

# Behavioral Control for Multi-Robot Perimeter Patrol: A Finite State Automata approach

Alessandro Marino, Lynne Parker, Gianluca Antonelli and Fabrizio Caccavale

**Abstract**— This paper proposes a multiple robot control algorithm to approach the problem of patrolling an open or closed line. The algorithm is fully decentralized, i.e., no communication occurs between robots or with a central station. Robots behave according only to their sensing and computing capabilities to ensure high scalability and robustness towards robots' fault. The patrolling algorithm is designed in the framework of behavioral control and it is based on the concept of *Action*: an higher level of abstraction with respect to the behaviors. Each Action is obtained by combining more elementary behaviors in the Null-Space-Behavioral framework. A Finite-State-Automata is designed as supervisor in charge of selecting the appropriate action. The approach has been validated in simulation as well as experimentally with a patrol of 3 Pioneer robots available at the Distributed Intelligence Laboratory of the University of Tennessee.

**Index Terms**— Behavioral control; Platoon of vehicles; Multi-robot systems; Border Patrol; Swarm Robotics.

## I. INTRODUCTION

The mission of patrolling a given region or border is of critical interest in modern societies. Border patrol is probably entering a new era since several countries are replacing human soldiers with robots with increasing autonomy. Examples where advanced testing facilities have been installed include United States [16], Israel and South Korea. In a few cases, armed robots are remotely controlled by a human operator. Similar projects, with guardian capabilities of civilian and military spaces, run in Japan, Singapore and Europe.

It is difficult to give an exact definition of robotic border patrol, in the sense that a patrolling mission may require different objectives to be fulfilled and may be subject to several constraints, depending on various conditions, such as the specific robot locomotion system, the kind and size of the border to patrol, the civilian or military applications, the number of available robots, their equipment and communication capabilities.

In [18] an analysis of the main patrolling task issues and some multi-agent-based solutions are presented. Several features, such as agents type, agents communication, coordination scheme, agents perception and decision-making,

A. Marino and F. Caccavale are with the Dipartimento di Ingegneria e Fisica dell'Ambiente, Università degli Studi della Basilicata, Viale dell'Ateneo Lucano 10, 85100, Potenza, Italy, {alessandro.marino, fabrizio.caccavale}@unibas.it

L. Parker is with the Department of Electrical Engineering and Computer Science, The University of Tennessee, 1122 Volunteer Blvd, TN 37996-3450, Knoxville, USA, lepark@utk.edu

G. Antonelli is with the Dipartimento di Automazione, Elettromagnetismo, Ingegneria dell'Informazione e Matematica Industriale, Università degli Studi di Cassino, Via G. Di Biasio 43, 03043, Cassino (FR), Italy, antonelli@unicas.it

are evaluated by using different evaluation criteria. The achievements in [18] have been further extended in [4], pointing out in more detail advantages and disadvantages of each multi-agent architecture, as well as the impact of the border geometry on the performance. In [17] and [10] graph-theory is used to find the optimal solution of a mathematical problem expressing a multi-robot surveillance problem. In [3] the authors analyze non-deterministic paths for a group of homogeneous mobile robots patrolling a frontier, under the assumption of an hostile agent trying to enter the area, where the latter has full knowledge of the algorithm.

According to our terminology, the term *robot* will denote a mobile machine, while *agent* may denote both a human or a robot. The following assumptions are adopted to develop the proposed solution to the multi-robot border patrol problem:

- Each robot can measure or estimate its position;
- Each robot knows or can estimate the geometric description of the border locally to its position;
- Each robot is characterized by a “visibility” area, where it recognizes the presence of another patrolling robot, a friend or an hostile agent;
- Each robot is characterized by its own safety area (contained in the visibility area) where other agents are not allowed to enter.
- Each robot is autonomous, it does not rely on a central computational unit; moreover, distributed algorithms such as consensus, that need an explicit exchange of information are not allowed.
- Each robot is aware of the existence of other patrolling robots, friends and hostile agents;
- The robots do not know the total number of patrolling robots;
- It is forbidden to the robots any kind of explicit communications.

It is worth noticing that the above assumptions are devoted at giving the maximum fault tolerance capability to the system; e.g., the robots might self-localize by adopting distributed SLAM algorithms, but this usually require explicit communications among them and eventually knowledge of the total number of robots in the patrolling team. Also, any kind of centralized approach here is not interesting, due to the inherently weakness of grouping all the computational effort in one single machine, even if remote.

The control objective is to patrol a given border. As stated above, this objective might be given in analytical form, and a proper functional could be designed to be minimized

by using a certain deterministic algorithm. This, however, is not necessarily the optimal solution from a practical point of view [3]. Let us consider, e.g., a functional that minimizes the time elapsed from two visits of a certain node; the definition of a node and the definition of an optimum criterion makes the patrolling algorithm predictable. On the other hand, a pure random movement of the robots is unlikely to be effective [18]. Although the presence of performance criteria/indexes may be important, it is usually difficult to model the environment in order to perform quantitative measurements; hence, the obtained results would necessarily be confined to the specific environments considered. With these characteristics in mind, the design of the control algorithm has been driven by the aim to transfer to the robots the heuristics in performing a patrol mission, rather than approaching the problem from a pure mathematical point of view. Moreover, one of the key points of this paper is the introduction of the concept of *action*, obtained by combining in a consistent way elementary behaviors. This is consistent with the adopted bottom-up approach in the actions selection mechanism, since it allows the developer to build an intelligent system on the basis of elementary components. Therefore, once a set of actions is defined, according to the requirements of the particular task, an *actions selection* mechanism need to be developed to properly select the best action; the optimum criterion may depend either on external stimuli or, eventually, on the internal state of the robotic team; the latter is often used to implement some form of learning and adaptivity.

The approach is first tested in simulation, by using the Player/Stage environment, and then is experimentally verified on a team of commercially available mobile robots, namely the Pioneer robots available at the Distributed Intelligence Laboratory of the University of Tennessee.

## II. PROPOSED APPROACH

The border patrol problem requires an high level of autonomy and the implementation of a reasoning method taken from those available in Artificial Intelligence (AI) literature [14]. Classical AI approaches generally consist in a deliberation based on symbols under first order predicate logic or probability theory.

In this paper a behavior-based approach, namely the Null-Space-based Behavioral control, is adopted [6]. An higher lever of abstraction, defined as Action is introduced to properly handle concurrent behaviors. Finally, a Supervisor, implemented as Finite State Automata, is in charge of selecting the proper Action. Figure 1 reports a sketch of the proposed control algorithm.

It is useful to clarify some aspects concerning behavioral control and the supposed *emergence* of an overall, macroscopic, behavior by connecting elementary behaviors. In an engineering problem, as the border patrol approached here, the control problem is given by its macroscopic definition; its decomposition into elementary behaviors, hence, is equivalent to the decomposition of a complex problem into simpler ones, given a set of technological constraints.

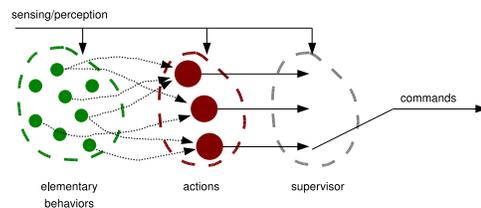


Fig. 1. Overall control schema.

On the other hand, in several biological or neurological studies, local interactions are first modeled and the overall, macroscopic, behavior of the dynamic system is seen as *emergent* and analyzed for what it is. Somehow, there is a kind of dichotomy between a *top-down* and a *bottom-up* behavioral approaches, where we do consider appropriate to follow the former.

Significant examples of behavioral control developed for multi-robot systems are given in [8] and [19]. According to the literature of multiple robots [20], the problem approached here is close to what is usually defined as *swarm*, i.e., a large group or robots that interact implicitly. The absence of explicit communication does not mean that there is not exchange of information; the robots may communicate indirectly, or *stigmergically*, leaving intentionally or not, some traces in the environment [12].

Although several paradigms are possible as in [9] or [7], in this work behaviors are handled in the framework of the Null-Space-based Behavior (NSB) approach, i.e., a competitive-cooperative approach recalled in Sect. II-B.

The use of the NSB approach introduces an higher level of abstraction (Sect. II-C) with respect to the elementary behaviors, thus allowing the developers to focus on the actions selection mechanism, via a suitably defined *Supervisor*, rather than on the command fusion problem.

### A. Elementary behaviors

In the following a brief review of the NSB approach is provided. Let  $\sigma \in \mathbb{R}^m$  be the mathematical representation of the behavior to be implemented (often referred as task) and  $p \in \mathbb{R}^n$  be the system configuration; in general, they are related via the following model

$$\sigma = f(p), \quad (1)$$

with the corresponding differential relationship

$$\dot{\sigma} = \frac{\partial f(p)}{\partial p} v = J(p)v, \quad (2)$$

where  $J \in \mathbb{R}^{m \times n}$  is the configuration-dependent task Jacobian matrix and  $v \in \mathbb{R}^n$  is the system velocity.

An effective way to generate motion references for the vehicles,  $p_d(t)$ , starting from the desired behavioral function,  $\sigma_d(t)$ , is to act at the differential level by inverting the (locally linear) mapping (2); in fact, this problem has been widely studied in robotics (see, e.g., [21] for a tutorial). A

typical requirement is to pursue minimum-norm velocity in a closed loop version, leading to

$$\mathbf{v}_d = \mathbf{J}^\dagger \left( \dot{\boldsymbol{\sigma}}_d + \boldsymbol{\Lambda} \tilde{\boldsymbol{\sigma}} \right), \quad (3)$$

where  $\boldsymbol{\Lambda}$  is a suitable constant positive-definite matrix gain and  $\tilde{\boldsymbol{\sigma}} = \boldsymbol{\sigma}_d - \boldsymbol{\sigma}$  is the task error.

### B. Composition of elementary behaviors via the NSB

As described earlier elementary behaviors are properly composed in more meaningful *actions*. Let the subscript  $i$  denote the degree of priority of the behavior (i.e., behavior 1 has the highest-priority); in the case of 3 tasks, according to [11], solution (3) is modified into

$$\mathbf{v}_d = \mathbf{v}_1 + \mathbf{N}_1 \mathbf{v}_2 + \mathbf{N}_{12} \mathbf{v}_3, \quad (4)$$

where

$$\mathbf{N}_1 = \left( \mathbf{I} - \mathbf{J}_1^\dagger \mathbf{J}_1 \right) \quad (5)$$

and  $\mathbf{N}_{12}$  is the Null space of the Jacobian obtained by stacking the higher priority Jacobians.

In this way, lower priority tasks are executed only in their components not affecting higher priority tasks; hence, differently from other command fusion approach, the output is predictable. Also, errors converge to zero can be guaranteed for properly defined tasks [5]. On the other hand, a differentiable analytic expression of the defined behaviors is required, so as to compute the required Jacobians.

### C. Specific Behaviors and Actions

In the case of a linear border it can be easily recognized that a set of elementary behaviors is:

- Stay on Frontier
- Patrol Frontier CW
- Patrol Frontier CCW
- Teammate Avoidance

whose semantics and analytical expressions are given in the Appendix A.

The definition of the elementary behaviors is conceived so as to satisfy one single behavioral function. As motivated above, it is appropriate, however, to compose the elementary behaviors into more complex behaviors; the latter are sometimes defined as *behaviors set* in the literature. Similar to the concept of behavior set, here an higher abstraction layer is introduced: the actions. As shown in Figure 1, an action is given by the proper union, via NSB, of several elementary behaviors and represents a macroscopic attitude of the robotic system. One single action can be active at once.

For the specific case of border patrol, with the elementary behaviors defined above, the actions are:

- Action Reach Frontier
- Action Keep Going
- Action Patrol Clockwise
- Action Patrol Counter-Clockwise
- Action Avoid Teammate

According to the definition in Sect. II-B, each action is given by elementary behaviors arranged in priority; e.g.,

the *Reach Frontier* action properly combines the elementary behaviors *Stay on Frontier* and *Teammate Avoidance*, depending on the sensed presence of other patrolling robots in the visibility range and the distance from the border. The analytical details on the specific actions are given in the Appendix B.

### D. The supervisor

Each robot decides the next action to be performed, based only on its sensing capabilities. The simplest way to achieve such a decisional process is based on the use of finite state automata playing the role of *Supervisor*. The main advantage is that all possible transitions between actions can be explicitly encoded; on the other hand, this can be hard to achieve as the number of actions increases and/or when facing highly dynamic environments.

A possible structure of the supervisor is shown in Figure 2. It is arranged in a hierarchical way by defining states and sub-states in the following way:

- State MS1: **If** the distance between the robot and the border is larger than the Visibility Range **and**
  - no teammate is in the safety area **then** *Action Reach Frontier* is active,
  - one or more teammates are in the safety area **then** *Action Avoid Teammate* is active.
- State MS2: **If** the distance between the robot and the border is smaller than the Visibility Range and larger than a given Threshold, **and**
  - no teammate is in the safety area **then** *Action Keep Going* is active,
  - one or more teammates are in the safety area **then** *Action Avoid Teammate* is active.
- State MS3: **If** the distance between the robot and the border is smaller than a given Threshold **and**
  - no teammate is in the visibility range **then** *Action Keep Going* is active,
  - there is a teammate on the left **then** *Action Patrol Clockwise* is active,
  - there is a teammate on the right **then** *Action Patrol Counter-Clockwise* is active.

The *Action Keep Going* is built in such a way that the patrol direction might vary according to a random variable every  $T$  seconds. At each time instant the robot can be in one of the *macro-state* MS1, MS2 or MS3, which correspond, respectively, to the condition in which the robot is far from the border, the robot is not far from the border (but its distance is large) and the robot is close to the border. The main reason behind this distinction is that in the state MS1 the robots try to reach the border (and avoid the teammates), in the state MS2 the robots behave as they are patrolling the border (and avoid the teammates), in the state MS3 the robots perform the patrolling mission and do not allow robots approaching the border to influence their motion; in the last state, avoidance of other teammates patrolling the border is achieved by activating the actions *Patrol CW* and/or *Patrol CCW*.

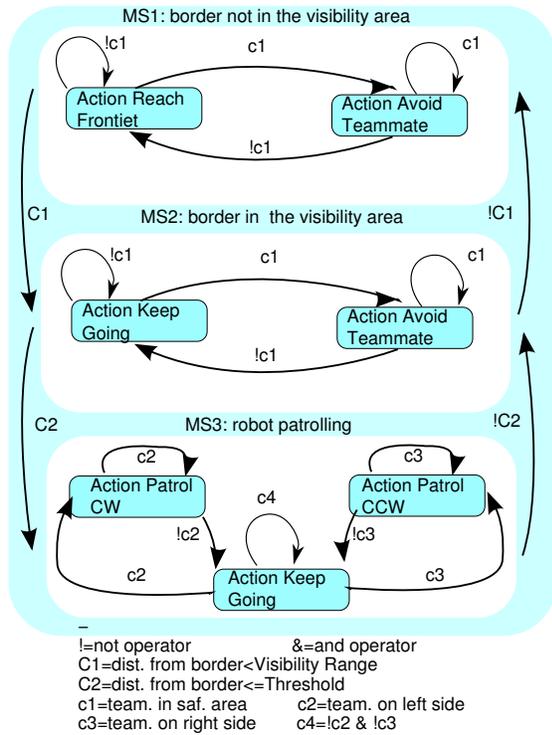


Fig. 2. Sketch of the supervisor.

### III. CASE STUDIES

Several simulations on closed and open border, with different sizes and shapes, have been carried out by using both Matlab [15] and Player/Stage [22] environments. Experiments have been carried out in various indoor environment. Both the simulations and the experiments yield to the same findings. Therefore, due to space constraints, only experimental results are discussed. Video of the experiments are available at [1], [2].

The robots team is composed by three Pioneer 2-DX robots (see Figure 3) 0.44 m long, 0.38 m wide, and 0.22 m tall, having a two-wheel drive along with a passive caster, equipped with two rings of sonars (8 front and 8 rear), a SICK laser range-finder, a pan-tilt-zoom color camera, on-board computation on a PC104 stack and Player control software [22].

Figure 4 shows a portion of the patrolled area and its representation in Player/Stage. The blue line represents the border, the red, green and cyan polygons represent the robots together with their visibility range. In particular the border is a closed line composed by segments joined by arcs; and its overall length is 51 m. The robots know the exact description of the border and they approach it at a speed of 0.5 m/s ( $\lambda_{rf} = 0.3$ ), patrol at a speed of 0.35 m/s ( $\lambda_{cw} = \lambda_{ccw} = 0.35$ ) and escape other teammates at a speed of 0.3 m/s ( $\lambda_{ta} = 0.3$ ). The localization in the environment is achieved by a pre-built map and the localization driver based on an adaptive particle filter ([13]) available in the Player control software. Visibility range and safety area are equal to 2.5 m, the threshold value for the state transitions



Fig. 3. A Pioneer 2-DX robot.

described in Sect. II-D is 0.6 m. Moreover, if a robot is in the patrolling state (see Sect. II-D), every 27 s it can decide to invert its motion direction, according to a random variable.

Figure 4 shows a portion of the patrolled area and its Player/Stage representation;



Fig. 4. On the left a portion of the environment. On the right its representation obtained by Player/Stage software, the path and the three patrolling robots.

Figure 5 shows the distance from the border for the three robots. The distances are within 0.1 m, this value is acceptable for the experimental conditions and requirements. Moreover, peaks are reached during rotation movements due to sensor noise and, above all, to the neglected robot dynamics in the control law.

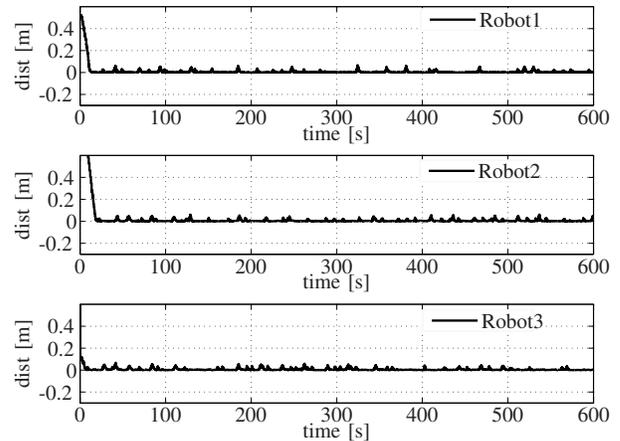


Fig. 5. Distance from the border.

Figure 6 shows the time histories of the robots action selection. In their initial positions, robots 1 and 2 are closer

than the safety range, in this way both of them will activate the *Avoid Teammate Action*. After a few seconds, all robots enter in the patrolling state. While patrolling, robots are usually in the state *KeepGoing* (*CW* or *CCW*). When two robots encounter each other, a sudden transition *Action Keep Going*  $\rightarrow$  *Action Patrol CW* for one robot always comes with a sudden transition to *Action Keep Going*  $\rightarrow$  *Action Patrol CCW* for the other robot; after the interaction they go back to the *Keep Going* state, proceeding in opposite directions. After the interaction they go back to the *Keep Going* state proceeding in opposite directions. A sequence of the movements occurring in this situation is shown in Figure 7.

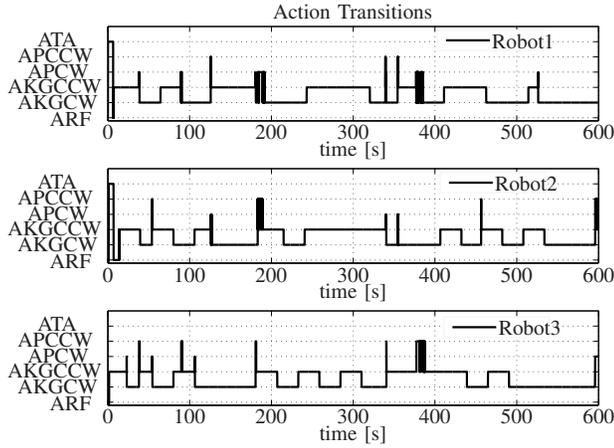


Fig. 6. Time history of actions selection. ARF: Action Reach Frontier. AKGCW: Action Keep-Going CW. AKGCCW: Action Keep-Going CCW. APCW: Actions Patrol CW. APCCW: Action Patrol CCW. ATA: Action Avoid Teammate.

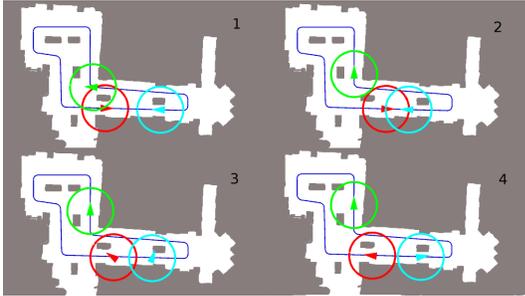


Fig. 7. Three robots team performing the patrol mission. The lower ones (in red and cyan) meet along the path and invert their motion directions. The arrows represent forward motion direction.

Finally, it is useful to show how the distance between two consecutive robots varies over time during the patrolling mission (Figure 8). According to the deterministic optimal policy in [10], in every time instant, the distance between two consecutive robots should be, in our case, one third of the overall border length; also, all robots should move in the same direction. On the contrary, according to [3], robots should move synchronously but in a nondeterministic way, in order to maximize the probability of intercepting an intruder. Both the cases require a centralized supervisor or information exchanging between robots, so it's straightforward to imagine

that the approach proposed can have better performance only increasing the number of vehicles or removing some constraints.

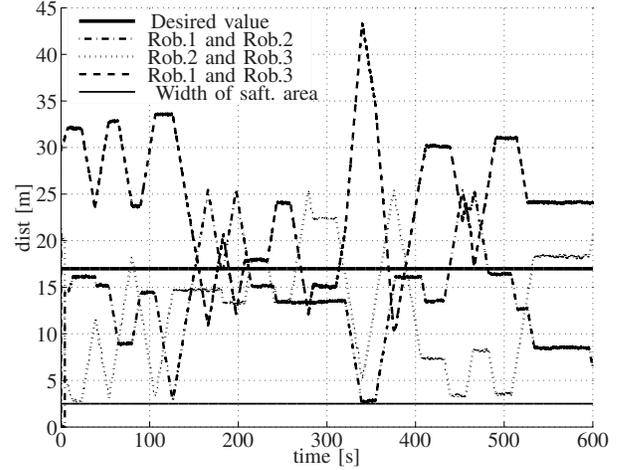


Fig. 8. Distance between two consecutive robots.

## IV. CONCLUSION

A fully decentralized algorithm for multi-robot border patrolling has been developed in the framework of the Null-Space-based Behavioral approach. An higher abstraction layer, the Action, has been introduced, and a Finite State Automata has been designed to properly select among the actions. The algorithm has been extensively tested in simulation and through experimental tests and provided satisfactory results. Preliminary results in demanding environments where the robots fail or where they are commanded to let it pass a *friend* agent are promising for the extension of the current implementation.

## APPENDIX

### A. Elementary Behaviors definition

1) *Reach Frontier*: Given the robot position  $\mathbf{p}_r \in R^2$  and the border  $B$ ,  $\mathbf{p}_B \in R^2$  is the closest point to  $\mathbf{p}_r$  belonging to  $B$ . The behavior reach frontier is simply defined as:

$$\begin{cases} \sigma_{rf} = \|\mathbf{p}_r - \mathbf{p}_B\|, \sigma_{rf,d} = 0, \\ \mathbf{J}_{rf} = \mathbf{r}_{rf}^T, \mathbf{J}_{rf}^\dagger = \mathbf{r}_{rf}, \mathbf{N}_{rf} = \mathbf{I}_2 - \mathbf{r}_{rf}\mathbf{r}_{rf}^T, \\ \mathbf{v}_{rf} = \lambda_{rf}\mathbf{r}_{rf}(-\sigma_{rf}), \end{cases} \quad (6)$$

where  $\mathbf{r}_{rf} = (\mathbf{p}_r - \mathbf{p}_B) / \|\mathbf{p}_r - \mathbf{p}_B\|$ ,  $\mathbf{J}_{rf}$  is the task Jacobian,  $\mathbf{I}_2 \in R^{2 \times 2}$  is the identity matrix,  $\mathbf{N}_{rf}$  is the null-space projection matrix and  $\lambda_{rf}$  is a positive scalar gain. It is worth noticing that computation of  $\mathbf{p}_B$  may require proper approximations [3].

2) *Patrol Frontier Clockwise*: Given the border  $B$  and a point  $\mathbf{p}_B$  belonging to  $B$ ,  $\mathbf{r}_{cw}$  is the unit vector tangent to the border in  $\mathbf{p}_B$  and oriented in the clockwise direction of the border. The behavior is then directly defined as:

$$\begin{cases} \mathbf{v}_{cw} = \lambda_{cw}\mathbf{r}_{cw}, \\ \mathbf{N}_{cw} = \mathbf{I}_2 - \mathbf{r}_{cw}\mathbf{r}_{cw}^T, \end{cases} \quad (7)$$

where  $\mathbf{r}_{cw}$  plays the role of the task Jacobian,  $\mathbf{N}_{cw}$  is the null-space projection matrix and  $\lambda_{cw}$  is a positive scalar gain.

3) *Patrol Frontier Counter-Clockwise*: This case is formally similar to the previous with the obvious difference to properly orient the vector tangent to the border in the counter-clockwise direction.

4) *Teammate Avoidance*: Given the robot position,  $\mathbf{p}_r$ , the obstacle position closest to the robot,  $\mathbf{p}_t$ , and the safety distance,  $d_s$ , the behavior obstacle avoidance is defined as:

$$\begin{cases} \sigma_{ta} = \|\mathbf{p}_r - \mathbf{p}_t\|, \sigma_{ta,d} = d_s, \\ \mathbf{J}_{ta} = \mathbf{r}_{ta}^T, \mathbf{J}_{ta}^\dagger = \mathbf{r}_{ta}, \mathbf{N}_{ta} = \mathbf{I}_2 - \mathbf{r}_{ta}\mathbf{r}_{ta}^T, \\ \mathbf{v}_{ta} = \lambda_{ta}\mathbf{r}_{ta}(d_s - \sigma_{ta}), \end{cases} \quad (8)$$

where  $\mathbf{r}_{ta} = (\mathbf{p}_r - \mathbf{p}_t) / \|\mathbf{p}_r - \mathbf{p}_t\|$ ,  $\mathbf{J}_{ta}$  is the task Jacobian,  $\mathbf{N}_{ta}$  is the null-space projection matrix and  $\lambda_{ta}$  is a positive scalar gain.

### B. Actions Definition

The elementary behaviors defined in Section A are the basis to build the actions defined in Section II-C. In the following, details of the actions defined for the patrolling problem are provided.

1) *Action Reach Frontier (ARF)*: This action allows the robot to reach the border, e.g., when it is far from it. In this case, the definition of the action simply coincides with the elementary behavior *Reach Frontier*:

$$\mathbf{v}_{Arf} = \mathbf{v}_{rf}, \quad (9)$$

2) *Action Patrol Clockwise (APCW)*: This action allows the robot to stay on the border, while covering it in the clockwise direction. This action is obtained by combining the *Reach Frontier* and the *Patrol Frontier Clockwise* behaviors in the NSB sense:

$$\mathbf{v}_{Apcw} = \mathbf{v}_{rf} + \mathbf{N}_{rf}\mathbf{v}_{cw}, \quad (10)$$

i.e., *Reach Frontier* is the higher priority behavior.

3) *Action Patrol Counter-Clockwise (APCCW)*: This case is formally similar to the previous, with the obvious difference to properly consider *Patrol Frontier Counter-Clockwise* behavior.

4) *Action Keep Going (AKG)*: This action allows the robot to stay on the border, while covering it in the clockwise or counter-clockwise direction. This action is obtained by combining the *Reach Frontier* and the *Patrol Frontier Clockwise (Patrol Frontier Counter-Clockwise)* behaviors in the NSB sense:

$$\mathbf{v}_{Akg} = \mathbf{v}_{rf} + \mathbf{N}_{rf}\mathbf{v}_p, \quad (11)$$

where  $\mathbf{v}_p$  is the vector tangent to the border in the closest point belonging to the border. It can be oriented clockwise or counter-clockwise, according to the current state.

5) *Action Teammate Avoidance (ATA)*: When a teammate vehicle enters the safety area of the robot, it needs to avoid the teammate, while trying to stay on the border or to reach it; in this way it can restart the patrol mission once the teammate-vehicle is far enough. This action can be obtained combining the behaviors *Teammate Avoidance* and *Reach Frontier* in the NSB sense:

$$\mathbf{v}_{Ata} = \mathbf{v}_{ta} + \mathbf{N}_{ta}\mathbf{v}_{rf}. \quad (12)$$

Since *Reach Frontier* is the secondary behavior, only its velocity components that do not conflict with the primary behavior will be executed.

### REFERENCES

- [1] <http://www.cs.utk.edu/dilab/movies/Marino-Video1c.avi>.
- [2] <http://www.cs.utk.edu/dilab/movies/Marino-Video2c.avi>.
- [3] N. Agmon, S. Kraus, and G. A. Kaminka. Multi-robot perimeter patrol in adversarial settings. In *Proceedings 2008 IEEE International Conference on Robotics and Automation*, pages 2339–2345, Pasaena, CA, May 2008.
- [4] A. Almeida, G. Ramalho, H. Santana, P. Tedesco, T. Menezes, and V. Corruble. Recent advances on multi-agent patrolling. *Proceedings of the Brazilian Symposium on Artificial Intelligence*, 2004.
- [5] G. Antonelli. Stability analysis for prioritized closed-loop inverse kinematic algorithms for redundant robotic systems. In *Proceedings 2008 IEEE International Conference on Robotics and Automation*, pages 1993–1998, Pasaena, CA, May 2008.
- [6] G. Antonelli, F. Arrichiello, and S. Chiaverini. The Null-Space-based Behavioral control for autonomous robotic systems. *Journal of Intelligent Service Robotics*, 1(1):27–39, online March 2007, printed January 2008.
- [7] R.C. Arkin. Motor schema based mobile robot navigation. *The International Journal of Robotics Research*, 8(4):92–112, 1989.
- [8] T. Balch and R.C. Arkin. Behavior-based formation control for multirobot teams. *IEEE Transactions on Robotics and Automation*, 14(6):926–939, 1998.
- [9] R.A. Brooks. A robust layered control system for a mobile robot. *IEEE Journal of Robotics and Automation*, 2:14–23, 1986.
- [10] Y. Chevaleyre. Theoretical Analysis of the Multi-agent Patrolling Problem. *Proceedings of the IEEE/WIC/ACM International Conference on Intelligent Agent Technology*, pages 302–308, 2004.
- [11] S. Chiaverini. Singularity-robust task-priority redundancy resolution for real-time kinematic control of robot manipulators. *IEEE Transactions on Robotics and Automation*, 13(3):398–410, 1997.
- [12] H. Nam Chu, A. Glad, O. Simonin, F. Sempe, A. Drogoul, and F. Charpillet. Swarm approaches for the patrolling problem, information propagation vs. pheromone evaporation. In *19th IEEE International Conference on Tools with Artificial Intelligence - ICTAI*, 2007.
- [13] D. Fox, W. Burgard, F. Dellaert, and S. Thrun. Monte carlo localization: Efficient position estimation for mobile robots. In *Proc. of the National Conference on Artificial Intelligence*, 1999.
- [14] H. Hertzberg and R. Chatila. *Springer Handbook of Robotics*, chapter AI Reasoning Methods for Robotics, pages 207–223. B. Siciliano, O. Khatib, (Eds.), Springer-Verlag, Heidelberg, D, 2008.
- [15] <http://www.mathworks.com>.
- [16] R.S. Inderieden, H.R. Everett, T.A. Heath-Pastore, and R.P. Smurlo. Overview of the mobile detection assessment and response system. In *DND/CSA Robotics and KBS Workshop*, St. Hubert, Quebec, October 1995.
- [17] A. Kolling and S. Carpin. Multi-robot surveillance: an improved algorithm for the graph-clear problem. In *Proceedings 2008 IEEE International Conference on Robotics and Automation*, pages 2360–2365, Pasaena, CA, May 2008.
- [18] A. Machado, G. Ramalho, J.D. Zucker, and A. Drogoul. Multi-Agent Patrolling: an Empirical Analysis of Alternative Architectures. In *Multi-Agent Based Simulation*, pages 155–170, 2002.
- [19] L.E. Parker. ALLIANCE: an architecture for fault tolerant multi-robot cooperation. *IEEE Transactions on Robotics and Automation*, 14(2):220–240, 1998.
- [20] L.E. Parker. *Springer Handbook of Robotics*, chapter Multiple Mobile Robot Systems, pages 921–941. B. Siciliano, O. Khatib, (Eds.), Springer-Verlag, Heidelberg, D, 2008.
- [21] B. Siciliano. Kinematic control of redundant robot manipulators: A tutorial. *Journal of Intelligent Robotic Systems*, 3:201–212, 1990.
- [22] R. T. Vaughan, B. Gerkey, and A. Howard. The player/stage project: Tools for multi-robot and distributed sensor systems. In *Proceedings of the 11th International Conference on Advanced Robotics, (ICAR)*, pages 317–323, Coimbra, Portugal, June 2003.