustic sensor (trinaural microphone array). Acoustic pulses of a certain amplitude ( $I_e = 9,000$ , in the scaled arbitrary units used within the firmware controller) are seen as events of interest, and are generated, at 1 second intervals, on this area at random locations by a 17<sup>th</sup> agent unrelated to the established network in any way.

#### D. Control Algorithm and Parameters

Let us build a finite state machine controller (shown in Fig. 4) for the system as follows. In our setup, we require an event to be detected by at least  $C$  nodes before it is reported. A node in the “listen” state that perceives an event (“detection”) will announce the tentative detection (“notification”) to a subset of the network (the nodes within communication range; default being single-hop broadcast), and wait in the “event” state for a similar notification message (“confirmation”) generated by another node in the network before reporting a significant event at the system level. If no confirmation is received within a specified “timeout,” it simply returns to the “listen” state.

As radio waves travel faster than sound waves, it is also possible to receive a “notification” before detecting an event; hence the left side of the diagram.

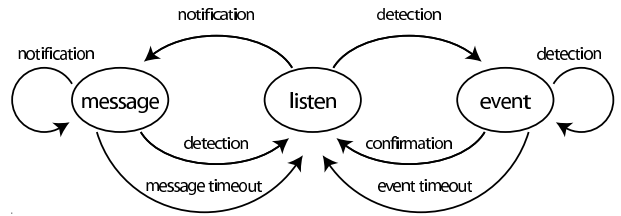


Fig. 4. Basic description of the individual controller algorithm as a finite state machine for  $C = 2$ . All nodes begin by “listening,” and upon either *detecting* an “event” or receiving a *notification* “message,” wait a short period of time for the complimentary signal before returning to the “listen” state either successfully (having observed a matched pair of the two signals) or unsuccessfully (timeout, most likely indicating a false positive). Therefore, an event is only reported at the level of the entire network if detected by at least two nodes (this occurs as part of the transition marked “confirmation”).

This adds three more control parameters into the system description:

- $T_e$  timeout after hearing an event, waiting for message
- $T_m$  timeout after receiving a message, waiting for event
- $C$  number of confirmations required for event acceptance

For the present setup,  $T_e$  and  $T_m$  can be selected as a function of the node spacing, the speed of sound, and the effective communication range. An event passing at the speed of sound will travel out of the area of interest in approximately 10 milliseconds; therefore even accounting for some potential processing and sending delay,  $T_e = T_m = 0.5$  seconds is more than sufficient, and still significantly below the expected delay between events.

In principle,  $C$  may be chosen arbitrarily, but there may be application specific instances in which certain values of  $C$  may be optimal, and others implausible (as a function of the field being monitored and the agents being used). Where necessary to further the example, we will continue to use  $C = 2$ , as in Fig. 4.

#### E. Implementation in Hardware

The three-state controller described in Section III-D was implemented to run in-situ on the individual nodes, and 16 nodes were arranged as shown in Fig. 2 and Fig. 3-Right. The event source is mobile, and wanders freely about the arena avoiding obstacles and emitting acoustic pulses, but does not interact in any other way with the observing network. One hundred events of each type (“desirable” and “undesirable”) were generated at random locations within the area of interest, and the response of the network to each event was recorded. (Results are presented in Section V.)

## IV. MICROSCOPIC MODELS

Microscopic models represent each agent in the system individually (in contrast with macroscopic models, which attempt to capture the dynamics of the aggregate system as a whole into a single model). The way in which these individual agents are modeled, however, can have a profound effect on both realism and complexity.

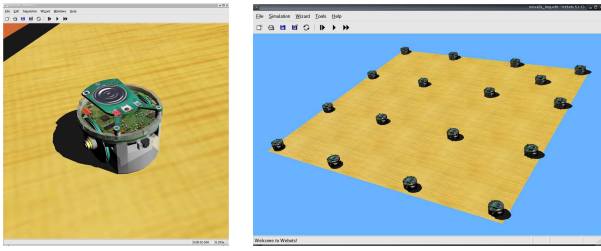


Fig. 5. LEFT: A simulated model of the e-puck in the realistic simulator Webots. RIGHT: A simulated version of the physical setup shown in Fig. 3. Radio communication is provided by a plug-in wrapper for the OMNeT++ network simulation engine, and acoustic dynamics are modeled using the Image-Source method (walls not visible in picture).

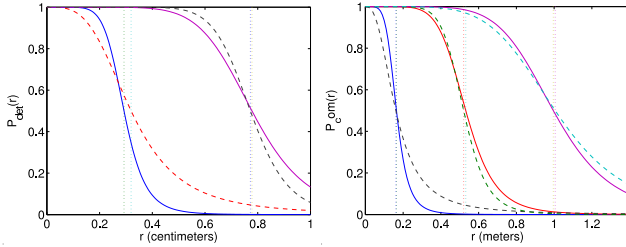


Fig. 6. The probabilities  $P_{det}(r)$  of detecting an event (left) and  $P_{com}(r)$  of successful communication (right) versus radial distance  $r$  to the robot for several emitted intensities or transmission powers. Sigmoidal regressions are shown for the physical system (dashed lines) and the module-based simulation (solid lines). The vertical dotted lines indicate threshold approximations at  $P(r) = \frac{1}{2}$ .

### A. Module-Based Microscopic Model

The first abstraction layer we will consider is the *module-based microscopic model*. While obviously a simplified version of the real world, this level still maintains as much realism as possible by preserving *intra-node* details, such as the individual sensors and sensing modalities, actuators, transceivers, etc.

In order to accurately capture the dynamics of a robotic sensor network engaged in the detection of acoustic events, it is necessary to include several key components in our module-based microscopic model. The first required component is a realistic description of the e-puck robotic platform in the Webots simulator [19] (Fig. 5-Left). Beyond the creation of the 3D model for the robots, it was necessary to properly calibrate the input and output responses of sensors and actuators (namely, infrared proximity sensors and motors) to match those of the real platform. This was accomplished via extensive experiments with real e-puck robots. The second required component is accurate modeling of acoustic dynamics. A modified version of the Image-Source sound propagation algorithm [20] which can simulate an environment with mobile speakers and microphones was implemented as a simulation add-on. This technique models reflection and attenuation (but not diffraction) and was found to be a good trade-off between accuracy and speed of simulation. The third required component is realistic radio communication between robots. We implemented this by wrapping the OMNeT++ [21] network simulation engine,

which handles channel coding and fading signal propagation, as a plug-in for the Webots simulator. A new 802.15.4/Zig-Bee module was developed to match the radio turret present on the real e-puck robots.

These modules were calibrated using data obtained from real-world experiments; a comparison between the responses of the simulated and real modules can be seen in Fig. 6. The experiment described in Section III was then performed in simulation (Fig. 5-Right); results shown in Section V.

### B. Continuous Spatial Microscopic Model

The module-based microscopic model can be further abstracted by omitting explicit descriptions of robot features and instead treating each robot as a basic state machine operating in a simplified environment; we denote this as an agent-based microscopic model. Three different abstraction levels of time-discrete agent-based microscopic models are considered: a continuous spatial model, a discrete spatial model, and a non-spatial model.

In the continuous spatial model, instead of a moving sound source, a Monte Carlo simulation was used for generating spatially and temporally random acoustic events which propagate for a fixed radial distance at the speed of sound. Radio communication between robots was assumed to be instantaneous and with a fixed radius of propagation. The results of a Matlab simulation of 1000 events using this model are shown as part of the comparison in Section V.

### C. Discrete Spatial Microscopic Model

It may also be possible to combine the benefits of the additional detail provided by a spatial representation with the simplicity and speed of a non-spatial model (which will be shown below), by recognizing certain regular structures in the system. Given that we have modeled sensing and communication radially, it is clear that these ‘overlapping circles’ will generate several regions (or in a regular deployment such as this one, classes of regions) which are covered by different numbers of nodes. When the node positions are known, these regions can be calculated exactly; and when the node placement is regular, they can even potentially be specified in a general form.

Indeed, Fig. 7 shows precisely how the relationship between the radius of detection and a regular grid spacing affects the percentage of the area of interest which is covered simultaneously by any given number of nodes. Obviously, in the presence of measurement noise, these functions will be perturbed slightly, but the basic characteristic will remain essentially unchanged. This then presents us with additional information, which we did not have previously, regarding the selection of  $C$  and how the various parameters of the system are interdependent. Looking at this graph, for our present case of  $C = 2$ , we can clearly see that a sensing radius of less than the grid spacing will result in certain areas where it is simply not possible for an event to be heard by two nodes. Now, in this case, particularly after describing it in this way, that particular conclusion may seem to be intuitive; however, the increasingly tangled mess of curves towards the right side

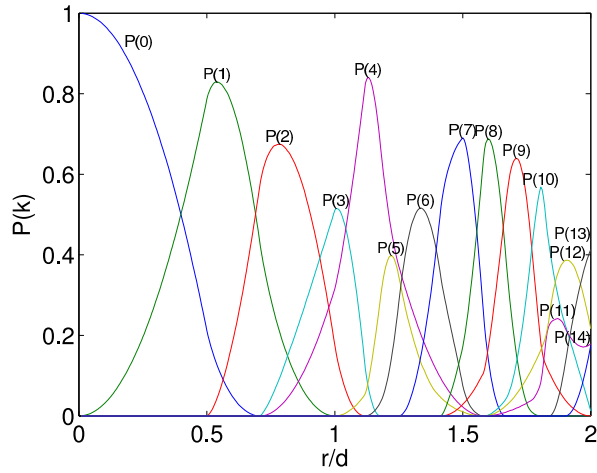


Fig. 7. Probability of an event at a random location being detected by  $k$  nodes, as a function of the sensing radius and the grid spacing. While it is possible to solve these for these curves in a piecewise analytically exact fashion, it rapidly becomes tedious. The data shown here was generated by a Monte Carlo simulation of 1,000,000 points.

of the plot should make it clear that this is not necessarily always the case.

Interestingly, as this also clearly implies a finite set of such regions for any specified  $\frac{r_{det}}{D}$ , there is even the possibility of modifying the state machine shown in Fig. 4 to include unique ‘event’ states for each class of event location, though it is not immediately clear that this would be to our advantage. It is possible (and a topic of ongoing inquiry) that such a formulation may even lend itself amenable to a macroscopic model, allowing us to make statements about a system of arbitrarily many agents using a single model for the collective.

The spatial discretization was performed as follows: whereas the graph shown in Fig. 7 is the result of 1,000,000 uniformly distributed random events, a quantization of the event space into only 900 evenly spaced fixed locations results in an average error of less than 0.78%. This allows us to pre-compute the network response to an event at each of a relatively small finite number of fixed points, for an additional gain in simulation speed, with very little effect on the quality of the simulation (see Fig. 9).

#### D. Non-Spatial Microscopic Model

For the non-spatial model, we no longer consider events to exist at a particular spatial location, but rather to be positionless occurrences which are detected by a node with some probability. The agent controllers then become Markov Chains which transition at discrete fixed time steps (see Fig. 8). Transition probabilities for these Markov Chains (i.e. the probability of event arrival  $P_e$  and the probability of message arrival  $P_m$ ) can be calculated analytically from system parameters using the geometrical methods shown in [22]; and as before these calculations were also verified by comparison with the results of the real system and the other microscopic models.

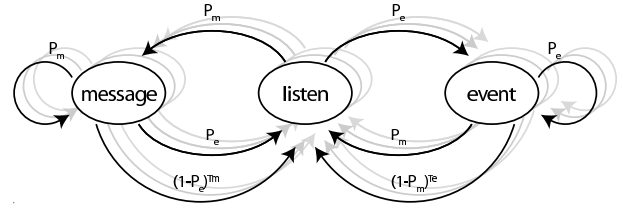


Fig. 8. The deterministic responsive controller from Fig. 4 modified with transitions labeled by the probability of encountering the associated stimulus, yielding a probabilistic finite state machine (PFMS). This Markov chain can then be iterated over in a non-spatial manner.

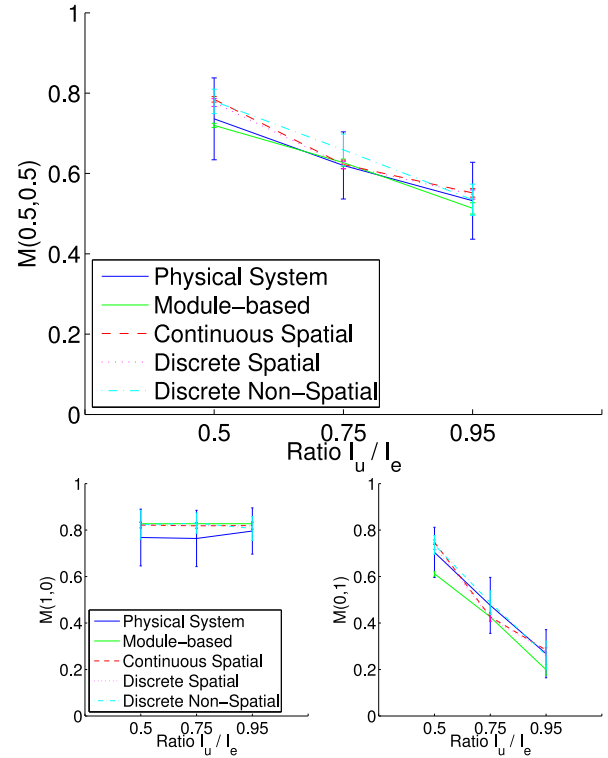


Fig. 9. Comparison of results for the physical system and all four of the modeling levels presented for the source discrimination experiment described in Section III. Mean and standard deviation over 5 runs of 50 events for the real system, and 20 runs each of 100, 1,000, 1,000, and 10,000 events for the module-based, continuous spatial, discrete spatial, and discrete non-spatial models, respectively.

## V. RESULTS AND COMPARISON OF MODELING LAYERS

Here we can finally see, in Fig. 9, the output of each modeling level side-by-side for the source discrimination experiment described in Section III. The ‘undesirable source’ was assigned an intensity  $I_u$  proportional to that of the target source  $I_e$  ( $\frac{I_u}{I_e} \approx \{0.5, 0.75, 0.95\}$ ) and  $C = 2$ . Five experimental runs of 100 events were performed on the real system, and each of the models were run 20 times; for 100 (module-based), 1,000 (continuous spatial), 1,000 (discrete spatial), and 10,000 events (discrete non-spatial), respectively. Fig. 9-Top shows  $M(\alpha = \frac{1}{2}, \beta = \frac{1}{2})$ , an even balance between the two contributing terms of the metric; the lower plots show the extreme cases  $M(1, 0)$  and  $M(0, 1)$ .

All four models reflect similar trends which closely re-



TABLE I

COMPARISON OF SIMULATION TIMES FOR THE VARIOUS MODELS.

Model	Speed Factor
Non-Spatial Microscopic	90.81x
Discrete Spatial Microscopic	23.27x
Continuous Spatial Microscopic	17.07x
Module-Based Microscopic	1.36x
Physical System	1.0x

semble the behavior of the real system—despite substantial differences in computational complexity (in some cases, more than an order of magnitude in execution time, see Table I). More experiments are needed to identify subtle effects of different modeling design choices on a given metric related to this case study.

## VI. REMARKS, CONCLUSIONS, AND FUTURE WORK

Here we have shown the application of a multi-level modeling methodology to a robotic wireless sensor network tasked with the reliable detection of acoustic events. Clear correspondence has been demonstrated at each abstraction, giving us objective grounds for justifying the applicability of experiments performed at higher abstraction levels to the target physical system. These higher levels can subsequently be used for extensive exploration of different possible parameters and controllers which would simply be infeasible otherwise. The results of such exploration can then be verified in a more detailed, lower-level model.

This is a basic formulation, with plenty of avenues open for refinement, but the essence of the multi-level framework is that it places overt and primary emphasis on maintaining a view of the whole system by providing an intuitive and incremental process for building descriptions and abstractions of the system at several levels.

Other obvious additions that we are currently studying include the adaptation of the models to deal with nonlinear systems in more a general way, particularly those involving spatial metrics. Eventually, we would also like to explore the possibility of applying a macroscopic model as well, incorporating all of the system dynamics into a single, concise representation, neglecting even the individuality of the agents, as previously done in swarm robotics systems.

## VII. ACKNOWLEDGMENTS

Christopher Cianci, Jim Pugh, and Alcherio Martinoli are currently sponsored by a grant from the Swiss National Science Foundation (Contract Number PP002-116913).

## REFERENCES

- [1] C. Y. Chong and S. P. Kumar, "Sensor networks: Evolution, opportunities, and challenges," *Proceedings of the IEEE*, vol. 91, no. 8, pp. 1247–1256, August 2003.
- [2] J. L. Hill and D. E. Culler, "Mica: A wireless platform for deeply embedded networks," *IEEE Micro*, vol. 22, no. 6, pp. 12–24, November/December 2002.
- [3] J. Polastre, R. Szewczyk, and D. Culler, "Telos: Enabling ultra-low power wireless research," in *Information Processing in Sensor Networks: Special track on Platform Tools and Design Methods for Network Embedded Sensors (IPSN/SPOTS)*, Los Angeles, CA, USA, April 2005.
- [4] A. Mainwaring, J. Polastre, R. Szewczyk, D. Culler, and J. Anderson, "Wireless sensor networks for habitat monitoring," in *ACM International Workshop on Wireless Sensor Networks and Applications (WSNA)*. Atlanta, GA, USA: ACM Press, September 2002, pp. 88–97.
- [5] R. Szewczyk, E. Osterweil, J. Polastre, M. Hamilton, A. Mainwaring, and D. Estrin, "Habitat monitoring with sensor networks," in *ACM Conference on Embedded Networked Sensor Systems (SenSys)*, J. A. Stankovic, A. Arora, and R. Govindan, Eds. Baltimore, MD, USA: ACM Press, November 2004, pp. 214–226.
- [6] S. Subramaniam, T. Palpanas, D. Papadopoulos, V. Kalogeraki, and D. Gunopulos, "Online outlier detection in sensor data using non-parametric models," in *International conference on Very large data bases (VLDB)*. Seoul, Korea: VLDB Endowment, September 2006, pp. 187–198.
- [7] C. Guestrin, P. Bodik, R. Thibaux, M. Paskin, and S. Madden, "Distributed regression: an efficient framework for modeling sensor network data," in *International symposium on Information processing in sensor networks (IPSN)*. Berkeley, CA, USA: ACM Press, 2004, pp. 1–10.
- [8] D. Chu, A. Deshpande, J. M. Hellerstein, and W. Hong, "Approximate data collection in sensor networks using probabilistic models," in *International Conference on Data Engineering (ICDE)*. Washington, DC, USA: IEEE Computer Society, 2006, pp. 48–59.
- [9] R. Jurdak, C. V. Lopes, and P. Baldi, "A framework for modeling sensor networks," in *Workshop on Building Software for Pervasive Computing (OOPSLA)*, Vancouver, Canada, October 2004.
- [10] S. Tilak, N. Abu-Ghazaleh, and W. Heinzelman, "A taxonomy of wireless micro-sensor network models," *SIGMOBILE Mobile Computing and Communications Review*, vol. 6, no. 2, pp. 1559–1662, 2002.
- [11] C.-F. Chiasserini and M. Gareto, "Modeling the performance of wireless sensor networks," in *Joint Conference of the IEEE Computer and Communications Societies (INFOCOM)*. IEEE Press, March 2004, pp. 231–242.
- [12] A. Martinoli, K. Easton, and W. Agassounon, "Modeling of swarm robotic systems: A case study in collaborative distributed manipulation," *The International Journal of Robotics Research*, vol. 23, no. 4–5, pp. 415–436, 2004.
- [13] J. Turek and D. Shasha, "The many faces of consensus in distributed systems," *IEEE Computer*, vol. 25, no. 6, pp. 8–17, 1992.
- [14] R. Olfati-Saber and R. M. Murray, "Consensus problems in networks of agents with switching topology and time-delays," *IEEE Transactions on Automatic Control*, vol. 49, no. 9, pp. 1520–1533, 2003.
- [15] R. Olfati-Saber and J. S. Shamma, "Consensus filters for sensor networks and distributed sensor fusion," in *IEEE Conference on Decision and Control, European Control Conference (CDC-ECC)*. IEEE Press, 2005, pp. 6698–6703.
- [16] J. C. March and W. E. Bentley, "Quorum sensing and bacterial crosstalk in biotechnology," *Current Opinion in Biotechnology*, vol. 15, no. 5, pp. 495–502, October 2004.
- [17] J. Hill, R. Szewczyk, A. Woo, S. Hollar, D. Culler, and K. Pister, "System architecture directions for network sensors," in *Architectural Support for Programming Languages and Operating Systems (ASPLOS)*, Cambridge, November 2000.
- [18] C. M. Cianci, X. Raemy, J. Pugh, and A. Martinoli, "Communication in a swarm of miniature robots: The e-puck as an educational tool for swarm robotics," in *Simulation of Adaptive Behavior (SAB-2006), Swarm Robotics Workshop*, Rome, Italy, October 2006, Lecture Notes in Computer Science (2007), vol. 4433, pp. 103–115.
- [19] O. Michel, "Webots: Professional mobile robot simulation," *Journal of Advanced Robotic Systems*, vol. 1, no. 1, pp. 39–42, 2004.
- [20] J. B. Allen and D. A. Berkley, "Image method for efficiently simulating small-room acoustics," *Journal of the Acoustic Society of America*, vol. 65, no. 4, pp. 943–950, 1979.
- [21] A. Varga, "Software tools for networking: "OMNeT++";" *IEEE Network Interactive (2002)*, vol. 16, no. 4, July 2002.
- [22] C. M. Cianci, T. Lochmatter, J. Pugh, and A. Martinoli, "Toward multi-level modeling of robotic sensor networks: A case study in acoustic event monitoring," in *International Conference on Robot Communication and Coordination (ROBOCOMM)*, Athens, Greece, October 2007, p. to appear.