

Autonomous Initialization of Robot Formations

Mathieu Lemay, François Michaud, Dominic Létourneau and Jean-Marc Valin
LABORIUS – Research Laboratory on Mobile Robotics and Intelligent Systems
Department of Electrical Engineering and Computer Engineering
Université de Sherbrooke, Sherbrooke (Québec) CANADA, J1K 2R1
{Mathieu.Lemay, Francois.Michaud}@USherbrooke.ca

Abstract—Real life deployment of robot formation cannot assume that robots are going to be correctly positioned to move in a particular configuration. To do so, we propose an approach that allows the group to determine autonomously the most appropriate assignment of positions in the formation. Our approach is distributed and uses directional visual perception to localize robots. Inter-robot communication allows them to share information on which robots are nearby, so that each can evaluate its ability to be the conductor of the group and assign formation positions to the other robots by minimizing repositioning. The assignment search is done using a distributed bounded depth-first with pruning search. The robot with the best score is selected as the conductor, and the other robots received from the conductor their assignment in the formation. Validation of our work is done in simulation and with Pioneer 2 robots.

I. INTRODUCTION

Formations are defined as groups of mobile robots establishing and maintaining some predetermined geometrical shape by controlling the positions and orientations of each individual robots relative to the group, while allowing the group to move as a whole. Such capabilities require to solve the following issues:

- 1) How to initialize and establish a formation ?
- 2) How to maintain formation shape while moving ?
- 3) How to avoid obstacles ?
- 4) How to change the shape of the formation ?
- 5) Is it possible to realize any types of formation ?

In this kind of problems, the intelligence of the robots is found in their capacity to answer these questions in an autonomous way. In this paper, we present a solution that addresses mostly the first four issues, with a particular focus on the first one. Compared to our previous work [1], this paper presents an improved method for position assignment in a formation, along with more detailed results. After giving a description of our formation control approach in Section II, our robots position assignment method is explained in Section III. The experimental results obtained in simulation and using Pioneer 2 robots are illustrated in Section IV, followed with an overview of related work in Section V. Finally, Section VI outlines future work and concludes this paper.

II. ROBOT FORMATION CONTROL ARCHITECTURE

Before presenting our approach, we find it necessary to define some specific terms and specifications of our work. A *Follower* is defined as a robot set to follow another robot (called a *Leader*) at a specified distance and angle (according

to its position assigned in the formation). Note that a *Follower* can also be the *Leader* of another robot. The robot leading the formation is called the *Conductor*. Also, we consider the heading as the orientation of the robot and the angle as angle of one robot relative to another. The robots used in the experiments have limited visibility of the environment: perception of the other robots is done using only color cylinders placed on their back. Knowing the width l_p of cylinder which the robot perceives by its camera, the robots derive their relative distances d between the cylinder and the camera using the geometrical model given by 1.

$$d = \frac{a}{l_p} + b \quad (1)$$

where a and b are constants determined experimentally (set to 21614 and -82 respectively, to obtain a distance in millimeter).

The precision on the approximated distance depends on the capacity of the system to recognize specific colors of the cylinders, which is influenced by illumination conditions. For an optimal approximation, good uniform lighting is necessary. Practically, when the cylinder is lit from the side, its color becomes not uniform, making only part of the width of the cylinders visible to robots. The approximated distance then becomes larger than the real distance. The relative angle between robots is determined according to the horizontal position of the center of the blob, the position of the camera and the position of the robot. Note that the camera angle is not fixed on the robot. The pan of the camera can be adjusted so that the colored cylinder of the *Leader* stays in the field of view. This provides flexibility to the positions of the robots in the formation, especially when avoiding obstacles.

Our approach follows the guidelines of a hybrid control architecture that combines local level control of a robot with global level deliberation over the overall states of the group [1]. Only two levels of the architecture are used, i.e, the behavioral level (bottom part) and the recommendation level (highest part). Figure 1 represents the architecture implemented.

A. Behavior Level

The behavior level is made of behavior-producing modules allowing a robot to react in particular ways to situations encountered in the environment. All modules run in parallel and their resulting commands are prioritized to generate the control actions of the robot. The behavioral modules used in our approach are : *Safe-Velocity*, to make the robot move

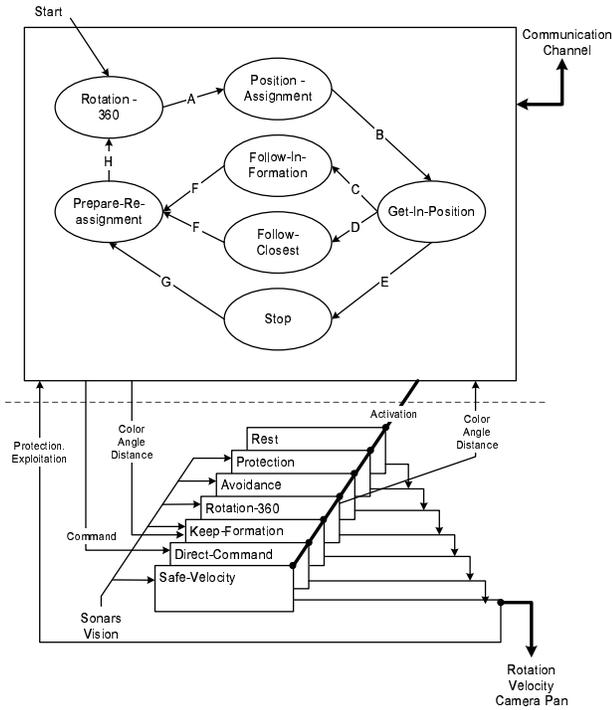


Fig. 1. Control architecture for our robot formation approach.

forward without colliding with obstacles; *Direct-Command*, to change the position of the robot according to specific commands; *Keep-Formation*, to allow a robot to follow its assigned leader; *Rotation-360*, to reinitialize the position of the camera toward the front, and then make the robot rotate slowly in order to localize other robots in its vicinity; *Avoidance*, to move the robot away from obstacles detected based on front sonar readings; *Protection*, to examine the presence of an object very close to the robot; *Rest*, to make the robot stop moving and wait.

Using these behavior-producing modules, a robot is capable of following another robot while avoiding obstacles. Since no absolute referencing system is used, robots have to rotate on themselves to detect the presence of other robots nearby. This information is used by the recommendation level to establish the formation. Also note that *Keep-Formation* infers distance with other robots using only visual information, and it can adjust the pan angle of the camera according to the position of the colored-blob in the image to keep the *Leader* in its field of view. The zoom of the camera is not used. Proportional controllers are used to maintain specified angle with the *Leader*, while a fuzzy controller is used to maintain the specified distance.

B. Recommendation Level

The recommendation level manages the different states allowed for the control of the formation, as represented by a finite state machine in Figure 1. Behavior-producing modules get activated based on these states.

Initially, each robot begins in the *Rotation-360* state and

rotates on itself to perceive nearby robots. Information gathered during this step is then shared between the robots of the group to determine their position in the formation. Details on this part is presented in Section III. Once all robots know their respective leader and at which angle and distance they must follow it (condition B), each robot enters into *Get-In-Position* state. In this state, robots are allowed to turn to the right heading to see their leaders (condition C). If a robot does not see its leader, then it set to follow its nearest neighbor (condition D). Otherwise, if a robot does not see any other robots, it is set to remain inactive and not to interfere with the other robots of the group (condition E). The formation can then get established by letting the robots move autonomously.

While establishing or moving in formation, robots may or may not have difficulties following the assigned robots. If one robot loses sight of its leader or if it is experiencing difficulties moving because it comes too close to an obstacle (having to use the *Protection* behavioral module), a request to reinitialize the formation is sent to the group (condition F). All inactive robot (in *Stop* state) are then allowed back to the group (condition G). In the *Reassignment* state, the robots position themselves at a reasonable distance of other robots and obstacles nearby (by using the behavior *Protection*) before stopping. Once the group has stopped moving (condition H), the *Rotation-360* state is repeated to reevaluate a new assignment of positions for the formation. Note that the *Conductor* is also allowed to request a reassignment if it experiences trouble moving. Any robot that is experiencing difficulties while in formation is set not to be considered a potential conductor for the group during the position assignment process.

Finally, the *Communication Channel* link in the Figure 1 represents all information that is communicated to and between robots. It can be commands from an operator to tele-operate the conductor of the formation or to request that the group to move in a specific formation. This link also includes information shared by the group, i.e., the visibility tables, a request for reassignment transmitted by a new robot or by a robot experiencing difficulties.

III. POSITION ASSIGNMENT

By rotating on themselves, robots in the *Rotation-360* state gather information on the nearby robots. The perceived information is broadcasted to all members of the group so that each robot can initialize its visibility table. This table is an array of N by N , where N is the number of robots in the group. Each entry (i, j) in the table is the angle and the distance of the robot j as seen by the robot i . The robot i initializes the line i of the table with the information derived from its camera. The other lines are filled with the information transmitted by other robots. The attribution of robot position in the formation is based only on information gathered in this visibility table. Note that the algorithms does not verify mutual perception of robots: it is possible to have robot 1 see robot 2, with robot 2 not seeing robot 1.

Once these visibility tables are initialized for each robot, the robots enter in the *Position-Assignment* state. The objective

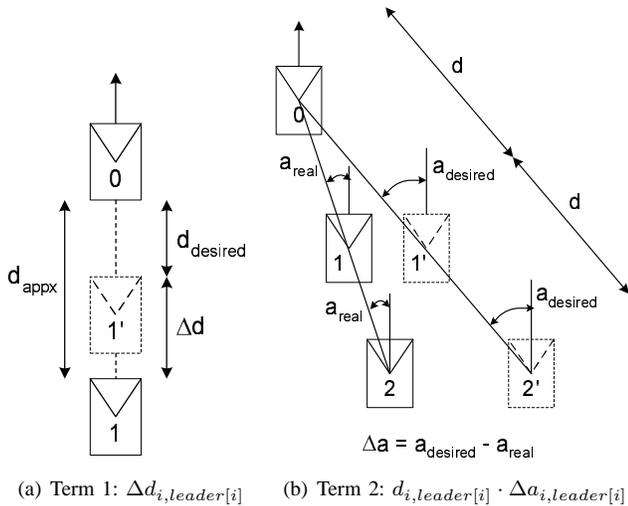


Fig. 2. Explanation of the assignment equation.

of this state is to search for a good assignment of position for robots in the formation. This search is distributed over the group. To do that, each robot assigns itself as being the *Conductor* of the desired formation and then searches the tree accordingly. It then uses a bounded depth-first search with pruning algorithm to find the best assignment of positions for others robots in the group. For each robot, the search algorithm minimizes the cost function given by Equation 2.

$$C = \sum_{i=1}^N \left[\frac{(\Delta d_{i,leader[i]})^2}{Level_i(f)} + (d_{i,leader[i]} \cdot \Delta a_{i,leader[i]})^2 \right] \quad (2)$$

where N is the number of robots in the formation f , Δ represents the difference between the desired and perceived parameter (distance d or angle a). The difference with our previous work [1] is that now the cost function considers the heading of the *Conductor*. $Level_i$ represents the level where the robot i is located in the formation. For example, for a line formation, each robot is side-by-side, all on level 1. For a column formation where each robot is placed one after the other, the conductor of the formation is on level 1, its follower is on level 2, and so on.

To better explain the cost function, Figure 2(a) represents $\Delta d_{i,leader[i]}$, which is the difference between the initial distance (approximated) and the desired distance of robots in the formation. This allows to favor the leader-follower assignment for nearest robots. Dividing this term by the $Level_i$, the robots assigned at the rear positions of the formation have a minimized impact on the cost. The robots at the first levels have more robots that depend on them (all the followers). So, it is better to minimize the movements of the front robots to give maximum chance to the followers to join their leaders quickly. Adding the square of the distance between robot i and its leader j , multiplied by the difference between the real angle and the desired angle between these robots, allows to minimize the angular displacement depending on the initial positions. As represented by Figure 2(b), if two robots have

to form an angle of 45° separated with the distance d , the best assignment is to have robot 1 follow robot 0 and not have robot 2 become the follower of robot 0 (to minimize the displacement of the robots). If the distance was not considered in the second term of Equation 2, robots 1 and 2 would get the same cost to follow the robot 0.

The bounded depth-first search assigns robot positions starting with the conductor of the formation and then assigning immediate followers first. A pruning is performed by using the monotonicity property of our cost function, i.e., that the cost never decreases when more terms are added. A branch is pruned as soon as the sum of the costs associated with the distance terms is greater than the current best score. The pruning could be made even more efficient by also taking into account the angular distance term.

In order to compute the second term, angles in the visibility table must be referenced to a common heading. Each robot considers and evaluates Equation 2 assuming its heading is the same as the direction of the formation. The values in the visibility tables are adjusted accordingly. Using the assignments from the cost computed with the directional displacement, the algorithm adds the angular displacement of each pair of leader-follower. Assuming that all robots in the group have the same heading, the angular displacement between a leader-follower pair is the difference between the desired angle and the actual angle as measured during the rotation on themselves.

The second term of Equation 2 is evaluated starting from the conductor, by summing the angular displacements for all follower-leader pairs. If a robot is assigned to follow a robot which it does not see (this robot does not have information concerning its leader), a huge score is added and the search continues with the next leader-follower pair. This score is set to 25 meters for the distance and to 360° for the angle.

In order to make the pruning more efficient, we also impose a higher bound on the cost function during the search. This bound prevents a robot from being assigned in a position for which it does not see its leader. Only if the search fails is the bound iteratively increased until a successful solution is found.

Moreover, if there is an incoherence in the visibility table (the leader sees its follower at a certain angle but its follower sees it at another angle) for a leader-follower pair, only the angular displacement computed for this pair is affected. Finally, it is possible that the best assignment found includes a follower-leader pair where the follower does not see its leader. Our algorithm considers this case as described in the Section II-B.

By having each robot search for the best assignment with itself as the formation conductor, the group performs a distributed search for the best assignment possible. The best result obtained by each robot is broadcasted to the others, and the one with the minimum cost is selected as the conductor of the formation, with positions assigned accordingly to the other robots. Once robots have reached a consensus (condition B), they all know which robot to follow (i.e., its color c), with angle a and distance d . They then enter the *Get-In-Position*

state to execute the algorithm described in Section II.

IV. EXPERIMENTAL EVALUATION

To validate our approach, we conducted experiments, both in simulation and with real robots. Our simulation experiments were done using the *Player* [2] and *Stage* [3] environment. In the real robot experiments, we used Pioneer 2 DX and AT robots equipped with Sony PTZ cameras, sonars and colored cylinders placed on their back to differentiate them.

A. Evaluation Criteria

To determine if robots are in formation or not, we need to define the notion of being in formation. Inspired by Fredslund [4], we propose the following criteria : given the positions of N mobile robots, an inter-robot distance $d_{desired}$, a desired angle a , and a geometric shape \mathcal{G} defined by a finite set of line segments and the angles between them, the robots are considered to be in formation \mathcal{G} iff:

- 1) *Inter-robot distance* : leader-follower pairs must respect the desired distance $dist(R_{i1}, R_{i2})$, with a maximal tolerance of ϵ_{d1} .
- 2) *Inter-robot angle* : the follower must respect the desired angle $\alpha(R_{i1}, R_{i2})$ with its leader, with a maximal tolerance of ϵ_a .
- 3) *Global shape* : the robots are in their assigned position depending on the desired formation with a maximal tolerance of ϵ_{d2} around this desired position.

Criteria 1 and 2 define tolerances for distance and angle measurements in order to account for perceptual noise and imprecise actuators. Criterion 3 states that it should be possible to adjust desired distances and angles over the predetermined position so that all robots form the desired shape. The robots can move into a radius of ϵ_{d2} from its desired position. With this criteria, we make sure that the desired shape is realized, even when each robot respects separately criteria 1 and 2.

B. Experimental Design

Our algorithm applies to many formations, but for the paper we focused on arrow, column, circle, diamond, wedge and line formations. Five different sets of random positions for a group of four robots, placed in a 4×4 -meter arena, were used in simulation. The first three sets are also used for the tests with real robots. Figure 3 illustrates these positions. The letter at front of each robot is their color ID: Blue, Orange, Green and Pink. These colors are used to identify robots in the assignment of position, according to the positions in a formation as illustrated in Figure 4. We ran five test conditions (1 trial for each test) for each formation in simulation and three tests for each formation using real robots (2 trials for each test are ran to validate the repeatability of our algorithm). Once all robots know its leader and are in the *Follow-In-Formation* state, the conductor is set to travel straight forward. ft corresponds to the distance travelled by the conductor up to the point where the robots establish the formation for the first time. For the experiments in simulation, $\epsilon_{d1} = 25\%$ of the desired inter-robot distance ($d_{desired} = 80\text{ cm}$) and $\epsilon_a = 5^\circ$. For

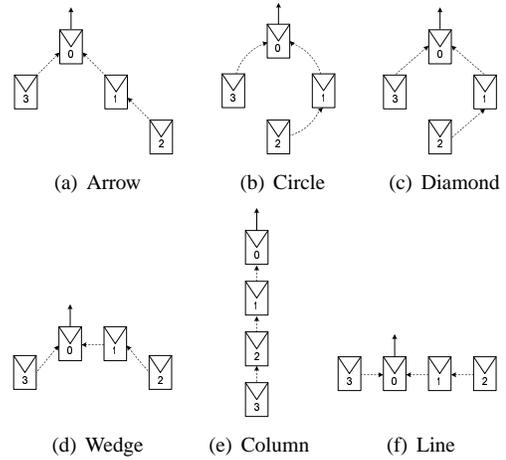


Fig. 4. Positions of robots in a formation ($N = 4$).

TABLE I

POSITIONS ASSIGNED BY THE ROBOTS (FOR THE FIVE TEST CONDITIONS), IN SIMULATION.

$N = 4$	Position's assignment (according to position 0-1-2-3)				
	Test 1	Test 2	Test 3	Test 4	Test 5
arrow	G-B-O-P	P-G-B-O	G-B-O-P	P-G-B-O	P-O-B-G
column	G-P-B-O	P-G-B-O	P-G-B-O	G-P-B-O	B-O-P-G
circle	B-O-P-G	P-G-B-O	B-P-G-O	G-O-B-P	P-B-O-G
diamond	B-O-P-G	P-G-B-O	B-P-G-O	G-O-P-B	P-B-O-G
wedge	G-B-O-P	P-G-B-O	G-B-O-P	P-G-B-O	P-O-B-G
line	P-B-O-G	P-G-B-O	G-B-O-P	P-G-B-O	P-O-B-G

real robots experiments, we are not able to obtain significant data since we did not have a way to determine the absolute positions of the robots during our experiments and there are severe perceptual and action limitations. Moreover, the robots do not have enough space in our laboratory (5×5 -meter arena) to initialize a formation according to some restrictive criteria. Both simulation and real robots experiments, criterion 3 was evaluated qualitatively by the experimenter.

C. Experimental Results

Table I reports, for tests in simulation, the position assigned by the robots to initialize the desired formation with minimum displacement. Assignments given into the table represent the color of the robot according to the position in the formation (see Figure 4) respectively. For example, in the first test of the arrow formation, the robot identified by a Green-blob is at the position 0 (it is the conductor), the robot identified by a Blue-blob is at the position 1 (it is the follower of the robot at the position 0) and so on.

We can observe that the found assignment minimizes the displacement: when the indicated conductor progresses, the desired shape is initialized without cross over of robot and the formation is formed directly. According to the Figure 3, we can see that both circle-diamond formation and wedge-line formation give the same assignment account to the similitude

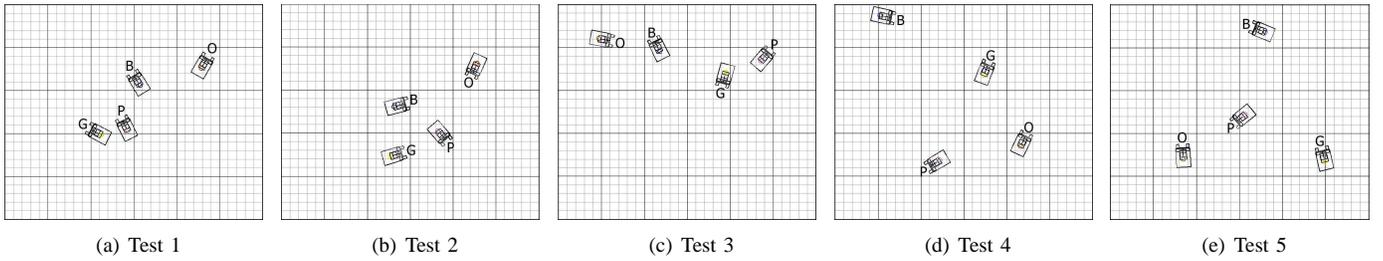


Fig. 3. Initials positions of robots for the five test conditions.

TABLE II

DISTANCE TRAVELLED BY THE CONDUCTOR (FIVE TEST CONDITIONS AND AVERAGE), IN SIMULATION.

N = 4	ft (m)					
	Test 1	Test 2	Test 3	Test 4	Test 5	Av.
arrow	0.24	0.47	0.16	1.52	1.94	0.86
column	0.34	1.35	0.06	2.59	1.41	1.15
circle	0.80	0.41	1.94	1.31	2.03	1.30
diamond	2.01	0.77	1.53	2.31	5.47	2.44
wedge	0.79	0.56	2.47	1.69	2.78	1.52
line	0.92	1.39	1.37	5.80	—	2.37

of their shape. This is right except for test condition 1 for the line formation and test condition 4 for the circle formation. These differences are due to an incoherence into the visibility table of a robot while it tried to find an assignment for the circle formation. It is possible to have difference in simulation because we have some glitches with the vision (a robot can see itself). While a robot does not have seen correctly nearby robots during its rotation, this erroneous information is transmit at all robot.

Table II reports the distance travelled by the conductor before the formations are established. Note that in test 4 of the diamond formation, a reassignment of positions occurred before the establishment of the formation. This was caused because robot G lost track of its leader, but still the same position assignment were preserved. A similar situation happened in test 5 of the diamond formation, but resulted in a new assignment of positions, which explains the large distance travelled by the conductor.

We can observe into this table that the formation is initialized quickly enough. For the line formation, the group was not able to initialize the formation in test 5. For tests 2 to 4, once the robots were in formation, they had difficulty maintaining the formation. This is due to inability of the robots to detect their orientations: even though a follower may see its leader on its side, it does not know the orientation of its leader. This physical limitation affect the ability to maintain formation which it has robots side-by-side, but this does not affect the capacity to find a good assignment.

Figure 5 illustrates what happened in test 4 for the circle formation. After rotating on themselves, the robots located

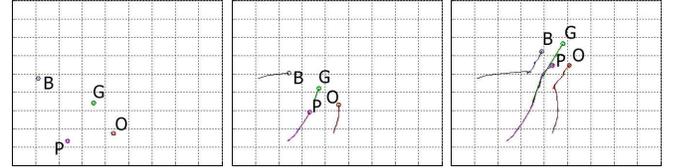


Fig. 5. Robot paths for test condition 4 for circle formation, in simulation.

TABLE III

POSITIONS ASSIGNED BY THE ROBOTS (FOR THREE TEST CONDITIONS), USING REAL ROBOTS.

N = 4	Position's assignment(according to position 0-1-2-3)					
	Test 1		Test 2		Test 3	
	Trial 1	Trial 2	Trial 1	Trial 2	Trial 1	Trial 2
arrow	G-B-O-P	O-P-G-B	P-B-G-O	P-B-G-O	B-G-P-O	B-G-P-O
column	G-P-B-O	O-B-G-P	P-G-B-O	P-G-B-O	P-G-B-O	P-G-B-O
circle	G-B-O-P	O-P-G-B	P-B-G-O	P-B-G-O	G-B-O-P	G-B-O-P
diamond	G-B-O-P	O-P-G-B	P-G-B-O	P-B-G-O	G-B-O-P	G-B-O-P
wedge	G-B-O-P	G-B-O-P	P-G-B-O	P-G-B-O	G-B-O-P	G-B-O-P
line	B-O-P-G	B-O-P-G	P-B-G-O	P-G-B-O	G-B-O-P	G-B-O-P

their leaders and the conductor travelled in straight line to initialize the formation.

Table III reports, for tests using real robots, the position assigned by the robots to initialize the desired formation with minimum displacement.

If we compare results of Table I with Table III, we can find that the robots found the same assignment 61% of the time. Also, considering only the Table III, we can observe that for both trials of the same test for each formation, real robots found the same assignment 67% of the time. Differences can be caused by perception problems with colors detection and lighting conditions. However, the assignment found by the search algorithm always gives assignment that minimizes displacement according to local perceptions of each robot.

Figure 6 illustrates what happened in test condition 1 for the arrow formation. As in simulation, after locating nearby robots, the robots located their leaders and the conductor travelled in straight line to initialize the formation.

V. RELATED WORK

In our earlier work [1], we proposed a taxonomy to characterize the robot formation control approaches, and we will use this taxonomy here to relate our work with others.



Fig. 6. Robot paths for test condition 1 for arrow formation, using real robots.

Our approach is characterized as using relative positioning with limited visibility and communication of local (such as the perception of other robots) and global (such as the assigned positions) information. It is based on neighbor-referenced positioning system (each robot maintains its position relative to one robot in proximity) with unidirectional influences, and uses a hybrid architecture (behavior-based and deliberative) to create flexible formation (i.e., that is allowed to change shape when avoiding obstacles).

The approach mostly related to ours is the one presented by Fredslund [4]. It is a distributed approach that uses behavioral modules to control formation. Each follower must keep its leader centered in the field of vision of its camera. Like us, each robot has a colored cylinder on its structure so that the other robots can recognize it. Contrary to our approach, the camera is fixed at an angle according to the formation. Consequently, formations are rigid, i.e., that the formation cannot change shape when dealing with obstacles. Also, a laser range finder is used to infer distance between robots. Compared to our approach, this provides much more accurate distance measurements than to just using vision. Position assignment in the formation is done based on the ID number of the robot. Switching formation is possible, but may require intermediate repositioning depending on the formation to switch to. Also, if the robots in the group are initially in an random position, the time required to initialize the formation shape may not be optimal.

Kumar et al. [5], [6] propose a hybrid distributed approach based on multiple controllers that can be seen as behavior selected according to a finite state machine. These states define control laws that consider nearby robots and the obstacles in proximity. The approach is vision-based with each robot identified by a color, but using omnidirectional cameras. Because the formation, its conductor and the allowed switch between formations must be predetermined, the approach can not be optimal depending if the environments are known or not, or if it is possible or not to initialize the position the robots in a good configuration for the desired formation.

Finally, Prencipe [7], [8] investigated pattern formation problems, including positions assignment, conductor selection and formation initialization with a group of N mobile robots. Robots are considered to be anonymous (each robot is indistinguishable), autonomous, oblivious (no memory of the past), asynchronous (no a priori synchronization) and using only implicit communication through the environment.

In our approach, each robot is autonomous, asynchronous and oblivious. We have shown that with minimal explicit

communication, a group of N (even or odd) robots can initialize and realize a predetermined formation based only on visual perception. Robots can be heterogeneous, but is not demonstrated in this paper. The robots, initialized at random positions, can select a conductor of the predetermined formation, select the best assignment to minimize displacement to configure the geometrical shape and establish and maintain the formation.

VI. CONCLUSION

This paper focuses on an approach allowing to assign positions for robot having to move in various formation. Using only directional visual perception, this paper demonstrates that robots can initialize and establish a predetermined formation autonomously and distributively using the proposed hybrid control architecture. Future work involves testing the approach in other conditions, improving the search algorithm to deal with robots that are assigned to follow a leader that is not directly visible, and integrating capabilities such as feasibility of formation, dynamic switching of formation, and merging of formations. The approach is also going to be used to validate relative positioning devices, providing more accurate readings of distance and angle between robots [9].

ACKNOWLEDGMENT

F. Michaud holds the Canada Research Chair (CRC) in Mobile Robotics and Autonomous Intelligent Systems. This research is supported financially by CRC, the Natural Sciences and Engineering Research Council of Canada (NSERC) and the Canadian Foundation for Innovation (CFI). Special thanks to Pierre Lepage for his help in this work.

REFERENCES

- [1] F. Michaud, D. Létourneau, M. Guilbert, and J.-M. Valin, "Dynamic Robot Formations Using Directional Visual Perception," in *Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, 2002.
- [2] B. Gerkey and M. Mataric, "Most Valuable Player: a Robot Device Server for Distributed Control," in *Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, 2001, pp. 1226–1231.
- [3] R. Vaughan, "Stage: a Multiple Robot Simulator," Institute for Robotics and Intelligent Systems, University of Southern California, Tech. Rep. IRIS-00-393, 2000.
- [4] J. Fredslund and M.J. Mataric, "A General Algorithm for Robot Formations Using Local Sensing and Minimal Communication," *IEEE Trans. Robotics and Automation*, vol. 18, no. 5, pp. 837–846, 2002.
- [5] R. Fierro, A. Das, V. Kumar, and J. Ostrowski, "Hybrid Control of Formations of Robots," in *Proc. IEEE Int. Conf on Robotics and Automation*, 2001, pp. 157–162.
- [6] A. Das, R. Fierro, V. Kumar, J. Ostrowski, J. Spletzer, and C. Taylor, "A Vision-Based Formation Control Framework," *Multi-Robot Systems: A Special Issue of IEEE Trans. Robotics and Automation*, vol. 18, no. 5, pp. 813–825, 2002.
- [7] P. Flocchini and G. Prencipe and N. Santoro and P. Widmayer, "Distributed Coordination of a Set of Autonomous Mobile Robots," in *Proc. IEEE Intelligent Vehicle Symposium*, 2000, pp. 480–485.
- [8] M. Cieliebak and G. Prencipe, "Gathering Autonomous Mobile Robots," in *Proc. VIII Int. Colloquium on Structural Information and Communication Complexity*, 2002, pp. 57–72.
- [9] J. Bisson, F. Michaud, and D. Létourneau, "Relative Positioning of Mobile Robots Using Ultrasounds," in *Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, 2003.