

Formation Maintenance for Autonomous Robots by Steering Behavior Parameterization

MAITE LÓPEZ-SÁNCHEZ, JESÚS CERQUIDES

WAI Volume Visualization and Artificial Intelligence Research Group, MAiA Dept.

Universitat de Barcelona

Gran Via de les Corts Catalanes 585, Barcelona

SPAIN

{maite,cerquide}@maia.ub.es <http://wai.maia.ub.es>

Abstract: Most often, autonomous robots maintain group formations by using global information such as the position of the group leader or even the position of all robots inside the formation. Alternative approaches to autonomous robot formations have considered local information, which is more realistic but presents some drawbacks such as troop deformation. In this paper we perform 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).

Key-words: Autonomous robotics, behavior-based robots, simulation.

1. INTRODUCTION

This paper presents an approach to group formations that considers simulated autonomous robots. These 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 the autonomous maintenance of three different well-known formations in motion (see figure 2.1): 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 ro-

man troop formation in military operations.

Most early work in formation control of robots [2] has assumed global knowledge. Balch and Arkin identified tree approaches to formation control [1]: 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 another 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 parameters’ values influence in the whole performance. In order to facilitate the set up and comparison of different settings, experiments have been conducted by simulation, based on the open source OpenSteer [9] C++ library ¹.

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”; “Limited passivity”; “Waiting for the follower”; and “Priority respect”. The first one actually moves the robot towards a target position which is computed by the “Reference neighbor following” behavior using the reference robot’s position. Nevertheless, one robot (the leader) lacks reference and therefore, it is given a trajectory to follow. Additionally, “Limited passivity” 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 they can be parameterized. Next section will afterwards show how three different formations are composed by defining different relative positions ².

We propose the following basic behaviors:

Reference neighbor following: When this behavior is active, robots do follow the trajectories of their reference neighbors keeping fixed angles and distances. Different formations require different angles and reference robots (see figure 2.1), 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.

¹A detailed description of the robot model can be found at [10].

²Although the paper focuses on the problem of formation maintenance in movement, it is worth noticing that robots with these behaviors are also able to distribute themselves to set up the formation.

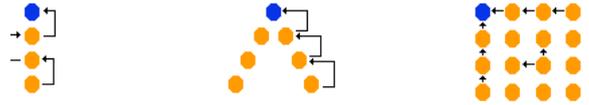


Figure 2.1: Robot’s references (black arrows) in our three different formations

Figure 2.2 depicts a white robot following an orange robot. Circled triangles represent robots, whose headings correspond to triangle top vertexes. This behavior has been implemented so that the follower robot computes its target position as the one located at the given distance d and defining a specific angle α with respect to the reference robot’s heading. In the figure, target position appears as a white dot at distance d and $\alpha = 0$ degrees. One robot within the troop lacks reference so that it is given a trajectory to follow and it is said to be the leader or conductor.

Limited passivity: “Reference neighbor following” implies the propagation and amplification of movements along the formation. Noisy movements must therefore be filtered. This is done by this “Limited passivity” basic behavior, which establishes a *minimum movement distance* the reference robot must advance before the follower reacts and starts following it. Under small values of this parameter noise and oscillations still appear. On the contrary, large values avoid noise and oscillations but introduce delays in the formation. Therefore, this behavior determines the degree of sensitivity of a robot regarding its reference.

Reaching a target position: When a robot tries to reach a position, it speeds up as much as possible (as long as there are no other behaviors slowing down the robot). Nevertheless, the robot must get to the target position and stop there, and therefore, it must reduce its velocity when approaching the target position. In this manner, a *braking distance* parameter has been specified for this behavior implementation. If this distance is too large, the distance that must be kept between robots in the formation is 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 does not allow stopping with a sudden braking, and as a result the robot surpasses the target position so that it must go back towards it. This undesired turnings retard robots and include loops in the trajectory that are afterwards propagated to following robots. Similarly, reaching an exact position while following a reference

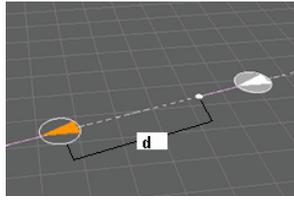


Figure 2.2: “Reference neighbour following” behaviour: a white robot follows the orange one.

robot may be too demanding for robots without much accuracy even when they do move slowly. This requires the addition of an extra parameter, the so-called *tolerance*. It enlarges the target position point up to a circle, so that it is easier for robots to reach it. Again, this *tolerance* should be balanced with the accuracy in maintaining the formation.

Waiting for the follower: when dealing with local information, robots can lose their references, especially when they have different speeds. This behavior forces the reference robot to reduce its velocity when its follower robot exceeds a threshold distance (named *maximum separation distance*), which is also a parameter. Obviously, this threshold distance should be larger than the separation distance that the follower must keep. This behavior, together with the next one, implement what could be interpreted as subconscious social courtesy.

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 (obstacle avoidance is provided as a repulsion force mechanism by the OpenSteer library, and therefore, we do not discuss it here). This behavior has two parameters: a *critical stopping distance* that makes the robot to stop in order to avoid an imminent collision and a larger precautionary distance that only requires a speed reduction. We named this last parameter as *critical braking distance*. Both distances have an angle of influence, so that, for example, a robot will not stop because of another robot with higher priority is approaching its back area. Furthermore, waiting deadlocks can be avoided by means of a priority system that establishes a total order relation among robots. This order relation can be as simple as assigning consecutive numbers to robots in the formation (being 1 the leader, 2 its follower, and so on), so that when a robot encounters in its neighborhood area another robot, it detects its number and, in case it is smaller than its own number, it gives it the priority.

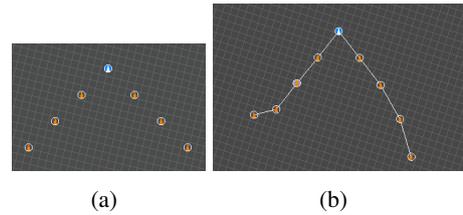


Figure 3.1: Inverted-V formation in movement. a) The troop leader follows a rectilinear trajectory, b) leader turns left.

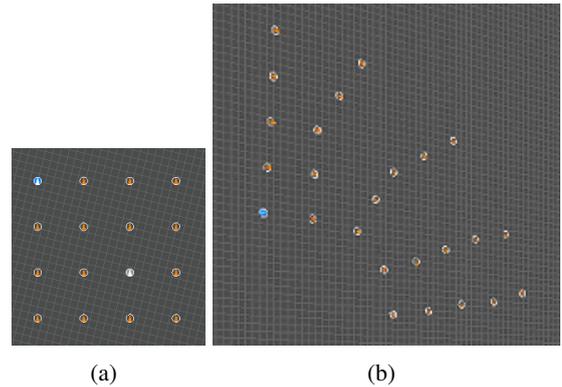


Figure 3.2: Rectangle formations a) 16 robots at its initial positions b) 25 robots when the leader has turned left.

3. FORMATION MAINTENANCE AS EMERGENT BEHAVIOR

From the combination of the 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 focus on the study of the Queue formation. However, preliminary results for Inverted-V (depicted in figure 3.1) and Rectangle (figure 3.2) formations are also provided. The emergence of these behaviors has been further described in [8].

When having a queue of robots, the reference robot

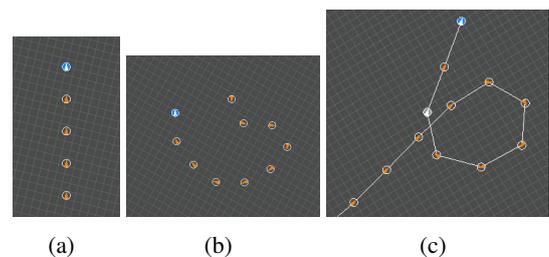


Figure 3.3: Queue formation in movement: a) The troop leader follows a rectilinear trajectory, b) a curved path, and c) a trajectory with a loop.

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 3.3 a) shows a snapshot of the formation in movement when the leader follows a rectilinear trajectory. Visually, angles and distances are kept rather constant. Figure 3.3 b) depicts a snapshot of a curved trajectory. In this case, robot headings must adapt to the curvature. Finally, figure 3.3 c) illustrates a crossed trajectory that requires robots to apply its “Priority respect” behavior. White lines linking successor robots are shown for clarity purposes only.

We shall recall that robots do only have local information that is managed by basic behaviors, so formation maintenance emerges from its combination. From these basic behaviors, the “Reference neighbor following” behavior together with the “Reaching a target position” behavior are the ones that generate the local movement of the robots. This local movement is propagated among robots so that the global formation movement emerges. On the other hand, “Limited passivity” behavior provides stabilization by avoiding the propagation of local oscillations. And finally, the “Waiting for the follower” behavior prevents robot segregation and is key for the global deformation recovery, which is an important aspect of the ‘robustness’ of the formation (in the formation maintenance sense).

Nevertheless, formations are not kept exactly. Some delays are introduced due to the propagation of the movement (that is, it takes some time from the first leader’s movement until last robot moves). But most importantly, robots’ errors do propagate with an accumulative effect, so that the last robot amplifies the accumulated error of its predecessor by adding its own error. As we mentioned before, basic behaviors do have some distance parameters such as *required* and *maximum separation distances* between consecutive robots. Next section presents some experiments we have performed with the aim of studying how these parameters can be set so that the error keeps as small as possible.

4. RESULTS

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. In our simulations, each

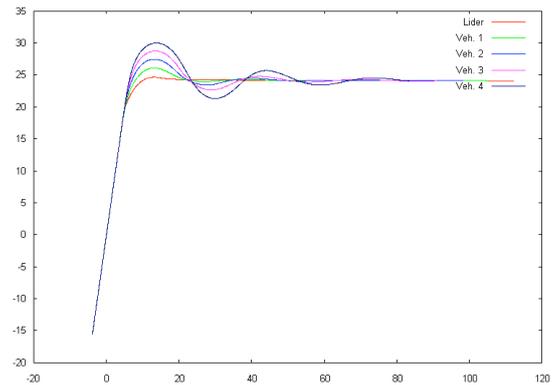


Figure 4.1: Trace of the trajectories of 5 robots in a queue formation. Leader’s red trajectory is the reference one. Last robot (veh. 4) has the larger deviation

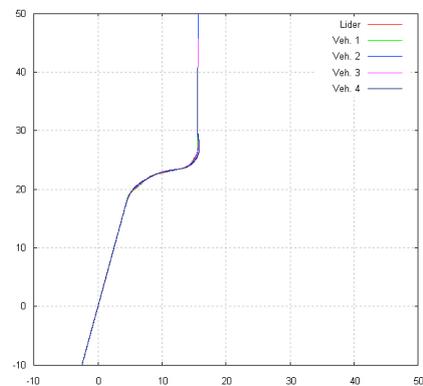


Figure 4.2: Trajectory traces in a 5-robot queue formation with braking distance equals to 2.0

robot only knows its own positions along time, so error measurement is an offline process that we have performed for completeness purposes rather than for correcting errors in run time.

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 4.1 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.

Behavior	Parameters	Val
“Reference neighbor following”	<i>separation distance</i>	4.0
“Limited passivity”	<i>minimum movement distance</i>	2.0
“Reaching a target position”	<i>braking distance tolerance</i>	{1, 2, 3} 0.1
“Waiting for the follower”	<i>maximum separation distance</i>	4.5
“Priority respect”	<i>critical stopping distance</i> <i>critical braking distance</i>	2.0 3.5

Table 4.1: Behavior parameters.

By tuning some parameters, it is possible to reduce these performance errors empirically. This paper focuses on the behavior parameters proposed in section 2, which are summarized in table 4.1. A detailed empirical study of the performance of these formations and justification for the parameter values can be found in [8].

We exemplify error reduction by presenting the case shown in figure 4.2. As before, we consider a queue formation composed by 5 robots, but in this case, the leader performs two consecutive turnings (right turn first, and left turn afterwards). Accuracy in following the trajectory (and thus, in maintaining the formation) 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.

In table 4.1 we can also see the parameter values that have been used in the testing. There, *braking distance* from “Reaching a target position” behavior is varied between 1 and 3. Figure 4.2 shows the case for value 2.0. 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 do also have an effect that is proportional with the *separation distance* that is actually kept between robots. In this manner, they introduce a divergence between the 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 fol-

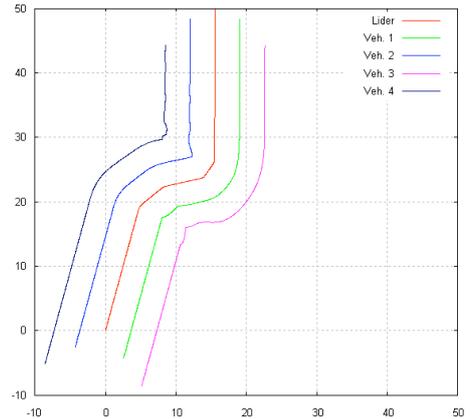


Figure 4.3: Trajectory trace of 5 robots in inverted-V formation. The leader (on the centre) performs one right and one left turn. External robots on the side of each turning present larger deviations.

lowing 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 : having its value equal to 1.0, the average error has increased up to 0.04, 0.07, 0.11, 0.18 for each of the four follower robots. In fact, value 2.0 is a minimum, because if we keep increasing it, accuracy decreases again (for example, a value of 3.0 involves average errors of 0.04, 0.5, 0.9, 0.1.)

Equivalent tests have also been performed for the inverted-V and rectangle formations (see [8] for more details). Figures 4.3 and 4.4 show two traces with the aim of providing an intuitive idea of how this kind of formations perform. As it can be seen, errors do propagate towards the external robots in the formation.

5. RELATED WORK

Multi-agent robotic systems have been intensively studied by the scientific community over the past decade ([3], [7]). 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

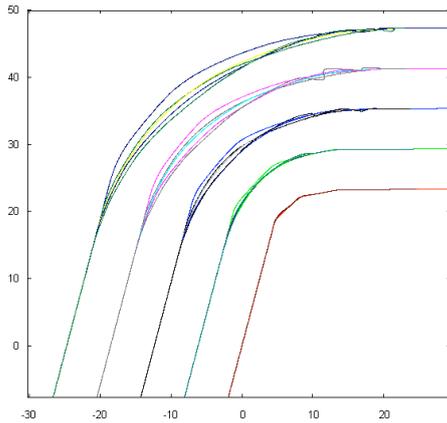


Figure 4.4: Trajectory trace of 25 robots in rectangle formation. Leader’s trajectory appears in red. Robots in the same column inside the rectangle do have superposing trajectories.

by coordinated groups of robots. Furthermore, some of these tasks can be outperformed when the group of robots form specific spatial distributions [5], 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 [6]. 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. [11], where attractions between robots are symmetrical. As in our case, the validity of their results was supported by computer simulations, but they study matematically 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. [4]. 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.

6. 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.

This paper allows us to envision parameter tuning as a feasible mechanism for formations to increase their performance autonomously. Therefore, future work will focus on the way this can be achieved automatically. Since we work on simulations, we envision genetic algorithms as an alternative, where the set of parameters codify the population and the error measure can be used as objective function to be optimized.

Acknowledgments. Bernat Grau’s implementation has been key for this work

REFERENCES

- [1] T. Balch and R. C. Arkin. Behaviour based formation control for multi-agent robot teams. *IEEE Ransactions on Robotics and Automation*, 14(6):926–993, 1998.
- [2] G. A. Bekey. *Autonomous Robots: From biological Inspiration to Implementation and Control*. The MIT Press, 2005.
- [3] R. A. Brooks, P. Maes, M. J. Mataric, and G. More. Lunar base construction robots. In *Proceedings of the 1990 IEEE/RSJ International Workshop on Intelligent Robots and Systems (IROS90)*, pages 389–392, 1990.
- [4] J. T. Feddema, D. Schoenwald, E. Parker, and J. S. Wagner. *Analysis and Control of Distributed Cooperative Systems*. Sandia Report SAND2004-4763, 2004.
- [5] J. Fredslung and M. Mataric. A general algorithm for robot formations using local sensing and minimal communication. *IEEE Transactions On Robotics And Automation*, 18(5), October 2002.
- [6] J. Fredslung and M. Mataric. Robots in formation using local information. In *Seventh International Conference on Intelligent Autonomous Systems (AIS-7)*, pages 100–107, 2002.
- [7] P. J. Johnson and J. Bay. Distributed control of simulated autonomous mobile robot collectives in payload transportation. *Autonomous Robots*, 2(1):43–63, 1995.
- [8] M. López-Sánchez and B. Grau. Parameterization of autonomous robot formations based on local information. Technical Report 2006/01, WAI. Volume Visualization and AI Reseach Group, 2006.
- [9] OpenSteer. Steering behaviours for autonomous characters. <http://opensteer.sourceforge.net/>.
- [10] C. Reynolds. Steering behaviors for autonomous characters. In *Proceedings of Game Developers Conference*, pages 763–782. Miller Freeman Game Group, San Francisco, California, 1999.
- [11] H. Yamaguchi, T. Arai, and G. Beni. A distributed control scheme for multiple robotic vehicles to make group formation. *Robotics and Autonomous Systems*, 36:125–147, 2001.