Leader-Following Graph-Based Distributed Formation Control

José Rodrigues, Dario Figueira, Carlos Neves, Isabel Ribeiro

Abstract— **This paper presents the distributed formation control of a multi-agent system using graph theory with emphasis on consensus and cooperation issues. The focal point is to achieve and maintain a formation from any initial condition, with and without a leader that the entire formation must follow. Our analysis framework is based on tools from algebraic graph theory, matrix theory and control theory. We present a brief derivation of multi-agent consensus in continuous-time and the corresponding iterative form stated in discrete-time, because while the real scenario is continuous, the implementation that we simulate is discrete. Based on the discrete-time algorithm, we propose a solution to obtain and uphold consensus when there is a leader to command the entire network. Simulation results are presented, indicating the capabilities and limitations of the algorithms.**

I. Introduction

This paper presents consensus based algorithms for the coordination of a networked multi-agent system that aims at achieving and preserving a formation amongst themselves.

A. Formation control

Networked Systems have lately been the focus of scientific attention due to the boom in computation speed and reliable communications. This provided a solid base for the development of several applications like formation flight control [1], [2], satellite clustering [3], and the control of groups of unmanned vehicles [1], [4], [5].

Advantages of interconnected multi agent systems over conventional systems include reduced cost, increased efficiency, performance, reconfigurability, robustness, and new capabilities.

A team of smaller robots to perform the same task of a larger single robot is at a distinct advantage in case of a malfunction. In one case the team of decentralized units will adapt to the loss of a team member and continue cooperating to accomplish the given task, on the other case the single robot is surely doomed as well as its given mission. Also, a space radar based on satellite clusters [6] is estimated to cost three times less than currently available systems, increase geolocation accuracy by a factor of 500, offer two-orders-of-magnitude smaller propulsion requirement, and be able to track moving targets through formation flight.

Undirected Graphs have been often picked to represent formations due to the instinctive way they describe the interconnection topology of a formation, *e.g.* in [6] and [7]. Moreover, directed graphs have been chosen to reflect the control structure [8], the constraint feasibility [9], the information flow [10], to quantify error propagation [11] and to reflect leader following inter-agents control specifications throughoutly scrutinized [12], [13], [14].

The authors are with the Institute for Systems and Robotics at Instituto Superior Técnico (IST), Av. Rovisco Pais, 1049-001 Lisboa, Portugal. Contact author: José Rodrigues. E-mail: jerasman@gmail.com.

The problem of coordination in multi-agent systems can be characterized naturally by a finite representation of the configuration space, namely by using graph-theoretic models to describe the local interactions in the formation, where nodes symbolize the physical entities (agents) and the edges represent virtual entities that support the information flow between the nodes.

B. Graph-based models to control a formation

This paper is mainly based on the notable results that have arisen since 2001. The groundwork on stating and solving consensus problems in networked dynamic systems appeared in [15] and [16], results that were later used in [17] and [18]. The issue of reaching an agreement without computing any objective functions was initially addressed in [19] and later extended in [20], [21]. These main results, which have a well described summary in [22] by Olfati-Saber *et al.*, are the base that supports the development in this work.

The problem of reaching a formation based on graph theory was already solved by Fax and Murray [16]. This theory consists on given an arbitrary initial position make the agents reach a consensus on a final common point. Then, a bias value is introduced, adding the feature that the final positions of the agents will not be a common point but a formation given by a desired geometric topology.

This framework, presented in Section II, consists in an introduction to the main problem discussed on this paper that consists on adding a leader to command the network and maintain the formation while performing the leader motion.

In the context of this paper, a formation is defined by relative positions between vehicles in a network interconnected by inter-vehicle communications. Multi-vehicle systems are an important category of networked systems due to their commercial and military applications. There are two broad approaches to handle distributed formation control: i) representation of formations as rigid structures [7], [23] and ii) representation of formations using the vectors of relative positions of neighboring vehicles and the use of consensus-based controllers with input bias [22]. In this paper we discuss this latter approach.

We explore graph-based models to control a desired formation, representing the interactions and the flow of information between the multiple agents in the graph. Graph and control theory support the formulation of the problem and help propose an elegant solution for the cases addressed in this paper.

C. Paper Organization

In Section II, we address the problem of reaching consensus in a distributed network. We present important theory results known from the literature. Section III solves the problem on reaching consensus in the presence of a leader that the rest of the network must follow, this being the paper's main contribution. In Section IV we describe simulation results that illustrate the behavior of the whole network using the methodology proposed. Section V concludes the paper and presents directions for future work.

II. Distributed Consensus without a Leader

A. Problem Statement

Our goal is to be able to control the general movement of a network of robots with only weak knowledge of other agents (the position of its neighbors). Here, in Section II, we analyze the consensus algorithm and how to achieve a robot formation with it and in Section III we introduce a leader that will control the general movement of the network.

Consider the problem of n holonomic robotic agents, moving in $\mathbb{R}^2$. The state vector that represents each robot location is a 2D space, because the knowledge of orientation is irrelevant from the view point of position control. Therefore, the location of each agent is represented by $\bar{x}_i \in \mathbb{R}^2, i = 1, 2, ..., n$ and the state of the network is defined as $\bar{x} = [\bar{x}_1^T \bar{x}_2^T ... \bar{x}_n^T]^T$.

The assumption of the robotic agents to be holonomic also allows the dynamics along each dimension to be decoupled. Hence, each dimension can be considered independently, being sufficient to analyze the performance along a single dimension. We will refer always just to one coordinate of $\bar{x}_i \in \mathbb{R}^2$, which we call x_i, since both coordinates can be described by similar equations. In this framework we consider network elements as integrators agents with dynamics $\dot{x}_i = u_i$.

B. Consensus in Continuous-time

Graph theory can provide a variety of useful notations and tools for analyzing some control strategies for such a system [24]. The interaction topology of a network of agents is represented using a directed graph $G = (V, E)$ with the set of nodes represented by $V = 1, 2, ..., n$ and with edges $E \subseteq V \times V$. Fig. 1 shows a graph-based representation for the interconnected system, considering that the agents have the dynamics of a pure integrator.

The neighbors of agent i are denoted by $N_i = \{j \in V : (i, j) \in E\}$ and the elements a_{ij} of the adjacency matrix A are 1 or 0 according to whether i and j are adjacent or not [24].

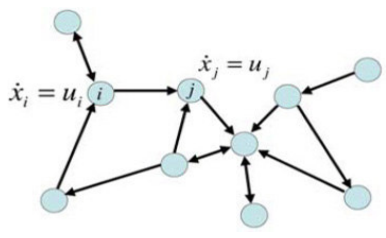

Fig. 1. Graph-based representation of the network of agents with integrator dynamics (reprinted from [22]).

A widely adopted distributed control strategy for driving n integrators agents to a desired formation can be expressed as an n^{th}-order linear system on a graph, [16],

$$\dot{x}_i(t) = u_i(t) = \sum_{j \in N_i} (x_j(t) - x_i(t)) + b_i(t) \qquad (1)$$

where $j \in N_i$ encodes the fact that the information is only allowed to flow from adjacent agents j and i, $x_i(0)$ is the initial position of the agent i and the bias value $b_i(t)$ provides information about the desired formation. If $b_i(t) = 0$, the n integrator agents will be driven to a common point that corresponds to the average of all agents' positions. Otherwise, the agents will reach the consensus not at a single point but in relative positions between each other, assuming a formation with a desired geometric topology encoded on $b_i(t)$.

In the first part of this section we consider $b_i(t) = 0$, and the agents are driven to a common point. In the last part $b_i(t) \neq 0$, we expose how the bias value encodes the geometric topology, leading to a given formation.

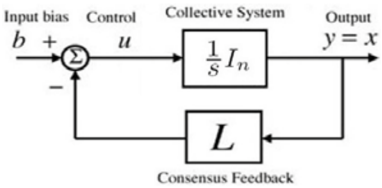

Fig. 2. Block diagram of the collective networked system (reprinted from [22]).

The collective dynamics of the group of agents following equation (1) can be written as

$$\dot{x} = -Lx \qquad (2)$$

where L in (2) is the graph Laplacian of the network with elements defined as

$$\begin{cases} -1, & j \in N_i \\ |N_i|, & j = i \end{cases} \qquad (3)$$

with $|N_i|$ the number of neighbors of node i.

Fig. 2 illustrates the consensus algorithm described by (1) and (2) as a multiple-input multiple-output (MIMO) linear system. The uniqueness and asymptotical stability of the equilibrium point of this consensus algorithm is proven in [23].

C. Discrete-time Consensus Algorithm

The equivalent consensus algorithm in discrete-time can be stated as

$$x_i(k+1) = x_i(k) + \epsilon \sum_{j \in N_i} (x_j(k) - x_i(k)). \qquad (4)$$

The collective dynamics of entire network under (4) can be written as

$$x(k+1) = Px(k) \qquad (5)$$

where $P = I - \epsilon L$ with $\epsilon > 0$ the step size in the discretization. With this algorithm formulation, it is demonstrated in [22], that we need to guarantee that

- $0 < \epsilon < 1/\Delta$, where Δ is the maximum degree of the network and
- the digraph is strongly connected [24],

for the consensus to be asymptotically reached for all initial states.

A strongly connected digraph is a directed graph in which it is possible to reach any node starting from any other by traversing edges in the directions they point to.

This means that the second condition is not reached when a leader is introduced in the network since a leader "doesn't care" about the rest of the network, but the entire network must follow it. This is the main contribution of this paper. A solution to this problem is discussed in the next section.

Differently from the problem addressed before when the consensus leads to a common point (the mean of the starting positions), for a formation to be obtained, a bias is added to (4) resulting in:

$$x_i(k+1) = x_i(k) + \epsilon \sum_{j \in N_i} (x_j(k) - x_i(k) - r_{ij}). \quad (6)$$

The value r_{ij} represents the desired inter-agent relative position between i and the neighbor j (example provided in Section IV). The collective dynamics can be written as

$$x(k+1) = Px(k) + \epsilon \ diag(AR). \quad (7)$$

The matrix R is composed by the elements r_{ij} that contain information about the formation and A is the adjacency matrix that constrains the information flow between agents. A is a global matrix, is considered static for the purposes of this paper and encodes what in a real application would be the inability of some robots to determine another robot's position. The operator $diag(AR)$ represents the vector whose elements are those in the diagonal of the matrix AR. The network is still driven to form around the mean point of the initial positions.

III. Leader-Based Multi-Agent Coordination

When a leader is inserted into the network, the digraph is no longer strongly connected, as shown in Fig. 3, but the consensus algorithm presented in (6) remains valid in the sense that, when the leader stops, the formation will be reached, and until then, the followers will always try to form up, using their current relative position to the leader and among each other to correct their locations. However, it is evident that the performance will degrade relative to the problem addressed in Section II due to the lack of information about the existence of a leader imposing a motion to the group of agents.

In this section we address a different problem, now with $n+1$ agents. We consider that there is an agent called "leader" whose movement is not determined by the consensus equation but instead by an external input or a predefined velocity and/or path pattern. The remaining n agents, named "followers" are ruled by the consensus equation and must follow the leader while preserving a formation. Therefor, with the introduction of the leader we gain the possibility to move the formation.

As in the previous section, we consider that the network may have any initial configuration, and therefore the followers must reach and maintain a formation while following the leader. As in general it is not possible to inform the followers of the leader's motion we continue to assume that the only information an agent collects is the position of its neighbors, since that can be easily obtained, for example, by communications or vision.

The solution proposed for this new problem is that every agent in the network will learn the average network velocity, which will converge to the average leader velocity.

Arbitrating that the leader is node n+1, this learning procedure is implemented, following the lines in [25],

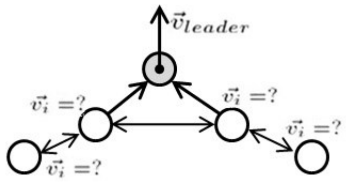

Fig. 3. Illustration of the problem caused by the insertion of a leader to the network.

through the inclusion of a new term in (6) and (7) leading to

$$x_i(k+1) = x_i(k) + \epsilon \textstyle\sum_{j \in N_i} (x_j(k) - x_i(k) - r_{ij}) \\ +\xi_i(k+1), \quad i = 1, 2, ..., n \quad (8)$$

$$x_{n+1}(k+1) = x_{n+1}(k) + v_{leader} \quad (9)$$

where N_i in (8) is evaluated for all the $n+1$ agents.

The collective dynamics of the followers can be written as

$$x_F(k+1) = Px_F(k) + \epsilon \ diag(AR) + \xi(k+1) \quad (10)$$

where $x_F(k)$ are the first n elements of $x(k) \in \mathbb{R}^{n+1}$ and the matrices A, R and P are the same as in (7). $\xi \in \mathbb{R}^n$ encodes the average velocity of the network and follows the dynamics

$$\xi(k+1) = \xi(k) + \gamma \left[Px_F(k) + \epsilon \ diag(AR) - x_F(k) \right]. \quad (11)$$

With some analysis we can verify that ξ_i is the average velocity of the neighbors for node i. Each node $i = 1, 2, ..., n$ will attempt to match its own velocity with that of its neighbors, which will cause the convergence of all agent's velocity to the same value. Since the leader disregards all this but the followers still see him as a fellow node, this referred value is ξ_i and moreover it converges to v_{leader}.

By analyzing the system in steady state we explain in the sequel why ξ_i converges to the leader's velocity. When consensus is reached, we have $\sum_{j \in N_i} (x_j(k) - x_i(k) - r_{ij}) = 0$, which causes (8) to be simplified into $x_i(k+1) = x_i(k) + \xi_i(k+1)$. Likewise (10) will be reduced to $x_i(k+1) = x_i(k) + \xi_i(k+1)$, $i = 1, 2, ..., n$. Finally (11) becomes $\xi(k+1) = \xi(k)$, which means that ξ has converged to a constant δ, and therefore $x_i(k+1) = x_i(k) + \delta$.

If the leader is standing still, reaching consensus means that the network agreed upon a final standing still position that can only be sustained if $\delta = \xi = 0$, which leads to $x_i(k+1) = x_i(k)$. If the leader is moving with a constant velocity, in order to $\sum_{j \in N_i} (x_j(k) - x_i(k) - r_{ij})$ remain equal to zero, each agent must be moving at the same speed, the leader's speed. Henceforth, $x_i(k+1) = x_i(k) + \delta$ implies that δ, and therefore ξ, has converged to the leader's velocity.

ξ being the average velocity of the network forcibly means that the followers will lag behind the leader until they learn the referred term. If the leader moves with constant velocity, ξ will be well learnt and consequently the followers will act as if the leader is stopped (because all agents will be moving with the same average speed) and will converge to the desired formation as in Section I-B.

The network dynamics in (7) can be considered as a particular case of (10) with $\xi = 0$. This makes sense since the network aims to converge to a stationary configuration. Now ξ is different from zero and is defined in such a way as to encode information about the leader velocity when its dynamics is constant.

In Section IV, results of simulation are presented showing that a high value for γ leads to undesired oscillations in the followers' movement.

IV. Results

In this section we present and discuss simulation results that illustrate the various aspects addressed in the previous sections. In Subsection IV-A, we present experiments that illustrate the behavior of the agents when trying to reach consensus without a leader. In Subsection IV-B we show the performance of the new consensus equations proposed in (8), (9) and (10). Then, in Subsection IV-C we discuss the influence of the parameters and their sensitivity. Finally, in Subsection IV-D we describe the relative location error that occurs when the leader has a non-constant velocity.

A. Distributed Consensus without a Leader

To present the results of the simulations, it is necessary to state the network configuration used in the experiments. In Fig. 4 the initial positions of the agents are represented by dots, expressing the nodes of the graph, and the arbitrarily chosen possible flows of communication between the nodes are represented by arrows, expressing the edges of the graph.

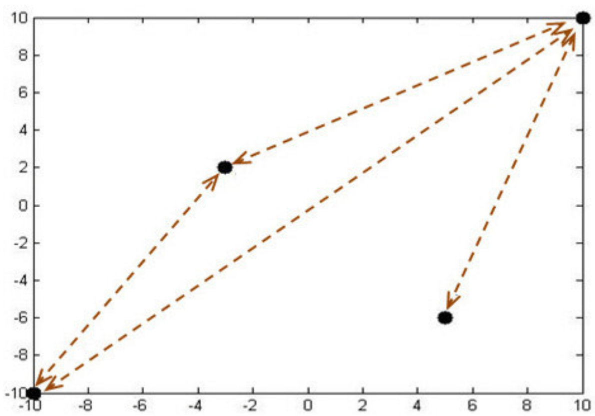

Fig. 4. Graph representation of the network placed in the initial position.

In Fig. 5 we represent the simulated results using the graph presented in Fig. 4. Equation (4) defines the distributed consensus algorithm used to drive the agents to the common final location, at $(0.5, -1)$, the mean point of the agents' initial position. Each agent's path is represented by the small dots, each dot representing a node at a certain time instant.

This is the simplest case, where the formation resumes to a single common point. Fig. 6 shows the geometry of a desired formation to the network. This topology is achieved using the elements r_{ij}:

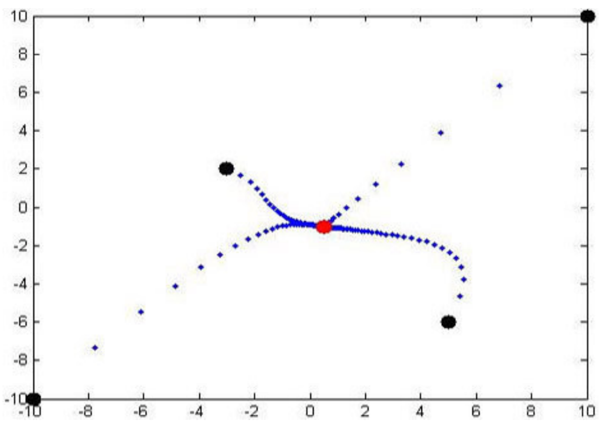

Fig. 5. Driving the agents to a common point. $\epsilon = 1/15\Delta$

$$R_x = \begin{bmatrix} 0 & 1 & 2 & 0 \\ -1 & 0 & 1 & -1 \\ -2 & -1 & 0 & -2 \\ 0 & 1 & 2 & 0 \end{bmatrix} R_y = \begin{bmatrix} 0 & 2 & 0 & -2 \\ -2 & 0 & -2 & -4 \\ 0 & 2 & 0 & -2 \\ 2 & 4 & 2 & 0 \end{bmatrix} \tag{12}$$

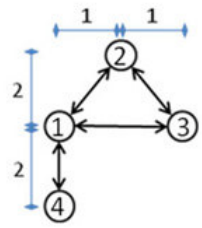

Fig. 6. Target relative positions of the formation.

Fig. 7 illustrates the time behavior of the system until it reachs the final formation in Fig. 6, using the consensus algorithm presented in (6) and (7). Note that the topology is centered at the mean point of the agents' position, which is independent from the topology of the network or its movement.

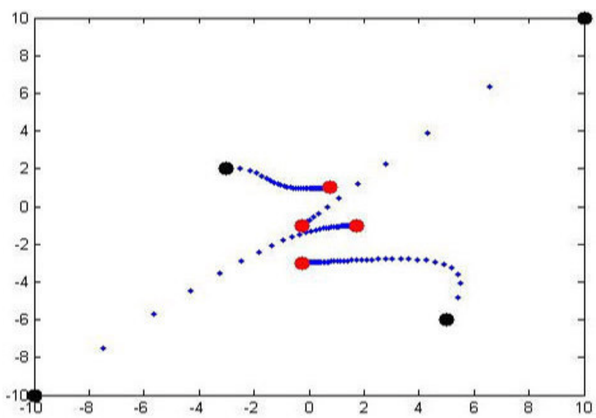

Fig. 7. Reaching a formation. Big dots are initial and final positions. $\epsilon = 1/15\Delta$

B. Leader-Based Multi-Agent Coordination

The main goal in these experiments is to present results of the consensus algorithm with different motions of the leader. The final formation that the network has to reach and maintain while following the leader is the one presented in Fig. 6.

In Figs. 8 to 12 particular positions of the agents are represented for given time instants with a square, for the leader, and big dots for the followers. In Fig. 8 the leader is moving with constant velocity and the final position is shown at instant 75. The agents in the network reach the desired formation and follow the leader. Using $\gamma = 0.6$, a low settling time is achieved with some oscillations.

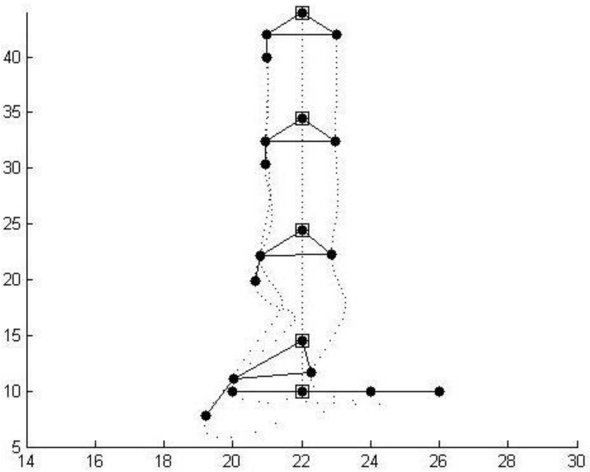

Fig. 8. Leader with a linear trajectory and constant velocity. $\gamma = 0.6$

This is a simple case, since the leader has a constant velocity along a linear trajectory. Fig. 9 presents the network's behavior when the leader is performing a non-linear trajectory, again with constant speed. The followers attempt to reach the formation and reproduce the same motion of the leader, with good results. Again a good balance between settling time and oscillations is achieved using $\gamma = 0.6$.

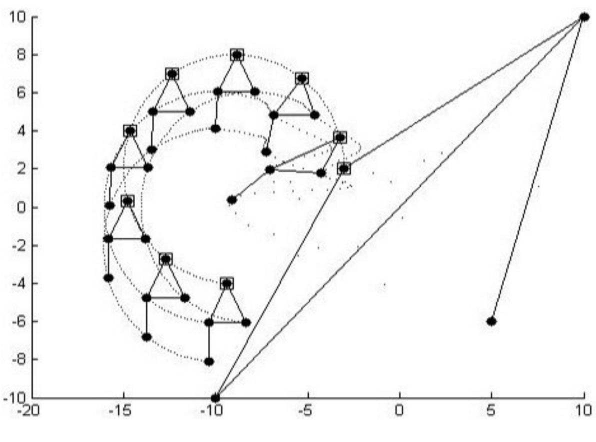

Fig. 9. Circular trajectory with $\gamma = 0.6$.

The final case considers a leader with constant speed along a sinusoidal trajectory as displayed in Fig. 10. The faster the variation of the movement direction, the harder it is for the network to maintain the formation. Still using $\gamma = 0.6$, and in spite of the oscillations of the sinusoidal trajectory, the constant changing of leader velocity, the followers are able to maintain the formation.

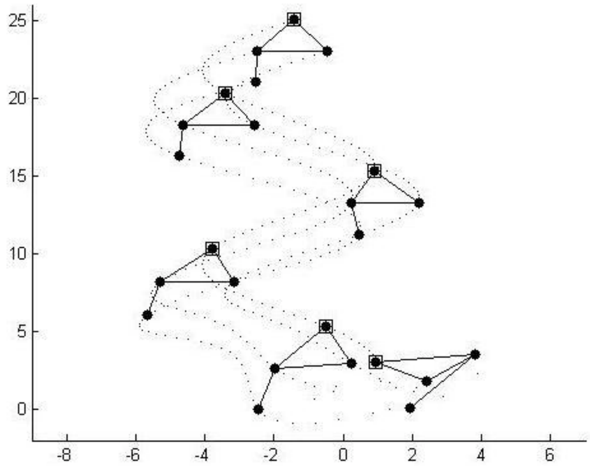

Fig. 10. Sinusoidal trajectory with $\gamma = 0.6$.

C. Parameters Influence and Sensitivity

A good value of the parameter γ in (11), that balances settling time and oscillations, was experimentally found to be 0.6. Fig. 8 and Fig. 9 show a good compromise between oscillation and settling time thanks to the tuning of γ.

Fig. 11 and Fig. 12 illustrate the effect of a lower value of gamma, in this case $\gamma = 0.03$, with little to none oscillations but a higher settling time. The difference in the settling time is highlighted in Fig. 13, where we compare the settling of the circular trajectory with $\gamma = 0.6$ and $\gamma = 0.03$.

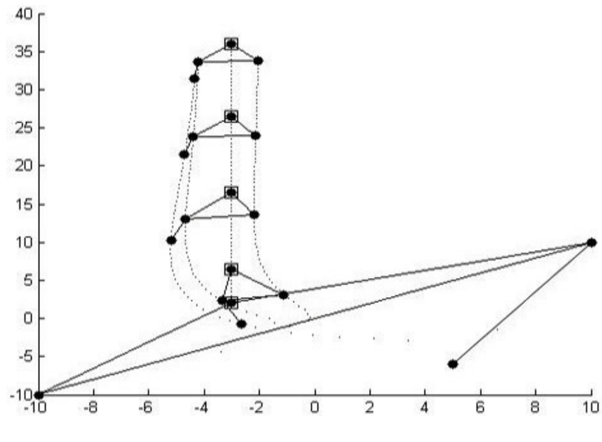

Fig. 11. Linear trajectory with a leader at constant velocity. $\gamma = 0.03$

Since (10) is a model of a constant velocity moving network, if the settling time increases so does the difficulty to adjust to non-linear trajectories. This is illustrated in Fig. 12, where the followers' delay trying to catch up to the leader is visible. And in Fig. 13 there is a clear non-null error due to this delay.

The higher the gamma's value, the larger the oscillations will be. In fact, a value of gamma larger than 1 causes the agents to diverge, due to the role it plays in (11).

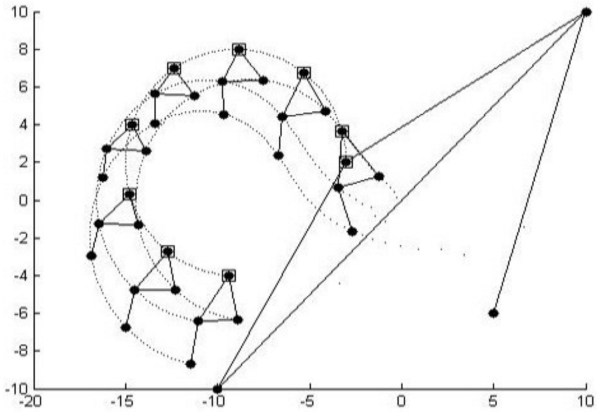

Fig. 12. Circular trajectory with $\gamma = 0.03$.

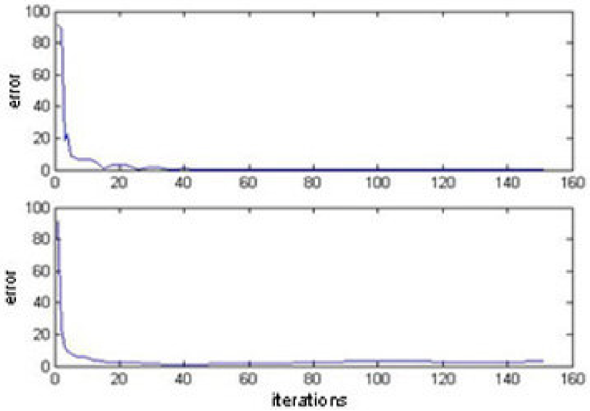

Fig. 13. Positional error comparison. Above: Positional error (13) of Fig. 9. Below: Positional error (13) of Fig. 12.

In (5) we needed to guarantee that $0 < \epsilon < 1/\Delta$. The experimental value that provides a good balance between oscillations and settling time was $\epsilon = 1/1.05\Delta$. The previous experiments (Fig. 8 up to Fig. 12) use this value. Lower values of ϵ lead to smaller step sizes and higher settling times as illustrated by the comparison of Fig. 7 and Fig. 14.

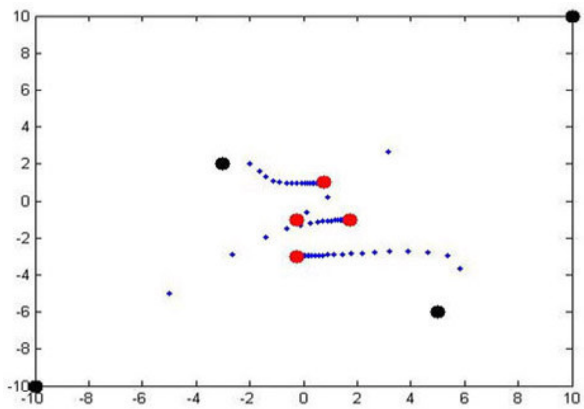

Fig. 14. Reaching a formation. $\epsilon = 1/2\Delta$.

In a similar way to the experiments with a variable gamma, larger settling times lead to poorer performance from the network, as we can see in Fig. 15.

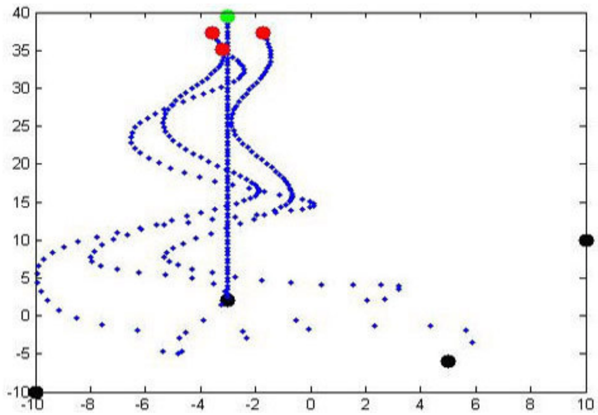

Fig. 15. Linear trajectory with constant velocity. $\gamma = 0.6, \epsilon = 1/3\Delta$

Values of ϵ larger than $1/\Delta$ cause immediate divergence, as it was predicted in [22]. This is due to the fact that epsilon needs to be in a certain range in order to uphold the necessary properties of the Perron matrix, P.

D. Position Error

Position error was calculated like this:

$$e = \sum_{i=1}^{4} \sum_{j \in N_i} (|x_j - x_i|) - r_{ij} \qquad (13)$$

Because (10) only models the first order motion, there is still an error in position associated with acceleration. Using the linear trajectory with different accelerations, the average position error converges to the results in Table I.

TABLE I

Position Error due to non-null acceleration in the leader.

Acceleration	Position Error
0,1	0,175
1	1,75
10	17,5

As it was expected, our experiments suggest that this error is proportional to the leader's acceleration.

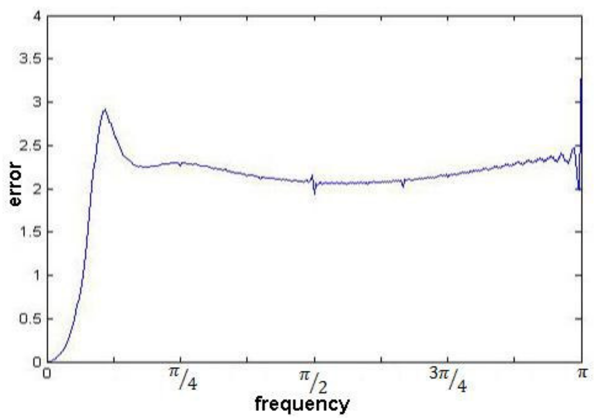

Fig. 16. Frequency response of the formation error when leader is excited sinusoidally. $w \in [0, \pi]$

Experimental results also show that there is an upper bound to the error when leader is excited with different frequencies. Figure 16 pictures that situation, representing the frequency response of the error when the leader's velocity variation is sinusoidal.

V. Conclusion

Previously published works have shown how to use the undirected and directed graphs to represent multi-agents networks, their topology and communication. It has also been studied how to control formations and achieve consensus using graph-theoretic models. Our work takes the research one step further, with the introduction of a moving leader into 2D formations of holonomic agents.

Based on the previous results, we had to look at the behavior of the network when adding a new element, that does not care about consensus. The solution presented is to add a learning term that gives a first order approximation of the general movement, that is, a velocity estimation of the leader.

The results show that it is possible to maintain a formation sharply when the leader moves smoothly, with constant speed, thanks to the first order approximation that allows velocity learning. However, when there are variations in velocity a position error proportional to the acceleration appears.

In the future, there are a few improvements to be made. First, tuning the algorithm for different kinds of non-holonomic agents. This will increase the complexity of the program, since the dimensions are not independent any more. Second, take into account movement restrictions, the rotation of the robots and obstacle avoidance. And third, extending it to different dynamics besides $\dot{x} = u$, since real robots will have dynamics modeled by different equations.

Also, this approach does not incorporate the expected uncertainties on the determination of robots position, so as additional future work would be the introduction of noise in the robots positions and the confirmation that asymptotic convergence is maintained as long as connections are not permanently lost.

VI. Acknowledgments

This work was partially supported by ISR—IST pluri-annual funds from FCT.

The authors of this paper have greatly profited from hints generously lavished in an Autonomous Systems course at IST from Professor Pedro Lima. Our cordial thanks.

A special recognition is also given to Professor António Pedro Aguiar of Institute for Systems and Robotics, that with great availability gave powerful hints in control theory to help solving leader-based problems.

References

[1] M. Mesbahi and F. Hadaegh, "Formation flying of multiple spacecraft via graphs, matrix inequalities, and switching," *AIAA J. Guid., Contr. Dynam.*, vol. 24, pp. 369–377, Mar 2001.

[2] R. W. Beard, J. Lawton, and F. Y. Hadaegh, "A coordination architecture for spacecraft formation control," *IEEE Trans. Contr. Syst. Technol.*, vol. 9, pp. 777–790, Nov 2001.

[3] C. R. McInnes, "Autonomous ring formation for a planar constellation of satellites," *AIAA J. Guid., Contr. Dynam.*, vol. 18, no. 5, pp. 1215–1217, 1995.

[4] L. P. F. Giulietti and M. Innocenti, "Autonomous formation flight," *IEEE Contr. Syst. Mag.*, vol. 20, pp. 34–44, June 2000.

[5] D. J. Stilwell and B. E. Bishop, "Platoons of underwater vehicles," *IEEE Contr. Syst. Mag.*, vol. 20, pp. 45–52, Dec 2000.

[6] R. Olfati-Saber and R. M. Murray, "Distributed structural stabilization and tracking for formations of dynamic multiagents," in *Proc. IEEE Conf. Decision and Control*, Las Vegas, NV, Dec 2002, pp. 209–215.

[7] P. N. B. T. Eren and A. S. Morse, "Closing ranks in vehicle formations based on rigidity," in *Proc. IEEE Conf. Decision and Control*, Las Vegas, NV, Dec 2002, pp. 2959–2964.

[8] A. K. Das, R. Fierro, and V. Kumar, *Cooperative Control and Optimization of Applied Optimization*. Norwell, MA: Kluwer, 2002, ch. 4 - Control Graphs for Robot Networks, pp. 55–73.

[9] P. Tabuada, G. J. Pappas, and P. Lima, "Feasible formations of multiagent systems," in *Proc. American Control Conf.*, Arlington, VA, June 2001, pp. 56–61.

[10] J. A. Fax and R. M. Murray, "Information flow and cooperative control of vehicle formations," in *Proc. 15th IFAC World Congr.*, Barcelona, Spain, July 2002, pp. 2360–2365.

[11] H. Tanner, V. Kumar, and G. Pappas, "Stability properties of interconnected vehicles," in *Proc. 15th Int. Symp. Mathematical Theory of Networks and Systems*, Notre Dame, IN, Aug 2002, [CD-ROM].

[12] J. P. Desai, J. P. Ostrowski, and V. Kumar, "Modeling and control of formations of nonholonomic mobile robots," *IEEE Trans. Robot. Automat.*, vol. 17, pp. 905–908, Dec 2001.

[13] R. Fierro, A. Das, V. Kumar, and J. P. Ostrowski, "Hybrid control of formation of robots," in *Proc. IEEE Int. Conf. Robotics and Automation*, May 2001, pp. 157–162.

[14] H. G. Tanner, G. J. Pappas, and V. Kumar, "Input-to-state stability on formation graphs," in *Proc. IEEE Int. Conf. Decision and Control*, Las Vegas, NV, Dec 2002, pp. 2439–2444.

[15] J. A. Fax, "Optimal and cooperative control of vehicle formations," Ph.D. dissertation, Control Dynamical Syst., California Inst. Technol., Pasadena, CA, 2001.

[16] J. A. Fax and R. M. Murray, "Information flow and cooperative control of vehicle formations," *IEEE Trans. Autom. Control*, vol. 49, no. 9, pp. 1465–1476, Sep 2004.

[17] R. Olfati-Saber and R. M. Murray, "Consensus protocols for networks of dynamic agents," in *Proc. 2003 Autom. Control Conf.*, 2003, pp. 951–956.

[18] ——, "Consensus problems in networks of agents with switching topology and time-delays," *IEEE Trans. Autom. Control*, vol. 49, no. 9, pp. 1520–1533, Sep 2004.

[19] A. Jadbabaie, J. Lin, and A. S. Morse, "Coordination of groups of mobile autonomous agents using nearest neighbor rules," *IEEE Trans. Autom. Control*, vol. 48, no. 3, pp. 988–1001, Jun 2003.

[20] L. Moreau, "Stability of multi-agent systems with time-dependent communication links," *IEEE Trans. Autom. Control*, vol. 50, no. 2, pp. 169–182, Feb 2005.

[21] W. Ren and R. W. Beard, "Consensus seeking in multi-agent systems under dynamically changing interaction topologies," *IEEE Trans. Autom. Control*, vol. 50, no. 5, pp. 655–661, May 2005.

[22] R. Olfati-Saber, J. A. Fax, and R. M. Murray, "Consensus and cooperation in networked multi-agent systems," in *Proc. of the IEEE*, vol. 95, no. 1, Jan 2007, pp. 215–233.

[23] R. Olfati-Saber and R. M. Murray, "Graph rigidity and distributed formation stabilization of multivehicle systems," in *Proc. 41st IEEE Conf. Decision and Control*, vol. 3, 2002, pp. 2965–2971.

[24] C. Godsil and G. Royle, *Algebraic Graph Theory*. Springers, 2001.

[25] A. P. Aguiar, I. Kaminer, R. Ghabcheloo, A. M. Pascoal, N. Hovakimyan, C. Cao, and V. Dobrokhodov, "Coordinated path following of multiple UAVs for time-critical missions in the presence of time-varying communication topologies," *to appear at IFAC'08 World Congress, Korea*, July 2008.