Consensus in leaderless and single-leader, connected and switching networks

Ananye Pandey and Esha Trivedi

Abstract

The concept of flocking through consensus has been a subject of study since swarms of insects and flocks of birds were observed in nature following a common direction with a constant velocity, without colliding with each other. This observation was first observed by Reynolds in 1987. In the case of leaderless networks, a group of local and global control laws are implemented to ensure the velocity values and directions converge along with inter-agent distances. These control policies are brought about by introducing a local sensor network and a global connections network. In this case, switching topology is implemented by generating random adjacency matrices over iterations.

1. INTRODUCTION

Jadbabaie et al. [1] define flocking as "a group of mobile agents is said to flock when all the agents attain the same velocity vector, distances between the agents is stabilized, and collision between agents is avoided." Most work in this field is motivated by the link of flocking with various aspects of biology, social behavior, robotics, distributed control and social behavior. Efforts to first understand and replicate this behavior was made by Reynolds [2] by studying schools of fish and flocks of birds, with him claiming that this behavior showed local as well as global synchronization. He further claimed in [3,4] that in order to align they needed a global picture, but for flocking to occur only local information was required.

Three basic properties need to be satisfied for flocking:

1. Separation between flockmates
2. Alignment in direction
3. Cohesion to move in the same direction

The superposition of these three properties ensures flocking.

Such a model can be generalized into leader-follower networks, where only the leader is capable of making a decision and the followers can only receive information and decide.

Each agent is modelled as a dynamical system with either single or double-integrator dynamics. Most average consensus algorithms consider cases of single integrator dynamics modelled as follows:

$$\dot{x} = -Lx$$

Where L is the graph Laplacian. Olfati-Saber et. al. [3] analyze this case of consensus and arrive at an average consensus algorithm.

In [7], one of the agents in a time-invariant multi-agent system interconnection graph is selected as a leader, and conditions are derived for such leadered graphs to be controllable by analysing the properties of the graph Laplacian matrix. The author proposes partitioning system interconnection graphs into leader controllable and uncontrollable classes to overcome the computational infeasibility of generating such graphs at present. Laplacian dynamics are then used to model system behavior for an iteratively-selected leader controllable topology for random initial states and desired end states.

The effect of connectivity and size on the cost of finding a leader controllable topology are also discussed, and it is shown that fully connected graphs are uncontrollable. Results are presented to show that it is computationally heavy to find leader controllable topologies for graphs with more than 13 nodes.

Liu et al. use the results of [7] and recursively manipulate the system dynamics to derive an expression for a discrete input sequence to take the system from a known initial state to a known final state in N steps for a graph with N+1 nodes as an auxiliary result in [8].

2. PROBLEM FORMULATION

2.0.1. Graph Theory.

In this section, we introduce some concepts of graph theory to realize the connections between the agents.

An undirected graph $G = (V, E)$ where V is the set of vertices (nodes) and E is the edge set where an edge is an unordered pair of distinct interconnected nodes. If $x, y \in V$ and $(x, y) \in E$, then x and y are said to be *neighbors* and are denoted as $x \sim y$. A graph where any two vertices are neighbors is called a *complete* graph. The number of neighbors of each vertex is known as it's *degree*. A graph is said to be *connected* if there is a path between any two vertices.

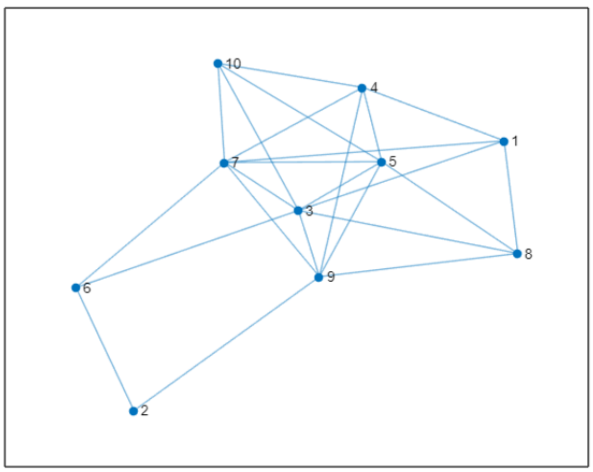

Figure 1. An undirected graph N = 10 nodes

The valency matrix is then defined as $\Delta(G)$ which is a diagonal N x N matrix whose i, i_{th} element is the degree of the i^{th} vertex.

The adjacency matrix $A(G)$ is defined as an N x N matrix whose $(i, j)^{\text{th}}$ element is -1 if $x \sim y$.

The Graph Laplacian $L(G)$ can be calculated as:

$$L = \Delta - A$$

Local interconnections are realized by introducing a sensing network $G_S = (V, E)$ where an edge is formed between x,y $\in V$ if and only if the distance between them is lesser than or equal to a specified distance R. In other words, R is the sensing radius of each agent to form local connections, and all it's neighbors formed are hence collectively referred to as the Neighbor Set N.

$$N_i = \{ j \mid (r_i - r_j) \leq R \}$$

For all purposes, it is assumed that all non-faulty nodes have a fixed sesning network. The communications network (defined by the Adjacency matrix A) is simulated with a fixed topology over all iterations as well as a switching topology where the Adjaceny matrix is semi-randomly generated during each step of the algorithm.

2.0.2. Leaderless System Model.

In order to simulate the network, we consider a group of N mobile agent with double-integrator dynamics, stated as follows:

$$\dot{r}_i = v_i$$
$$\dot{v}_i = u_i$$
$$\text{where } i = 1, 2, \ldots, N,$$

And r_i is the position of agent i, v_i it's velocity and u_i the control policy input given to agent i.

For a 2-Dimensional case, $r_i = (x_i, y_i)^T$, $v_i = (\dot{x}_i, \dot{y}_i)^T$ and $u_i = (u_x, u_y)^T$.

It is evident that u_i is the acceleration input to each agent. It consists of two parts, α_i and a_i:

$$u_i = \alpha_i + a_i$$

The first component controls the alignment of the velocity vectors and the second term avoids inter-agent collision and sets a desired distance between them.

2.0.3. Flocking Algorithm.

We first explain the need for an Artificial Potential Function (V) in order to maintain desired inter-agent distances and prevent collision.

The potential function V_{ij} is a non-negative, differential and radially unbounded function of distance between any two agents i and j, $\|r_{\text{ij}}\|$ and needs to satisfy two conditions:

1. $V_{ij}(\|r_{ij}\|) \to \infty$ as $\|r_{ij}\| \to 0$
2. V_{ij} attains it's unique minimum when agents i and j are at a particular distance.
3. The derivative of V_{ij} is zero for $\|r_{ij}\| >$ R

The above definitions make it evident that minimizing the potential of each agent provides cohesion and collision-avoidance. The total potential of an agent i, V_i is hence calculate as:

$$V_i = \sum_{j \in N_\mathrm{i}} V_{ij}(\|r_{ij}\|)$$

The inter agent Potential Function used by us is:

$$V_{\mathrm{ij}} = \begin{cases} -a_1\|r_{\mathrm{ij}}\| + log\|r_{\mathrm{ij}}\| + a_2/\|r_{\mathrm{ij}}\| & \|r_{\mathrm{ij}}\| \leq R \\ -a_1R + logR + a_2/R & \|r_{\mathrm{ij}}\| > R \end{cases}$$

Where a_1 and a_2 are selected to have the minimum at a desired distance. The control input to each agent u_i is hence given by:

$$u_i = - \sum_{i\sim j}(v_\mathrm{i} - v_\mathrm{j}) - \sum_{j \in N_i} \nabla_{\mathrm{r_i}} V_{\mathrm{ij}}$$

Where the gradient of V_{ij} is taken in the direction of the position vector of agent i, r_i.

The fact that the potential function V_{ij} is symmetric with respect to r_i and r_j can be utilized to compute the gradient of the function easily.

2.0.4. Leadered Systems.

Consider a graph with N nodes given by $G = (V, E)$ with a Laplacian given by $L = \Delta - A$. The dynamics of a system of N agents given by be described by this graph as:

$$\dot{x} = - \Delta^{-1/2}\mathrm{L}\Delta^{-1/2}x$$

We can choose any of the agents as a leader without loss of generality. Let the agent indexed by N be the leader for this system. We can rewrite the system dynamics as:

$$\begin{bmatrix} \dot{y} \\ \dot{z} \end{bmatrix} = - \begin{bmatrix} F & r \\ 0 & 0 \end{bmatrix} \begin{bmatrix} y \\ z \end{bmatrix} + \begin{bmatrix} 0 \\ u_N \end{bmatrix}$$

Where F is generated by deleting the last row and column of L, r is a vector containing the first $N-1$ elements of the deleted column, y is the stack vector of the first $N-1$ agents, z is the agent indexed by N, and u_N is the input generated by the leader agent acting as an input.

The system behavior is altered as follows:

1. Interconnections with the leader are now unidirectional, i.e., the leader is unaffected by the other agents
2. Follower agents are affected by the leader as well as other followers.

In [7], it is proved that for a system to be controllable by an agent acting as a leader, the following must hold true:

1. All eigenvalues of F must be unique
2. All the eigenvectors of F must be orthogonal to r

To generate controllable leadered topologies, it is also necessary that the graph be connected. The multiplicity of the zero eigenvalue of the Laplacian of a connected graph is one.

3. RESULTS AND SIMULATIONS

3.0.1. Leaderless Systems.

This section presents the results of the implementation of the above example on a group of N dynamical mobile agents. The number N is kept small for the purposes of clarity or presentation and CPU stability of Matlab.

For the case of leaderless networks with a fixed topology, the initial positions and velocities are initialized randomly for their x and y coordinates. The adjacency matrix A is randomly generated with [0, -1]. The system is then simulated over T (max. 1000) iterations and the results of the velocities and positions off the agents is shown in figures 2 and 3.

A point to note is that if the initial positions of the nodes are far away from each other and the interconnected graph is not $strong$ then it takes much longer for the algorithm to converge. In the case of switching networks, the algorithm can be shown to converge as long as there exists a *spanning tree* connecting the graph. The delayed convergence problem of fixed topology can be avoided if the adjency matrix generated in this case spans the entire network. These results are presented in figures 4 and 5.

For the case of leaderless networks with a switching topology, we implemented a piece of code that generates an Adjacency matrix A depending on the strength of the overall connection we wanted to simulate. The initial positions were initialized similar to the above case with fixed topology, and the algorithm presented was simulated over T (max. 1000) iterations. The inter-agent distances and velocity vectors converge when there exists a *spanning tree* connecting the graph. It noticably takes longer for convergence.

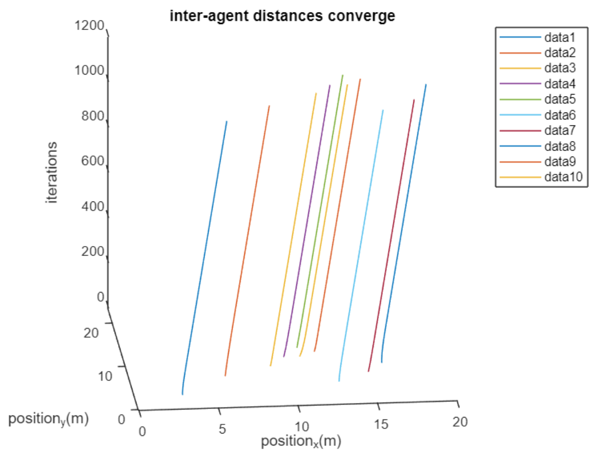

Figure 2. Fixed topology, N = 10 agents, Inter-agent distances converge

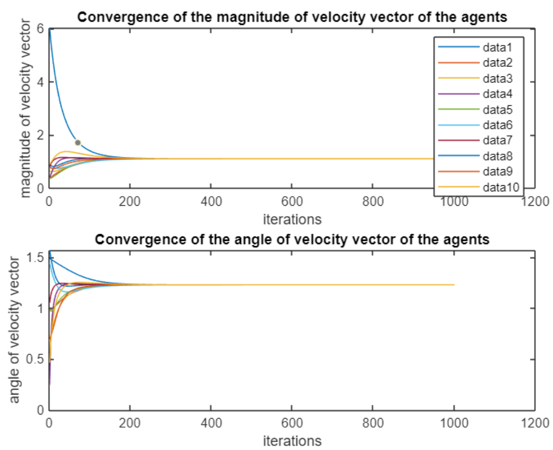

Figure 3. Fixed topology, N = 10 agents, Agent velocities converge

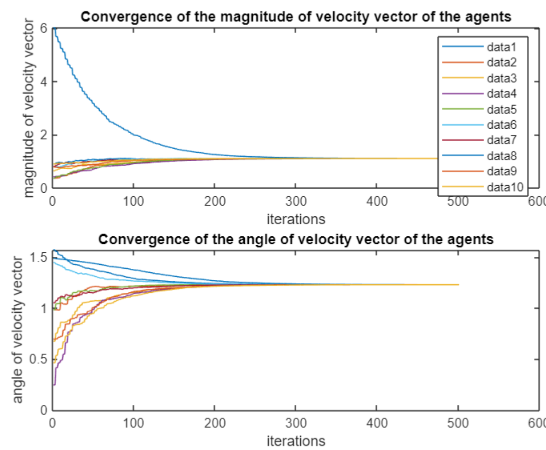

Figure 4. Switching topology, N = 10 agents, Inter-agent velocities converge

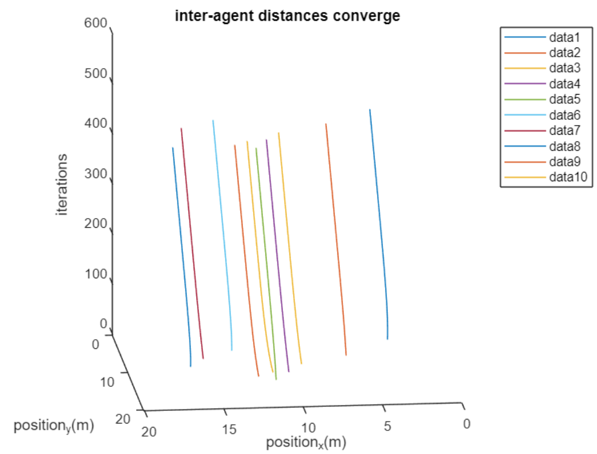

Figure 5. Switching topology, N = 10 agents, Inter-agent distances converge

3.0.2. Leadered Systems.

Using the results from above, a controllable graph is found by:

1. Generating a matrix of zeros and sparsely populating its upper triangle with -1 with some random probability

2. Adding the upper triangular matrix to its transpose to obtain adjacency matrix

3. Generating the degree matrix and Laplacian

4. Ensuring connectivity by checking the eigenvalues of the Laplacian

5. Deleting rows and corresponding columns to generate various values of F

6. Deleting rows and corresponding columns to generate various values of F

Controllable leader topologies were found using Matlab for graphs with 4 to 14 nodes, and some truncated results are presented below.

The first table gives the number of controllable topologies found at different probabilities over a million iterations for graphs with node size varying from 4 to 14. It was found that sparsity is a larger factor than size for leader controllability, and the range 0.3-0.5 is ideal for graphs of the sizes given below. For probabilities greater than 0.6, it is virtually impossible to reasonably find a controllable topology regardless of the size of the graph.

The second table gives the average time (in seconds) taken to find the first controllable topology

Probabilities	4	5	6	7	8	9	10	11	12	13	14
0.24459	5		6	12	6	6	6	11	8	9	8
0.59037	4		2	2	2	2	2	2	1	2	3
0.94043	0		0	0	0	0	0	0	0	0	0
0.69238	0		1	1	2	1	2	1	2	2	2
0.61859	3		1	2	1	2	1	1	3	2	1

Number of Nodes	p = 0.2	p = 0.3	p = 0.4
4	0.000649	0.004828	0.000109
5	0.000355	0.000766	0.000297
6	0.000072	0.000316	0.000059
7	0.00013	0.000509	0.00014

against number of nodes and probabilities of occupation (0.2, 0.3, and 0,4).

The third table gives the average time taken for converge to the mean of all agent states over a hundred iterations.

4. CONCLUSION AND FUTURE WORK

Such an algorithm is a commonly used base for expanding into more real-life situations like robot swarm control and other distributed consensus algorithms. One such modification includes modifying the potential function so as to include obstacle avoidance. Yuanguo Bi et.al. [5] implemented an obstacle avoidance flocking algorithm based on an external sensing radius of each agent called ”danger zone” and steering laws to avoid collisions. Xiaoyuan Luo in [6] implemented a variation of the potential function to include obstacle avoidance. Such technology is now being implemented in drone control in both leader-less situations and situations under leaders.

5. ACKNOWLEDGMENTS

The authors gratefully acknowledge Professor James Anderson for his support and guidance during our work on this project.

References

[1] Herbert G. Tanner, Ali Jadbabaie and George J. Pappas, ”Flocking in Fixed and Switching Networks” *IEEE Transactions on Automatic Control*, vol. 52 , issue 5 , May 2007.

[2] C. W. Reynolds, ”Flocks, herds, and schools: a distributed behavioral model” *ACM SIGGRAPH Computer*, vol. 21, no. 4, pp. 25–34, 1987.

[3] C. W. Reynolds, ”Steering behaviors for autonomous characters” *Proceedings of the of Game Developers Conference*, pp. 763–782, San Francisco, Calif, USA, 1999.

[4] C. W. Reynolds, “Interaction with a group of autonomous characters”, *Proceedings of the of Game Developers Conference*, pp. 449–460, San Francisco, Calif, USA, 2000.

[5] Jialiang Wang, Hai Zhao, Yuanguo Bi, ”An Improved Fast Flocking Algorithm with Obstacle Avoidance for Multiagent Dynamic Systems”, *J. Applied Mathematics 2014*, 2014.

[6] Xiaoyuan Luo, Dan Liu, ”Flocking and obstacle avoidance for multi-agent based on potential function”, *Proceedings of the 29th Chinese Control Conference*, 29-31 July 2010.

[7] H.G. Tanner, “On the controllability of Nearest Neighbor Interconnections”, *IEEE Conference on Decision and Control*, 2004.

[8] Bo Liu, Guangming Xie, Tianguang Chu, Long Wang, “Controllability of Interconnected Systems via Switching Networks with a Leader”, ıIEEE Conference on Systems, Man and Cybernetics, 2006.

Number of Nodes	Average Number of Iterations for Convergence
4	557.1000
5	480.3000
6	437.5000
7	404.0000
8	370.2000