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

Scalable, Fault-Tolerant and Distributed Multi-Robot Patrol in Real World Environments

David Portugal, *Student Member, IEEE* and Rui P. Rocha *Member, IEEE*

***Abstract*— Despite the focus that multi-robot patrolling has received recently, there is a manifest lack of practical real-world implementations of such systems. Beyond that, the existing ones have been mainly focused on centralized policies to coordinate the team of agents. The present work addresses realistic patrol in indoor environments with teams of arbitrary number of autonomous robots performing a distributed, scalable and fault-tolerant strategy for multi-robot coordination in patrolling missions. Agents decide their actions locally and adapt to the system's needs using distributed communication. The work is validated through experiments in a large indoor real-world environment with a team of autonomous mobile robots.**

I. INTRODUCTION

Over the past couple of decades, research in multi-robot systems (MRS) has witnessed progress as never before. More particularly, robots have increasingly been used in military and security applications, taking advantage of space distribution, parallelism, task decomposition and even redundancy. In this context, there have been several advances in multi-robot coverage and patrolling.

In the coverage problem, the environment is usually modeled as a grid-like map requiring the team of robots to sweep all cells of the environment. Whereas in the area patrolling problem, it is common to abstract the environment through a topological, graph-like map and robots are expected to have improved sensing abilities, meaning that they need to visit regularly all important places in the environment (*i.e.*, vertices of the graph), without necessarily going everywhere.

This work addresses MRS in cooperative patrolling missions in realistic scenarios. Being monotonous and repetitive, these missions may also be dangerous (*e.g.*, patrolling in hazardous environments). Therefore, using MRS in this context can be advantageous to secure human lives in areas of application like mine clearing, rescue operations or surveillance, enabling human operators to be occupied in nobler tasks like monitoring the system from a safe location.

In the next two sections, a literature review is conducted; the Multi-Robot Patrolling Problem (MRPP) is defined, the performance metric is presented and the contributions of the paper are described. The following section describes the patrolling strategy for teams of robots adopted in this paper. Sections 5 and 6, present the experimental scenario, results and discussion of the facets of the problem. Finally, the article ends with conclusions and future work.

II. RELATED WORK

Research on the patrolling problem has focused on three different fronts: adversarial patrol, perimeter patrol and area patrol. In adversarial patrol, the team of robots assumes the existence of an intruder and the aim is to coordinate itself to quickly capture the opponent. On the other hand, perimeter and area patrol are concerned with monitoring, collecting information, searching for objects or detecting anomalies, while at the same time, guaranteeing frequent visits to strategic places in the environment. In perimeter patrol, the agents move in the boundaries of the environment, whilst in area patrol, agents conduct their tasks throughout the environment [1]. Henceforth, the focus is on the latter.

Several contributions to the MRPP at a theoretical level have already been presented [2], [3]. It has been shown that the problem is NP-Hard. Nevertheless, based on topological representations of the environment and using global/centralized information, it is acknowledged in the literature that optimal patrolling can be obtained for a single robot by computing a TSP cycle[1], in the patrol graph. In the multi-robot case, computing optimal partitions of the graph and having each robot following TSP cycles inside each region usually lead to superior performance when compared to having all robots follow the same global TSP cycle in the graph, equally distributed in time and space. Yet, TSP cycles are not trivial to compute in sparse topologies (as most real world environments), not even existing in most cases.

Beyond these contributions, some authors have proposed distinct strategies for multi-robot coordination in patrolling missions based on a variety of concepts. Simple pioneer architectures using agents guided to locations that have not been visited for a while were firstly introduced in [4]. Other models have been explored subsequently, *e.g.*, based on task allocation [5], auction-based coordination [6] or Markov decision processes [7].

Despite the diversity of methods proposed, there is an evident lack of implementation using physical MRS. Only sporadic studies have gone beyond simulations. Any simulator uses simplifying approximations to some extent, which may perhaps jeopardize the validity of the outcome. Furthermore, the MRPP is mainly a practical problem and it is essential to validate convincing real world solutions.

This work was supported by a PhD scholarship (SFRH/BD/64426/2009), the CHOPIN research project (PTDC/EEA-CRO/119000/2010) and by the Institute of Systems and Robotics (project PEst-C/EEI/UI0048/2011), all of them funded by the Portuguese science agency "Fundação para a Ciência e a Tecnologia" (FCT).

David Portugal and Rui P. Rocha are with the Institute of Systems and Robotics (ISR), Dept. of Electrical and Computers Eng., University of Coimbra, 3030-290 Coimbra, Portugal {davidbsp, rprocha} at isr.uc.pt

[1]TSP stands for the well-known Traveling Salesman Problem (an NP-hard problem).

In the past, Cabrita *et al.* [8] successfully employed a team of Roomba robots, which navigated through indoor corridors, aiming at monitoring the environment by collecting samples of alcohol concentration and temperature. To this end, the Multilevel Subgraph Patrolling (MSP) algorithm, which partitions the environment in regions and assigns a region to each robot [9], was adopted. Similarly, Iocchi *et al.* [10] tested both cyclic and partitioning strategies, addressing coordinated robot behavior, and validated their experiments with realistic simulations and through Erratic platforms in an indoor environment. Finally, Pasqualetti *et al.* [11] focused on constructing tours using graph-theoretic techniques, instructing the robots to travel according to an *Equal-Time-Spacing* trajectory. Experiments were conducted in an indoor lab scenario also using Erratic mobile robots.

These approaches have in common the fact that the patrolling routes for each robot were computed *a priori* using global information and passed on to the robots. In this work, robots have the capability to decide online their own patrol route according to the state of the system at the moment, without requiring a central planner. Thus, a distributed approach for coordination is verified in a large real-world indoor facility using a team of patrolling robots.

III. Preliminaries

A. Problem Definition and Performance Metric

As seen before, it is common to represent the area to patrol by an undirected connected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with $v_i \in \mathcal{V}$ vertices and $e_{i,j} \in \mathcal{E}$ edges. Therefore, $\mathcal{G}$ corresponds to the topological map for the patrolling mission, which is obtained from a metric representation, assumed to be known *a priori*, by means of a graph extraction algorithm [12].

Having a graph representation, vertices correspond to important places or landmarks and edges represent the connectivity between those locations. Hence, the MRPP can be reduced to coordinate robots in order to visit frequently all vertices of the graph, ensuring the absence of atypical situations, with respect to a predefined optimization criterion.

Diverse criteria have been previously proposed to assess the effectiveness of multi-robot patrolling strategies. Typically, these are based on the idleness of the vertices, the frequency of visits or the distance traveled by agents [10]. In this work, the first is considered, because it measures how long it has been since the last visit from any agent in the team to a specific location, being intuitive to analyze and brought into confrontation with the possibility of attacks to the system. It is assumed that in the beginning of the experiments, all vertices $v_i \in \mathcal{V}$ have an idleness value of zero, *i.e.*, $\mathcal{I}_{v_i}(0) = 0$.

B. Contributions

This work describes the implementation of a system for multi-robot patrol in realistic scenarios, aiming to fill a gap in the present state-of-the art by validating a distributed approach, where fully autonomous agents decide locally their patrol routes according to the state of the system. This is done using a strategy recently proposed [13] and briefly described in the next section, which is inspired on a Bayesian-based formalism. It is shown that agents can coordinate effectively, using distributed communication and different teamsizes. Additionally, it is also demonstrated that the approach is robust to robot failures.

IV. State-Exchange Bayesian Strategy

In a previous study of the authors [13], it was shown that methods based on Bayesian inspiration can effectively solve the MRPP. These methods can deal with uncertainty and actions are selected according to the state of the system at the time, resulting in adaptive and distributed cooperative patrolling. More particularly, the State-Exchange Bayesian Strategy (SEBS) prevents robots from interfering with other teammates' actions while, at the same time, aims to maximize a local gain function, which guarantees that all positions in the environment are visited regularly by all agents. Stage, a recognized simulator by the robotics community, together with ROS [14], which is used to program individual robots, were adopted in [13] to compare the performance of diverse state-of-the art multi-robot patrol techniques. SEBS outperformed the remaining methods. As a consequence, it is currently chosen as the preferred strategy for coordination of a team of mobile robots, in a patrolling mission in the real world.

SEBS is reviewed in this section. Note that agents decide asynchronously which place to move next when they reach their current location. To that end, a fundamental random variable, which simply represents the act of moving (or not) to a neighbor vertex is defined as:

$$move = \{true, false\}. \tag{1}$$

The variables which influence each robot's individual decision are presented in the next sections with special focus on the selection of proper statistical distributions to model the data, in order to ensure the quality of the results. Afterwards, it is shown how the local decision-making process is automated, applying Bayes Rule.

A. Gain

When reaching a vertex of the navigation graph, each robot is faced with a decision stage, where it must decide the direction it should travel next. Therefore, the Gain G_A of moving from the current vertex (v_0) to a neighbor vertex (v_A) is proportional to the difference of the idleness values when moving to v_A at a constant speed (c):

$$G_A(t) = c \cdot \left(\frac{\mathcal{I}_{v_A}(t) - \mathcal{I}_{v_A}(t + \Delta t)}{|e_{val}|} \right), \tag{2}$$

where t is the current instant, $t + \Delta t$ is the arrival time in v_A and $\Delta t = |e_{0A}|/c$. Note that $\mathcal{I}_{v_A}(t + \Delta t) = 0$ when the robot reaches v_A, therefore G_A does not take negative values.

In most cases, $|e_{val}|$ takes on the value of $|e_{0A}|$, which is the distance between the two vertices, given by the weight of the edge that connects v_0 to v_A. However, constraint (3) is imposed in order to dimension $|e_{val}|$, avoiding occasional

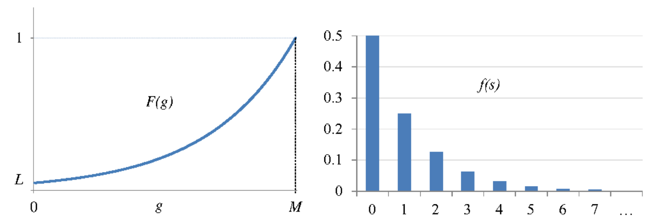

Fig. 1. a) Distribution function of Gain.
b) Distribution function of State.

situations where robots may get trapped in local optima (*i.e.*, repeatedly visiting vertices that are very close to each other):

$$|e_{val}| = \begin{cases} |e_{min}|, \textbf{ if } \max\{e_{0A},\ldots,e_{0\beta}\} > 2\min\{e_{0A},\ldots,e_{0\beta}\} \\ \qquad\qquad\qquad\qquad\qquad \wedge\ |e_{0A}| < |e_{min}| \\ |e_{0A}|, \textbf{ otherwise}. \end{cases} \tag{3}$$

Logically, higher values of gain rapidly have more influence in the robot's decision. Therefore, the distribution function $F(g)$ of G_i is defined as a monotonically increasing function, following the exponential model shown on Fig. 1:

$$F(g) = ae^{bg}; \quad a > 0, \tag{4}$$

Which is equivalent to:

$$F(g) = L \cdot \exp\left(\frac{\ln(1/L)}{M} g\right), \tag{5}$$

with:

$$L, M > 0 \quad \text{and} \quad g < M. \tag{6}$$

L and M are constants that control the distribution function. More specifically, L is the y-intercept, which controls the probability values for lower gains and M is the gain saturation, beyond which the probability values are maximum; $F(g \geq M) = 1$. These constants are simply defined as a value close to 0 for L, *e.g.*, 0.1 was used in the experiments; and M is calculated through (2) using an upper bound of $\mathcal{I}_{v_A}$.

B. State

In collective operations with a common objective, coordination between agents plays a fundamental role in the success of the mission. Particularly in this context, it is highly undesirable that agents move to the same positions. Hence, vertex state S_i is defined as a discrete variable that represents the number of robots that intend to visit a given vertex v_i involved in the decision process of robot r, which is currently located in vertex v_0:

$$S_i \in \mathbb{N}_0 \cap [0, R-1]; \quad R > 1. \tag{7}$$

As before, it is necessary to define a statistical distribution to model the vertex state. The greater the number of teammates in the vicinity of a robot, it becomes increasingly unlikely for the robot to move in that direction. To describe this behavior, the following discrete probability distribution, which uses a geometric sequence of ratio ½ has been defined:

$$f_{S_i}(s)_{R\to\infty} = P(S_i = s)_{R\to\infty} = \frac{1}{2^{s+1}}, \tag{8}$$

as shown in Fig. 1. This geometric sequence is used to guarantee that the total probability for all S_i equals 1:

$$\sum_{s=0}^{R-1} f_{S_i}(s) = 1, \tag{9}$$

Eq. (8) assumes that the number of robots R is unknown and can be arbitrarily high. However, since the robots communicate among themselves, it is more realistic to consider R as known and with finite values. Therefore, the following approximation to (8) is assumed:

$$f_{S_i}(s) = P(S_i = s) = \frac{2^{R-(s+1)}}{2^R - 1}; \quad R > 1. \tag{10}$$

C. Robot Decision

Each decision to move from a vertex v_0 to v_A is considered independent and agents have the ability to choose the action which has the greatest expectation of utility, weighted by the effects of all possible actions.

To this end, robots locally update the instantaneous idleness time values online, by communicating to other robots when they reach another vertex of the navigation graph, in a distributed way. This enables the calculation of the Gain of moving from the current vertex to any of its neighbors. Likewise, by receiving other robots' intentions, agents can calculate vertex state.

Having the likelihood distribution models $P(G_i|move)$ and $P(S_i|move)$ defined respectively by $F(g)$ and $f(s)$; and assuming the prior knowledge as uniform, $P(move)$, where all decisions are equiprobable; agents calculate the probability of moving to a specific vertex i given its gain G_i and the vertex state S_i, applying Bayes rule:

$$P(move|G_i, S_i) = \frac{P(move)P(G_i|move)P(S_i|move)}{P(G_i)P(S_i)}, \tag{11}$$

where the denominator term is regarded as a normalization factor, being often omitted for simplification purposes. The move to the neighbor vertex with the maximum a posteriori (MAP) probability is the one chosen by the agent:

$$move_{MAP} = \underset{move}{\operatorname{argmax}}\ P(move|G_i, S_i) \tag{12}$$

V. Setting Up the Experiments

Experiments were conducted in a large indoor scenario, namely the floor 0 of the Institute of System and Robotics (ISR), in the University of Coimbra. Fig. 2 shows the extracted topological map on top of the 67.85 × 26.15 meters environment and a few snapshots of the corridors of the ISR.

The topology obtained is a non-complete, connected and sparse graph, as most real world environments, with a low algebraic connectivity: 0.025, given by the Fiedler Value λ_1 - the smallest non-zero eigenvalue of the graph's Normalized Laplacian matrix [1].

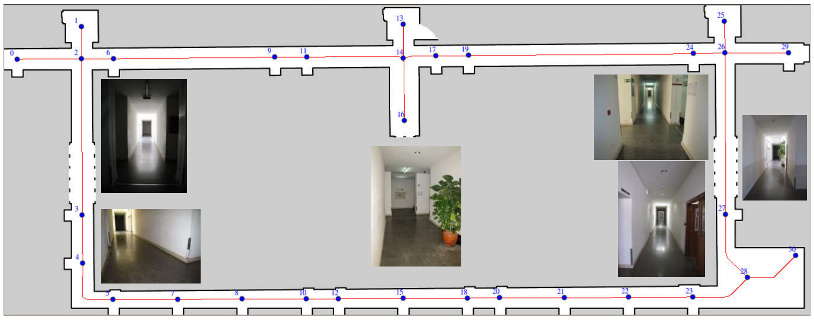

Fig. 2. Topological map of the "ISR-Floor0" Environment.

Fig. 3. Robots used in the experiments.

TABLE I

EXPERIMENTS WITH 1 TO 3 ROBOTS (ALL VALUES IN SECONDS)

Teamsize R	$\overline{\mathcal{I}_{\mathcal{G}}}$	$\max(\overline{\mathcal{I}_{\mathcal{V}}})$	$\widetilde{\mathcal{I}_{\mathcal{G}}}$	σ	τ
1	336.676	412.207	370.994	78.769	1648.828
	332.745	407.897	366.677	77.892	1631.590
	331.615	406.387	365.345	77.626	1625.550
2	168.921	309.455	137.267	64.210	1237.821
	180.761	296.085	180.293	56.064	1184.341
	170.267	328.300	146.890	62.603	1313.201
3	128.875	273.670	116.269	54.893	1094.682
	116.248	216.020	95.150	44.356	864.081
	112.954	200.030	101.923	36.066	800.121

Experiments in a large indoor environment demonstrate the potential of the patrolling strategy proposed as a solution for real-world multi-robot surveillance. To that end, one must overcome noisy sensor readings, localization issues and even robot failures, which are usually ignored or not precisely modeled in simulation experiments. Therefore, a team of three Pioneer-3DX robots, equipped with an Hokuyo laser in the front and a laptop on top was used, as seen in Fig. 3. Each laptop runs the ROS's navigation stack using the Adaptive Monte Carlo (AMCL) algorithm for localization, and assuring that all robots navigate safely by heading towards their goals. Robots avoid collisions with walls and other robots, reaching speeds of up to 1 m/s.

For communication, a distributed publish/subscribe mechanism has been used, due to its built-in integration in ROS. Moreover, each robot runs its own ROS master node (*roscore*) and multimaster communication is provided using the *wifi_comm*[2] package. This means that there is no central point of failure in the system. Also, given that robots only share their current and future immediate goals, the bandwidth requirements are negligible even with large teams.

In the beginning of each test, the graph of the environment is loaded by every robot. A ROS node (*i.e.*, a ROS application) is responsible for advertising the start of the mission and collect results during the experiments[3]. These results are examined in the next section.

VI. RESULTS AND DISCUSSION

In this section, the results of the multi-robot patrolling strategy employed in the ISR corridors are analyzed. Firstly experiments with one, two and three robots were conducted. Each experiment was repeated 3 times. Afterwards, in order to further demonstrate the scalability of the approach, virtual robots were added to the team, and 3 trials with 6 agents $(3+3)$ and 9 agents $(3+6)$ were also conducted. Finally, to prove its robustness, experiments which included failures in the robots at different time instants are analyzed.

Not only is the average graph idleness along time, $\overline{\mathcal{I}_{\mathcal{G}}}$, examined as a global performance metric [1], but also the median $\widetilde{\mathcal{I}_{\mathcal{G}}}$, standard deviation σ, and the maximum average idleness of all vertices, $\max(\overline{\mathcal{I}_{\mathcal{V}}})$. Each experiment finishes after 4 complete patrolling cycles, *i.e.*, every $v_i \in \mathcal{V}$ is visited at least 4 times. It is shown that this stopping condition is adequate, as the $\overline{\mathcal{I}_{\mathcal{G}}}$ converges in all experiments, when no faults occur. Additionally, a value of $|e_{min}| = 7.5m$ was used in all experiments.

During the course of experiments, the estimated sum of distances traveled by the physical robots was 23 Kms.

A. Initial Experiments

Table I summarizes the first set of experiments using one to three robots. It can be seen that the $\overline{\mathcal{I}_{\mathcal{G}}}$ values as well as the total mission time τ decreases with teamsize, as expected. In all cases the median is fairly close to the average value, meaning that most data is divided around the mean.

A particularly interesting result is shown by the $\max(\overline{\mathcal{I}_{\mathcal{V}}})$, which is low for the case of 1 robot. This happens because of the existence of a main loop in the environment, which results in fairly uniform visits to all vertices of the graph. In the cases of 2 and 3 robots, the distance to the average value increases due to robots occasionally meeting in the environment and coordinating themselves. This often results in changes to their heading direction and consequently the frequencies of visits are not as balanced. This can be confirmed by the standard deviation, which is around 23% using 1 robot and 35% and 37% for a teamsize of 2 and 3 robots respectively. In addition, the $\overline{\mathcal{I}_{\mathcal{G}}}$ value in the end of the experiment with one robot is lower than $\widetilde{\mathcal{I}_{\mathcal{G}}}$, meaning that the distribution is negatively skewed, whereas in the case of 2 and 3 robots, it is positively skewed[4].

Figure 4 shows the evolution of the idleness in three different experiments with 1 to 3 robots. It is shown that $\overline{\mathcal{I}_{\mathcal{G}}}$

[2]Available at http://www.ros.org/wiki/wifi_comm

[3]The code is available at http://www.ros.org/wiki/patrol

[4]A video demonstrating a partial experiment with 3 robots is available at: http://isr.uc.pt/~davidbsportugal/videos/IROS2013

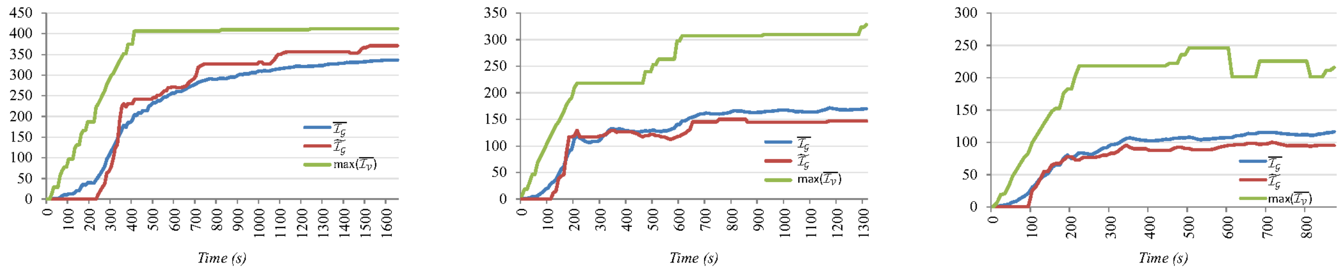

Fig. 4. Evolution of the idleness along time: a) with 1 robot, b) with 2 robots and c) with 3 robots.

TABLE II

EXPERIMENTS WITH 6 AND 9 ROBOTS (ALL VALUES IN SECONDS)

Teamsize R	$\overline{\mathcal{I}_G}$	$\max(\overline{\mathcal{I}_V})$	$\widetilde{\mathcal{I}_G}$	σ	τ
6 (3+3)	71.097	152.625	65.483	27.130	610.500
	72.165	140.725	67.043	24.418	562.900
	77.332	150.145	72.938	27.350	600.580
9 (3+6)	48.623	102.305	47.395	16.499	409.220
	50.239	90.580	54.157	16.083	362.320
	51.687	105.12	52.271	19.622	420.480

converges in all cases, meaning that it is no longer affected by the initial conditions, seeing as all vertices start with a null value of idleness.

B. Scalability

In the previous subsection, the number of robots R is limited to the physical robots available. Nevertheless, the distributed patrolling method used supports an arbitrary high teamsize.

In order to test the approach with greater teamsize and evaluate its scalability, virtual agents, running in the stage simulator, were added to the team, resulting in a mixed team of real and simulated robots. Three trials were conducted with a total of 6 agents composed by 3 physical robots and 3 simulated ones; and three more trials were performed with a teamsize of 9, composed by 3 physical robots and 6 simulated ones. Similarly to [10], the software layer is used unchanged both on real robots and in simulation, since each agent is running ROS.

Results in Table II show that the overall values of $\overline{\mathcal{I}_G}$, $\max(\overline{\mathcal{I}_V})$, $\widetilde{\mathcal{I}_G}$, σ and τ are within the expected, following the trend shown in the cases of two and three robots.

In order to assess the scalability of the approach, Balch's speedup measure $\upsilon(i)$ [15], a classical scalability metric, was calculated:

$$\upsilon(i) = \frac{\Psi(1)/i}{\Psi(i)}, \tag{13}$$

$\Psi(i)$ is the performance for i robots, given by $\overline{\mathcal{I}_G}$. Fig. 6 presents the speedup chart using different teamsizes. Despite improvement of performance with teamsize as seen previously, this chart shows that the contribution of adding robots progressively decreases, therefore the system enters in sublinear performance ($\upsilon(i) < 1$), due to the more frequent

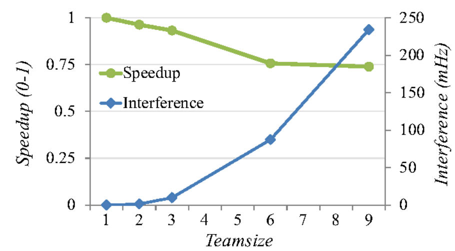

Fig. 6. Interference and Speedup against Teamsize.

existence of spatial limitations, which, in turn, increases the interference between robots. Interference is measured as the frequency of different agents sharing nearby areas and affecting each others' decisions. This increasing trend of the interference with teamsize is common to all MRPP approaches, as shown in a benchmark comparison in [13]. However, SEBS presents a smoother slope when compared to other approaches, because it takes into account teammates' intentions.

C. Fault-Tolerance

One of the main advantages of providing the patrol robots with means for deciding their moves in the environment is the absence of a centralized coordinator, which would represent a critical point of failure. A distributed autonomous robotic system, such as the herein presented, enables redundancy, remaining functional if some of the agents fail.

To demonstrate the robustness of the approach, three experiments using the Pioneer 3-DX robots available were planned. In these experiments a robot is shutdown at different instants of time, aiming at studying the effect of the faults in the overall performance, as well as how the system evolves.

In the first experiment a robot is shutdown after 200 seconds from the beginning of the experiment. Similarly, in the second and third experiment, a robot is shutdown after 400 and 600 seconds respectively. The other robots assume that a teammate has failed when no message has been received from it in a period of 2 minutes.

Generally, it can be seen in Table III that the results obtained in the first experiment resemble those obtained with two robots, as most of the experiment is spent with only two agents, due to the failure occurring in the beginning. On the other side, the results shown in the second and third experiment are closer to those obtained using three robots,

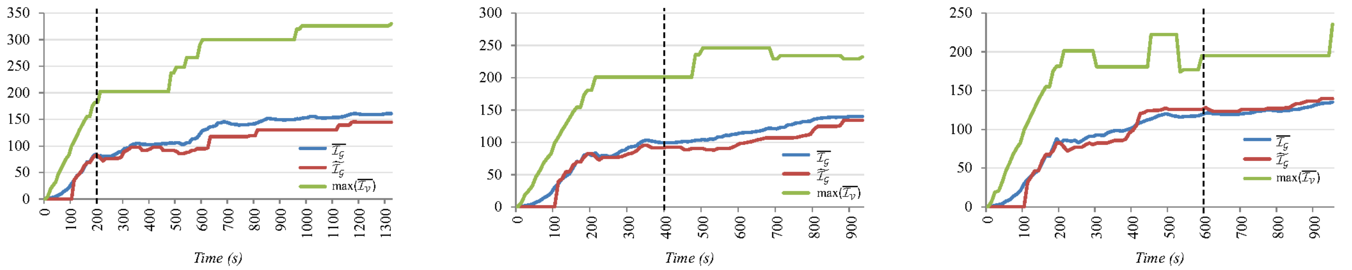

Fig. 5. Evolution of the idleness along time in experiments with robot failures. a) Failure at 200s. b) Failure at 400s. c) Failure at 600s.

TABLE III

EXPERIMENTS WITH 3 ROBOTS WITH FAILURE OF A ROBOT IN DIFFERENT INSTANTS OF TIME (ALL VALUES IN SECONDS)

Failure Time	$\overline{\mathcal{I}_G}$	max($\overline{\mathcal{I}_V}$)	$\widetilde{\mathcal{I}_G}$	σ	τ
200 s	160.975	330.225	144.846	62.825	1320.901
400 s	140.128	232.290	134.177	45.934	929.161
600 s	135.209	235.700	139.797	41.262	942.801

even though the performance is slightly inferior, as expected. In these experiments, especially in the third case, most of the mission takes place with three agents and the failure occurs towards the end, which is reached after 4 patrolling cycles. As a consequence, Fig. 5c shows that there was not enough time, after occurring the failure, for the values to converge to the two robots situation.

Analyzing now the influence of the failures in the evolution of the results, one can verify that in all three cases, when the failure occurs, the values of $\overline{\mathcal{I}_G}$ and $\widetilde{\mathcal{I}_G}$ tend to increase after a while, which is particularly visible in Fig. 5a and Fig. 5b. These results prove the robustness of the system, enabling graceful degradation, as long as one robot remains operational.

VII. CONCLUSIONS AND FUTURE WORK

In this work, the implementation of a distributed multi-robot system for patrolling an indoor infra-structure is presented. Making use of a simple Bayesian-based formalism, autonomous robots decide their patrol routes according to the state of the system at a given time.

Previous results had shown the superior performance of the approach when compared with other MRPP strategies. In this work, the results obtained have demonstrated that the approach is able to scale to a high number of robots, being robust to failures and having the ability to adapt to constraints, since the decision-making is done online with the information that each agent has collected about the system. Experiments were conducted using physical robots and mixed teams of both virtual and real agents, in a real-world environment, proving the effectiveness of the approach and the potential to use it in the real world.

In the future, the patrolling strategy could be extended in order to deal with breaks in the communication between robots. In addition, it would be interesting to study the influence of unforeseen dynamic obstacles in the real world, *e.g.*, people moving in the corridors. Finally, we intend to implement an estimation method to dimension a team of robots in a patrolling mission according to the environment topology and temporal constraints.

ACKNOWLEDGMENT

The authors gratefully acknowledge João B. Campos, Amadeu Fernandes, Micael Couceiro and Gonçalo Cabrita (ISR, University of Coimbra) for their contribution.

REFERENCES

[1] D. Portugal and R. P. Rocha, “Multi-Robot Patrolling Algorithms: Examining Performance and Scalability”. *Advanced Robotics Journal*, 27 (5), 325-336, March 2013.

[2] Y. Chevaleyre, “Theoretical analysis of the multi-agent patrolling problem”. *2004 Int. Conf.e on Agent Intelligent Technologies (IAT’04)*, Beijing, China, Sep. 20-24, 302-308, 2004.

[3] Y. Elmaliach, N. Agmon and G. Kaminka, “Multi-Robot Area Patrol under Frequency Constraints”. *2007 Int. Conf. on Robotics and Automation)*, 385-390, Rome, Italy, April 10-14, 2007.

[4] A. Machado, G. Ramalho, J. Zucker and A. Drogoul, “Multi-Agent Patrolling: an Empirical Analysis of Alternative Architectures”. Lecture Notes in Comp. Science, Vol. 2581, 155-170, Springer, 2003.

[5] F. Sempé and A. Drogoul, “Adaptive Patrol for a Group of Robots”. *Int. Conf. on Intelligent Robots and Systems*, Las Vegas, 2003.

[6] C. Pippin, H. Christensen and L. Weiss, “Performance Based Task Assignment in Multi-Robot Patrolling”. *Symposium on Applied Computing (SAC 2013)*, Coimbra, Portugal, March 18-22, pp. 70-76, 2013.

[7] J. Marier, C. Besse and B. Chaib-draa, “Solving the Continuous Time Multiagent Patrol Problem”. *2010 Int. Conf. on Robotics and Automation*, Anchorage, Alaska, USA, May, 2010.

[8] G. Cabrita, P. Sousa, L. Marques, A. de Almeida, “Infrastructure monitoring with multi-robot teams”, *Int. Conf. Intel. Robots and Systems*, WS Robotics for Env. Monitoring, Taiwan, Oct.18-22, 2010.

[9] D. Portugal and R. Rocha, “MSP Algorithm: Multi-Robot Patrolling based on Territory Allocation using Balanced Graph Partitioning”. *Symp. on Applied Computing*, 1271-1276, Sierre, Mar. 22-26, 2010.

[10] L. Iocchi, L. Marchetti and D. Nardi, “Multi-Robot Patrolling with Coordinated Behaviours in Realistic Environments”. *Int. Conf. on Intelligent Robots and Systems*, S. Francisco, USA, Sep.25-30, 2011.

[11] F. Pasqualetti, J. Durham and F. Bullo, “Cooperative Patrolling via Weighted Tours: Performance Analysis and Distributed Algorithms”. In *IEEE Transactions on Robotics*, 9, 1-8, 2012.

[12] D. Portugal and R. Rocha, “Retrieving Topological Information for Mobile Robots Provided with Grid Maps”. *Agents and Artificial Intelligence*, Comm. in Comp. Inf. Science, Vol. 358, Springer, 2013.

[13] D. Portugal and R. P. Rocha, “Decision Methods for Distributed Multi-Robot Patrol”. *Int. Symp. on Safety, Security and Rescue Robotics*, College Station, TX, USA, Nov.8-11, 2012.

[14] M. Quigley, B. Gerkey, K. Conley, J. Faust, T. Foote *et al.*, “ROS: an open-source Robot Operating System”, *Int. Conf. on Robotics and Automation*, WS Open Source Soft., Kobe, Japan, May 12-17, 2009.

[15] T. Balch and R. Arkin, “Communication in reactive multiagent robotic systems”. In *Autonomous Robots*, 1(1): 27-52, 1994.