17th Mediterranean Conference on Control & Automation
Makedonia Palace, Thessaloniki, Greece
June 24 - 26, 2009

Fuzzy Behavioral Control for Multi-Robot Border Patrol

Alessandro Marino
and Fabrizio Caccavale

DIFA
Università degli Studi della Basilicata
Viale dell'Ateneo Lucano 10,
85100, Potenza, Italy
alessandro.marino@unibas.it
fabrizio.caccavale@unibas.it

Lynne E. Parker

EECS
The University of Tennessee
1122 Volunteer Blvd,
TN 37996-3450, Knoxville, USA
leparker@utk.edu

Gianluca Antonelli

DAEIMI
Università degli Studi di Cassino
Via G. Di Biasio 43,
03043, Cassino (FR), Italy
antonelli@unicas.it

***Abstract*—This paper deals with the problem of multi-robot border patrolling. The patrolling algorithm is designed by resorting to the behavioral control framework and is organized in a hierarchical structure. Several Elementary Behaviors are defined, which are the basis of the concept of Action, placed at a higher level of abstraction with respect to the behaviors. Each Action is obtained by properly combining multiple Elementary Behaviors via the Null-Space-Behavioral control framework. For the sake of robustness, the overall patrolling algorithm is fully decentralized, since explicit communication between robots is not needed. A a Fuzzy Inference System is designed to select the proper Action according to local sensor information only. The algorithm has been validated in simulation as well as experimentally on a setup composed by three Pioneer robots.**

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

I. Introduction

Border patrol is gaining more and more a critical interest in modern society. Patrolling missions 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 application context (e.g., civilian or military), the number of available robots, their equipment and their communication capabilities.

In [16], the main patrolling task issues are investigated and a few multi-agent-based solutions are proposed. Several features, such as agents type, agents communication, coordination scheme, agents perception and decision-making, are evaluated by using different index criteria. The work in [16] has been further extended in [2], pointing out in more detail the features of each multi-agent architecture, as well as the impact of the border geometry on the performance. In [15] and [8] graph-theory is used to find the optimal solution of a mathematical problem expressing a multi-robot surveillance problem. In [1] the authors analyze non-deterministic paths for a group of homogeneous mobile robots patrolling a frontier, under the assumption of an hostile agent (intruder) trying to enter the area, where the latter has full knowledge of the algorithm.

This paper is aimed at developing a fully decentralized patrolling algorithm. 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 [1]. 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 [16].

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 estimates) 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;
- 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 intruders;
- 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; any kind of centralized approach here is not interesting, due to the inherent weakness of grouping all the computational effort in one single machine, even if remote. A key point of this paper is the introduction of the concept of *Action*,

978-1-4244-4685-8/09/$25.00 ©2009 IEEE

obtained by combining in a consistent way a set of elementary behaviors. This is consistent with the adopted bottom-up approach in the action 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 *action 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. In [17] a Finite-State-Automata approach is adopted to build the action selection mechanism. In this paper, instead, a Fuzzy Inference System, is in charge of selecting the proper action. Due to lack of space, theoretical and experimental comparison of the two approaches will be the subject of a different paper.

The approach has been first tested in simulation, by using the Player/Stage environment; then, it has been 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 a high level of autonomy and the implementation of a reasoning method taken from those available in the Artificial Intelligence (AI) literature [13], where, usually, deliberation based on symbols under first order predicate logic (or probability theory) is adopted.

According to the literature [19], the problem approached here is close to what is usually defined as *swarm*, i.e., a large group of 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 [10].

In this paper a behavior-based approach is adopted to solve the border patrol problem. Significant examples of behavioral control developed for multi-robot systems are given in [6] and [18]. Although several paradigms are possible (e.g., [7], [5]), in this work behaviors are handled in the framework of the Null-Space-based Behavior (NSB) approach [4], i.e., a competitive-cooperative approach recalled in Section II-B.

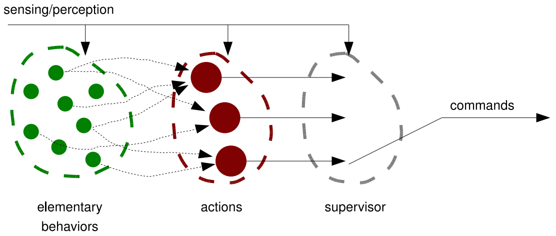

Fig. 1. Overall control schema.

In order to to properly handle concurrent behaviors, a higher level of abstraction with respect to the elementary behaviors, defined as *Action* (Section II-C), is introduced. This choice allows the developers to focus on the action selection mechanism, via a suitably defined *Supervisor*, rather than on the command fusion problem. The supervisor in charge of selecting the proper action is implemented as a Fuzzy Inference System.

Figure 1 reports a sketch of the proposed control structure.

A. Mathematical description of the Elementary Behaviors

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

$$\boldsymbol{\sigma} = \boldsymbol{f}(\boldsymbol{p}), \tag{1}$$

with the corresponding differential relationship

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

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

An effective way to generate motion references for the vehicles, $\boldsymbol{p}_d(t)$, starting from the desired behavioral function, $\boldsymbol{\sigma}_d(t)$, 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., [20] for a tutorial). A typical requirement is to pursue minimum-norm velocity in a closed loop version, leading to

$$\boldsymbol{v}_d = \boldsymbol{J}^{\dagger} \left(\dot{\boldsymbol{\sigma}}_d + \boldsymbol{\Lambda} \widetilde{\boldsymbol{\sigma}} \right) = \boldsymbol{J}^T \left(\boldsymbol{J} \boldsymbol{J}^T \right)^{-1} \left(\dot{\boldsymbol{\sigma}}_d + \boldsymbol{\Lambda} \widetilde{\boldsymbol{\sigma}} \right), \tag{3}$$

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

B. NSB approach to the definition of the Actions

As described earlier elementary behaviors are properly composed in more meaningful *Actions* by adopting a priority-based approach. Elementary behaviors are arranged in such a way that the index i related to the i-th behavior denotes its degree of priority (i.e., behavior 1 has the highest priority). In this paper, no more than 2 behaviors are used to build the actions; in this case, according to [9], solution (3) is modified into

$$\boldsymbol{v}_d = \boldsymbol{v}_1 + \boldsymbol{N}_1 \boldsymbol{v}_2, \tag{4}$$

where v_i is the velocity contribution (in the form (3)) of the i-th behavior,

$$\boldsymbol{N}_1 = \left(\boldsymbol{I}_n - \boldsymbol{J}_1^{\dagger} \boldsymbol{J}_1 \right) \tag{5}$$

is the null space projection matrix of $\boldsymbol{J}_1$ and $\boldsymbol{I}_n$ is the $(n \times n)$ identity matrix .

In this way, lower priority behaviors are executed only in their components not affecting higher priority behaviors; hence, differently from other command fusion approach, the output is predictable. Also, convergence to zero of task errors can be guaranteed for properly defined tasks [3]. On the other hand, a differentiable analytic expression of the defined behaviors is required, so as to compute the required Jacobians.

C. Elementary Behaviors and Actions

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

- Reach Frontier
- Patrol Frontier Clockwise
- Patrol Frontier Counter-Clockwise
- Teammate Avoidance

whose semantics and analytical expressions are given in the Section II-D.

As motivated above, it is appropriate to compose the elementary behaviors into more complex behaviors; the latter are sometimes defined as *Behavior Sets* in the literature. Similar to the concept of behavior set, here a higher abstraction layer is introduced: the *Action*. As shown in Figure 1, an action is given by the proper composition, achieved 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, by combining the elementary behaviors defined above, the following set of actions is obtained:

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

According to the definition in Section 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 Section II-E. It is worth noticing that such actions require that each robot is able to recognize other agents and their nature (friends or teammates) and localize itself in the environment or with respect to the border.

D. Elementary Behaviors definition

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

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

where $\boldsymbol{r}_{rf} = (\boldsymbol{p}_r - \boldsymbol{p}_B)/\|\boldsymbol{p}_r - \boldsymbol{p}_B\|$, $\boldsymbol{J}_{rf}$ is the task Jacobian, $\boldsymbol{N}_{rf}$ is the null-space projection matrix and λ_{rf} is a positive scalar gain. It is worth noticing that computation of $\boldsymbol{p}_B$ may require to resort to proper geometrical approximations [1].

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

$$\begin{cases} \boldsymbol{v}_{cw} = \lambda_{cw}\boldsymbol{r}_{cw}, \\ \boldsymbol{N}_{cw} = \boldsymbol{I}_2 - \boldsymbol{r}_{cw}\boldsymbol{r}_{cw}^{\mathrm{T}}, \end{cases} \tag{7}$$

where $\boldsymbol{r}_{cw}$ plays the role of the task Jacobian, $\boldsymbol{N}_{cw}$ is the null-space projection matrix and λ_{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, $\boldsymbol{p}_r$, the obstacle position closest to the robot, $\boldsymbol{p}_t$, and a suitably defined safety distance, d_s, this behavior is defined as:

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

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

E. Actions definition

The elementary behaviors defined in Section II-D 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*:

$$\boldsymbol{v}_{Arf} = \boldsymbol{v}_{rf}, \tag{9}$$

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:

$$\boldsymbol{v}_{Apcw} = \boldsymbol{v}_{rf} + \boldsymbol{N}_{rf}\boldsymbol{v}_{cw}, \tag{10}$$

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:

$$\boldsymbol{v}_{Akg} = \boldsymbol{v}_{rf} + \boldsymbol{N}_{rf}\boldsymbol{v}_{p}, \tag{11}$$

where $\boldsymbol{v}_p$ 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:

$$\boldsymbol{v}_{Ata} = \boldsymbol{v}_{ta} + \boldsymbol{N}_{ta}\boldsymbol{v}_{rf}. \tag{12}$$

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

F. Fuzzy supervisor

According to the assumptions made in Section I, for the sake of robustness and fault tolerance, each robot decides the next action to be performed based only on its sensing capabilities. Therefore, the set of actions (see Section II-C) defines only the system's skills for reacting to the situations encountered in its environment; i.e., there is the problem of which action needs to be executed next. Although several paradigm might be used, an appealing tool is represented by Fuzzy Logic [11]. A Fuzzy Logic Engine works by encoding an expert's knowledge into a set of linguistic rules which are smoothly interpolated; the result is then defuzzified to provide a crisp actuation output. Each rule is specified as a properly shaped function and assigned to a proper range of input variable. The main advantage presented by this technique is, of course, the possibility that the solution to the problem can be cast in terms that human operators can understand, so that their experience can be incorporated into the design process. This makes it easier to mechanize tasks that are already successfully performed by humans, delete or add new linguistic rules. With regards to the i-th vehicle, the Fuzzy Inference System (FIS), based on the local sensors information, calculates the degree of activation $\alpha_j^i \in [0,1], j = 1, 2 \ldots n$ of each action, based on a classical Mamdami fuzzy type system. As it may happen that sensory data matches with several behavior rules (i.e., the conditional part of the fuzzy rules), more than one action might fire at the same time; hence, an arbiter needs to be build in order to deal with such a situation. However, the FIS is conceived in such a way that only one action may fire at each time step, since the arbiter selects the j-th action characterized by the maximum activation level α_j^i.

The three crisp inputs of the FIS are the distance from the border d_b^i, the distance from the closest teammate d_{ct}^i, and a variable γ^i whose expression is:

$$\gamma^i = \begin{cases} 0 & if\; d_b^c > \theta_{th2} \\ \dfrac{\theta_{th2} - d_b^c}{\theta_{th2} - \theta_{th1}}\lambda^i & if\; \theta_{th1} \le d_b^c \le \theta_{th2} \\ 1 & if\; d_b^c < \theta_{th1} \end{cases} \tag{13}$$

where d_b^c is the distance from the border of the closest teammate, indicated with the subscript c, and θ_{th1} and θ_{th2} are given thresholds. Once a curvilinear abscissa on the border is established, s^i represents the value of curvilinear abscissa of the point on the border closest to the i-th vehicle. The variable λ^i is -1 if $s^c < s^i$, 1 otherwise. Therefore, γ^i is zero if the closest teammate is *far* from the border, negative if it is *close* to the border and $s_c < s^i$, positive if it is *close* to the border and $s_c \geq s^i$. In sum, γ^i represents the minimal information about the closest teammate needed to describe its status and its position with respect to the i-th vehicle. Depending on the values of γ^i and d_{ct}^i, the FIS can decide to avoid the teammate, reverse its motion direction, or keep going on its motion.

As stated before, the outputs of the FIS are the actions degrees of activation. The linguistic variables corresponding to the crisp inputs are

- *DistanceFromBorder* = {low, medium, high}
- *DistanceFromNeighbor* = {low, medium, high}
- *NeighborState* = {PatrollingOnMyLeft, notPatrolling, PatrollingOnMyRight}

while the output linguistic variables are

- *ReachFrontierLevel(RFL)* = {low, medium, high}
- *KeepGoingLevel(KGL)* = {low, medium, high}
- *PatrolClockwiseLevel(PCWL)* = {low, medium, high}
- *PatrolCounterClockwiseLevel(PCCWL)* ={low, medium, high}
- *AvoidTeammateLevel (ATL)* = {low, medium, high}.

It is worth noticing that for an open border also distance from the left and right end of the border need to be considered, for brevity here we refer to a closed border. As membership functions, trapezoids and triangles have been chosen, together with min-max method for and-or and implication-aggregation operations, and centroid as defuzzification method. In Figure 2 are shown the input and output membership functions.

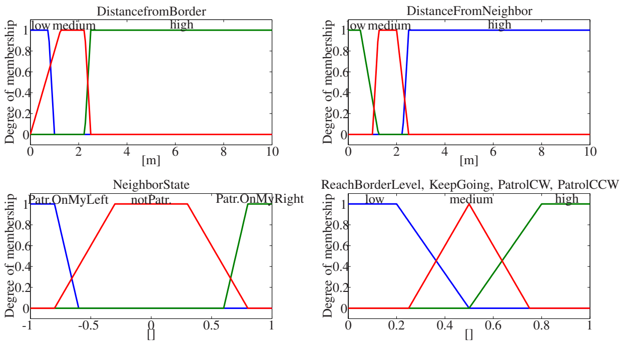

Fig. 2. Membership functions of the input and output linguistic variables.

The number of possible states of the Supervisor is 27; the fuzzy rules depending on the state of *DistanceFromBorder* linguistic variable are reported in Table I.

The main reason of this distinction is that when the *DistanceFromBorder* is *high* the robots should only try to reach the border avoiding other teammates, when the *DistanceFromBorder* is *medium* the robots can start to patrol the border clockwise or counterclockwise but always avoiding other teammates, finally, when the *DistanceFromBorder* is *low*, the robots are performing the patrolling mission and do not allow robots approaching the border to influence their motion; this is helpful in case a large number of vehicles are trying to reach the border; in this case, avoidance of other teammates patrolling the border, is achieved by activating the actions *Patrol CW* and/or *Patrol CCW* depending on the values of the parameter γ and on the distance from the teammate. It can be noticed that in any case, for security

DistanceFromBorder is high			
DistanceFromNeighbor NeighborState	low	medium	high
PatrollingOnMyLeft	ATL high RFL low KGL low PCWL low PCCWL low	ATL high RFL low KGL low PCWL low PCCWL low	ATL low RFL high KGL low PCWL low PCCWL low
notPatrolling	ATL high RFL low KGL low PCWL low PCCWL low	ATL high RFL low KGL low PCWL low PCCWL low	ATL low RFL high KGL low PCWL low PCCWL low
PatrollingOnMyRight	ATL high RFL low KGL low PCWL low PCCWL low	ATL high RFL low KGL low PCWL low PCCWL low	ATL low RFL high KGL low PCWL low PCCWL low

DistanceFromBorder is medium			
DistanceFromNeighbor NeighborState	low	medium	high
PatrollingOnMyLeft	ATL high RFL low KGL low PCWL low PCCWL low	ATL high RFL low KGL low PCWL medium PCCWL low	ATL low RFL low KGL high PCWL low PCCWL low
notPatrolling	ATL high RFL low KGL low PCWL low PCCWL low	ATL medium RFL low KGL high PCWL low PCCWL low	ATL low RFL low KGL high PCWL low PCCWL low
PatrollingOnMyRight	ATL high RFL low KGL low PCWL low PCCWL low	ATL high RFL low KGL low PCWL low PCCWL medium	ATL low RFL high KGL low PCWL low PCCWL low

DistanceFromBorder is low			
DistanceFromNeighbor NeighborState	low	medium	high
PatrollingOnMyLeft	ATL high RFL low KGL low PCWL low PCCWL low	ATL low RFL low KGL low PCWL high PCCWL low	ATL low RFL low KGL high PCWL low PCCWL low
notPatrolling	ATL high RFL low KGL low PCWL low PCCWL low	ATL low RFL low KGL high PCWL low PCCWL low	ATL low RFL low KGL high PCWL low PCCWL low
PatrollingOnMyRight	ATL high RFL low KGL low PCWL low PCCWL low	ATL low RFL low KGL low PCWL low PCCWL high	ATL low RFL low KGL high PCWL low PCCWL low

TABLE I
SUPERVISOR LINGUISTIC RULES

reasons, if the distance between robots is *low* then *Action Avoid Teammate* is activated.

III. EXPERIMENTS

Several simulations on closed and open borders, with different sizes and shapes, have been carried out by using both Matlab [14] and Player/Stage [21] environments. Both the simulations and the experiments yield the same findings. Therefore, due to space constraints, experimental results are discussed in an extensive way, while simulations are omitted. Videos of the experiments are available[1] [2].

The robots team is composed by three Pioneer 2-DX robots 0.44 m long, 0.38 m wide, and 0.22 m tall, having a two-wheel drive along with a passive caster. Sensing capabilities include 16 front and rear sonars, a SICK laser range-finder, encoders and a color camera; on-board computations are performed on a PC104 stack, by using the Player/Stage control software [21].

Figure 3 shows a picture of the indoor environments to be patrolled and the corresponding graphical representation in Player/Stage, where the blue line represents the border, the red, green and cyan polygons represent the robots together with their visibility range. For brevity, in the following we refer only to the closed border, that is composed by segments joined by arcs and whose overall length is about 51 m. Each robot knows 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 ($\lambda_{ta} = 0.3$). To accomplish such a mission, the robots need to localize themselves in the environment, to this aim the robots use a pre-built map and a localization driver based on an adaptive particle filter [12], available within the Player control software. Visibility range is supposed to be equal to 2.5m, and parameters θ_{th1} and θ_{th2} in (13), are 0.7m and 2m, respectively. Moreover, if for some robot *Action Keep Going* is active, every 27 s it can decide to reverse its motion direction, according to a random variable: this should confer some unpredictability with respect to enemies' actions.

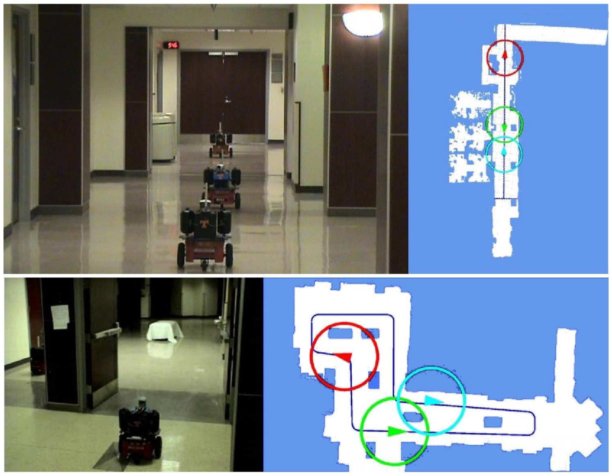

Fig. 3. Experimental scenarios. Left: pictures of the environments. Right: environments representations in the Player/Stage software, patrolled borders, patrolling robots (with their visibility areas).

To demonstrate the robots localization capability with respect to the border and the correctness of actions' definition, Figure 4 shows the time histories of the distances from the border for the three robots. Once the robots reach the border, the distances stay within 0.1 m; this value is acceptable for the experimental conditions and requirements.

Figure 5 shows the time histories of the robots action selection. In their initial positions, robots are *far* from the border, then *Action Reach Frontier* is active. When on the border, usually *Action Keep Going* is active for each robot; in this state, as already stated, transitions form one versus to the other may occur randomly. When two robots encounter each other, a sudden transition *Action Keep Going* → *Action Patrol CW* for one robot always comes with a sudden transition to *Action Keep Going* → *Action Patrol CCW* for the other robot; after the interaction they go back to the *Keep Going* state proceeding in opposite directions. For sake of clarity, a sequence of the movements occurring in this situation is shown in Figure 6. Such motion direction changes appear as spikes in Figure 5, due to the wide time scale of the graphs

[1] http://www.cs.utk.edu/dilab/movies/MED2009_1.avi

[2] http://www.cs.utk.edu/dilab/movies/MED2009_2.avi

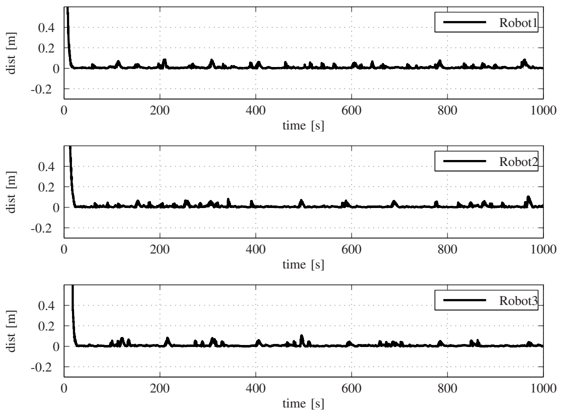

Fig. 4. Distance from the border.

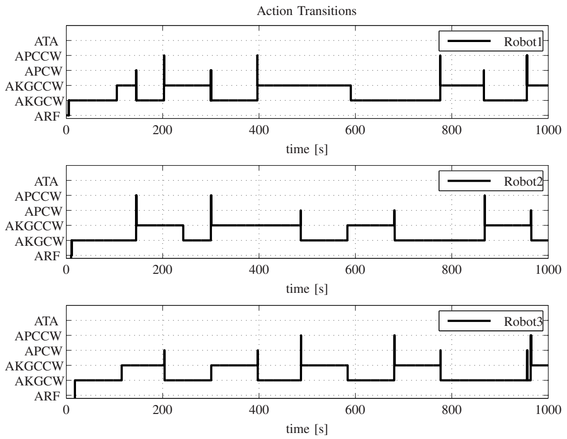

Fig. 5. Time history of action selection. ARF: Action Reach Frontier. AKGCW: Action Keep-Going (CW). AKGCCW: Action Keep-Going (CCW). APCW: Actions Patrol CW. APCCW: Action Patrol CCW. ATA: Action Avoid Teammate. Apparent spikes are due to the wide time scale.

IV. Conclusion

In this paper a fully decentralized algorithm for multi-robot border patrolling has been developed. A Fuzzy Inference System has been designed in order select the proper *Action* to be performed, according to only local sensor information. In more detail, an action introduces a higher abstraction layer, combining elementary behaviors in the framework of the Null-Space-based-Behavioral control. The proposed approach has been extensively tested in simulation and through experimental tests providing satisfactory results.

References

[1] 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, Pasadena, CA, May 2008.

[2] 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.

[3] 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.

[4] 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, January 2008.

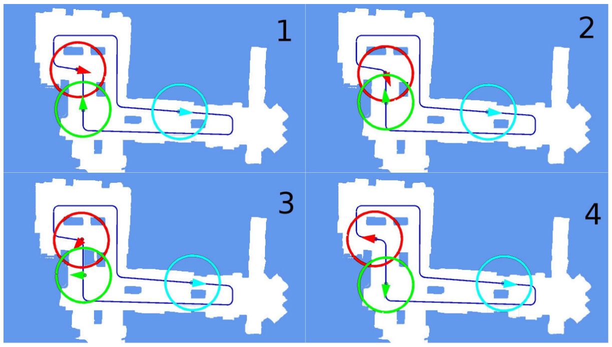

Fig. 6. Three robots team performing the patrol mission. The green and red ones meet along the path and invert their motion directions. The arrows represent forward motion direction.

[5] R.C. Arkin. Motor schema based mobile robot navigation. *The International Journal of Robotics Research*, 8(4):92–112, 1989.

[6] T. Balch and R.C. Arkin. Behavior-based formation control for multirobot teams. *IEEE Transactions on Robotics and Automation*, 14(6):926–939, 1998.

[7] R.A. Brooks. A robust layered control system for a mobile robot. *IEEE Journal of Robotics and Automation*, 2:14–23, 1986.

[8] Y. Chevaleyre. Theoretical Analysis of the Multi-agent Patrolling Problem. *Procedings of the IEEE/WIC/ACM International Conference on Intelligent Agent Technology*, pages 302–308, 2004.

[9] 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.

[10] 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.

[11] D. Driankov, H. Hellendoorn, and M. Reinfrank. *An Introduction to Fuzzy Control*. Springer-Verlag, Berlin, D, 1995.

[12] 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.

[13] J. 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.

[14] http://www.mathworks.com.

[15] 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, Pasadena, CA, May 2008.

[16] 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.

[17] A. Marino, L.E. Parker, G. Antonelli, and F. Caccavale. Behavioral control for multi-robot perimeter patrol: A finite state automata approach. *Proceedings 2009 IEEE International Conference on Robotics and Automation*, 2009. Accepted.

[18] L.E. Parker. ALLIANCE: an architecture for fault tolerant multirobot cooperation. *IEEE Transactions on Robotics and Automation*, 14(2):220–240, 1998.

[19] 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.

[20] B. Siciliano. Kinematic control of redundant robot manipulators: A tutorial. *Journal of Intelligent Robotic Systems*, 3:201–212, 1990.

[21] R. T. Vaughan, B. Gerkey, and A. Howard. The player/stage project: Tools for multi-robot and distributed sensor systems. In *In Proceedings of the 11th International Conference on Advanced Robotics, (ICAR)*, pages 317–323, Coimbra, Portugal, June 2003.