2016

Leader-based Multi-Agent Systems

Daniel J. Loikits
Lehigh University

Leader-based Multi-Agent Systems

By

Daniel J. Loikits II

A Thesis

Presented to the Graduate and Research Committee

of Lehigh University

in Candidacy for the Degree of Master of Science

in

Electrical Engineering

Lehigh University

May 2016

Certificate of Approval

This thesis is accepted and approved in partial fulfillment of the requirements for the Master of Science.

____
Date

____

Nader Motee, Thesis Advisor

____

Filbert J. Bartoli, Co-Advisor

____

Filbert J. Bartoli, Chairperson of Department

Acknowledgements

I would like to thank my advisor, Professor Nader Motee, for the opportunity to write this thesis. I greatly appreciate that he gave me the honor of working in his research group and surrounding me with people much smarter than I. I also would thank my friends and family for supporting me throughout this endeavor; without them I would not have the fortitude to complete such an achievement. I would like to express my gratitude to my research group for motivating me to not settle for mediocrity. They are truly an inspiration and I am thankful for the opportunity to work with every one of them.

I would like to thank Donna B. Rovito for her guidance in the art of writing throughout my entire academic career, especially for this thesis. I would also like to thank my stepmother Cathy for helping to guide me through the process which eventually led me to attend graduate school at Lehigh; without her I would not be here. I want to express my sincere gratitude to my beautiful girlfriend Alana for her unwavering support and love.

To my parents, thank you for providing me with all the tools for success. I am eternally grateful for all that you have done for me; without you I would not be the man I am today. I love you and I dedicate this thesis to you.

Table of Contents

List of Figures

Abstract

In this thesis, we analyze multi-agent systems under the leader/follower control scheme. We take a graph-theoretic approach to defining the system which allows us to create a state-space representation of the agents. Using this model we can consider the group of agents as a linear time-invariant system under point mass dynamics. Linear control theory is used to examine the controllability of these systems. Uncontrollability from graph topology and symmetry is also explored. The process of electing both an optimal leader and set of optimal leaders to bring the agents to a consensus is investigated. Conditions of optimality require the leaders to minimize a cost function while simultaneously leading to a controllable network. At the end, we decompose the cost index in such a way as to show its relation to the underlying commutation graph of the network and the desired location of the agents. Finally, we demonstrate how varying the weights and the leader configuration affects the performance of the network.

Chapter 1: Background

1.1 Introduction

Over recent decades networked systems have dramatically increased in size [15]. Consequently, so has their computational complexity [16] and [17]. Problems such as scalability, controllability and energy management arose to the forefront, which required a new computing paradigm to solve [11], [14], [18], [19] and [21]. Hence, the multi-agent system emerged as a viable model to tackle these serious and important problems. A multi-agent system, in this context, is a computerized system that is comprised of multiple intelligent agents who are able to cooperate, communicate and exercise control over their behavior to reach a consensus. An agent can either be a human or autonomous robot. Other than computational efficiency and robustness, another advantage to using a multi-agent system is that it is a decentralized architecture, thereby providing immunity to the "single point of failure" problem.

Some of the many applications which utilize multi-agent systems include formation achievement and control of mobile robotics, traffic telematics and intelligent manufacturing systems (IMS) [20] and [23]. There exists three main methods to control a multi-agent system: Leader/follower, Virtual Structure and Behavioral. [2], [3] and [7] use the leader/follower approach where a single agent is selected as a leader while the remaining agents are designated as followers which move based on the leader's movements. The virtual structure method dictates that the agents attempt to maintain a semi-rigid geometric relationship with respect to each other and to a frame of reference [4]. Finally the behavioral control scheme focuses on goal oriented behaviors which is analogous to the flocking of birds and schooling of fish [5]. This thesis contains aspects of all three approaches but primarily deals with the leader/follower dynamic under which the follower agents execute a linear consensus algorithm [22].

Energy management plays a crucial role in the efficacy of multi-agent systems under the leader/follower control paradigm. An optimal leader or leaders must be chosen for this particular model in order to minimize the energy expelled by the agents. A leader is optimal if it leads to a controllable network and simultaneously minimizes the cost of reaching a target for all the other agents in the network [7]. The process of choosing an optimal leader or leaders is not simple and becomes more complicated as the size of the network increases. There are many ways to approach this optimization problem. The algebraic Riccati equation can be applied in order to find an optimal leader [8]. However, this method has a major drawback in that it becomes computationally impractical for large-scale networks. One can use matroid optimization to find the optimal leaders, but again, this is an iterative process which can fall victim to computational problems [14]. However, with use of submodular relaxation the matroid optimization framework can be computationally viable.

1.2 Algebraic Graph Theory

Algebraic graph theory serves as the mathematical tool to describe the interactions and information exchange between the agents. It is the mathematical foundation for analyzing the dynamics of multi-agent systems under leader/follow control. The purpose of this section is to present the preliminaries of algebraic graph theory as it pertains to these specific topics. For a more detailed and expansive review of this branch of mathematics refer to [9].

An undirected graph G consists of a node set N and an edge set E. An edge is an unordered pair of two distinct nodes in the graph G such that if $i, j \in N$ and $(i, j) \in E$ then i and j are neighbors denoted by $i \sim j$. For a weighted graph each edge is assigned a weight w_{ij}. The weights discussed in this thesis are non-negative numbers representing the strength of sensing between neighboring nodes.

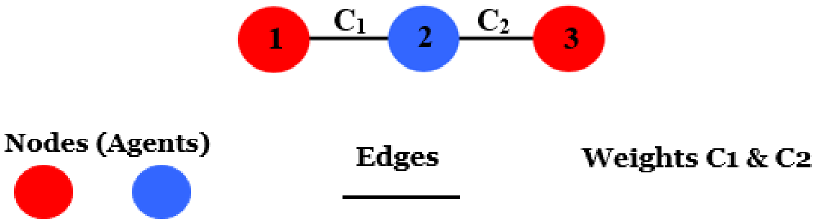


Figure 1: Depiction of nodes and edges

The degree or valency of a node is the number of neighbors it has. We define the valency matrix $\Delta(G)$ of a graph G as a diagonal matrix in which the (i,i)-entry is the valency of node i. The adjacency matrix $A(G)$ holds the information on the weights in a graph G. The adjacency matrix $A(G)$ is defined as

$$A(G)_{(i,j)} = \begin{cases} w_{ij} & (i,j) \in E \\ 0 & otherwise \end{cases}$$

where $w_{ij} \in \mathbb{R} :\ w_{ij} > 0$. This matrix is symmetric; therefore, it has the property $(i,j) = (j,i)$. A graph G is connected if there exists a path between any two nodes. The Laplacian of a graph is defined as

$$L(G)_{(i,j)} = \begin{cases} \sum\limits_{i \neq j} w_{ij} & i = j \\ -w_{ij} & (i,j) \in E \\ 0 & otherwise \end{cases}$$

The Laplacian has many special properties such as always being positive semi-definite, symmetric and where the multiplicity of its zero eigenvalue is equal to the number of connected components in the graph.

1.3 Common Graph Topologies

There exists around eight basic network topologies. I will present four of the most common ones. If the reader is interested they may find all of them in [9].

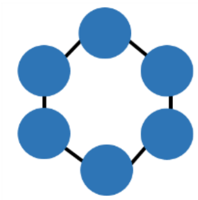


Figure 2: Ring Graph

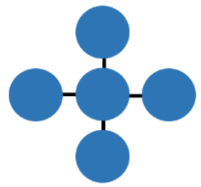


Figure 3: Star Graph

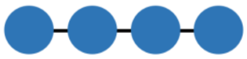


Figure 4: Path Graph

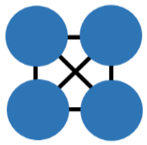


Figure 5: Complete Graph

1.4 General Dynamics of Agents

Each agent is modeled as a point mass with consensus dynamics defined as

$$\dot{x}_i = n_i$$

where

$$n_i = -\sum_{i\sim j} w_{ij}(x_i - x_j)$$

which can be rewritten as

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

where $L \in \mathbb{R}^{N\times N}$ is the Laplacian of the system and N is the number of agents. The effect of n_i is that neighboring agents will try to converge to each other's position. How fast they converge is dependent on the weight w_{ij} and the distance between them.

1.5 Dynamics of Agents under Leader/Follower Control

When an agent is chosen as a leader it no longer converges to other agents. More precisely, if agent α is chosen as a leader then

$$\dot{x}_\alpha = 0$$

now the system can be rewritten as

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

where $S \in \mathbb{R}^{N\times N}$ is the Laplacian matrix where the row associated with the leader is replaced with zeros. The control input u is given only to the leaders. Therefore, the group dynamics are written as

$$\dot{x} = -Sx + B_l u$$

where B_l is the input matrix defined as

$$B_{l(i,j)} = \begin{cases} 1 & i = j \; and \; node \; i \; is \; a \; leader \\ 0 & otherwise \end{cases}$$

Using this scheme the leaders can only manipulate the movements of the followers through movement alone. Hence, the leaders show the agents where to go rather than telling them directly.

1.6 Controllability Gramian

In control theory, the controllability Gramian W_c is used to determine whether a LTI (linear time-invariant) system is controllable or not. Given a LTI system with initial conditions equal to zero

$$\dot{x} = Ax + Bu$$

the controllability Gramian is

$$W_c = \int_0^{t_f} e^{A\tau} B B^T e^{A^T \tau} d\tau$$

The LTI system is controllable if and only if W_c is positive definite. In order to be positive definite W_c must satisfy

$$x^T W_c x > 0$$

If all the eigenvalues of A lie in the left-half plane then W_c is the unique solution to the Lyapunov equation

$$AW_c + W_c A^T = -BB^T$$

1.7 Minimum Energy Control

For a LTI system a control input u can be constructed such that it will take the system to a desired state x_f with a minimum expenditure of energy. With initial conditions set to zero, this control input u is defined as

$$u = B^T e^{A^T(t_f - t)} Wc^{-1} x_f$$

Note that W_c^{-1}only exists if and only if W_c is positive definite. Therefore, this input can only be constructed for controllable systems.

Chapter 2: Controllability of Leader Based Networks

2.1 Structural Controllability

Before the question of how to control a system is posed, we must ask if the system is able to be controlled. To find out the controllability of the Multi-Agent System we must analyze its graph topology along with the weights of the edges. The topology is dependent on the arrangement of agents and which of them are sensing each other. And the weight assignment is based upon how strong the sensing is between particular nodes.

A system is considered to be structurally controllable if there exists non-zero weights such that the system is able to be controlled [7]. The graph topology determines if there can exist a weighting scheme such that will lead to a structurally controllable system [7]. Consider the Leader-based system with fixed weights w_{ij}

$$\dot{x} = -Sx + B_l u$$

Lemma 1 from [7] states that this system is controllable if the controllability matrix K defined as

$$K = [B_l \;\; SB_l \;\; \dots \;\; S^{N-1}B_l]$$

is full rank where N is the number of agents in the network. This lemma is analogous to the controllability matrix found in linear system control theory [10] where a given LTI system

$$\dot{x} = Ax + Bu$$

is controllable if the matrix

$$[B \;\; AB \dots A^{N-1}B]$$

has an inverse which is equivalent to having full rank. Note that once the weighting is fixed the leader-based system is a LTI system. Therefore, additional conditions are necessary to determine structural controllability for leader-based multi-agent systems when weights are not fixed. Lemma 3 from [7] states that the pair (S, B_l) is structurally controllable if and only if it is neither reducible or can be written as

$$Q = \begin{pmatrix} Q_{11} \\ Q_{22} \end{pmatrix}$$

where $Q_{22} \in \mathbb{R}^{(N-p)\times N}$ and $Q_{11} \in \mathbb{R}^{p\times N}$ with at maximum $(p-1)$ nonzero entries and the rest of the columns are all zero. Reducible is defined in [11] as

$$L = \begin{bmatrix} L_{11} & 0 \\ L_{21} & L_{22} \end{bmatrix}, \qquad B_l = \begin{bmatrix} 0 \\ B_{l22} \end{bmatrix}$$

where $L_{11} \in \mathbb{R}^{p\times p}$, $L_{21} \in \mathbb{R}^{(N-p)\times p}$, $L_{22} \in \mathbb{R}^{(N-p)\times(N-p)}$ and $B_{l22} \in \mathbb{R}^{(N-p)\times N}$. If a system satisfies any of these two conditions, no weighting scheme can make the system controllable. As a consequence some topologies are impossible to control, no matter the weighting scheme, using leader/follower control. This section makes clear that the topology of the network affects the ability to control the system.

2.2 Popov-Hautus-Belevitch (PHB) Test

From linear systems theory, we can use the Popov–Hautus–Belevitch (PHB) test to determine controllability of our LTI system [10]. The system

$$\dot{x} = -Sx + B_l u$$

is uncontrollable if and only if there exists a left eigenvector v^T of S such that

$$v^T B_l = 0$$

Since S is a symmetric matrix, its left and right eigenvectors are the same. Thus, the condition transforms into if any eigenvector of $-S$ is orthogonal to B_l the system is uncontrollable. Note that this is equivalent to the controllability matrix

$$[B_l \;\; SB_l \;\; \dots \;\; S^{N-1}B_l]$$

being full rank since S can be eigen-decomposed into $V\Omega V^{-1}$ where V is defined as the matrix comprising the eigenvectors of S

$$V = [v_1 \quad v_2 \quad \dots \quad v_N]$$

and Ω is the diagonal matrix comprising the eigenvalues of S

$$\Omega = \begin{bmatrix} \lambda_1 & 0 & \dots & 0 \\ 0 & \lambda_2 & \dots & 0 \\ \vdots & 0 & \ddots & \vdots \\ 0 & 0 & \dots & \lambda_N \end{bmatrix}$$

Now using the fact that S is a real symmetric matrix $V\Omega V^{-1}$ can be re-written as $V\Omega V^T$ where V^T is

$$V^T = \begin{bmatrix} v_1^T \\ v_2^T \\ \vdots \\ v_N^T \end{bmatrix}$$

Thus, the controllability matrix can be written as

$$[B_l \;\; V\Omega V^T B_l \dots V\Omega^{N-1}V^T B_l]$$

where $V^T B_l$ is defined as

$$\begin{bmatrix} v_1^T \\ v_2^T \\ \vdots \\ v_N^T \end{bmatrix} B_l = \begin{bmatrix} v_1^T B_l \\ v_2^T B_l \\ \vdots \\ v_N^T B_l \end{bmatrix}$$

if one of the eigenvectors of S is orthogonal to B_l it is obvious that

$$[B_l \quad V\Omega V^T B_l \dots V\Omega^{N-1} V^T B_l]$$

cannot be full rank. Therefore the PHB test and the controllability matrix rank test are equivalent.

2.3 Uncontrollability from Leader Symmetry

Sometimes a graph topology might be controllable only if certain agents are selected as leaders. With these networks we need find conditions to test if an agent is a viable leader candidate. Propositon 5.8 from [12] states that the system

$$\dot{x} = -Sx + B_l u$$

is uncontrollable if it is leader symmetric. A system is leader symmetric if it can be written as

$$JS = SJ \quad and \quad JB_l = {B_l}^T J = B_l$$

where $J \in \mathbb{R}^{N \times N}$ is a $\{0,1\}$ non-identity permutation matrix with a single non-zero entry in each row and column [12]. Through eigendecomposition of S it can be proved this will lead to an uncontrollable system. From

$$JS = SJ$$

we can write S as

$$S = JSJ^{-1}$$

thereby we can find a right eigenvector v of S by

$$S(Jv) = \lambda(Jv)$$

Thus, Jv is an eigenvector of S corresponding to the eigenvalue λ. Since λ is distinct and all of S's eigenvectors are orthonormal to each other, there exists another right eigenvector of S defined $v - Jv$ [12]. Now using the PHB test we have

$$(v - Jv)^T B_l = v^T B_l - v^T J^T B_l = v^T B_l - v^T B_l = 0$$

which results in an uncontrollable system.

Note that all leader symmetric systems are uncontrollable. However, the inverse is not true. You can have an uncontrollable system that is leader asymmetric. Hence, Proposition 5.9 from [12] states that leader symmetry is not a necessary condition for the uncontrollability of a system. The proof can be found in [12] if the reader is interested.

2.4 Uncontrollability due to Graph Topology

There exists graph topologies that cannot be controlled by a single leader in an unweighted network, the complete graph being the most well-known among these special cases. Theorem IV.1 from [3] states that the system

$$\dot{x} = -Sx + B_l u$$

is controllable if and only if the eigenvalues of S are all distinct and the corresponding eigenvectors are not orthogonal to B_l. Now, take a complete graph G_c with L_c as its resulting Laplacian where L_c can be written as

$$L_c = NI_N - 1^T{}_N 1_N$$

where N is the number of agents, I_N is a N-dimenisonal identity matrix and 1_N is the column vector of ones. Now, no matter which agent is chosen as a leader the resultant S_c matrix will have spectra $\{\frac{1}{N}, 1^{(N-2)}\}$ [3]. As a consequence this would produce a multiplicity of the eigenvalue value 1 which violates Theorem IV.1 from [3], thus making the system uncontrollable. Note that if $N = 2$, the trivial case, this result does not hold meaning the system would be controllable.

Another uncontrollable network topology is the ring graph. Proposition 5.15 from [12] states that a ring graph, with one leader, is never controllable. This is due to the fact that no matter what agent is chosen as a leader the system

$$\dot{x} = -Sx + B_l u$$

is leader symmetric and thus uncontrollable. The proof is derived using Proposition 5.13 in [12] which states that the system

$$\dot{x} = -Sx + B_l u$$

is leader symmetric if and only if there is a nonidentity automorphism for the follower graph G_f such that the indicator function remains invariant under its action. An automorphism is a term that describes a mapping of a system to itself while simultaneously preserving its structure. The indicator function δ is a way to track which nodes are neighbors with the leader. Thus, the indicator function δ for this system is defined as the column vector

$$\delta(i) = \begin{cases} 1 & if\ node\ i \sim leader\ node \\ 0 & otherwise \end{cases}$$

Now consider a ring graph with N nodes. Without loss of generality, choose a node to be a leader and label it “1”. Then index the remaining follower nodes in a clockwise fashion. This results in an indicator function vector

$$\delta = [1,0, \dots 0,1]^T$$

since the leader is neighbors with node 2 and node N. Using the permutation

$$i \rightarrow N - i + 2 \ \text{ for } i = 2, \dots., N$$

is an automorphism of the follower graph G_f [12]. During this permutation, the leader is still connected to node 2 and N thus the indicator function is time-invariant. We can conclude that the system is leader symmetric and thus uncontrollable.

Chapter 3: Leader Selection Processes

3.1 Optimal Leader Election using Riccati Equation

The process of finding an optimal leader begins once the controllability of the network has been established and verified. Neglecting the trivial case in which only one node can become leader, an optimization problem is formulated to seek out the best leader in the group. When discussing the topic of optimality one must create an objective function to minimize or maximize and constraints to bound the problem in order have proper conditions of optimal. In [8] the objective function to be minimized is

$$J = \int_0^\infty (u^T R u + x^T Q x) dt$$

where u is the control input, R is a positive definite symmetric matrix, Q is a positive semi-definite symmetric matrix and x is the state of the system defined as

$$\dot{x} = -Sx + B_l u$$

[8] assumes the velocity of the agents to be unconstrained and the time allotted for the agents to obtain the desired states is infinite. From [13] the optimal control input u which minimizes J is

$$u = -R^{-1} B_l^T P x$$

where P is found by solving the algebraic Riccati equation

$$-S^T P - PS - P B_l R^{-1} B_l^T P + Q = 0$$

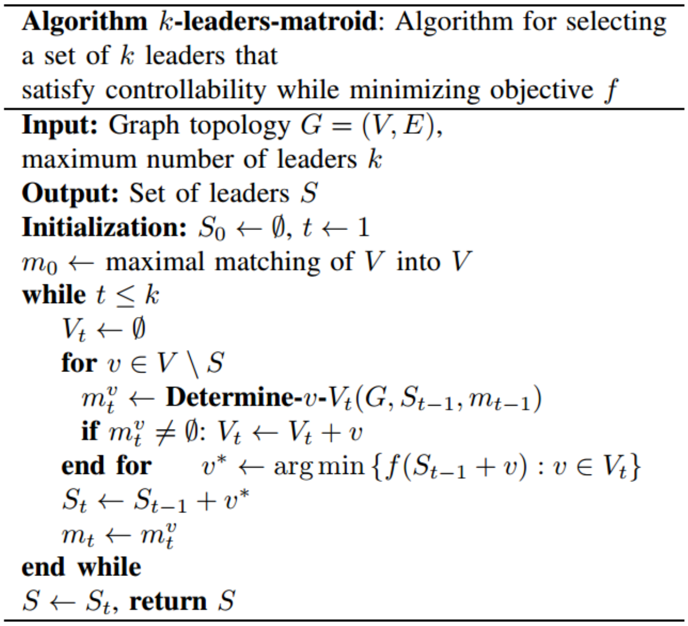
Any leader candidate who minimizes J is elected as the optimal leader. However, for a network with many nodes this process of election becomes computationally impractical. Thus, development of a method which utilizes the graph's properties to determine optimality might be more efficient. [8] poses that the leader node with highest degree, most neighbors, and closeness centrality, connected to the most important edges, is most cost-optimal. It should be noted that this conclusion was found using statistical methods.

3.2 Optimal Set of Leaders Election using Matroid Optimization Framework

In [14] they design their matroid optimization framework with the following criteria: the leader set S cannot exceed a fixed number k, the system should be controllable under the selected leaders and the leaders should minimize the supermodular objective function $f(S)$. Thus, the optimization problem is

$$\begin{aligned} & \underset{S}{minimize}\ f(S) \\ & subject\ to \quad |S| \leq k \\ & S \in C \end{aligned}$$

Where C is the set of leaders that lead to a controllable network. They pose a general algorithm which uses a graph theoretic approach to determine controllability at each iteration, and then check if they minimize the cost function when added to the set of leaders. If they do not minimize the function they are discarded and the algorithm moves to the next leader. This process continues until the set of leaders is exhausted or the maximum number of leaders k are elected. The details of [14] are not presented here due to the extensive amount of mathematical background needed to cover this topic. 4.2 acts as devil’s advocate to the optimization problem of leader selection using linear algebra methods.



Algorithm k-leaders-matroid: Algorithm for selecting a set of k leaders that
satisfy controllability while minimizing objective f

```
**Input:** Graph topology $G = (V, E)$,
maximum number of leaders $k$
**Output:** Set of leaders $S$
**Initialization:** $S_0 \leftarrow \emptyset$, $t \leftarrow 1$
$m_0 \leftarrow$ maximal matching of $V$ into $V$
while $t \leq k$
    $V_t \leftarrow \emptyset$
    for $v \in V \setminus S$
      $m_t^v \leftarrow$ Determine-$v$-$V_t(G, S_{t-1}, m_{t-1})$
      if $m_t^v \neq \emptyset$: $V_t \leftarrow V_t + v$
    end for    $v^* \leftarrow \arg\min \{f(S_{t-1} + v) : v \in V_t\}$
    $S_t \leftarrow S_{t-1} + v^*$
    $m_t \leftarrow m_t^v$
end while
$S \leftarrow S_t$, return $S$
```

Figure 6: Pseudocode for selecting a set of leaders [14]

3.3 Matroid Optimization with Submodular Relaxation

There are some cases in which a fixed number of leaders k cannot fully control a network. Thus, the aim is then to find the set of leaders S that can control most of the follower nodes. [14] introduces a term called the graph controllability index (GCI) which represents the fraction of nodes controlled by a particular set of leaders. It is defined as

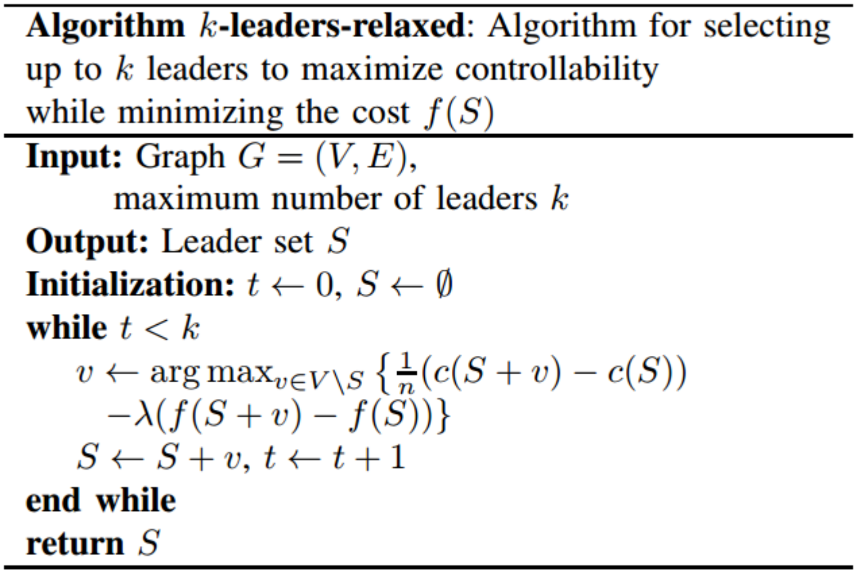
$$GCI = \frac{c(S)}{N}$$

where N is the number of nodes in the graph and $c(S)$ is the number of nodes that can be controlled by the set of leaders. Note that GCI is equal to 1 when the leader set can control all the nodes in the graph. Using the notation of 4.2 we can formulate this optimization problem as

$$\underset{S}{maximize}\ \frac{c(S)}{N} - \Omega f(S)$$

$$subject\ to \quad |S| \leq k$$

where Ω is a nonnegative constant. Under this formulation the goal is to find the set of leaders that controls the most nodes, maximizing controllably, while being subject to the penalty of the cost function. The severity of the cost penalty is tuned using Ω. Now if $\frac{c(S)}{N}$ is a submodular function of S and $\Omega f(S)$ is supermodular function of S then $\frac{c(S)}{N} - \Omega f(S)$ is submodular and as a result efficient algorithms can now be used to solve such a problem [14].



```
Algorithm $k$-leaders-relaxed: Algorithm for selecting
up to $k$ leaders to maximize controllability
while minimizing the cost $f(S)$
Input: Graph $G = (V, E)$,
        maximum number of leaders $k$
Output: Leader set $S$
Initialization: $t \leftarrow 0$, $S \leftarrow \emptyset$
while $t < k$
    $v \leftarrow \arg\max_{v \in V \setminus S} \{\frac{1}{n}(c(S+v) - c(S))$
      $-\lambda(f(S+v) - f(S))\}$
    $S \leftarrow S + v$, $t \leftarrow t + 1$
end while
return $S$
```

Figure 7: Pseudocode for selecting a set of leaders to maximize controllability [14]

Chapter 4: Results and Conclusions

4.1 Cost Index Minimization via Minimum Energy Control

Consider a controllable system of the form

$$\dot{x} = -Sx + B_l u$$

where initial conditions are equal to zero and given control input u

$$u = {B_l}^T e^{-S^T(t_f - t)} Wc^{-1} x_f$$

where x_f is the desired state of the agents. With the cost index J

$$J = \int_0^\infty u^T u \, dt = {x_f}^T W_c^{-1} x_f$$

we can relate J to the eigenvalues and eigenvectors of the Gramian matrix. Since the system is controllable the Gramian is full rank and spans $\mathbb{R}^N$. Thus, the desired state of the agents can be written as a weighted linear sum of the Gramian's eigenvectors.

$$x_f = \alpha_1 v_1 + \alpha_2 v_2 + \cdots + \alpha_N v_N$$

where α_i are constants and v_i are the eigenvectors of the Gramian matrix. Now, J can be rewritten as

$$J = \int_0^\infty u^T u \, dt = {x_f}^T W_c^{-1} x_f = \frac{{\alpha_1}^2}{\lambda_1} + \frac{{\alpha_2}^2}{\lambda_2} + \cdots + \frac{{\alpha_N}^2}{\lambda_N}$$

where λ_i are the eigenvalues of the Gramian. From here it is clear that a x_f which lies in the direction of the eigenvector corresponding with the largest eigenvalue minimizes J. The eigenvalues of the Gramian can be altered by changing the weights on the edges connecting the nodes. Also, the eigenvalues can be affected by which agents are leaders. Thus, from here we have two ways to minimize J given a desired state x_f.

4.2 Simulation Results from varying Leaders

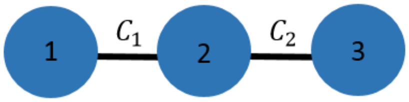


Figure 8: Path graph of three nodes

Without loss of generality, we can simulate in the 2-D case where the agents' x and y positions are controlled by the same dynamics. Thus, the system has the form

$$\dot{x} = -Sx + B_l u_x$$

$$\dot{y} = -Sy + B_l u_y$$

Case 1: Consider Figure 8 with node 1 being selected as the leader and assume weights $C_1 = C_2 = 1$. Therefore the resultant S and B_l matrices will be

$$S = \begin{bmatrix} 0 & 0 & 0 \\ -1 & 2 & -1 \\ 0 & -1 & 1 \end{bmatrix}$$

$$B_l = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

Now take x_f and y_f to be

$$x_f = \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix} \qquad y_f = \begin{bmatrix} 5 \\ 5 \\ 5 \end{bmatrix}$$

With control input u respectively

$$u_x = {B_l}^T e^{-S^T(t_f - t)} Wc^{-1} x_f \qquad u_y = {B_l}^T e^{-S^T(t_f - t)} Wc^{-1} y_f$$

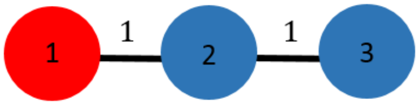


Figure 9: Path graph of three nodes with node 1 selected as a leader

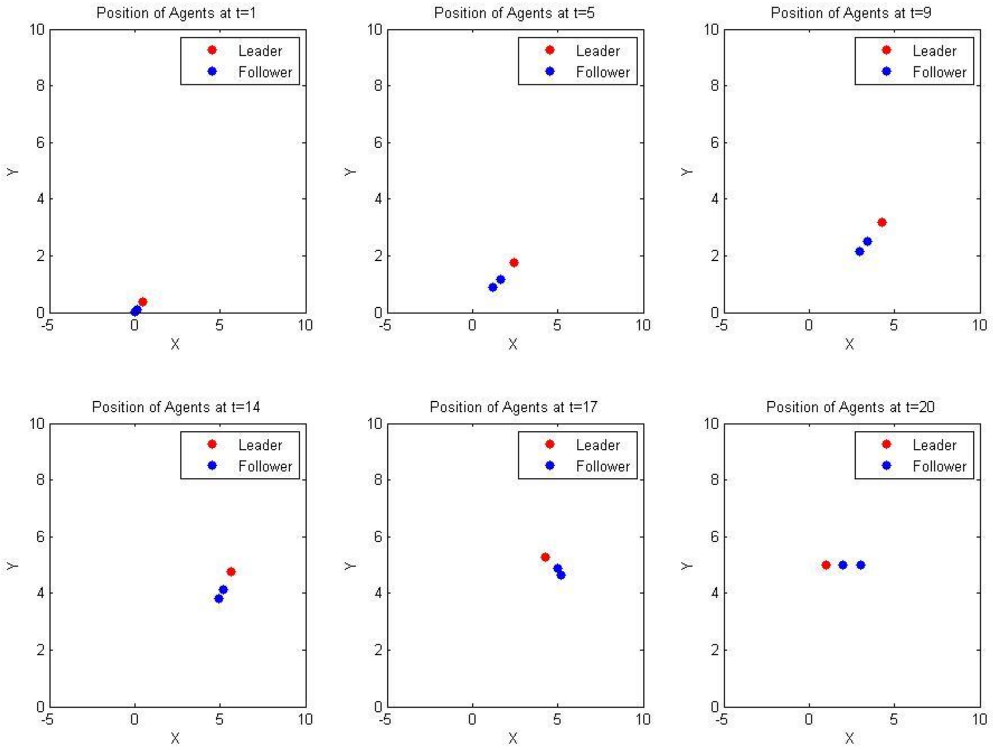


Figure 10: Position of agents at specific time intervals for Case 1

17.2575.

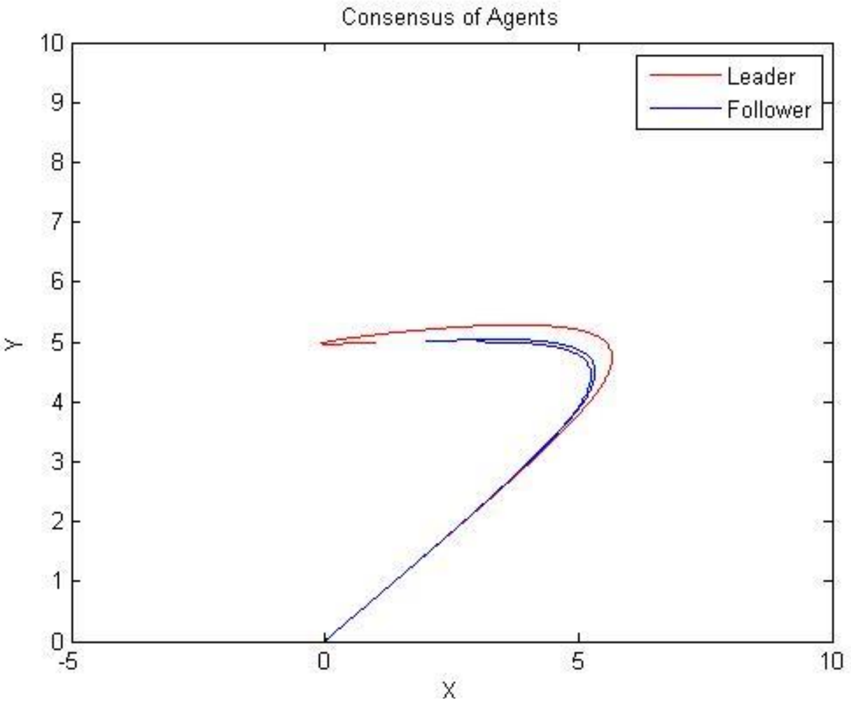


Figure 11: Trajectory of agents for Case 1

The eigenvalues of the Gramian are

$$\lambda = \begin{bmatrix} 51.7427 \\ 0.7511 \\ 0.0100 \end{bmatrix}$$

and the α_i constants that relate the desired x and y locations to the eigenvectors of the Gramian are

$$\alpha_x = \begin{bmatrix} -3.3899 \\ 1.5451 \\ 0.3482 \end{bmatrix} \quad \alpha_y = \begin{bmatrix} -8.6494 \\ 0.4330 \\ 0.0300 \end{bmatrix}$$

Note that in Figure 10 and Figure 11, the leader node needs to swing a considerable distance from its optimal trajectory (straight-line) to put the followers in their desired locations. This consequently requires the leader to expend excess energy. The cost index for this case is $J_1 = 17.2575$.

Case 2: Consider Figure 8 with nodes 1 and 3 being selected as the leaders and assume weights $C_1 = C_2 = 1$. Therefore the resultant S and B_l matrices will be

$$S = \begin{bmatrix} 0 & 0 & 0 \\ -1 & 2 & -1 \\ 0 & 0 & 0 \end{bmatrix}$$

$$B_l = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Now take x_f and y_f to be

$$x_f = \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix} \qquad y_f = \begin{bmatrix} 5 \\ 5 \\ 5 \end{bmatrix}$$

With control input u respectively

$$u_x = B_l{}^T e^{-S^T(t_f - t)} Wc^{-1} x_f \qquad u_y = B_l{}^T e^{-S^T(t_f - t)} Wc^{-1} y_f$$

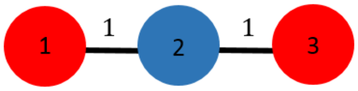


Figure 12: Path graph of three nodes with node 1 and 3 selected as leaders

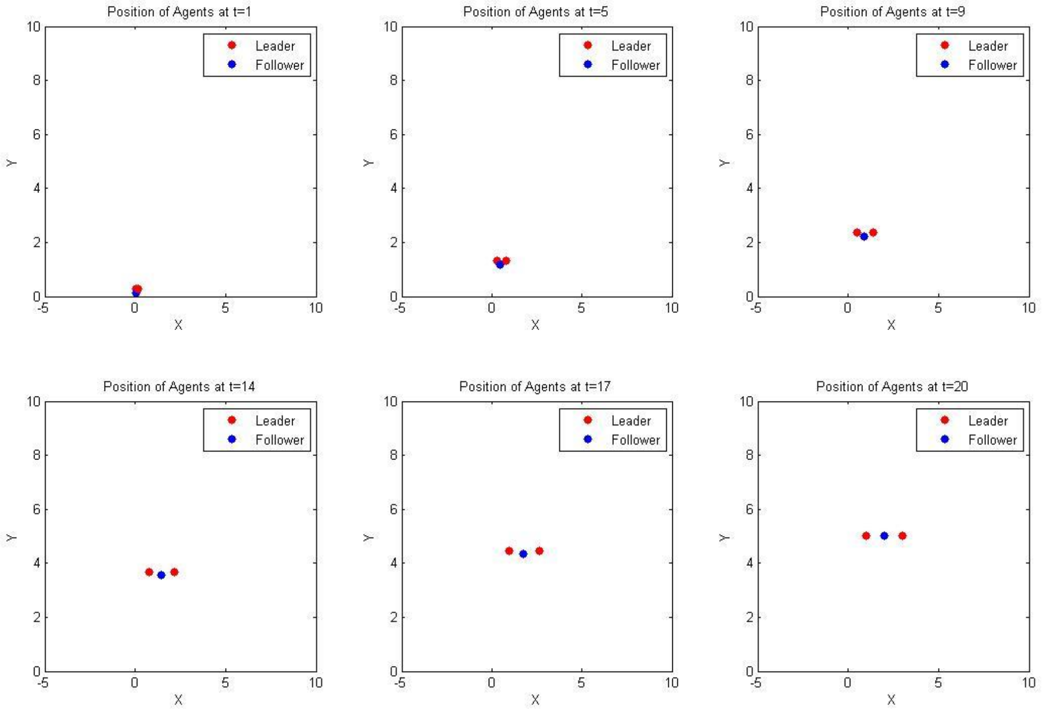


Figure 13: Position of agents at specific time intervals for Case 2

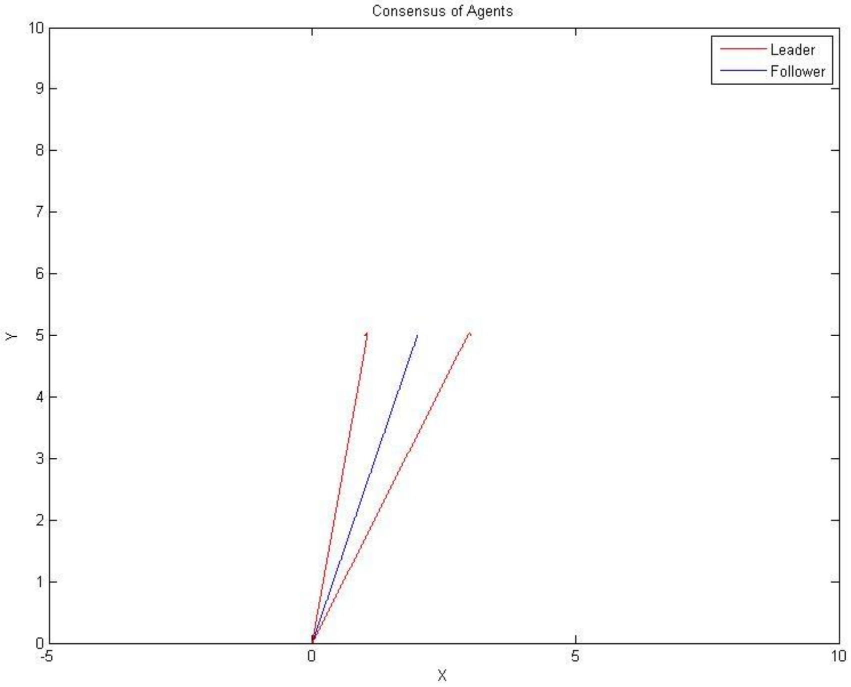


Figure 14: Trajectory of agents for Case 2

The eigenvalues of the Gramian are

$$\lambda = \begin{bmatrix} 0.0805 \\ 20.00 \\ 29.5442 \end{bmatrix}$$

And the α_i constants that relate the desired x and y locations to the eigenvectors of the Gramian are

$$\alpha_x = \begin{bmatrix} -0.0347 \\ 1.4142 \\ 3.4639 \end{bmatrix} \quad \alpha_y = \begin{bmatrix} -0.0867 \\ 0 \\ 8.6598 \end{bmatrix}$$

Note that in Figure 13 and Figure 14, the leaders and followers follow an optimal trajectory. Comparing with Figure 10 and Figure 11 it seems that the two leader case allows the agents to get to the desired location better than the single leader case. This observation is proven by looking at the cost index for this case which is $J_2 = 3.1526$.

The α_i constants for case 2 are of lesser value than in case 1. Also, the numerical eigenvalues of the Gramian Matrix are distributed more uniformly across the eigenvectors than in case 1. These two facts are the reasons why the cost index J_2 is much less than J_1. We can then conclude that having two leaders placed at the ends of the graph will lead to a more optimal system when it comes to minimizing energy of the leaders.

4.3 Simulation Results from varying Weights

Case 1: Consider Figure 8 with node 3 being selected as the leader and assume weights $C_1 = C_2 = 1$. Therefore the resultant S and B_l matrices will be

$$S = \begin{bmatrix} 1 & -1 & 0 \\ -1 & 2 & -1 \\ 0 & 0 & 0 \end{bmatrix}$$

$$B_l = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Now take x_f and y_f to be

$$x_f = \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix} \qquad y_f = \begin{bmatrix} 5 \\ 5 \\ 5 \end{bmatrix}$$

With control input u respectively

$$u_x = {B_l}^T e^{-S^T(t_f - t)} Wc^{-1} x_f \qquad u_y = {B_l}^T e^{-S^T(t_f - t)} Wc^{-1} y_f$$

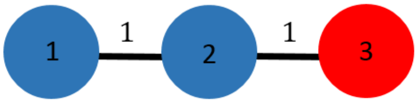


Figure 15: Path graph of three nodes with node 3 selected as a leader

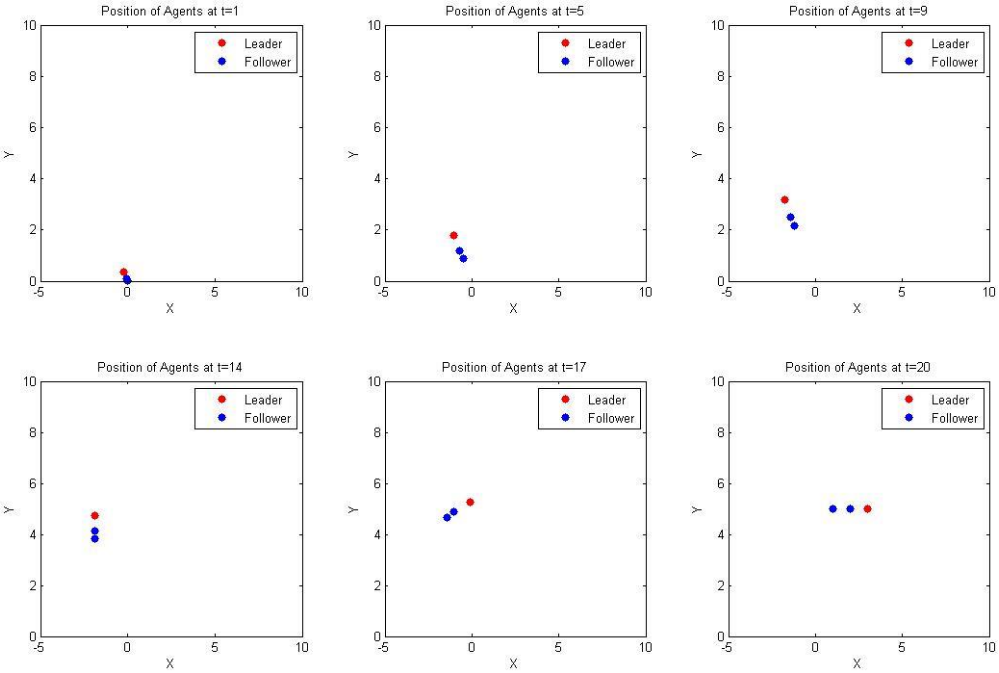


Figure 16: Position of agents at specific time intervals for Case 1

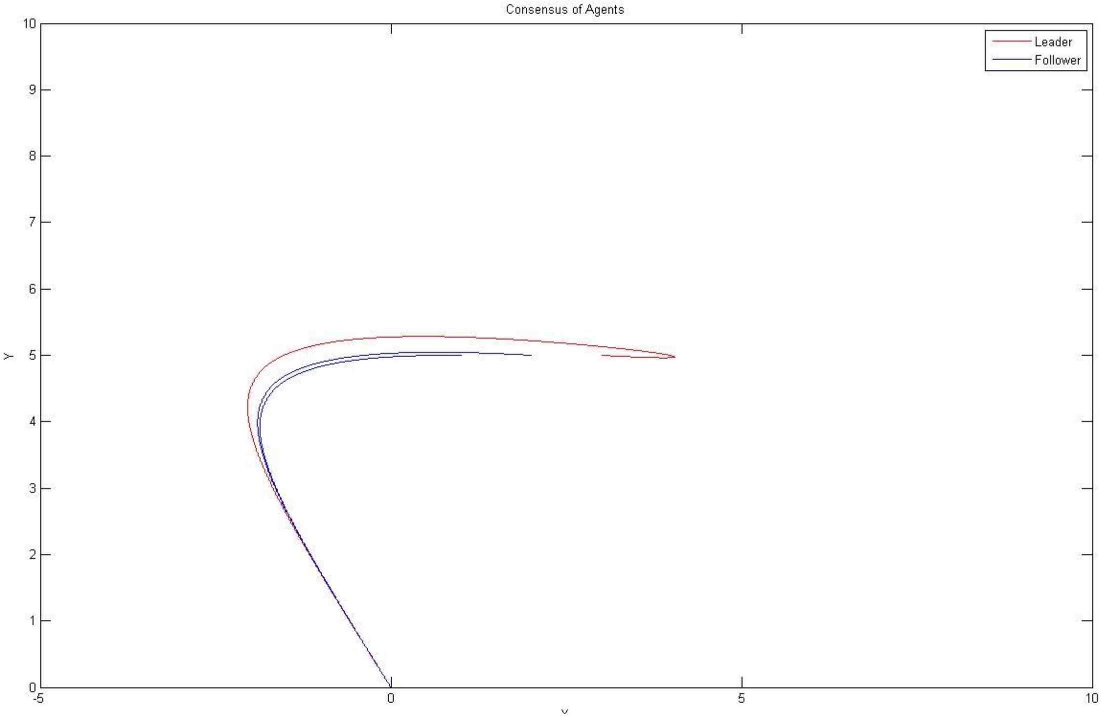


Figure 17: Trajectory of agents for Case 1

The eigenvalues of the Gramian are

$$\lambda = \begin{bmatrix} 51.7427 \\ 0.7511 \\ 0.0100 \end{bmatrix}$$

And the α_i constants that relate the desired x and y locations to the eigenvectors of the Gramian are

$$\alpha_x = \begin{bmatrix} -3.5296 \\ 1.1987 \\ -0.3242 \end{bmatrix} \quad \alpha_y = \begin{bmatrix} -8.6494 \\ -0.4330 \\ 0.0300 \end{bmatrix}$$

Note the similarities between case 1 of section 4.2 and case 1 of section 4.3. As such the cost index for this case is $J_1 = 14.4062$.

Case 2: Now assume weights $C_1 = C_2 = 5$. Therefore the resultant S and B_l matrices will be

$$S = \begin{bmatrix} 5 & -5 & 0 \\ -5 & 10 & -5 \\ 0 & 0 & 0 \end{bmatrix}$$

$$B_l = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Now take x_f and y_f to be

$$x_f = \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix} \qquad y_f = \begin{bmatrix} 5 \\ 5 \\ 5 \end{bmatrix}$$

With control input u respectively

$$u_x = {B_l}^T e^{-S^T(t_f - t)} Wc^{-1} x_f \qquad u_y = {B_l}^T e^{-S^T(t_f - t)} Wc^{-1} y_f$$

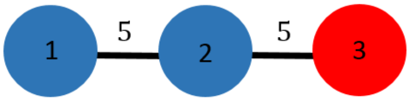


Figure 18: Path graph of three nodes with node 3 selected as a leader

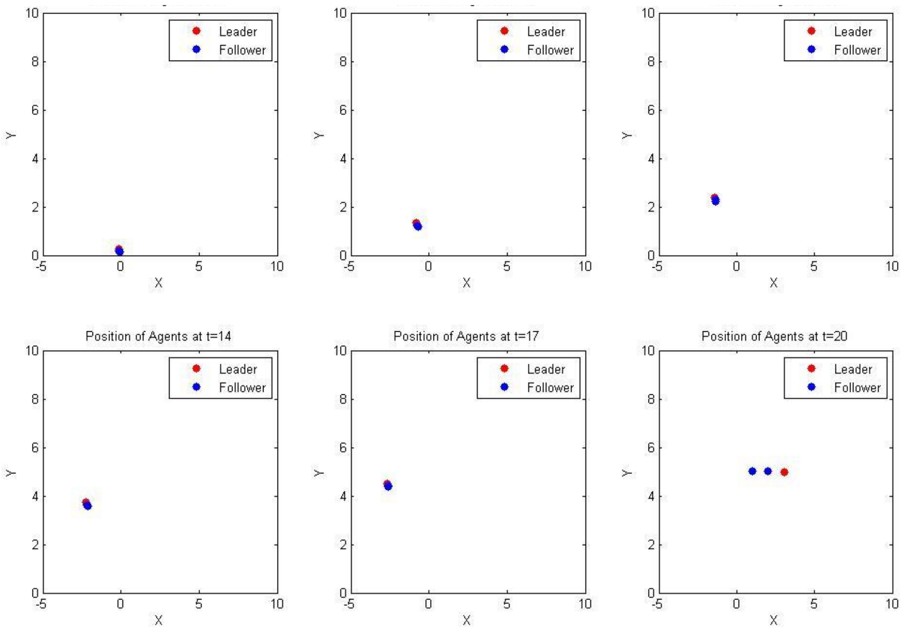


Figure 19: Position of agents at specific time intervals for Case 2

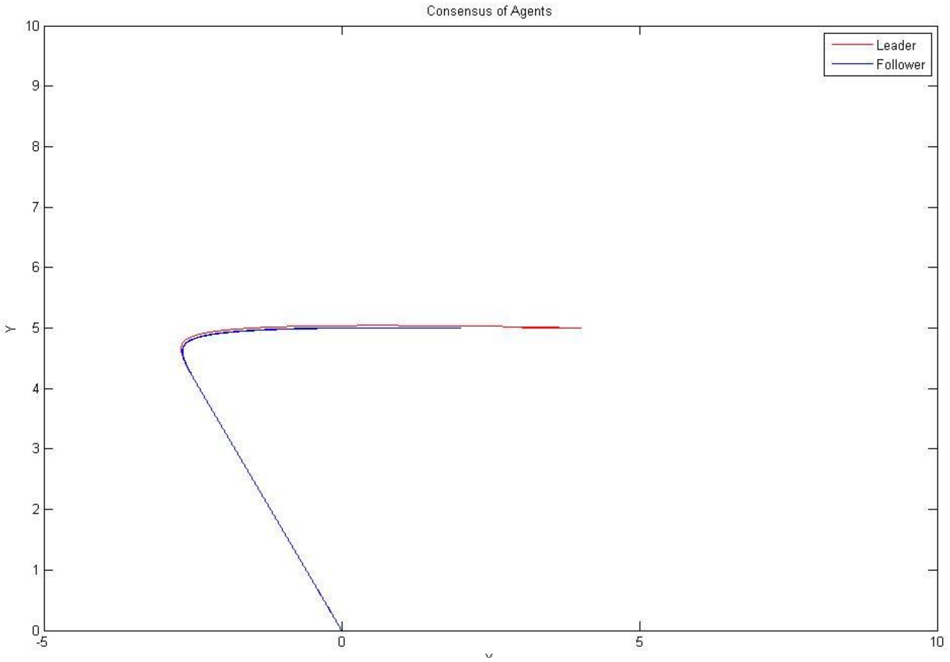


Figure 20: Trajectory of agents for Case 2

The eigenvalues of the Gramian are

$$\lambda = \begin{bmatrix} 58.3254 \\ 0.1715 \\ 0.0021 \end{bmatrix}$$

And the α_i constants that relate the desired x and y locations to the eigenvectors of the Gramian are

$$\alpha_x = \begin{bmatrix} -3.4762 \\ 1.3459 \\ -0.3231 \end{bmatrix} \quad \alpha_y = \begin{bmatrix} -8.6599 \\ -0.0760 \\ 0.0047 \end{bmatrix}$$

The cost index for this case is $J_2 = 61.3575$ which is significantly higher than case 1. Therefore, increasing the weights consequently increases the cost index for these two cases. As observed, by increasing the weights the α_i increased and the second and third eigenvalues of the Gramian dramatically decreased, thereby causing the cost index to increase. This is a sub-optimal leader configuration, as proven in section 4.2. Thus, it makes intuitive sense that increasing the weights has a negative effect on the performance of the network.

Case 3: Consider Figure 8 with node 1 and 3 being selected as the leaders and assume weights $C_1 = C_2 = 5$. Therefore the resultant S and B_l matrices will be

$$S = \begin{bmatrix} 0 & 0 & 0 \\ -5 & 10 & -5 \\ 0 & 0 & 0 \end{bmatrix}$$

$$B_l = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Now take x_f and y_f to be

$$x_f = \begin{bmatrix}1\\2\\3\end{bmatrix} \qquad y_f = \begin{bmatrix}5\\5\\5\end{bmatrix}$$

With control input u respectively

$$u_x = {B_l}^T e^{-S^T(t_f-t)} Wc^{-1} x_f \qquad u_y = {B_l}^T e^{-S^T(t_f-t)} Wc^{-1} y_f$$

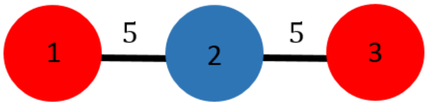


Figure 21: Path graph of three nodes with node 1 and 3 selected as leaders

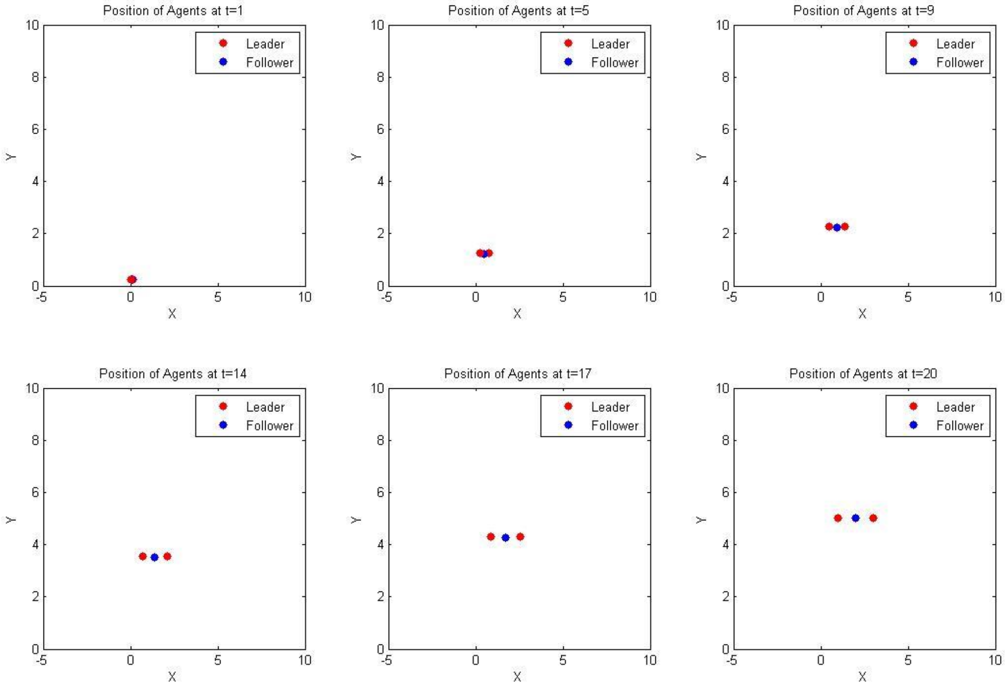


Figure 22: Position of agents at specific time intervals for Case 3

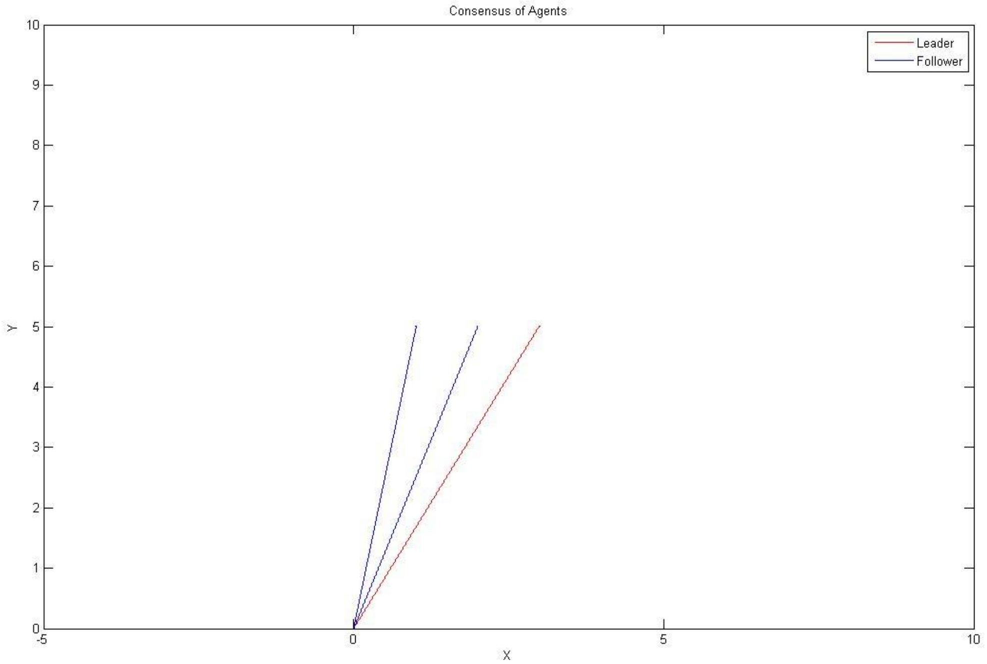


Figure 23: Trajectory of agents for Case 3

The eigenvalues of the Gramian are

$$\lambda = \begin{bmatrix} 0.0167 \\ 20.00 \\ 29.9080 \end{bmatrix}$$

And the α_i constants that relate the desired x and y locations to the eigenvectors of the Gramian are

$$\alpha_x = \begin{bmatrix} -0.0069 \\ 1.4142 \\ 3.4641 \end{bmatrix} \quad \alpha_y = \begin{bmatrix} -0.0171 \\ 0 \\ 8.6602 \end{bmatrix}$$

The cost index for this case is $J_3 = 3.0293$. So here increasing the weights reduced the cost index because the α_i were reduced when the weights were increased. This is an optimal leader configuration, as proven in section 4.2. Thus, it also makes intuitive sense that increasing the weights in this case has a positive effect on the performance of the network. Thus, increasing weights in a graph controlled by a sub-optimal configuration leads to a higher cost index. On the other hand, increasing the weights of a graph controlled by an optimal leader configuration leads to a lower cost index.

4.4 Closing Remarks

Numerous methods exist on determining controllability of leader based systems due to the fact that these systems can be transformed or viewed as LTI systems which are extensively studied in control theory. Consequently, finding conditions of controllability from a linear algebra point of view are vast and more work should be done to approach controllability from other branches of mathematics such as graph theory. Using a graphic theoretic approach would be more scalable from a computation standpoint and will provide more insight on how the structure of a network affects its ability to be controlled, which would be extremely useful in future research.

To our knowledge, leader selection processes mainly exist in mathematical contexts which do not account for various dynamics inherent in the systems with which most engineers in the field of robotics deal. Thus, additional work on making connections between these processes and practical engineering applications would be beneficial. It would also be interesting to see additional research considering dynamics such as non-linear and time varying and how those particular dynamics change optimality conditions on leaders. Likewise, it would be valuable to study leaders with time varying weights so that they are only leading for a finite time and then switching the responsibility to other nodes, i.e., switching topologies. It would also be valuable to create performance measures on the effects of removing follower nodes from the graph given a fixed set of leaders.

There is an obvious link between the number of nodes in the graph and the number of leaders needed to control the network. Also, there is a relation between the underlying communication graph and who can be chosen as leaders. In some cases, particular graphs cannot be controlled due to who the nodes communicate with. Thus, future work on reconstructing the communication graph to an optimal form to satisfy controllability standards and to meet performance measures might be of interest, as well as formulation of a semi-definite program (SDP) that solves the optimal weights for a graph in order to reduce the leader's energy. Also, it would be instructional to conduct more research on how changing weights affects the Gramian's eigenvalue and eigenvectors.

It is interesting to note that increasing the weights for graphs controlled by a sub-optimal leader configuration causes the leaders to expel more excess energy while on the other hand, increasing the weights for graphs under optimal leader control causes the leaders to expel less energy. Future work could also be done on how to develop trade off curves that plot the diminishing returns of increasing the weights to increase performance. It is obvious that the more leaders which exist in a network, the more controllable the nodes will be. However, there exists diminishing returns on adding leaders to the group. Finally, development of an optimization problem that determines the optimal number of leaders to have in a network under certain performance and energy criterion should be investigated.

Bibliography

[1] Alonso, Eduardo. "From Artificial Intelligence to Multi-Agent Systems: Some Historical and Computational Remarks." Artificial Intelligence Review 21.1 (1998): 3-24.

[2] Lozano, Rogelio, et al. "Controllability and observability of leader-based multi-agent systems." Decision and Control, 2008. CDC 2008. 47th IEEE Conference on. IEEE, 2008.

[3] Tanner, Herbert G. "On the controllability of nearest neighbor interconnections." Decision and Control, 2004. CDC. 43rd IEEE Conference on. Vol. 3. IEEE, 2004.

[4] Lewis, M. Anthony, and Kar-Han Tan. "High precision formation control of mobile robots using virtual structures." Autonomous Robots 4.4 (1997): 387-403.

[5] Balch, Tucker, and Ronald C. Arkin. "Behavior-based formation control for multirobot teams." Robotics and Automation, IEEE Transactions on 14.6 (1998): 926-939.

[6] Oliveira, Eugenio, Klaus Fischer, and Olga Stepankova. "Multi-agent systems: which research for which applications." Robotics and Autonomous Systems 27.1 (1999): 91-106.

[7] Zamani, Mohsen, and Hai Lin. "Structural controllability of multi-agent systems." American Control Conference, 2009. ACC'09.. IEEE, 2009.

[8] Borsche, Theodor, and Sid Ahmed Attia. "On leader election in multi-agent control systems." Control and Decision Conference (CCDC), 2010 Chinese. IEEE, 2010.

[9] Godsil, Chris, and Gordon F. Royle. Algebraic graph theory. Vol. 207. Springer Science & Business Media, 2013.

[10] Phillips, Charles L., and Royce D. Habor. Feedback control systems. Simon & Schuster, 1995.

[11] Lin, Ching Tai. "Structural controllability." Automatic Control, IEEE Transactions on 19.3 (1974): 201-208.

[12] Rahmani, Amirreza, et al. "Controllability of multi-agent systems from a graph-theoretic perspective." SIAM Journal on Control and Optimization 48.1 (2009): 162-186.

[13] Kwakernaak, Huibert, and Raphael Sivan. Linear optimal control systems. Vol. 1. New York: Wiley-interscience, 1972.

[14] Clark, Andrew, Linda Bushnell, and Radha Poovendran. "On leader selection for performance and controllability in multi-agent systems." Decision and Control (CDC), 2012 IEEE 51st Annual Conference on. IEEE, 2012.

[15] Zhang, Jiang, et al. "Scaling behaviours in the growth of networked systems and their geometric origins." Scientific reports 5 (2015).

[16] Olfati-Saber, Reza, Alex Fax, and Richard M. Murray. "Consensus and cooperation in networked multi-agent systems." Proceedings of the IEEE 95.1 (2007): 215-233.

[17] Olfati-Saber, Reza. "Flocking for multi-agent dynamic systems: Algorithms and theory." Automatic Control, IEEE Transactions on 51.3 (2006): 401-420.

[18] Zampieri, Sandro. "Trends in networked control systems." 17th IFAC World Congress. 2008.

[19] Wijngaards, Niek JE, et al. "Supporting internet-scale multi-agent systems." Data & Knowledge Engineering 41.2 (2002): 229-245.

[20] Cao, Y. Uny, Alex S. Fukunaga, and Andrew Kahng. "Cooperative mobile robotics: Antecedents and directions." Autonomous robots 4.1 (1997): 7-27.

[21] Siami, Milad, and Nader Motee. "Systemic measures for performance and robustness of large-scale interconnected dynamical networks." Decision and Control (CDC), 2014 IEEE 53rd Annual Conference on. IEEE, 2014.

[22] Bolouki, Sadegh, et al. "A geometric approach towards linear consensus algorithms." Decision and Control (CDC), 2014 IEEE 53rd Annual Conference on. IEEE, 2014.

[23] Siami, Milad, and Nader Motee. "Fundamental limits on robustness measures in networks of interconnected systems." Decision and Control (CDC), 2013 IEEE 52nd Annual Conference on. IEEE, 2013.

Vita

Daniel J. Loikits II was born to Diana and Daniel Loikits in Allentown, Pa on November 21, 1991. He graduated from Allentown Central Catholic High School and then went on to study at The Pennsylvania State University. He completed a Bachelor of Science in Electrical Engineering in May of 2014. He then continued his academic career at Lehigh University and will graduate in May of 2016 with a Master of Science in Electrical Engineering.