2013 IEEE/RSJ International Conference on
Intelligent Robots and Systems (IROS)
November 3-7, 2013. Tokyo, Japan

An Experimental Study in Wireless Connectivity Maintenance Using up to 40 Robots Coordinated by an Institutional Robotics Approach*

José N. Pereira[1,2], Porfírio Silva[2], Pedro U. Lima[2], and Alcherio Martinoli[1]

Abstract— **This work is developed in the framework of Institutional Robotics (IR), an approach to cooperative distributed robotic systems that draws inspiration from the social sciences. We consider a case study concerned with a swarm of simple robots which has to maintain wireless connectivity and a certain degree of spatial compactness. Robots have local, bounded communication capabilities and have to execute the task (running an IR controller) using exclusively as information their current number of wireless connections to neighbors. For the very same case study, we previously introduced an IR-based macroscopic model for the behavior of a large number of robots, validated using a submicroscopic model implemented through a realistic simulator. In this work, we go a step further and validate our submicroscopic model with real world experiments, duplicating accurately the conditions used, including a large number of robots and noisy communication channels. The main conclusions of this paper are two-fold. First, the IR approach was able to maintain the wireless connectivity of a swarm of 40 real, resource-constrained robots. This speaks in favor of the robustness and scalability of such approach. Second, the submicroscopic model implemented is faithfully capturing the reality and can be used to further optimize the performances of distributed control strategies using an IR approach.**

I. Introduction

Institutional Robotics (IR) [17], is an approach to cooperative distributed robotic systems that draws inspiration from the social sciences, namely from Institutional Economics' concepts [7]. It combines the notions of institution, coordination artifact, and environment, aiming to provide a comprehensive strategy for specifying social interactions (e.g., norms, roles, hierarchies) among robots. Under IR, robots are situated not only in a physical but also in an institutional environment, where their interactions are guided by institutions. Cooperation is achieved by this regulation of social interactions since the robots know not only how to behave in a given scenario but also what to expect from other robots and the environment.

On the one hand, one of the goals of our research is to develop IR models that predict the system performance both quantitatively and qualitatively, and allow us to analyze their intrinsic limitations, performance bounds, and general system properties. On the other hand, real world validation remains a fundamental task in robotics when assessing the soundness of models and algorithms, grounding the studied methods in reality. Although this is true for all fields of robotics, it is of critical importance for the field of distributed robotic systems (DRS). The behavior of DRS with a large number of robots is difficult to model, since these are often stochastic, dynamical, and non-linear in nature. Traditionally, when implementing these systems in reality, researchers tend to use small dimensions and low-cost robots, allowing a large number of robots on limited physical space and limited budget. Such robots are prone to noise in sensing and actuation, presenting additional difficulties when comparing models and reality.

Modeling techniques for large DRS, capable of predicting their performance and allowing for verification of relevant properties, are of critical importance. They allow researchers to test a broad range of parameters and design choices that would take too long to test with the large number of robots considered. In previous work [16], we introduced an IR highly abstracted Generalized Stochastic Petri Net (GSPN) model for the behavior of a large number of robots in a distributed robotics case study, enabling qualitative and quantitative model-based analysis, and allowing us to quickly test relevant parameters and design choices. Ideally, our aim would be to validate our GSPN model with real world experiments, i.e., to make a direct grounding of a model with a high degree of abstraction in reality. However, it is not always possible to make a direct correspondence between these paradigms. Closing the gap between highly abstracted models and physical reality is essential, and submicroscopic models, characterized by a low degree of abstraction and capturing intra-robot details such as individual sensors and actuators, are fundamental in this goal.

We consider a swarm robotics case study introduced in [14], concerned with a robot swarm which has to maintain wireless connectivity and a certain degree of spatial compactness using a decentralized control algorithm. Robots are equipped with radio frequency communication modules to achieve local, bounded communication, and the communication radius is considerably less than the global diameter of the swarm. Robots use exclusively as information their current number of wireless connections to neighbors. This case study has been tested extensively in simulation and using various modeling abstraction levels [20]. Moreover, in [19] the authors report an implementation using a small number of real robots (4-8 robots) where local communication was

*This work was partially supported by Fundação para a Ciência e a Tecnologia (FCT) through grants SFRH/BD/33671/2009 (first author) as part of the Joint Doctoral Program IST-EPFL and SFRH/BPD/35862/2007 (second author), as well as by FCT ISR/IST Pluriannual funding through the PIDDAC program funds, and by FCT project grant PTDC/EEACRO/104658/2008.

[1]José N. Pereira and Alcherio Martinoli are with the Distributed Intelligent Systems and Algorithms Laboratory, EPFL, Lausanne, Switzerland {jose.pereira, alcherio.martinoli} at epfl.ch

[2]José N. Pereira, Porfírio Silva and Pedro U. Lima are with the Institute for Systems and Robotics, Instituto Superior Técnico, Lisbon, Portugal {jpereira, porfiriosilva, pal} at isr.ist.utl.pt

achieved with a combination of a global wireless network and an overhead camera delimiting the communication range.

A first goal of this work is to move one step further with the realism of the physical implementation by using real local communication channels and a large number of robots (tests were performed with sets of 20 and 40 robots). We chose to use 40 robots in order to maintain as much as possible a parallel with the original case study experiments, where 40 simulated agents were used [20].

A second goal of such implementation is to show that the IR approach is able to handle such realism, and in particular maintain the wireless connectivity of a swarm of 40 real, resource-constrained robots, further increasing our confidence in the approach's robustness and scalability.

A third goal of this work is to go further in validating the GSPN macroscopic model which we introduced in [16] but only validated using a submicroscopic model. We will do so by recreating the same conditions present in our submicroscopic model in real world experiments, and performing multiple trials in order to obtain significant statistics for our metrics of interest. This will allow us in turn to conclude that if the submicroscopic model provides a good description of reality, it can be used to correctly validate our highly abstracted GSPN macroscopic model.

In Section II we present some background and related work for this study. In Section III we present the wireless connected swarm case study in detail, and give details on our robotic and simulation platforms, and on the experimental setup. Performance metrics for the case study and results from submicroscopic model and real robot experiments are presented and discussed in Section IV. Section V gathers some conclusions and future work possibilities.

II. Background and Related Work

When modeling large DRS, we are interested in following a bottom-up multi-level modeling methodology [10]. Starting with a real implementation of a particular case study, this methodology builds a series of modeling layers increasing in their abstraction level. The lower layer model (*submicroscopic*) is typically implemented using embodied, realistic simulation tools capable to represent intra-robot details in a very detailed way (e.g., individual sensors and actuators are modeled separately with their noise characteristics, nonlinear response, placement and orientation). A formal representation of the controller used to execute the case study in real robots and submicroscopic simulations (for instance, a Finite State Automaton (FSA)) is used to build the two following layers of models. In the second layer (*microscopic*), most of the intra-robot details are abstracted and an aggregated representation of each individual robot is used (e.g., a probabilistic FSA). In such abstraction process, the actual robot controller can serve as blueprint for the model structure. The higher layer (*macroscopic*) uses the same formal representation of the controller to generate, often in a mean field approach (although not necessarily), an estimate of the number of robots in each state of the controller. Performance metrics can be studied on the several layers of models and cross-validation between these layers provides solid grounding for all modeling levels.

This methodology was proposed in [10] for a collaborative swarm robotics case study and has been applied to other case studies concerned with robot aggregation [5], [6] and wireless connectivity [20]. This methodology has also been compared with other approaches based on multiple levels of modeling but proceeding in a top-down fashion in terms of model-building and control design [2], [1]. As shown in [11], a bottom-up modeling and control design appears to be particularly indicated to deal with resource-constrained robots.

In [15], we formalized institutions - the central concept in IR - using an abstract representation (executable Petri Nets [EPNs]), allowing their design and execution for distributed robotic systems, so as to obtain behaviors capturing the social interactions of interest. Our method composes a set of institutions, to create an institutional robot controller (defined as an EPN) able to execute a desired task and observe the specified social interactions. In [16], we follow the multi-level modeling methodology, using a robot controller to build higher abstraction models. However, in contrast with the work mentioned above on multi-level modeling, the very same Petri net structure that is used to execute a task in submicroscopic simulations is also used as a GSPN macroscopic model, without the need of employing extra analytical tools (e.g., macroscopic difference equations describing state transitions) to accurately describe the system.

III. Institutional Agent Controllers

In the IR approach, we model institutions using a formal representation, leading to a standard design and execution platform (in real robots, submicroscopic realistic simulations, and microscopic multi-agent systems). Institutions encapsulate relevant behavioral rules for robots, specifying social interactions of different types among actors in a given scenario. They represent the basic building blocks for creating shared coordinated working environments. Moreover, concurrent execution of institutions has to be regulated since not all behaviors can be executed simultaneously. We use *Petri Nets* (PNs) as the formal framework and follow their usual definition as described in [3].

Formalizing institutions for modeling and execution of robot controllers means that we need to take into account robot actions and sensor readings. *Executable Petri Nets* (EPNs) are PNs that have actions and boolean conditions (verifiable by sensor readings) associated with places and transitions, respectively. The basic intuition behind this definition is that by associating actions with places we are able to define which actions are to be executed at each time step. This is done simply by checking if the corresponding place is marked. By associating transitions with conditions verified by sensor readings we trigger state changes in the EPN due to changes in the robots environment.

We represent each institution by an EPN that can be executed independently or together with other institutions. We also represent robot's *individual behaviors* by EPNs.

While the institutions specify behaviors that have a *social nature*, i.e., they relate the robot to other robots in some way, the individual behaviors specify a set of basic behaviors that have exclusively an *individual nature*, i.e., they relate the robot with the surrounding environment and its own goals. The composition of the individual behavior with a set of institutions generates a robot controller.

Definition: An *Institution* I is a four-tuple $(Inst,\ initial_I,\ final_I,\ d_I)$ where:

- $Inst$ is an EPN;
- $initial_I,\ final_I \in Cdt$ are initial and final conditions for the execution of $Inst$;
- $d_I \in D = \{AllowAll,\ StopInd,\ StopInst,\ StopAll\}$ is the associated deontic operator.

The EPN *Inst* specifies the desired behavior that should be performed by the robot. This behavior is not always being executed, its start and end are dictated by conditions $initial_I$ and $final_I$, which the robot verifies at each time step. Thus, we say that an institution I at each time step can be *active* or *idle*. Each institution also includes a deontic operator d_I which is used when combining it with the robot individual behavior and further institutions, allowing or stopping the concurrent execution of institutions and/or individual behavior. *Inst* must be designed, but institutions can be kept simple and further behavioral complexity is the result of composition, in a modular fashion.

EPNs can be represented by macro places in a hierarchical fashion, using two distinct layers. We consider that each institution I is part of a lower layer and is represented by one macro place m_I in the higher layer. By adding bidirectional arcs between each transition in I and m_I, we guarantee that if m_I is marked, I is active, otherwise it is idle. This allows us to compose our institutions at the higher layer where relationships among the institutions and the individual behavior should be specified while keeping relationships between actions and conditions separated in the lower layer.

The composition of individual behaviors and institutions is performed algorithmically by adding, in the higher layer, places and transitions that restrict their concurrent execution, according to the specification provided by the deontic operators. Both layers can be then merged algorithmically to obtain a full EPN that can be used as controller. This EPN is designated as the *Institutional Agent Controller* (IAC). Each robot in a social collective setting mediated by institutions runs its IAC. This IAC is used as the starting point for our GSPN model.

IV. Wireless Connected Swarm Case study

In this section we present the wireless connected swarm case study, previously investigated in [14], [19], [20]. We perform real world experiments of the case study and compare results obtained in reality with those obtained in submicroscopic simulation.

A. Materials and Methods

Our platform is the *e-puck* robot [13], a differential drive robot of 7 cm in diameter. In order to endow the robots with scalable wireless communication capabilities, we use a radio communication module developed at DISAL [4]. This module is ZigBee-compliant and uses TinyOS [8]. A bounded communication range is obtained using software-controllable power emission and a dedicated hardware attenuator.

Our goal is to assess the validity of our submicroscopic model of this case study. For implementing this model we used *Webots* [12], a flexible, 3D realistic simulator, and considered kinematic models of the *e-puck* robot. The original case study considered a perfect circular bounded communication radius and perfect package reception inside that radius (radial disk model). In this work, communication between *e-pucks* is also simulated realistically using the network simulation engine OMNeT++ [18] as a plugin for *Webots*. The OMNet++ engine handles channel coding, noise, fading signal propagation, as well as a non-circular communication footprint. Fig. 1-(a) offers a visualization of the *Webots* submicroscopic simulations. Fig. 1-(b) displays an image of the arena during execution taken with the overhead camera.

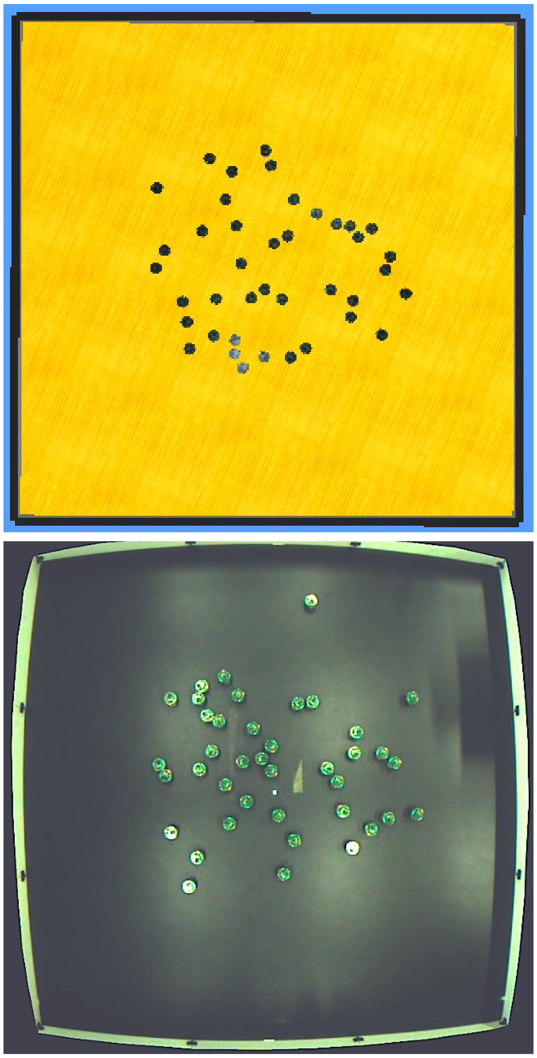
B. Task Description & Decentralized Control Algorithm

In the wireless connected swarm case study a decentralized control algorithm is implemented to maintain wireless connectivity and a certain degree of spatial compactness of a robotic swarm (with N robots) in an unbounded arena using exclusively, as information at the robot level, the current number of wireless connections to the neighbors. The communication is local and its bounded range is a parameter of the robotic system. Let X be the number of connections perceived by a robot. In the default state (defined as $forward$), the robot simply moves forward. If at any time the robot senses the loss of a connection and X falls below a threshold α (where $\alpha \in \{0,\dots,N-1\}$), the robot assumes it is going in the wrong direction and switches to state *coherence*. In this state the robot performs a 180° turn in order to recover the lost connection. Upon recovering the lost connection, the robot performs a random turn and moves back to the default state. If the connection is not recovered, the robot simply moves to the default state. If an obstacle is detected the robot immediately switches to state *avoid*, where it performs obstacle avoidance for a given number of time steps, after which it returns to its previous state.

While this simple algorithm has limited robustness, it allows the swarm to maintain its connectivity to a certain extent, with its spatial compactness being controlled by the communication range and by the threshold α. It is implemented in [20] using a FSA controller with states defined as above.

C. Institutional Agent Controller

In our IAC implementation, robots execute an individual behavior *IndAv* (*Individual Avoidance*) and two institutions $T180$ (*Turn 180 degrees*) and TR (*Turn Random*), all specified by EPNs shown in the lower layer of Fig. 2. Individual behavior *IndAv* specifies a behavior relating the robot to its environment, consisting on simple obstacle avoidance.



(a)

(b)

Fig. 1. (a) *Webots* simulation screenshot, 40 *e-puck* robots simulated. (b) Real world experiment screenshot, 40 *e-puck* robots.

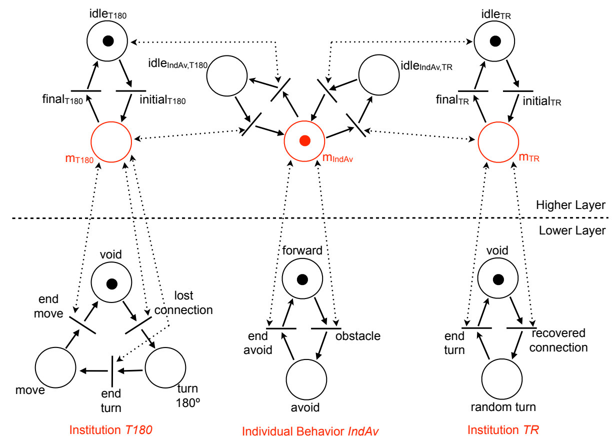


Fig. 2. IAC for wireless connected swarm. Lower layer: EPNs for individual behavior *IndAv* and institutions $T180$ and TR. Higher layer: composition of individual behavior and institutions.

Institutions $T180$ and TR implement the social rules, dealing with loss and recovery of connections. $T180$ specifies that upon losing a connection the robot performs a 180° turn followed by moving forward for a small number of steps. Institution TR specifies that if a connection is recovered the robot performs a random degree turn.

To consider institutions as defined in Section 3, we need initial and final conditions and deontic operators. For institution $T180$ we say that initial condition $initial_{T180}$ is "loss of connection detected and number of connections is less than α" and the final condition $final_{T180}$ is "move forward procedure has ended". For institution TR we say that initial condition $initial_{TR}$ is "recovery of connection detected and previous number of connections is less than α" and the final condition $final_{TR}$ is "random turn procedure has ended". The deontic operator associated with both institutions is $StopInd$, specifying that institutions and individual behavior cannot be executed concurrently.

We now have all the elements needed to obtain the IAC that specifies our desired behavior. The composition of the individual behavior *IndAv* and institutions $T180$ and TR (specified separately by EPNs shown in the lower layer of Fig. 2) is shown in the higher layer of Fig. 2. The final controller is the full EPN of Fig. 2, obtained after merging the two layers.

D. Experimental Setup

We replicated, to the best extent possible, the conditions of the original case study presented in [20]. Therein, the authors considered 40 robots in an unbounded arena performing the task over 10 000 seconds. In this work, we carry out experiments (both real robot experiments and *Webots* simulations) with sets of $N = 20$ and $N = 40$ robots in a 3 by 3 meters bounded arena performing the task over 1800 seconds. The connection threshold is dependent on the size of N and is set to $\alpha = 8$ for $N = 20$ and $\alpha = 16$ for $N = 40$. The communication radius of the *e-puck* is intended to be 0.7 meters, instead of the original 2.0 meters, in order to keep the ratio between communication and physical radius presented in the original paper. We set the transmission power of the *e-puck* communication module to an appropriate value that allows us to roughly achieve the desired communication radius.

To compare the performance of our submicroscopic model and real world experiments we performed 100 runs of the simulation for each $N = 20$ and $N = 40$, and 10 runs of real world experiments for $N = 20$ and 5 runs for $N = 40$. During runs we stored the number of time steps robots spent with each number of connections (between 0 and $N-1$). We also recorded videos of the arena during the real world experiments using an overhead camera and the *SwisTrack* software [9]. We processed the videos offline, using *SwisTrack* to perform background subtractions and blob detection, in order to extract and store the position of each robot in each frame. We also stored information about the position of robots at each time step of our simulations.

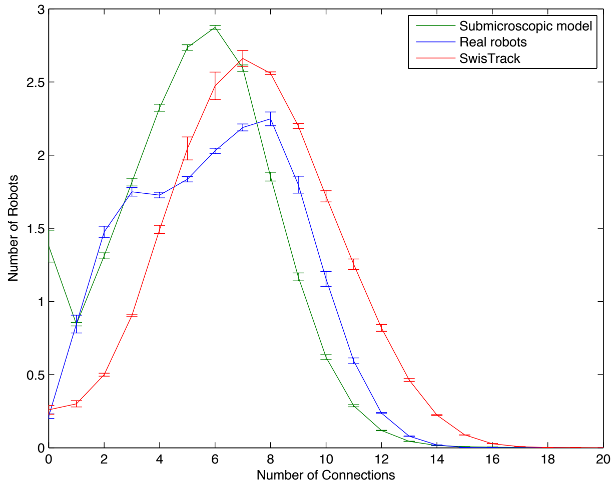


Fig. 3. Connectivity metric: average number of robots with a particular number of connections during a run. Variance shown for different runs. Results for 20 robots and $\alpha = 8$.

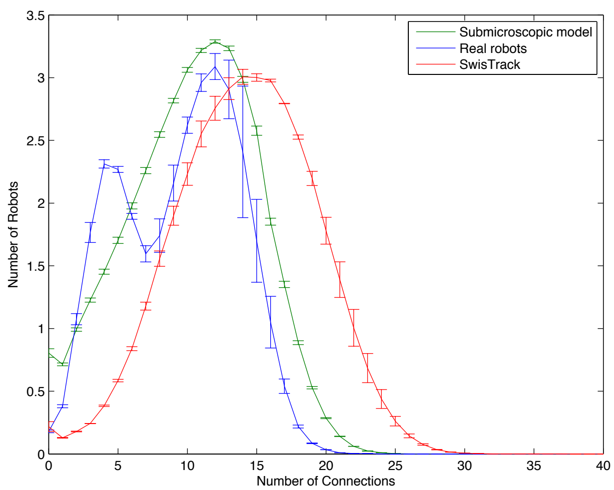


Fig. 4. Connectivity metric: average number of robots with a particular number of connections during a run. Variance shown for different runs. Results for 40 robots and $\alpha = 16$.

V. Results

In this work, we are interested in three main metrics that represent and allow us to analyze different aspects of the swarm (and individual) behavior: connectivity, dispersion, and displacement.

Connectivity tells us, on average, how many robots have a particular number of wireless connections during the time needed to perform a run of the experiment. To measure connectivity we use data gathered by the robots about the number of time steps spent with each number of connections. Robots with α or more connections are not concerned with recovering lost connections and are likely to be moving away from the swarm. On the other hand, robots with less than α connections are actively trying to regain connections and are likely to be moving towards the swarm. Thus, we can expect the swarm connectivity to peak at α, i.e., at each time step we will have more robots with α connections than with any other number of connections.

Dispersion measures the average distance of robots to the swarm center of mass. It gives us an indication of how spread out the swarm is across the arena. Ideally we would like this value to be as close to zero as possible, bounded by the communication radius, and constant throughout the run. To compute dispersion we use data about the position of robots gathered either in simulation or through *SwisTrack* in the real world experiments.

Displacement measures the distance between the swarm center of mass and the center of the arena. Given the stochastic nature of the movement of the robots, displacement will start close to zero (runs start with robots gathered closely in the center of the arena) and will increase throughout the run. The motion of the swarm as a whole resembles a random walk through the arena. This metric would be somewhat different if considered in the original case study, given that an unbounded arena was considered.

In Fig. 3 and Fig. 4 we present the connectivity metric results for $N = 20$ and $N = 40$. In green we display results obtained with submicroscopic simulations, while in red and blue we display results with real robots. The blue line was obtained with data about number of connections as perceived and recorded by the robots. On the other hand, the red line was obtained in offline processing using *SwisTrack* by counting, for each robot, how many other robots were present in a 0.7 meters range, somehow emulating a perfectly radial communication disk. The differences in these two lines can be explained by the spatially irregular coverage of the wireless radio communications. The blue line reflects more accurately this noisy nature by spreading the number of robots more evenly between 3 and 9 connections in Fig. 3 and producing a second local maximum for 4 connections in Fig. 4. This maximum can be explained by the increase in N and α. The increase in α forces robots to try to keep more neighbors in their communication radius, leading to robots aggregating in a smaller space. This effect is magnified by the increase of robots in the swarm. Thus, when robots lose or gain connections they lose 1 or 2 connections with $N = 20$ but they lose 4 or 5 connections with $N = 40$. The video data processed with *SwisTrack* always gives the correct number of neighbors since all robot positions are known, thus the red line better reflects the overall swarm behavior. We can see that connectivity measured with *SwisTrack* has a very good agreement with the connectivity measurements obtained in our submicroscopic simulations. The slight shift of the curve of the simulations in relation to the curve of *SwisTrack*, representing that robots have on average slightly less connections, is most likely a product of the inclusion of wireless communication realism (noisy fading and ellipsoidal communication area) in the simulations through the OmNET++ plugin. These results also show a very good agreement with the results presented in the original case study work [20].

Fig. 5 and Fig. 6 display the dispersion metric results. Real robots results are obtained only using the video data

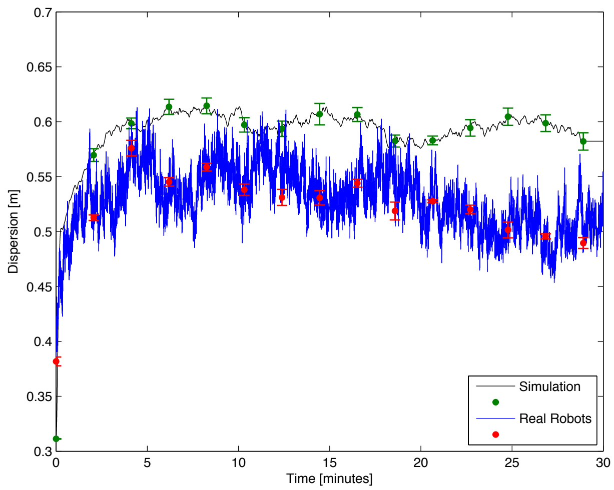


Fig. 5. Dispersion metric: average distance of robots to swarm center of mass throughout a run. Results for 20 robots.

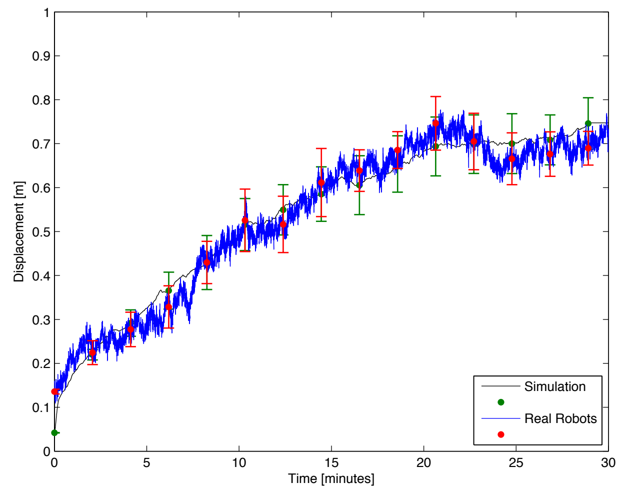


Fig. 7. Displacement metric: average distance of swarm center of mass to arena center throughout a run. Results for 20 robots.

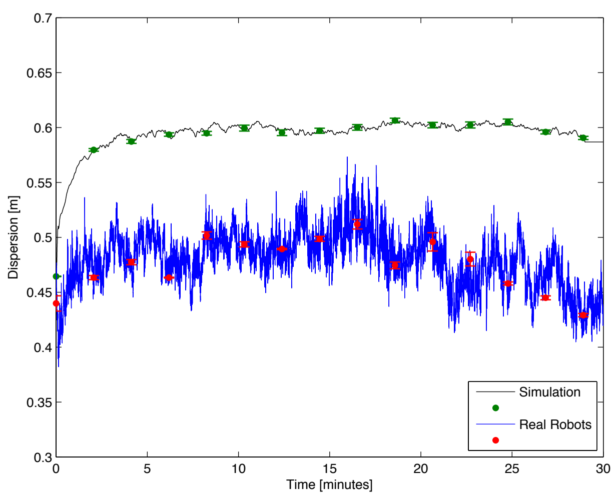


Fig. 6. Dispersion metric: average distance of robots to swarm center of mass throughout a run. Results for 40 robots.

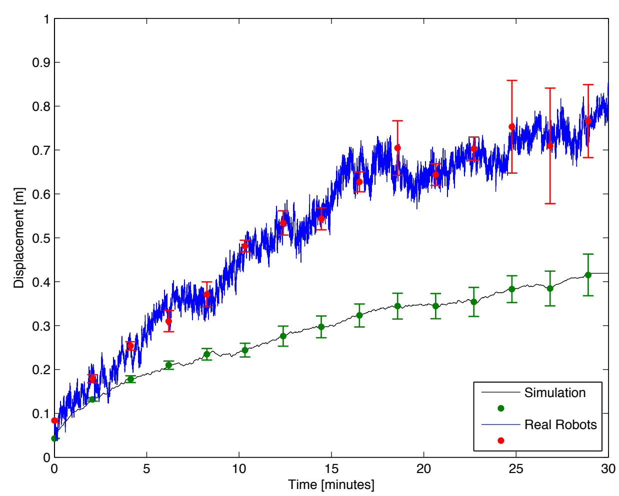


Fig. 8. Displacement metric: average distance of swarm center of mass to arena center throughout a run. Results for 40 robots.

processed with *SwisTrack*, since robots do not have localization capabilities and are unaware of their own location as well as the location of others. We can see that despite a small difference between real robots and submicroscopic simulations, the results still show a good agreement. As expected, the distance to the swarm center of mass is close to zero, smaller than the communication radius (0.7 meters) and constant (within some bounds) throughout the run. The small variations in this distance indicate an expansion and contraction motion performed by the swarm while losing and consequently trying to regain connections. This can be observed mainly in the real robots results, since the elevated number of runs performed in simulations diminishes the effect.

In Fig. 7 and Fig. 8 we present the displacement metric results for $N = 20$ and $N = 40$. Again, real robots results are obtained only using the video data processed with *SwisTrack*, for the reasons previously stated. As expected, displacement distance is close to zero at the beginning and increases throughout the run. For $N = 20$, submicroscopic simulations and real robot experiments show perfect agreement. However, for $N = 40$, despite distance increasing in both simulation and real robots, we observe that the rate of increase is doubled from simulation to real robots. A possible explanation for this effect is the difference in the obstacle avoidance behavior. While in submicroscopic simulations *e-pucks* are considered as perfect cylindrical blocks, in reality *e-pucks*' bodies are translucent. This leads to some collisions between robots, being this effect greatly increased when the number of robots is doubled and they are forced to aggregate in a smaller space (because α is also doubled). Robots motion becomes less predictable and more stochastic and as a result the displacement of the whole swarm is increased, much in the same manner as a random walk with increased turning probability. This difference also helps explain the slightly worst matching (with respect to Fig. 5) between reality and simulation in Fig. 6.

VI. Conclusion

We obtained a real world implementation of the wireless connected swarm case study, following an IR approach. We observed that such approach was able to maintain the wireless connectivity of a swarm of 40 real, resource-constrained robots. After careful experimentation we were also able to validate our submicroscopic model in a real world scenario with real robots.

Connectivity is the most fundamental of the metrics chosen to evaluate performance, since it relates most directly to the main objective of the swarm - to maintain wireless connectivity - and to the only information available at robot level - the number of wireless connections to neighbors. Results on connectivity show that, despite the high influence of noise in real world wireless communication, the overall swarm behavior implemented using an IR distributed control approach is able to maintain the expected connectivity. The submicroscopic model implemented using a realistic simulator is able to capture the swarm behavior and maintain connectivity but, despite considering noise in communications (with the OmNET++ plugin), are not able to faithfully recreate the perception of the number of wireless connections at the individual robot level.

The two remaining metrics, dispersion and displacement, relate to the spatial distribution of the swarm in the arena. Observing the dispersion metric results we conclude that, both in simulation and in reality, the swarm is able to maintain spatial compactness, since the average distance of robots to the swarm's center of mass is constant (or at least bounded). The displacement metric results show a perfect agreement between simulation and reality for $N = 20$. This result gives us further confidence that our submicroscopic model is an accurate representation of the real implementation of the wireless connected swarm case study. For $N = 40$, the results for simulation and reality are not in perfect agreement, leading us to believe that further refinement in the parametrization of the submicroscopic model could still be performed in order to improve our lower layer model.

In the future, we intend to further improve the macroscopic GSPN model in a corresponding multi-level methodology, so as to obtain a full range of models increasing in abstraction where properties of the system can be analyzed and verified quantitatively and qualitatively. We would like to validate our abstraction methodology also on a slightly more complex algorithm for the wireless connected swarm case study, considering the sharing of neighborhood information among robots, as presented in [14]. We intend to apply our IR formalism and our multi-level modeling methodology to this algorithm, in order to investigate situations where the microscopic-to-macroscopic (or individual-to-swarm) mapping might be less straightforward to capture accurately because of the additional complexity of the coordination algorithm.

References

[1] S. Berman, A. Halasz, and M. A. Hsieh. Ant-Inspired Allocation: Top-Down Controller Design for Distributing A Robot Swarm among Multiple Tasks. In Y. Xiao, editor, *Bio-inspired Computing and Networking*. CRC Press, New York, 2011.

[2] S. Berman, A. Halasz, M. A. Hsieh, and V. Kumar. Optimized stochastic policies for task allocation in swarms of robots. *IEEE Transactions on Robotics*, 25(4):927–937, 2009.

[3] C. G. Cassandras and S. Lafortune. *Introduction to Discrete Event Systems*. Springer, second edition, 2008.

[4] C. 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 Behaviour (SAB-2006), Swarm Robotics Workshop*, pages 103–115, Rome, Italy, 2006.

[5] N. Correll and A. Martinoli. Modeling and designing self-organized aggregation in a swarm of miniature robots. *International Journal of Robotics Research*, 30(5):615–626, 2011.

[6] W. C. Evans, G. Mermoud, and A. Martinoli. Comparing and modeling distributed control strategies for miniature self-assembling robots. In *International Conference on Robotics and Automation*, pages 1438–1445, Anchorage, Alaska, USA, 2010.

[7] G. M. Hodgson. What Is the Essence of Institutional Economics ? *Journal of Economic Issues*, 34(2):317–329, 2000.

[8] P. Levis, S. Madden, J. Polastre, R. Szewczyk, A. Woo, D. Gay, J. Hill, M. Welsh, E. Brewer, and D. Culler. Tinyos: An operating system for sensor networks. In *Ambient Intelligence*. Springer Verlag, 2004.

[9] T. Lochmatter, P. Roduit, C. Cianci, N. Correll, J. Jacot, and A. Martinoli. SwisTrack - a flexible open source tracking software for multi-agent systems. In *2008 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 4004–4010. Ieee, Sept. 2008.

[10] A. Martinoli, K. Easton, and W. Agassounon. Modeling swarm robotic systems: A case study in collaborative distributed manipulation. *Int. Journal of Robotics Research*, 23(4):415–436, 2004.

[11] G. Mermoud, U. Upadhyay, W. C. Evans, and A. Martinoli. Top-Down vs Bottom-Up Model-Based Methodologies for Distributed Control: A Comparative Experimental Study. In *Proc. of the Twelfth Int. Symp. Experimental Robotics, December 2010*, New Delhi, India, 2014. Springer Tracts in Advanced Robotics, Vol. 79. To appear.

[12] O. Michel. Webots TM : Professional Mobile Robot Simulation. *Advanced Robotic*, 1(1):39–42, 2004.

[13] F. Mondada, M. Bonani, X. Raemy, J. Pugh, C. Cianci, A. Klaptocz, S. Magnenat, J. Zufferey, D. Floreano, and A. Martinoli. The e-puck, a robot designed for education in engineering. In *Proceedings of the 9th conference on autonomous robot systems and competitions*, volume 1, pages 59–65, 2009.

[14] J. Nembrini, A. F. T. Winfield, and C. Melhuish. Minimalist Coherent Swarming of Wireless Networked Autonomous Mobile Robots. *From animals to animats 7: Proceedings of the seventh international conference on simulation of adaptive behavior*, pages 273–282, 2002.

[15] J. N. Pereira, P. Silva, P. U. Lima, and A. Martinoli. Formalizing Institutions as Executable Petri Nets for Distributed Robotic Systems. In T. Lenaerts, M. Giacobini, H. Bersini, P. Bourgine, M. Dorigo, and R. Doursat, editors, *Advances in Artificial Life, ECAL 2011*, pages 646–653, Paris, France, 2011. MIT Press.

[16] J. N. Pereira, P. Silva, P. U. Lima, and A. Martinoli. Formalization, Implementation, and Modeling of Institutional Controllers for Distributed Robotic Systems (in print). *Artificial Life*, 20(1), 2013.

[17] P. Silva and P. U. Lima. Institutional Robotics. In *Proc. of ECAL 2007 - 9th European Conference on Artificial Life*, pages 157–164, Lisboa, Portugal, 2007.

[18] A. Varga. Software tools for networking: OMNeT++. *IEEE Network Interactive*, 16(4), 2002.

[19] A. Winfield and J. Nembrini. Safety in numbers : fault-tolerance in robot swarms. *International Journal of Modelling, Identification and Control*, 1(1):30–37, 2006.

[20] A. F. T. Winfield, W. Liu, J. Nembrini, and A. Martinoli. Modelling a wireless connected swarm of mobile robots. *Swarm Intelligence*, 2(2-4):241–266, Sept. 2008.