ROBOT BEHAVIOR ADAPTATION FOR FORMATION MAINTENANCE

Maite López-Sánchez
WAI Volume Visualization and Artificial Intelligence Research Group, MAiA Dep
Universitat de Barcelona. Gran Via de les Corts Catalanes 585, Barcelona, Spain

Keywords: Autonomous robotics, behavior-based robots, simulation.

Abstract: The autonomous robot formation maintenance problem can be approached by considering local information only. This approach is more realistic than using global information, but presents a troop deformation drawback. This paper performs a step forward in local information usage for formation maintenance by analyzing a parameterization of different basic behaviors. Formation maintenance emerges from the combination of these simple behaviors, and its overall accuracy is empirically optimized by tuning behavior parameters. In particular, we study and characterize three different formations: queue or column (as for ants), inverted V or wedge (as for birds or planes) and rectangle (for manipulus antique roman troop formations). This paper describes simulated robots that incorporate a unique set of basic behaviors from which formation maintenance emerges. These simple behaviors provide formation robustness and are parameterized in order to minimize deformation while following a trajectory.

1 INTRODUCTION

In this paper simulated robots implement a series of basic behaviors that use local information to allow the emergence of a global behavior that maintains the group formation without having the notion of it embedded in the individuals. In particular, we consider an autonomous maintenance of three different well-known formations in motion (see figure 1): queue, also known as line or column, is the simplest; wedge – or inverted V-formation– has aerodynamic advantages so it is usually adopted by birds and planes; and rectangle, which is much more condensed, corresponds to the 'manipulus' antique roman troop formation in military operations.

Most early work in formation control of robots (Bekey, 2005) has assumed global knowledge. Balch and Arkin identified tree approaches to formation control (Balch and Arkin, 1998): unit centre referenced, leader referenced and neighbor reference. They differ in the information that each robot requires to compute its desired position. Every robot in a unit centre referenced formation uses as reference the centroid position of the whole robot group, so robots require global information. Similarly, for leader referenced formations, robots always know the position of the leader regardless its position, thus this formation also entails a global scope. On the contrary, neighbor reference is the only that is considered to use local information since a robot can take as reference a robot in its vicinity and gather information about it (such as its position or distance to it) by using its own sensors.

Although simulations usually have access to global information, it is much more realistic to use local information when modeling physical formations such as robotic or biological groups, where the access to the overall information is hardly possible mainly due to sensing capabilities and to limitations on communication.

Therefore, our formation simulations consider local information only, assuming a neighbor reference approach. Furthermore, our pure local information approach lacks of a "formation notion". In this manner, a robot only knows about its neighbors and does not have the concept of group nor the group ability to keep the formation (since its measurement would require some sort of global information).

Unfortunately, local information presents the problem of error propagation among robots in the formation, whose main consequence is the deformation of the troop. This is an important issue that we tackle by parameterising the basic behaviors and performing experiments to study how these parameter values influence in the whole performance. In order to facilitate the set up and comparison of different settings, experiments have been conducted by sim-

López-Sánchez M. (2006).
ROBOT BEHAVIOR ADAPTATION FOR FORMATION MAINTENANCE.
In *Proceedings of the Third International Conference on Informatics in Control, Automation and Robotics*, pages 283-288
DOI: 10.5220/0001211502830288

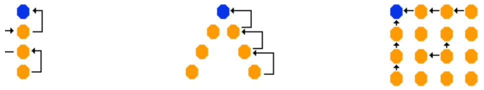


Figure 1: Robot’s references (black arrows) in our three different formations.

ulation, based on the open source OpenSteer (Steer, 2004) C++ library.

2 BASIC BEHAVIORS

We consider formations as specific distributions of robots with regular relative positions. Additionally, if formations are to be maintained while moving, they require a robust adaptation in order to keep these local relations as constant as possible. Simplicity is often related to robustness, and therefore, we propose that all robots in the troop do rely on a reduced set of basic behaviors to maintain formations.

Briefly, these simple behaviors are: “Reaching a target position”; “Reference neighbor following”; “Reactivity”; “Waiting for the follower”; and “Priority respect”. The former is the one that actually moves the robot towards a target position that is computed by the “Reference neighbor following” behavior based on the reference robot’s position. Nevertheless, one robot lacks of reference so that it is given a trajectory to follow and it is said to be the leader or conductor. Additionally, “Reactivity” behavior determines the degree of sensitivity of a robot regarding its reference. Finally, “Waiting for the follower” and “Priority respect” behaviors implement what could be interpreted as social courtesy.

This section describes these simple behaviors individually, giving a hint of their different complexity degrees and how can they be parameterized. Next section will afterwards show how three different formations are composed by defining different relative positions.

We propose the following basic behaviors:

Reference neighbor following: Robots follow the trajectories of their reference neighbors keeping fixed angles α and distances d (see Figure 2). Different formations require different angles and reference robots (see figure 1), so they can be treated as fixed formation properties. On the contrary, the *separation distance* depends on other factors such as robot visibility range, speed or reaction capabilities, so it has been used as a parameter to tune the overall performance.

Reactivity: Reference neighbor following implies the propagation and amplification of movements along the formation. Noisy movements must therefore be filtered. This is done by this “Reactivity” basic behavior, which establishes a *minimum movement distance* the reference robot must move before the follower reacts and follows it. Small values for this parameter do not avoid noise and emergence of many oscillations. On the contrary, large values introduce delays in the formation.

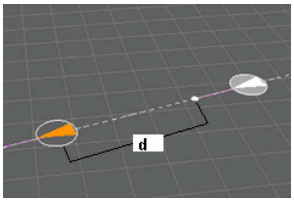


Figure 2: Reference neighbour following behaviour: a white robot follows the orange one.

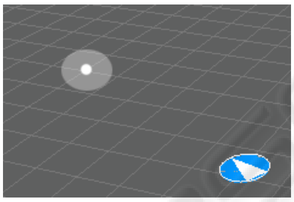


Figure 3: Tolerance for reaching a target position.

Reaching a target position: When a robot tries to reach a position, it must get to the target position and stop there, and therefore, it must reduce its velocity when approaching the target position at a certain *braking distance* (i.e. a parameter). If this distance is too large, robot separation distances are never accomplished, since the follower robot moves significantly slower than the reference robot. On the other hand, if this braking distance is too small, the inertia of a robot moving at high speed causes the robot to surpass the target position and to include loops in the trajectory that are afterwards propagated to following robots. Similarly, reaching an exact position may be too demanding for robots without much accuracy. This requires a *tolerance* parameter (see figure 3) so to enlarge the target position point up to a circle without loosing much accuracy in maintaining the formation.

Waiting for the follower: This behavior forces the reference robot to reduce its velocity when its follower robot exceeds a threshold distance (that is, before it can be lost). Figure 4 shows this *maximum separation distance* as the radius of blue circumferences centered on each reference robot. Obviously, this threshold distance should be larger than the separation distance parameter.

Priority respect: Leader’s trajectories can have loops that force following robots to cross their ways. Robots should thus avoid to collision with crossing ones. As figure 5 shows, this behavior has two parameters: a *critical stopping distance* that makes the

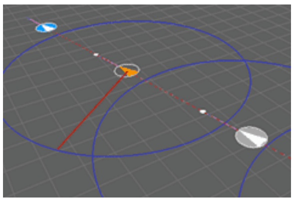


Figure 4: Waiting for the follower behavior.

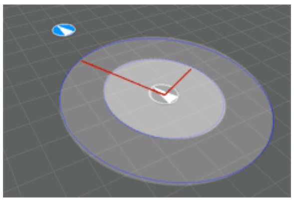


Figure 5: Priority respect behavior.

robot to stop in order to avoid an imminent collision and a larger precautionary distance that only requires a speed reduction (the *critical braking distance*). Both distances have an angle of influence and there is a priority system that establishes a total order relation among robots, so that when a robot encounters in its neighborhood area another robot, it detects its ordering and, and decides to give it the priority in order to avoid waiting deadlocks.

3 FORMATION MAINTENANCE AS EMERGENT BEHAVIOR

From the combination of previous basic behaviors we can obtain complex behaviors that allow the robots to maintain different formations. Each type of formation just emerges by specifying reference robots and the angle to form with them. Here we study three of them.

Queue: when having a queue of robots, the reference robot is the foregoer and the angle is zero degrees. The only exception is the leader, positioned on the first place, which follows its own trajectory. As a consequence of the “Reference neighbor following” behavior, the formation propagates the movement of the leader. In this manner, all robots in the queue pass eventually through the same positions. Figure 6 a) shows a snapshot of the formation in movement when the leader follows trajectories that are rectilinear (a), curved (b) or crossed (c).

Inverted V: birds and planes usually adopt inverted V-formations due to its aerodynamic advantages. Leaders are located at the centre of the formation and angles must be +/-45 degrees for those robots on the left/right side. As before, the reference robot is the foregoer. Rectilinear trajectories do not deform formations (see Figure 7 a)). On the contrary, deformations appear for turnings. Figure 7 b) shows the consequences of a left turn (right turns perform analogously): right-side robots must follow a wider trajectory so that distances between robots increase before robots can adapt whilst left-side robots must move slower because foregoers become closer to followers.

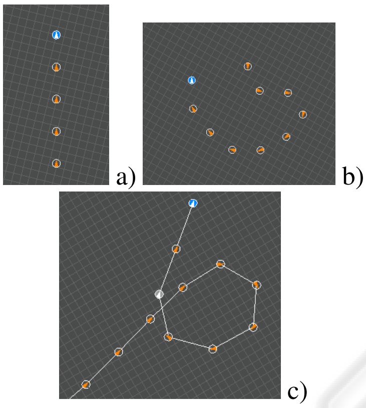


Figure 6: Queue formation in movement: a) The troop leader follows a rectilinear trajectory, b) a curved path, and c) a trajectory with a loop (white lines clarify successor relations).

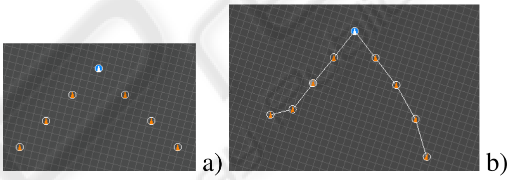


Figure 7: Inverted-V formation in movement. a) The troop leader follows a rectilinear trajectory, b) leader turns left.

Rectangle: This formation comes from the ‘manipulus’ antique roman troop formation in military operations and is characterized by its density of individuals. Figure 8 a) illustrates a formation with its leader located on the top left position. This facilitates left turns such as the one depicted in figure 8 b). Rectangle formations require robots to have two reference robots: the one in front and the robot on the left hand side. Therefore, angles are 0 degrees and -90 degrees respectively. As we can see, the deformation during turns becomes obvious, since robots on the right side of the formation must cover much longer distances, whereas robots behind the leader behave quite similarly to robots in a queue. Finally, just mention that, for right turns, the robot on the top right position should become the leader, just as soldier troops do in real settings.

Although queue formations adapt to trajectory

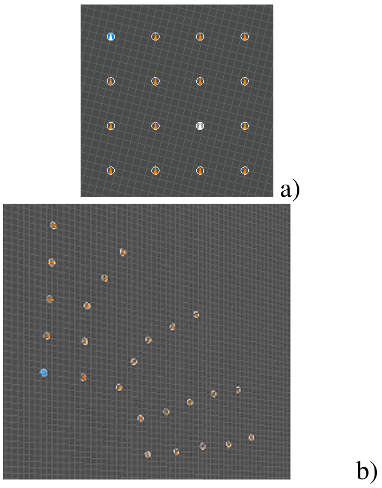


Figure 8: Rectangle formations a) 16 robots at its initial positions b) 25 robots when the leader has turned left.

changes faster than inverted-V and these faster than rectangles (we could say they present an increasing 'rigidity'), all three formation distributions do recover from deformations, especially when the leader follows a rectilinear trajectory for some time. They are also able to restore their topology once the leader stops. They do it naturally in an ordered manner: since changes propagate trough the formation, the successors of the leader are the first ones in reaching their target position, which, corresponds to their target position in the static formation. And this same process propagates until the last robot in the formation reaches its target position, so that the whole formation topology is recovered.

Nevertheless, formations are not kept exactly. Some delays are introduced due to the propagation of the movement and robots' errors do propagate with an accumulative effect. Next section presents some experiments we have performed with the aim of studying how basic behavior parameters can be set so that the error keeps as small as possible.

4 PERFORMANCE EVALUATION

In order to evaluate the formation maintenance performance of our different formations, we have considered an error measure that provides the maximum distance between robot actual trajectories and the ones that should have followed instead.

More concretely, for the queue case, every robot should follow the leader trajectory, and thus we measure, for each robot, the maximum distance between its trajectory and the leader's trajectory. Furthermore, robots can return to previously visited positions so that distance measures among trajectories are performed taking time into account. Therefore, we measure the distance between a robot's position and the equivalent position in the leader's trajectory for this specific instant.

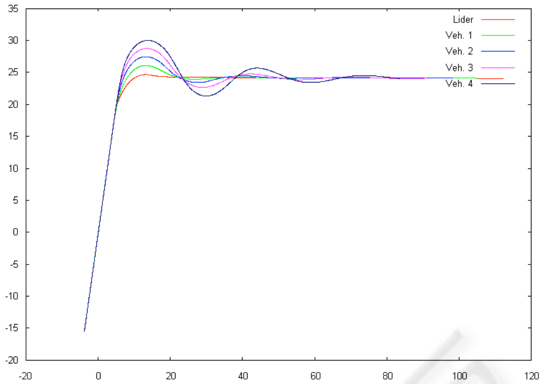


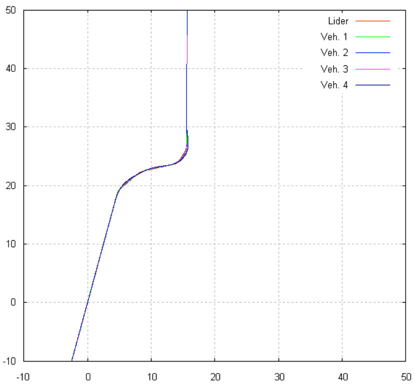
Figure 9: Trace of the trajectories of 5 robots in a queue formation. Leader's red trajectory is the reference one. Last robot (veh. 4) has the larger deviation.

5 RESULTS

Considering error measurements and behavior parameters described in previous sections, we have performed a series of tests about the formation maintenance performance in terms of the resulting error. We have done it by changing a single parameter for each test so that we can isolate its influence in the overall performance.

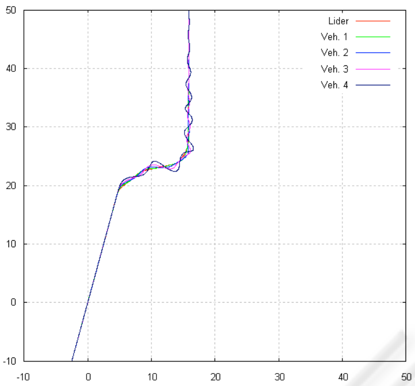
Figure 9 plots an example of how does perform a queue formation of 5 robots. In this case, the leader follows a trajectory that starts with a rectilinear movement, performs a right turning, and ends with a new straightforward movement. Consecutive robots (veh. 1 to veh. 4) do deviate along the turning and recover during the second rectilinear movement. For this specific example, the maximum error is performed by robot 4 at position (14.6, 29.9) where there is a distance of 5.37 to the reference leader position (14.03, 24.6). The average error for each of the 4 robots is 0.19, 0.50, 0.85, and 1.5 respectively.

By tuning some parameters, it is possible to reduce these performance errors empirically. Due to the lack of space we cannot present all conducted studies. Nevertheless, we exemplify error reduction by presenting the case shown in figure 10. We consider a queue formation composed by 5 robots where leader performs two consecutive turnings (right turn first, and left turn afterwards). Accuracy in following the trajectory (and thus, in maintaining the formation)



a)

Figure 10: Trajectory traces in a 5- robot queue formation with braking distance parameter = 2.0.



b)

Figure 11: Trajectory traces in a 5- robot queue formation with braking distance parameter = 1.0.

has visibly increased. In fact, the average error for each of the 4 robots is 0.03, 0.04, 0.05, and 0.07 respectively. These values can be considered especially accurate considering that a robot is simulated as a circle of diameter 1 in OpenSteer environment units.

Getting into more detail, this error reduction has been accomplished by setting the *separation distance* to 4.0 for the “Reference neighbor following” behavior. Although this distance could be thought as a formation parameter rather than a behavior parameter, our tests have proven that the performance drops considerably when this distance is smaller than 4 times the size of the robot. This is mainly due to the fact that, if robots do not have enough maneuverability, their turns generate oscillations that propagate errors. Nevertheless, in order to avoid robot losses, separation distance values should be proportional to the *maximum separation distance* parameter in the “Waiting for the follower” behavior. In our case, this last parameter has been set to 4.5 so that robots do not move at high speeds when its successors cannot follow them.

On the other hand, *minimum movement distance* parameter has been set to 2.0 for the “Reactivity” behavior. In general, being this value twice the robot’s size is enough to prevent for small local oscillations that do propagate along the formation. Values higher than that do introduce undesired delays that result in the deformation of the formation (usually, elongation). Another parameter that helps in reducing local oscillations is *tolerance*. Similarly, it should be kept small (it has been set to 0.1 in the example) to avoid global deformation. “Priority respect” behavior does also have parameters that help in the avoidance of undesired situations such as robot losses or collisions. These parameters are *critical stopping* and *critical braking* distances, which have been set to 2.0 and 3.5 respectively. These values, which must be correlated, are applied within an influence angle that has been set to 90 degrees in order to implement the right side priority traffic norm.

Finally, *braking distance* from “Reaching a target position” behavior has been set to 2.0 in the case shown in figure 10. This is a key parameter that affects three significant factors. Firstly, *braking distance* values do have an overall effect in the formation that is inversely proportional to the formation velocity: large *braking distance* values slow down the whole formation advance (robots start reducing its velocity unnecessary early) whilst small values allow the formation to advance faster. Secondly, its values introduce a divergence between the *separation distance* that should be kept between robots during formation displacements and the one that is actually kept. And thirdly, and most important, *braking distance* values do also affect into the accuracy in following the trajectory. On one hand, small values position robots so near to their target position that they are not able to react smoothly to turnings, and therefore, local oscillations are propagated and amplified among robots in the formation. On the other hand, large *braking distance* values enlarge target positions distances to an extent that causes robots to perform rectilinear shortcuts in tunings, and therefore, the accuracy in following the trajectory (and thus, maintaining the formation) is reduced.

Additional experiments have been performed for this *braking distance* parameter. In this manner, figure 11 shows the trace for a formation of 5 robots having value equal to 1.0 for this parameter and keeping those values mentioned before for the remaining parameters. In this case, the average error has increased up to 0.04, 0.07, 0.11, 0.18 for each of the four follower robots. As mentioned before, if we increase *braking distance* values up to 2.0, then errors decrease (figure 10: 0.03, 0.04, 0.05, 0.07), but it is a minimum, because if we keep increasing it, accuracy decreases again. In this case, for example, a *breaking distance* value of 3.0 involves average errors of 0.04, 0.5, 0.9, 0.1.

6 RELATED WORK

Multi-agent robotic systems have been intensively studied by the scientific community over the past decade ((Brooks et al., 1990) (Johnson and Bay, 1995)). The main reason for this is that, despite the limitations of single robots for accomplishing general tasks such as foraging, transportation, construction or surveillance, these tasks can be successfully achieved by coordinated groups of robots. Furthermore, some of these tasks can be outperformed when the group of robots form specific spatial distributions (Fredslung and Mataric, 2002a), what it is usually known as robot formations.

This paper presents a parameterization of basic behaviors whose combination yields to the emergence of a more complex global behavior that consists on formation maintenance while following a trajectory. In particular, robots have proven to be able to maintain three different formations just by using local information and without having the concept of formation explicitly. Local information refers to reference robots in the neighborhood, similarly to friend robots in (Fredslung and Mataric, 2002b). Our "Priority respect" behavior is also analogous to its robot ID ordering. Nevertheless, following its 'friendship' nomenclature, the "Waiting for the follower" behavior results in a more tight double-linked chain (i.e., reciprocal-friendship) than the single-linked chain of friendships of Fredslund and Mataric.

On the other hand, this "Waiting for the follower" behavior is related to the unsupervised formation maintenance work by Yamaguchi et al. (Yamaguchi et al., 2001), where attractions between robots are symmetrical. As in our case, the validity of their results was supported by computer simulations, but they study mathematically the stabilization of the formation by means of formation vectors that do apply in the formation creation rather than in the formation maintenance in movement. These formation vectors are also related to the attractive and repulsive gradient forces implemented by Feddema et al. (Feddema et al., 2004). Their work has a system control perspective that focuses on stability rather than, as in our case, in following accurately a trajectory while maintaining the formation.

7 CONCLUSIONS AND FUTURE WORK

Our work is based on the parameterization of basic behaviors to optimize the performance of robot formations empirically. Despite the potential loss of generality, this tuning strategy applies for different queue, inverted V and rectangle formations, and tries to pose a step forward in the solution of the formation maintenance problem when using local information. Future work will focus on the way adaptation can be achieved automatically: since we work on simulations, we envision genetic algorithms as an alternative, were the set of parameters codify the population and the error measure can be used as objective function to be optimized.

ACKNOWLEDGEMENTS

Bernat Grau's implementation has been key for this work.

REFERENCES

Balch, T. and Arkin, R. C. (1998). Behaviour based formation control for multi-agent robot teams. *IEEE Ransactions on Robotics and Automation*, 14(6):926–993.

Bekey, G. A. (2005). *Autonomous Robots: From biological Inspiration to Implementation and Control*. MIT Press.

Brooks, R. A., Maes, P., Mataric, M. J., and More, G. (1990). Lunar base construction robots. In *Proceedings of IEEE/RSJ International Workshop on Intelligent Robots and Systems (IROS)*, pages 389–392.

Feddema, J. T., Schoenwald, D., Parker, E., and Wagner, J. S. (2004). *Analysis and Control of Distributed Cooperative Systems*. Sandia Report SAND2004-4763.

Fredslung, J. and Mataric, M. (2002a). A general algorithm for robot formations using local sensing and minimal communication. *IEEE Transactions On Robotics And Automation*, 18(5).

Fredslung, J. and Mataric, M. (2002b). Robots in formation using local information. In *Seventh International Conference on Intelligent Autonomous Systems (AIS-7)*, pages 100–107.

Johnson, P. J. and Bay, J. (1995). Distributed control of simulated autonomous mobile robot collectives in payload transportation. *Autonomous Robots*, 2(1):43–63.

Steer, O. (2004). *Steering behaviours for Autonomous Characters*. http://opensteer.sourceforge.net/.

Yamaguchi, H., Arai, T., and Beni, G. (2001). A distributed control scheme for multiple robotic vehicles to make group formation. *Robotics and Autonomous Systems*, 36:125–147.