Multi-robot cognitive formations

Miguel Sousa[1], Sérgio Monteiro[1], Toni Machado[1], Wolfram Erlhagen[2] and Estela Bicho[1]

Abstract— **In this paper, we show how a team of autonomous mobile robots, which drive in formation, can be endowed with basic cognitive capabilities. The formation control relies on the *leader-follower* strategy, with three main pair-wise configurations: *column*, *line* and *oblique*. Furthermore, *non-linear attractor dynamics* are used to generate basic robotic behaviors (i.e. follow-the-leader and avoid obstacles). The control architecture of each follower integrates a representation of the leader (*target*) direction, which supports leader detection, selection between multiple leaders (decision) and temporary estimation of leader direction (short-term memory during occlusion and prediction). Formalized as a *dynamic neural field*, this additional layer is smoothly integrated with the motor movement control system. Experiments conducted in our 3D simulation software, as well as results from the implementation in middle size robotic platforms, show the ability for the team to navigate, whilst keeping formation, through unknown and unstructured environments and is robust against ambiguous and temporarily absent sensory information.**

I. Introduction

A team of agents is considered to be moving in formation when the relative distance between each element is kept constant. That is, from an outside observer's point of view, the team of robots must exhibit a fixed pattern while driving along a certain path. Formations can be acknowledged as a special case of swarms, where, in the latter, motion between agents is less structured, and such agents are only required to maintain themselves in the vicinity of one another.

Formations are important because there are many applications. Examples include military applications, where a set of aircraft flies in formation, distributed sensor networks, where a group of agents equipped with sensors with a small detection range can mimic a wider range sensor, when arranged in a rigid formation, or even payload transportation, with a group of robots carrying an object from a place to another.

Keeping a team of agents in formation represents a challenge that has interested researchers, whom have proposed several approaches to address such problem. Solutions may be distinguished, not only by the location of the controller (centralized [4] or decentralized [2]), but also regarding the method used. *Behavior based* [2], *virtual structures* [4], [5] and *leader-follower* [7], [8] are the three main approaches found in the literature.

In [1], a decentralized formation control architecture was presented, which is of leader-follower type and relies on a behavior-based approach for the distributed controller implementation. These controllers were designed and implemented using non-linear dynamical systems. The architecture introduces three key robot configurations (leader-follower pairs): line, column and oblique. The ability to stabilize a desired formation from any initial state, static and dynamic obstacle avoidance, implicit split and join formations (in the presence of obstacles), as well as ordered formation switches, represent the main features of the solution. However, cases of ambiguous or absent sensory information have not been considered. This way, the work presented in this paper aims to extend the architecture in order to endow the robots with some cognitive capabilities that allow to address this problem. A dynamic neural field for each robot, which is used to represent its leader direction, endows the robot with the capacity do *detect, decide, memorize, forget and predict* its leader direction. The ability to track a leader, select between multiple leaders, discard false leaders, along with robustness against temporary leader occlusion and estimation of leader direction, are described and validated by means of computer simulations and real robot experiments.

The remainder of the paper is organized as follows: section II presents some related work. Section III introduces the attractor dynamics approach to behavior generation as the base for the formation control architecture. Section IV briefly describes the concept of dynamic neural fields and its main properties. In Section V we describe the test beds and present some results coming from the extension of the control architecture. Finally, Section VI concludes the paper and Section VII gives some pointers for future development.

II. Related work

Among behavior-based solutions, the work presented in [2] considers four different reactive motor schemas, move-to-goal, avoid-static-obstacle, avoid robot and maintain formation, where each behavior generates a vector (direction and magnitude), which corresponds to the current sensory input, contributing to the total behavioral response. [3] extend the previous work by presenting a new form of potential functions, called social potentials. Their inspiration derived from the way molecules are drawn to their places when forming crystal-like structures. This way, each robot is not strictly assigned to a position in the formation. Instead, it is "attracted" to the closest position available.

Toni Machado would like to thank the Portuguese Science and Technology Foundation for providing his Ph.D. scholarship (ref. SFRH/BD/38885/2007). This work is also supported by FEDER Funds thru Competitivity Factors Operational Program - COMPETE and National Funds by FCT Portuguese Science and Technology Foundation under the Project FCOMP-01-0124-FEDER-022674.

[1] Dept of Industrial Electronics, University of Minho, 4800-058 Guimarães, Portugal Email: {msousa, sergio, tmachado, estela.bicho}@dei.uminho.pt

[2] Department of Mathematics for Science and Technology, University of Minho, 4800-058 Guimarães, Portugal Email: wolfram.erlhagen@mct.uminho.pt

An early work using the virtual structures approach to solve the formation control problem is the work by [4]. Here, they define the virtual structure concept as a rigid body, where the relative positions of each robot are kept constant. This way, each robot's position is fixed with respect to a reference frame, which is moved along the plane. Also taking advantage of the virtual structure concept is the work done by [5]. They apply it to the formation control of spacecraft and extend the approach with the introduction of formation feedback. This overcomes the problem where the virtual structure can evolve too fast and single elements are not able to track it, which may result in formation loss. The same authors extend the method with a decentralized implementation in [6].

One of the first to use the concept of leader-follower in formation control problems was the work by [7]. The author introduces several navigation strategies (nearest-neighbor tracking, multi-neighbor tracking, inertially referenced movements and a mixed approach between the first and last) which can be used to solve the formation control problem. [8] and [9] present two main strategies for leader follower based formation control: $l-\psi$ (distance and angle to one leader) and $l-l$ (distance to two leaders). Here, they define a unique lead robot for the whole formation, whereas many other leaders can be used to guide followers. A similar approach is used in [10], but augmented with a discrete event system to control the execution of primitive behaviors. In [11] the followers incorporates the leader dynamics and estimates the linear and angular velocities of the leader. Yet, as in previous works, it is not clear what are the consequences of missing sensory information.

[12] also presents the leader-follower strategy, along with the behavior-based approach. Much like in our work, each robot has the domain of its heading direction modeled as a dynamic neural field. However, their global goal consists of generating the basic behaviors (*target acquisition*, *obstacle avoidance* and *formation keeping*) through stimuli to the field. Such stimuli may be the global target to reach or a leader target, which are excitatory, or obstacles, which are inhibitory. Afterwards, the decision for the robot to move in certain direction comes from the location of the activation peak on the field. Conversely, our approach aims to generate more abstract forms of behavior (i.e. *selection*, *memory*, *prediction*) related to the representation of target information, through the exploitation of different properties of the *dynamic neural fields*, with the basic behaviors emerging from the *attractor dynamics* approach, after an estimation of target direction is made.

Several solutions for the formation control problem can be found in the literature (refer to [13] and [14] for a more extensive review). The control architecture developed in previous work [1] follows a decentralized implementation, without the need for explicit communication and knowledge about the absolute position and orientation of the agents in the world, thus easily employable in unknown and unstructured, real world scenarios. However, by simply using the robot's vision system to recognize the target (*follower* tracking the *leader*), such unpredictable scenarios bring about certain problems. For instance, false leaders (same coloured blobs) may appear in the followers field of view, which may either be temporary, thus needing to be filtered out, or persistent, implying the need for the robot to keep tracking its leader, albeit having multivalued information about the direction of said leader. Furthermore, cases of leader occlusion (i.e. when an obstacle lies between the follower and its leader, or when the team turns around a corner) are of importance. This way, our work proposes an approach to address these cases of ambiguous and intermittent sensory information.

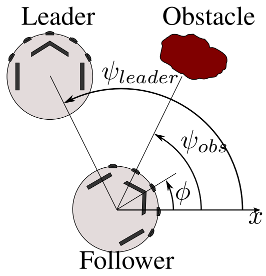


Fig. 1. Target aquisition and obstacle avoidance tasks constraints.

III. Attractor dynamics approach to formation control

We start with a brief description of the principles of the *attractor dynamics approach* to behavior generation. How this theoretical framework was applied to the case of configurations with two robots, follows, with the generalization to N agents being presented immediately after.

The robots behavior is modeled by using the *attractor dynamics approach*, as originally proposed by Schoner [15]. More specifically, behavior is generated by providing values to two control variables: the robot's heading direction, ϕ, and path velocity, v. These values are generated by dynamical systems. To design our control system, we are interested in specific solutions of these dynamical systems, entitled *attractor fixed points* and *repeller fixed points* solutions. In particular, we model the behavior by erecting attractors in desired states of the system and by erecting repellers in undesired states. In case of the heading direction dynamics, we define the directions in which leader lies as attractors and the direction in which obstacles lie as repellers (see figure 1). Desired velocity is defined as an attractor for the path velocity dynamics. When the robot moves, its perceived sensorial information changes. As a consequence, the resultant attractors move. In order to guarantee that the system is stable, one must limit the rate of shift of these attractors such that the system is able to track them. This is accomplished by limiting the robot's path velocity and by design parameters.

Three *leader-follower* configurations are defined: *column*, *line* and *oblique*. For the case of a *column* configuration (follower drives behind the leader), the equations that govern the robots' behavior are as follows:

1) Heading direction:

$$f_{col}(\phi) = -\lambda_{col}\sin(\phi - \psi_{leader}) \quad (1)$$

$$f_{obs,i}(\phi) = \lambda_{obs,i}(\phi - \psi_{obs,i})exp\left(-\frac{(\phi - \psi_{obs,i})^2}{2{\sigma_{obs,i}}^2}\right) \quad (2)$$

$$F_{obs}(\phi) = \sum_{i=1}^{N_{sectors}} f_{obs,i}(\phi) \quad (3)$$

$$\dot{\phi}(t) = f_{col}(\phi) + F_{obs}(\phi) + f_{stoch} \quad (4)$$

2) Path velocity:

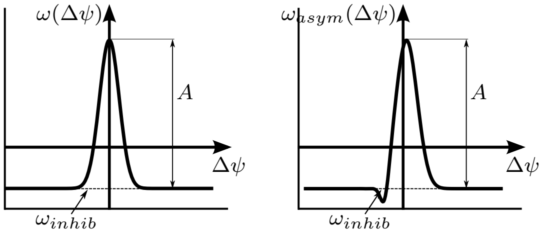
$$\dot{v}(t) = -\alpha_{col}(v - v_{col}) - \alpha_{obs}(v - v_{obs}) \quad (5)$$

Equation (1), represents the target aquisition task contribution to the heading direction dynamics, which erects an attractive force-let at ψ_{leader}, the direction of the target (direction in which the leader lies), where λ_{col} is the strength of attraction. Conversely, (2) accounts for the obstacle avoidance task, which erects a repulsive force-let at the direction in which an obstacle is sensed by sensor i, ψ_{obs_i}, with λ_{obs_i} representing the strength of repulsion, related to the distance to the obstacle. The final non-linear dynamics, in (4), results from the integration of the target acquisition and obstacle avoidance behaviors (the contribution of *N* obstacle sensors is defined in (3)). Equation (5) encodes the velocity dynamics, by simply defining a linear dynamical system, which erects an attractor at the desired velocity, either v_{col} or v_{obs}, depending on the direction in which the obstacles are detected (the robot decreases its velocity when obstacles are on its path or when it is too close to the leader). Similar controllers are employed for the case of *oblique* and *line* configurations, but with the desired bearing to keep to the leader affecting the dynamics. The overall formation is achieved by definig sets of leader–follower pairs using the above mentioned configurations (see [1] for more details). To control the robot, angular velocity, ω, is obtained directly from (4), and path velocity, v, results from integrating (5) with the forward Euler method. For further detail, please refer to [15], [16], [17].

IV. Dynamic neural fields for leader representation

In the architecture described above, simple robotic behaviors were generated directly in response to sensory information. However, when such sensory input is ambiguous or intermittent, more abstract forms of behavior are required. For instance, *detection*, *decision*, *memory*, *forgetting* and *prediction* are examples of such behaviors. These processes require representations of information. In the attractor dynamics approach, each variable has a unique value at all times and changes continuously over time, therefore being unable to represent graded amounts of information. For instance, absence of information about the target direction, or multiple targets, can not be encoded.

The capabilities stated above, and associated representation of information, are achieved by means of *dynamic neural fields* (see [18], [20]). Presented by Amari [19], the neural field theory approach, in the form of a *dynamic neural field* equation with lateral inhibition, models the cortical activation occurring in neural tissues. Amari proposed that a high number of interacting neurons, creating a homogenous network, can be approximated by a continous neural field of activation. Such equation features recurrent interactions between neurons, allowing for an amplification of stimuli received and self-stabilization of activation states, and reads:



(a) Gaussian (b) Asymetric

Fig. 2. Interaction kernels.

$$\begin{aligned}\tau\dot{u}(\psi,t) &= -u(\psi,t) + S(\psi,t) + h \\ &+ \int_0^{2\pi} \omega(\psi-\psi')f(u(\psi',t))\mathrm{d}\psi'\end{aligned} \quad (6)$$

in which τ represents the time scale of the field, $u(\psi,t)$ the field's activation, h the resting level and $S(\psi,t)$ the input. With only these terms, the field simply relaxes to the input pattern plus the resting level ($S(\psi,t)+h$). The integral term describes the lateral interaction between neurons on the field, which can be excitatory or inhibitory, and is given by an interaction kernel $\omega(\psi-\psi')$ affected by a firing rate function, $f(u(\psi',t))$. A Gaussian shape (7) and step functions (8) are possible choices for the interaction kernel and firing rate, respectively.

$$\omega(\psi-\psi') = Aexp\left(-\frac{(\psi-\psi')^2}{2\sigma^2}\right) - \omega_{inhib} \quad (7)$$

$$f(u) = \begin{cases} 0 & \text{for } u \leq u_0 \\ 1 & \text{for } u > u_0 \end{cases} \quad (8)$$

This type of interaction kernel (Fig. 2a) dictates that sites closer to each other (within the cooperative area) are excited, while sites further away are inhibited. The firing rate function grants that only positively activated neurons ($u(\psi,t) > 0$) contribute to interaction.

Several properties arise from the dynamic field, which can be used to generate more abstract behaviors. (i) *Detection*: with the field relaxed to the resting level, $h < 0$, only an input sufficiently strong to make the activation surpass the treshold, $u(\psi,t) > u_0$, thus making interaction take place, is acknowledge as valid. (ii) *Decision*: when multimodal inputs are presented, that is, input patterns with multiple peaks, the distance between the location of these peaks is key to the decision process. On one hand, if two peaks are within the cooperative range of the interaction kernel, an excitatory interaction occurs, where the localized activation results from

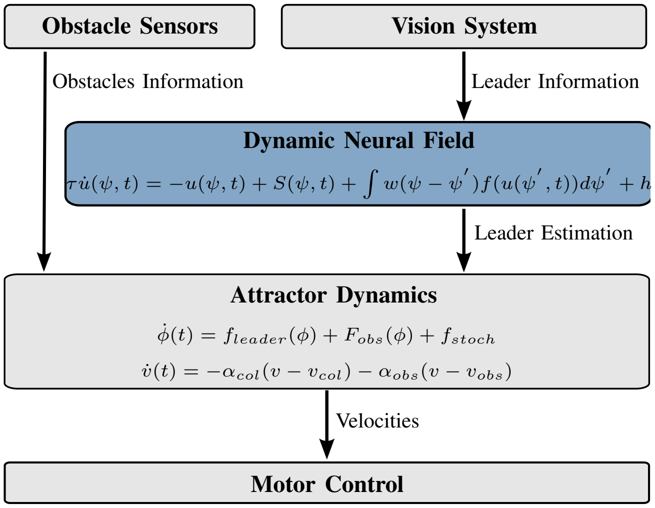

Obstacle Sensors
Vision System
Obstacles Information
Leader Information
Dynamic Neural Field
$\tau\dot{u}(\psi,t) = -u(\psi,t) + S(\psi,t) + \int w(\psi-\psi')f(u(\psi',t))d\psi' + h$
Leader Estimation
Attractor Dynamics
$\dot{\phi}(t) = f_{leader}(\phi) + F_{obs}(\phi) + f_{stoch}$
$\dot{v}(t) = -\alpha_{col}(v-v_{col}) - \alpha_{obs}(v-v_{obs})$
Velocities
Motor Control

Fig. 3. Extension of the control architecture with the addition of a target representation layer (blue shaded).

the interpolation of both peaks. On the other hand, if the peaks are separated by more than this cooperative range, a decision between which one to select is affected by the strength of each, with stronger activation peaks being more likely to win the competition. Furthermore, selection can be biased by a pre-activation of a place on the field, for cases of inputs with the same strength. (iii) *Memory*: for certain values of the interaction kernel and resting level (see [18]), a peak of activation has the ability to sustain itself through self-excitation, even after the input has been removed. When a new input, of sufficient strength, is received on a different location, the peak is shifted to match this new location, or simply replaced by a peak of activation over the new location, thus updating the stored information. (iv) *Forgetting*: by defining the global inhibition on the field, h, as a dynamical system (9), the memorized peak of information can be deleted after a particular time span.

$$\dot{h} = -r_{h,min}c_h(h-h_{min}) - r_{h,max}(1-c_h)(h-h_{max}) \quad (9)$$

When no input is present ($c_h = 1$), the system lowers the resting level at a rate $r_{h,min}$. Stored information is effectively discarded when the localized self-excitation of the peak is no longer able to overcome this global inhibition. The restoring process ($c_h = 0$), in which the field is capable of memorizing new information, takes place by increasing the resting level at a rate $r_{h,max}$, which is faster than the forgetting process, so the filed is quickly capable of storing new information. (v) *Prediction*: when a shifting input, which creates a shifting peak of activation, is removed, the memorized peak can continue to shift at the same rate. This can be accomplished by changing the interaction kernel of the field to an asymmetric one (10) (see [20]), which biases the interaction to one direction over the other (Fig. 2b), creating a traveling wave solution, whose velocity can be adjusted.

$$\omega_{asym}(\psi-\psi') = \omega(\psi-\psi') + \eta(t)\omega'(\psi-\psi') \quad (10)$$

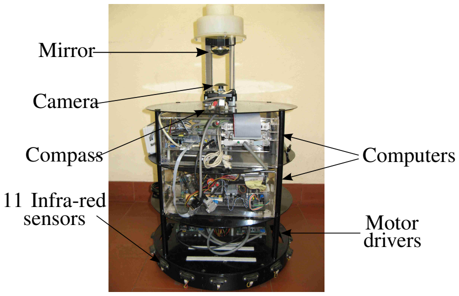


Fig. 4. The robotic plattform.

where $\omega'(\psi)$ is the kernels derivative and $\eta(t)$ controls the shift rate.

A dynamic neural field is used to represent the direction in which a leader lies in the follower's reference frame (see Fig. 3). We discretize the field with a sampling distance of 1 deg., thus $0 < \psi \leq 360$. The input to the field results from the vision system, which returns the relative bearing of the leader. This input can have noise or even be absent. The estimated target direction, i.e. the ψ_{leader} to be used in the motor movement dynamics control, is obtained from the center of mass of the activation peak on the field (see [18] for details). The main advantage of adding the target representation layer, in the form of a dynamic neural field, is that it allows for a continuous valid representation of the target direction even if it not momentarily there. In our previous architecture, without this layer, the absence of target detection caused that the last known direction to be used, even if this direction was changing. Now it is possible to take into account all this aspects, and even control the rate of change of the represented target. Furthermore, one can also control for how long the memory persists. Of course memorization time is a trade off: when using the memorized representation, one assumes that the leader is evolving similarly to the moments prior to being absent. This is not always true so the memorized information should fade away.

V. Results

In this section we present some results, both in simulation and real implementation, of the addition of this higher level layer to the control architecture. Videos of the experiments can be found in our server[1]. We focus on cases of leader occlusion, in which memory and forgetting processes, as well as prediction, are of importance. For the simulations we use *CoopDynSim* [21], our physics enabled 3D simulator, which mimics the middle-size robotic platforms available in our laboratory (Fig. 4), used for the real implementation. The control application is implemented in MATLAB and the same code (and user interface) used for simulation is seamlessly transferred to the robots in the real experiments.

A. Simulation

Fig. 5 shows a scenario where two robots should navigate in a column formation, starting in a corridor that leads into

[1] https://marl.dei.uminho.pt/Public/CognitiveFormations/

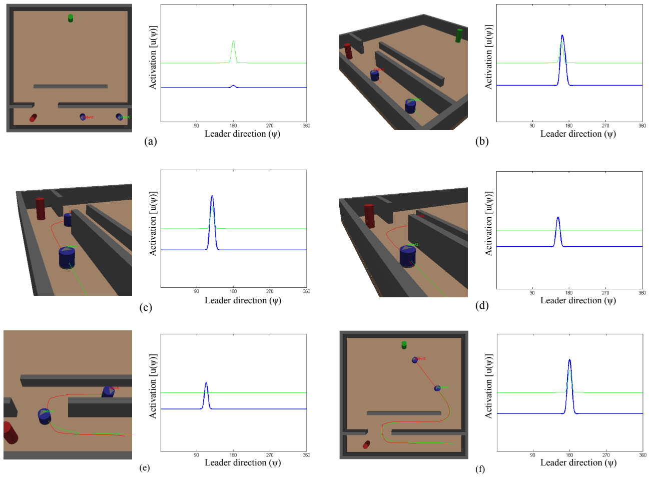


Fig. 5. Simulation of a room entrance scenario. Each panel represents an instant of the simulation, with the correspondent snapshot on the left and the relative leader direction representation (by the follower) on the right. This experiment shows the follower's ability to track its leader, recurring to the memorized information. Despite the leader entering the room (c), thus disappearing from the followers line of sight (no sensory input), a self-stabilized peak of activation in the field dynamics (d) encodes the leader's previous direction, making the follower turn towards the room entrance (e) in order to reacquire formation.

an entrance to a room. The left images show the snapshots of the simulation, while the right ones represent the state of the dynamic neural field, with the (dark) blue line representing the field activation and the (light) green line the input stimulus. In this scenario, the goal of the lead robot is to move towards the red target and, when it reaches its location, change to the green one (these targets are always visible to the leader), while the follower simple needs to drive in a column configuration (panel (b)). Panels (c) encodes the instant right before the leader disappears from the line of sight of the follower. (d) shows the follower driving in the correct direction towards the room entrance, with the memorized target direction information (however, nothing relevant is shown here, since the same would happen by simply maintaining the robots trajectory and make it move forward). The key moment in the simulation lies in panel (d). Here, the robot still drives into the room (caused by the shift in the memorized direction), since it "saw the leader turn in that direction" before disappearing. One can see by the follower's trajectory that the prediction of its leader's direction is not entirely correct, since the latter found a wall along its path and abruptly changed its direction. Nonetheless, a decision to enter the room was still taken, instead of just driving forward or simply stopping, when no sensory input was available. In (e) the follower reacquires sensory information, updating the target representation and correcting its trajectory in order to match the desired configuration, with the team effectively reaching the target location in (f).

To demonstrate the scalability of the approach we also present in fig. 6 a brief simulation with a team of five robots navigating in a triangle formation, in an environment with obstacles that cause some occlusions.

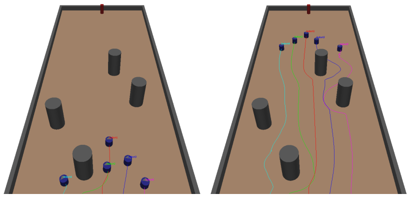


Fig. 6. Simulation of a team with 5 robots in a triangle. On the left snapshot, the leftmost robot does not detect its leader due to the obstacles.

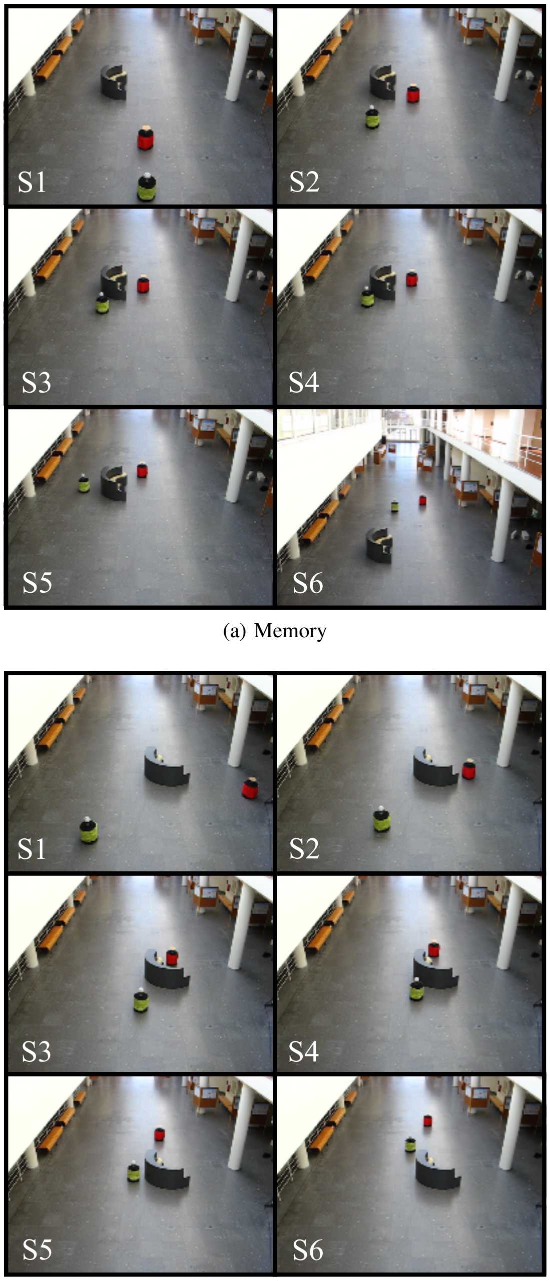
B. Real implementation

Moving to the implementation on the real platforms, two scenarios will be presented. Only two robots are used, because are the only available at the time. Since this is a leader–follower architecture, and we are adding just an extra layer to our previous architecture, we argue that the results in terms of scalability still hold. The first (Fig. 7a) shows the robots stabilizing, from an initial column configuration (S1), an oblique formation (S2). In S3, the follower loses sight of its leader. At the same time, the obstacle makes the robot turn left (S4), in order to avoid it. However, the ability to stabilize the same formation pattern (instead of just driving forward), even without current sensory information about the leader, is shown in S5, with the robots reaching the destination in S6. This experiment depicts the case where the follower estimates the direction of its leader solely from the memorized information on the field dynamics.

In the second scenario (Fig. 7b), the robots should stabilize a column formation. Starting from the initial positions depicted in S1, one can see the follower driving towards the leader (S2), when it becomes occluded by the obstacle in the environment. However, instead of simply going to the last memorized direction of its leader, one can see the follower estimating the future direction of such leader, adjusting its path accordingly (S3 and S4). In S5 and S6, and after avoiding the obstacle, the leader becomes visible again and the formation is stabilized to the desired distance. This scenario clearly depicts the future leader relative direction estimation (prediction) capability of the robots. Instead of navigating through the right hand side of the bench, the follower predicted the direction of its leader and, as a result, adjusted its trajectory and (in this scenario) took the shortest path in order to stabilize the formation.

VI. Conclusion

In this work, we extended the architecture developed in [1] with target representation applied to the formation control, by using (for each robot) a dynamic neural field, thus endowing the autonomous robot with some basic cognitive capabilities. Concretely, *target (leader) detection and selection*, *short-term memory* and *prediction* emerge. This representation allows for an estimation of leader information, even in the absence of an input, somehow drifting away from the input-to-output reactive behavior (always dependent on the current sensory input). Results show the ability of the follower to temporarily estimate its leader direction, when the latter becomes occluded, which can represent an improvement over



(a) Memory

(b) Prediction

Fig. 7. Formation stabilization of the middle size platforms in the presence of an occlusion. The leader is the dark red robot and the follower is the light green robot.

the case of the follower simply stopping or wandering when its leader disappears. Albeit only presenting results for two robots, we argue that the architecture naturally grows to *N* agents.

VII. Future Work

As future work, the problem of leader distance estimation, when it is occluded, must be addressed. In the work presented we only aim to solve the problem of temporary leader occlusion regarding leader direction, assuming the distance is the desired one. Nevertheless, a complete representation of the target could lead to better results. Optimization in the robot place assignment in the formation, from initial positions, as well as leader-follower pairs' designation, in order to minimize the overall energy consumption of the team, poses an open question. An automatic generation of the formation pattern, depending on the mission and environmental constraints, represents yet another challenge. Furthermore, a learning mechanism, to allow the robot to autonomously tune its design parameters, is desirable.

References

[1] Monteiro, S and Bicho, E. "Attractor dynamics approach to formation control: theory and application," *Autonomous Robots*, *29*, pp. 331-355, 2010

[2] Balch, T. and Arkin, R. "Behavior-based formation control for multi-robot teams," *IEEE Transactions on Robotics and Automation*, *14(6)*, pp 926-939, 1998

[3] Balch, T. and Hybinette, M. "Social potentials for scalable multi robot formations," *Proceedings of the IEEE International Conference on Robotics and Automation*, pp 73-80, 2000

[4] Lewis, M. and Tan, K. "High precision formation control of mobile robots using virtual structures," *Autonomous Robots*, *4*, pp. 387-403, 1997

[5] Ren, W. and Beard, R. "Virtual structure based spacecraft formation control with formation feedback," *AIAA Guidance and Control Conference*, pp. 2002-4963, 2002

[6] Ren, W. and Beard, R. "A decentralized scheme for spacecraft formation flying via the virtual structure approach," *Proceedings of the 2003 American Control Conference*, 1746-1751, 2003

[7] Wang, P. "Navigation strategies for multiple autonomous mobile robots moving in formation," *Journal of Robotic Systems*, *8*, pp. 177-195, 1991

[8] Desai, J., Ostrowsky, J. and Kumar, V. "Modeling and control of formations of nonholonomic mobile robots," *IEEE Transactions on Robotics and Automation*, *17*, pp. 905-908

[9] Desai, J. "A graph theoretic approach for modeling mobile robot team formations," *Journal of Robotic Systems*, *19*, pp. 511-525, 2002

[10] Gamage, G., Mann, G., Gosine, R. "Discrete event systems based formation control framework to coordinate multiple nonholonomic mobile robots," *Proceedings of the 2009 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 4831-4836, October 11-15, St. Louis, USA, 2009

[11] Dierks, T., Jagannathan, S. "Neural Network Output Feedback Control of Robot Formations," *IEEE Trans. on Systems, Man, and Cybernetics–Part B: Cybernetics*, 40(2), pp. 383-399, April 2010

[12] Oubbati, M. and Palm, G. "Neural fields for controlling formation of multiple robots," *Proceedings of the IEEE International Symposium on Computational Intelligence in Robotics and Automation*, pp. 90-94, 2007

[13] Chen, Y. and Wang, Z. "Formation control: a review and a new consideration," *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 3664-3669, 2005

[14] Gazi, V. and Fidan, B. "Coordination and control of multi-agent dynamic systems: models and approaches," *Proceedings of the 2nd International Conference on Swarm Robotics*, pp. 71-102, 2007

[15] Schöner, G., Dose, M. and Engels, C. "Dynamics of behavior: theory and applications for autonomous robot architectures," *Robotics and Autonomous Systems*, *16*, pp. 213-245, 1995

[16] Bicho, E. and Schöner, G. "The dynamic approach to autonomous robotics demonstrated on a low-level vehicle platform," *Robotics and Autonomous Systems*, *21*, pp. 23-35, 1997

[17] Bicho, E. and Monteiro, S. "Formation control for multiple mobile robots: a non-linear attractor dynamics approach," *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, *2*, pp. 2016-2022, 2003

[18] Bicho, E., Mallet, P. and Schöner, G. "Target representation on an autonomous vehicle with low-level sensors," *International Journal of Robotics Research*, *19*, pp. 424-447, 2000

[19] Amari, S. "Dynamics of pattern formation in lateral inhibition type neural fields," *Biological Cybernetics*, *27*, pp. 77-87, 1977

[20] Erlhagen, W. and Bicho, E. "The dynamic neural field approach to cognitive robotics," *Journal of Neural Engineering*, *3*, pp. 36-54, 2006

[21] Machado, T., Sousa, M., Monteiro, S. and Bicho, E. "CoopDynSim: a 3D robotic simulator," *12th International Conference on Autonomous Robot Systems and Competitions*, manuscript accepted, 2012