Methods Toward the Design of Estimation and Control for Networked Multiagent Systems

Anthony J. Savas

A Dissertation
Presented to the Faculty
of Princeton University
in Candidacy for the Degree
of Doctor of Philosophy

Recommended for Acceptance
by the Department of
Mechanical and Aerospace Engineering

Advisers: Naomi Ehrich Leonard, H. Vincent Poor

September 2022

Abstract

This dissertation explores distributed estimation and control of networked multiagent systems. We consider collaborative systems in which agents access information from their neighbors and use this information to better understand and manipulate the environment.

We consider distributed filtering of a scalar linear stochastic process over a networked multiagent system and focus on the setting where communication between agents is corrupted by Gaussian noise. We show that communication noise is not easily handled by a two-stage consensus filter in the literature and propose a novel algorithm which does not suffer performance degradation under such communication noise. We discuss how to optimally tune two fixed gains to minimize the asymptotic error covariance of the filter.

We consider designing a network of agents to perform distributed estimation *and* control of linear time-invariant systems. We develop a framework in which agents have the flexibility to change their feedback control parameters without needing to redesign their estimation strategies. We use the small-gain theorem and the bounded real lemma to characterize conditions under which this is possible using linear matrix inequalities. We show how linear consensus dynamics can be used to further extend the operating regime of this framework.

Leveraging our distributed estimation and control framework, we develop a methodology to distinguish agents for the purpose of actuator and sensor selection. We use hypothesis testing to rigorously compare a set of centrality measures as selection tools, focusing on the actuator selection problem. We show that under our framework there is strong statistical evidence that, given sufficient edge density, betweenness centrality is a good metric for actuator selection over Erdos-Renyi random graphs in terms of minimizing a key matrix norm. We further show that these results broadly extend to other graph generation methods and are robust to actuator failure as well as network scale. When considering scenarios in which individual agents do not have full controllabilty of the system, we show that there is strong statistical evidence that degree centrality, rather than betweenness centrality, is a good selection heuristic for actuator placement.

Acknowledgements

My time in graduate school has been filled with intellectual stimulation as well as personal growth. I not only was able to learn a great deal about the subject matter within my field of study, but I also was able to learn a great deal about myself. I am extremely grateful to have been afforded the opportunities to explore and grow both professionally as well as personally. Of course, this would not have been possible without the help of many colleagues and friends.

I would first like to thank my advisors Naomi Leonard and H. Vincent Poor. Naomi's passion for research is incredibly inspiring, and this passion is equally matched by her kindness. Naomi has shown me endless patience and understanding, and for that I am forever grateful. She has pushed me to be a better researcher and collaborator, and has instilled in me values which I will take and nurture throughout the rest of my career and beyond. Likewise, Vince has provided me with irreplaceable feedback on my research. In meetings, Vince would always ask very deep questions that caused me to really reflect on aspects of our work together and ultimately strengthened the research to a great degree. I felt like Vince's metaphorical door was always open, and no matter how many directions he was pulled in at any given moment he would without fail provide feedback or get back to me almost instantaneously. Thank you, Naomi and Vince, for all that you have provided to me over my time at Princeton.

Throughout the course of my research I had the privilege of collaborating with Vaibhav Srivastava and Shinkyu Park. During my first semester at Princeton I began collaborating with Vaibhav, and I remember being so appreciative of how welcoming Vaibhav was to me. As I was adjusting not only to graduate school but also to returning to school in general as I had been in the workforce for over two years prior, Vaibhav provided me with a sounding board for research questions or just any questions I had about life at Princeton. Similarly, when working with Shinkyu I always felt comfortable to ask just about any question regarding our work together, no matter how trivial it might seem. Shinkyu was always there to provide valuable feedback and center my thinking back to the right space whenever I might get lost in the weeds. I am a better researcher for having worked with both Vaibhav and Shinkyu. Thank you both for everything.

I would like to thank my Ph.D. committee members N. Jeremy Kasdin, Clancy Rowley, and Anirudha Majumdar for their invaluable insight over the years. I want to especially thank Ani and Shinkyu for taking the time to read and provide feedback on this thesis. Many thanks also to Luigi Martinelli and Vaibhav for serving as my dissertation examiners.

I must also extend a warm thanks to the former and current members of the Leonard Lab that I have been privileged to work alongside: Vaibhav, Shinkyu, Biswadip Dey, Kaynan Ozcimder, Karla Kvaternik, Zahra Aminzare, Katie Fitch, Will Scott, Peter Landgren, Liz Davidson, Bec Gray, Renato Pagliara, Desmond Zhong, Anastasia Bizyaeva, Udari Madhushani, Mari Kawakatsu, Christine Allen-Blanchette, Kenza Hamidouche, María Santos, Yunxiu Zhou, Justice Mason, Charlotte Cathcart, and Justin Lidard. Thank you all for the camaraderie and support. Although I was never one to do a large portion of my work in the office spaces, I always felt energized whenever I would interact with my lab mates. Thank you all for being some of the most compassionate and driven people I have met.

I would like to thank the MAE department at Princeton. Jill Ray was always available to answer any questions I had regarding navigating graduate school. Even before formally enrolling as a graduate student when I visited Princeton to tour the campus along with other accepted students, I remember asking Jill for a good dinner location to grab something to eat with my family. Jill, as always, was able to help out and let us know her favorite pizza place in the area.

Finally, I truly owe so much thanks and gratitude to my family. Ruoqiao, thank you for your advice and support over the years. Whenever I was feeling stressed out or out of sorts, I knew that I could talk to you and I would instantly feel better. You have taught me how to be a better man. I could not have gotten to this point without you. I love you. Also, thank you Mom and Dad for always giving me unending love and encouragement. Going home for holidays and breaks was always a time to relax and recharge, as well as to eat some wonderful home cooking! Even if I wrote something like $1 + 1 = 2$ in a manuscript my parents were always my biggest cheerleaders. Thank you for raising me to be intellectually curious and morally strong.

This dissertation carries the number T#3428 in the records of the Department of Mechanical and Aerospace Engineering.

To my parents, John and Francine.

Contents

List of Tables

List of Figures

Part I

Design of Distributed Estimation and Control for Networked Multiagent Systems

1
Introduction

Networked systems are ubiquitous throughout our society. Examples include transportation networks, power grids, robotic networks, financial markets, as well as social interaction networks[1,2]. In the most basic sense, a *network* is a collection of *nodes* and *edges* where an edge represent an identifiable connection between a pair of nodes. In the case of a social network, nodes could represent people on a social networking site and edges could represent friendship. In the context of dynamical systems theory it is common to refer to nodes as *agents*, where agents are considered to be entities which can perceive the environment using sensors or act on the environment using actuators[3,4]. We refer to a collection of agents which can potentially interact with one another as a *multiagent system*. When such agents are connected in a network according to an underlying structure, we refer to this as a *networked multiagent system*. A great benefit of multiagent systems is that, while individual agents can be relatively simplistic in terms of their sensing or actuation capabilities, through sophisticated feedback dynamics and inter-agent interactions the collective group is capable of completing sophisticated tasks[5]. When considering a large-scale system, it is evident that utilizing multiple agents may afford unique benefits[6]. If a small number of agents become inoperative, it is possible to design into the system a smooth degradation in performance which allows for built-in robustness. Furthermore, if it is relatively simple to add additional agents into the system, then it allows for easy scalability within the system. If individual agents are relatively cheap to produce with respect to the entire system architecture, then we can also see benefits in terms of cost reductions.

We are interested in problems related to utilizing networked multiagent systems to perform estimation and control tasks. Just as multiagent systems can provide unique benefits as previously mentioned, they also offer unique challenges in terms of design. With respect to estimation, typically the information that is collected by sensors in a multiagent system is *distributed* in nature. There are a variety of reasons for this. Individual agents could be measuring data that differs either spatially or temporally (or both). Individual agents could also be equipped with different types of sensors and tasked to measure fundamentally different attributes of some external process of interest. Because of the fact that agents could be measuring disparate data in this sense, the environment could be *partially observable* to an individual agent. Partial observability brings with it particular challenges for decision making in multiagent systems. One example is that optimal planning of a multiagent system under partial observability can be intractable[7]. There is also the issue of *sensor fusion*, which refers to the problem of determining how best to combine the sensor measurements of different agents so as to increase the collective knowledge of the group as a whole[8,56,92,9,10,76]. In Chapter 2 we investigate novel ways to achieve sensor fusion in the presence of communication noise. We show how communication noise degrades the performance of a distributed filtering algorithm outlined in[56] and propose an innovative algorithm which avoids such degradation. The study summarized in Chapter 2 has been published and appears as Savas, Srivastava, and Leonard[11]. The paper can be found in Chapter 8 of Part II of this dissertation. The study was done in collaboration with Vaibhav Srivastava and my advisor Naomi Leonard.

With respect to performing control tasks, typically the control in multiagent systems is *decentralized*. In such schemes there is not a centralized hub that communicates with every agent in the architecture and provides commands, but rather each individual agent is responsible for its own decision-making. One obvious benefit to decentralized control is there is built-in robustness to the system that is not afforded by centralized methods. In well-designed systems if a small number of agents are removed from the architecture it is possible to still meet a desired control objective, which is not true if a centralized controller were to be removed from within a centralized framework. One popular method to address the general problem of decision-making in multiagent systems has been the field of game theory, where each agent is assumed to be rational and strategic[12]. In this dissertation we focus on collaborative multiagent systems where the agents work together to achieve a common control objective. Examples might include formation control of mobile agents[13,14,88], multiagent rendezvous problems[15], or cooperative manipulation by robots[16]. Often in such networked systems agents have limited (local) information regarding the process of interest or the data known to the other agents in the network. If an agent is influenced by the actions of other agents in the network but does not have access to those other agents' data, then the system has a *nonclassical* information structure[89]. Optimal solutions to such nonclassical decentralized systems differ greatly from well-known results in the centralized setting. For example, in[90] the author shows that for a linear system subject to Gaussian noise when the objective is to minimize the expectation of some quadratic criterion, a nonlinear control law exists which outperforms all affine laws when the information structure is nonclassical. Finding the analytical form for the optimal nonlinear control law to this celebrated Witsenhausen's counterexample is still an open problem today[91].

Another well-known result from the centralized setting which is not generally applicable when the control and estimation is distributed over a network is the *separation principle*. In feedback systems theory, the separation principle states that solving for the optimal controller and state estimator can be decoupled under certain conditions[17]. This notion of a separation principle was introduced in[18,19]. The underlying idea is that one chooses as a control at each instant in time the conditional expectation of what would be the optimal control if there was no uncertainty. In other words, one designs the control law assuming perfect information, and then separately designs the estimator such that the estimator errors converge to zero. This is closely related to the idea of *certainty equivalence*[20]. The pioneering idea to use a class of control laws which are functions of the estimates was proposed in[21].

There have been efforts to extend this notion of a separation principle to networked systems. In[22] authors propose and prove the existence of a separation principle for observer-based discrete-time networked control systems with random packet drops. A separation principle is proved for a linear quadratic control problem in a team-based distributed setting in[23]. In Chapter 3 we consider a distributed control problem and propose a new framework which allows us to design the control and estimation parameters separately under certain conditions. We also make use of linear consensus dynamics to extend the operating regime of our algorithm. The study summarized in Chapter 3 is in preparation for submission and appears as Savas, Park, Poor, and Leonard[24]. The paper can be found in Chapter 9 of Part II of this dissertation. The study was done in collaboration with Shinkyu Park and my advisors H. Vincent Poor and Naomi Leonard.

We are also interested in the problem of sensor and actuator selection. Typically, one has a configuration of sensors and actuators and then designs controllers and observers to best use that given configuration to achieve some desired objective. However, in large-scale systems utilizing all of the sensors and actuators in a given architecture could be infeasible both from a computational standpoint or an economic one. Thus, it is a worthwhile endeavor to develop methods to select appropriate subsets of available sensors and actuators for practical use cases. Much of the literature considers the problem of graceful degradation in performance relative to the optimal Kalman filter or Linear Quadratic Regulator that makes use of all of the sensing or actuating capabilities available, which in general is a difficult combinatorial problem. A convex sensor selection methodology for a problem with linear measurements corrupted by additive noise is given in[25] which utilizes heuristics to approximately solve the problem. In[26] the author proposes using a genetic algorithm for actuator selection. In this instance an aerospace application is considered in which there are many possible candidate locations for actuator placement, and the genetic algorithm is used to determine the minimum number of actuators needed to meet a design objective. Obviously enumerating all of the possible combinations in such an application quickly becomes cumbersome and infeasible as the size of the decision space grows. Looking at the joint sensor and actuator selection problem for vibration control of flexible structures, Powell's iterative method for multidimensional function minimization has been used to optimize sensor and actuator placement in an H_2 sense [27] and gradient-based techniques based on the gradient of the controlled system's H_2 norm are used in [28] to inform actuator and sensor placement. In a fluid dynamics application, H_2 optimal sensor and actuator placement was explored for the linearized Ginzburg-Landau system [29]. An alternating direction method of multipliers algorithm for joint sensor and actuator placement is proposed in [30]. Joint sensor and actuator selection is also evident in the leader selection problem in consensus networks. A convex relaxation is employed in [31] and an analytical solution is given for one or two leaders in [32].

We consider the problem of sensor and actuator selection for our novel distributed control framework summarized in Chapter 3 and detailed in Chapter 9, focusing primarily on actuator selection. Rather than tackling the problem of optimally selecting subsets of sensors and actuators, we instead explore how various centrality measures from graph theory might be utilized as heuristics. In Chapter 4, we use hypothesis testing to compare between select centrality measures with the goal being to minimize a key norm. A benefit to such an approach is that one could use these statistically discovered heuristics as seeds to "warm-start" a given numerical optimization procedure. As previously discussed, solving the optimal sensor and actuator selection problem is computationally intensive and can become infeasible as the size of the architecture grows. Utilizing a suboptimal solution as a seed for such optimizations could vastly improve computation times. In addition, a heuristic such as these centrality measures could be employed in online algorithms that must respond to changes in the network architecture where one does not have the time to reoptimize. We explore situations related to this, specifically irrecoverable actuator failure, in Chapter 5.

1.1 Outline and Contributions

In this dissertation we focus on the design of estimation and control frameworks for distributed multiagent systems. We begin by considering the distributed filtering problem with the added caveat that communication between agents is a noisy process. We show the detrimental effects such communication noise has on consensus filters in the literature and propose a novel filtering algorithm which is resilient to such effects. We then consider the joint problem of distributed estimation and control over a network of agents. By utilizing the small-gain theorem we are able to provide conditions under which the design of the estimation and control parameters can be done independently akin to the celebrated separation principle in optimal control theory. We then use this distributed control framework as a basis to develop heuristics for sensor and actuator placement. We employ statistical hypothesis testing to compare heuristics and analyze the robustness and scalability of our results.

The dissertation is separated into two parts. Part I contains a summary of all key work as well as unpublished results. Part II, consisting of Chapters 7-9, contains work that has been published or that is in preparation for submission.

In Chapter 2 we introduce the distributed linear filtering problem and summarize results from Chapter 8. We consider two-stage consensus filters and demonstrate that communication noise inherently destabilizes filters from the literature. We develop an innovative two-stage filter which is robust to communication noise and rigorously analyze its properties. We also provide analytical results on how the performance of individual agents differs depending on their location in the network. We conclude with a preliminary investigation of the convexity of a proposed cost function integral to our algorithm design.

In Chapter 3 we consider distributed control of linear time-invariant systems and summarize results from Chapter 9. We employ the small-gain theorem and characterize linear matrix inequality conditions under which a weaker notion of the separation principle holds for the design of the estimator and control parameters. We also explore how to utilize linear consensus dynamics to extend the region of applicability of our framework. We show how our framework can handle nonlinear control laws and consider a multi-vehicle platooning example as an application.

In Chapter 4 we use statistical hypothesis testing to compare heuristics for sensor and actuator selection with respect to the distributed control framework proposed in Chapters 3 and 9. We compare a set of centrality measures against one another and compute statistics over Erdos-Renyi random graphs. We show that there is strong statistical evidence that, among the centrality measures considered, betweenness centrality is the best heuristic given a sufficient edge density.

In Chapter 5 we explore how our statistical results extend to additional graph generation methods, actuator failure, sparse control authority, as well as network scale. We find strong statistical evidence that betweenness centrality remains the most effective heuristic for actuator selection given sufficient edge density except in the paradigm of sparse control authority. We consider systems in which individual agents do not have full controllability of the process; rather, there is only joint controllability at the network level. In such cases we find strong statistical evidence that degree centrality becomes the most effective heuristic for actuator selection.

In Chapter 6 we summarize the main results and provide commentary on possible future directions of this work.

2

Distributed Filtering with Noisy Communication

We now investigate the problem of distributed filtering in a networked multiagent system of a scalar linear stochastic process under noisy communication. Sections 2.1-2.5 summarize results which are presented in Part II: Chapter 8 which appears as Savas, Srivastava, and

Leonard[11]. In Section 2.1 we define precisely what we mean by the distributed linear filtering problem. In Section 2.2 we provide an overview of linear consensus dynamics which are a key component of our algorithm. In Section 2.3 we introduce one two-stage distributed filter from the literature. We define what we mean by noisy communication within the sensor network and analyze how such communication noise impacts the performance of said filter. In Section 2.4 we propose and rigorously analyze a novel algorithm to mitigate such issues.

Sections 2.6-2.7 cover new results, not published in[11] of Chapter 8. In Section 2.6 we explore how the performance of individual nodes is related to their location in the network. In Section 2.7 we provide preliminary analysis on the convexity of a pertinent cost function.

2.1 Distributed Linear Filtering Problem Setup

Consider the following scalar linear stochastic process

$$x(k+1) = ax(k) + w(k), \quad x(0) = X_0, \tag{2.1}$$

for each $k \in \mathbb{Z}_{\geq 0}$, where $a \in \mathbb{R}$ is a constant, $\{w(k)\}_{k \in \mathbb{Z}_{\geq 0}}$ is a sequence of independent identically distributed (i.i.d.) zero-mean Gaussian noise with variance $q \in \mathbb{R}_{>0}$, and X_0 is a Gaussian random variable with mean x_0 and variance σ. For simplicity, we will consider the case with $a = 1$ but note that the following analysis is generalizable to the case $a \neq 1$. An example evolution for the stochastic process defined in (2.1) with $a = 1$, process noise variance $q = 1$, mean of initial condition $x_0 = 0$, and variance of initial condition $\sigma = 1$ is shown in Figure (2.1).

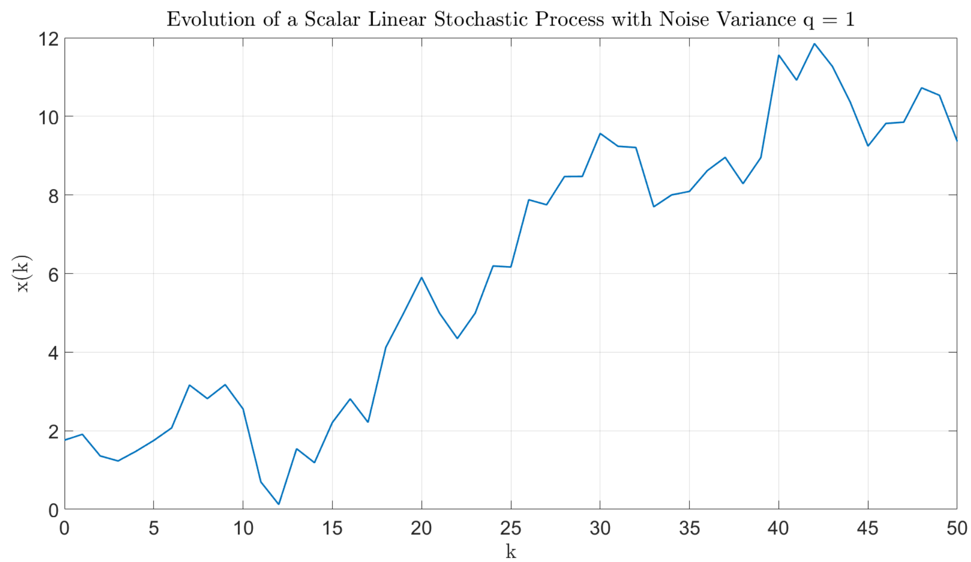


Figure 2.1: Evolution of a scalar linear stochastic process given by (2.1) with $a = 1$, noise variance $q = 1$, mean of initial condition $x_0 = 0$, and variance of initial condition $\sigma = 1$.

Now suppose that a sensor is tasked to sample such a stochastic process at each discrete time k to obtain a noisy measurement given by

$$y(k) = x(k) + n(k), \quad \text{for each } k \in \mathbb{Z}_{\geq 0}, \tag{2.2}$$

where $\{n(k)\}_{k \in \mathbb{Z}_{\geq 0}}$ is a sequence of i.i.d. zero-mean Gaussian noise with variance $r \in \mathbb{R}_{>0}$. Using sensor measurements $y(k)$ given in (2.2) to estimate the state $x(k)$ in (2.1) is known as the standard scalar Kalman filtering problem[70]. We now extend these ideas to a network of sensing nodes which can communicate over a fixed undirected graph $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$, where $\mathcal{V} = \{1, \ldots, N\}$ is the vertex set, $\mathcal{E} \in \mathcal{V} \times \mathcal{V}$ is the edge set, and N is the total number of nodes. In this analysis we only consider connected graphs, meaning that every node has an existing path to every other node in the network. Now assume that each node $i \in \{1, \ldots, N\}$

samples the stochastic process (2.1) at discrete times k and generates noisy measurements defined by

$$y_i(k) = x(k) + n_i(k), \quad \text{for each } i \in \{1, \ldots, N\}, \tag{2.3}$$

where $\{n_i(k)\}_{k \in \mathbb{Z}_{\geq 0}}$ are i.i.d. zero-mean Gaussian noises with variance r. Note that each sensing node is assumed to have the same noise variance r. We further assume the noise sequences $n_i(k)$ are independent for different $i \in \{1, \ldots, N\}$. We can write (2.3) in vector form as

$$y(k) = x(k)1_N + n(k), \tag{2.4}$$

where $y(k)$ and $n(k)$ are the N-column vectors of $y_i(k)$'s and $n_i(k)$'s, respectively, and 1_N is the N-column vector of all ones. As an illustrative example, consider a network of $N = 3$ nodes which each take noisy samples of the process $x(k)$ seen in Figure (2.1). If each sensor has noise variance $r = 25$, which indicates a relatively high sensor-to-process noise ratio, then typical sensor output described by (2.3) is shown in Figure (2.2).

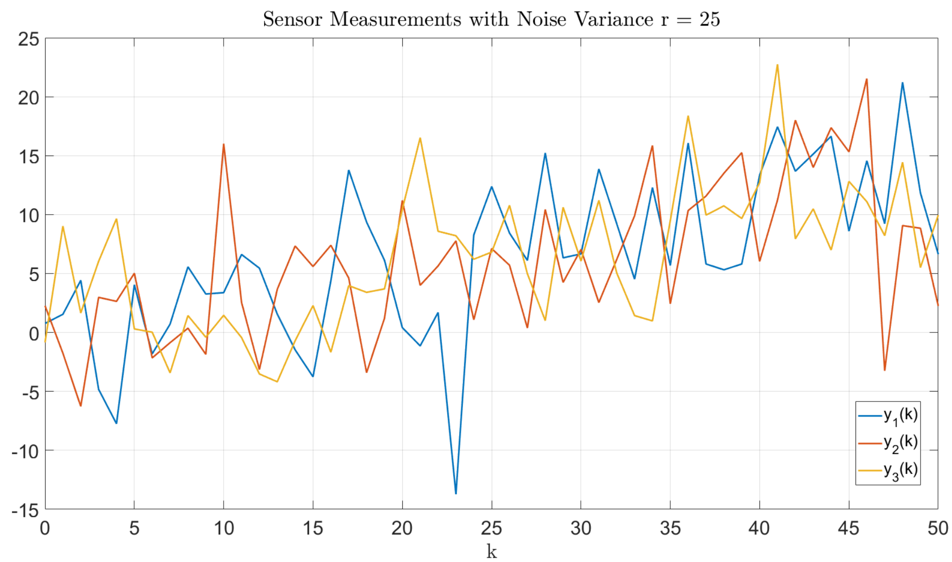


Figure 2.2: Samplings of the stochastic process seen in Figure (2.1) by a network of $N = 3$ sensors each with noise variance $r = 25$.

We will now focus on the distributed estimation problem in which the sensing nodes can only use local communication within the network to construct and update their estimates of the process $x(k)$. In other words, each node in the network does not have global knowledge of the other nodes' estimates for any given discrete time and must instead use information flow within the network to improve their individual estimates. Specifically, we focus on consensus-based dynamics and their applications to distributed estimation[59,60,71].

2.2 Consensus Dynamics

In consensus dynamics each node at each discrete time will average its state (in this case, its estimate) with that of its neighbors in the communication graph[59,68,72]. Let $F \in \mathbb{R}^{N\times N}$ be the matrix of convex weights a node assigns to its neighbors. Note that F_{ii} is node i's self-weight. To construct such a consensus matrix we employ the adjacency matrix $\mathcal{A} \in \mathbb{R}^{N\times N}$

of the graph, defined as

$$\mathcal{A}_{ij} = \begin{cases} 1 & \text{if } (i,j) \in \mathcal{E} \\ 0 & \text{otherwise} \end{cases} \tag{2.5}$$

The graph Laplacian $\mathcal{L} \in \mathbb{R}^{N \times N}$ is then defined as

$$\mathcal{L}_{ij} = \begin{cases} -\mathcal{A}_{ij} & \text{for } i \neq j \\ \sum_{k=1}^{N} \mathcal{A}_{ik} & \text{for } i = j \end{cases} \tag{2.6}$$

We can then construct a consensus matrix F as[60]

$$F = \mathcal{I}_N - \varepsilon \mathcal{L} \tag{2.7}$$

where $\mathcal{I}_N$ is the identity matrix of order N and $0 < \varepsilon < \frac{1}{\max\limits_i \left(\sum_{k=1}^{N} \mathcal{A}_{ik} \right)}$. It is well-known that this matrix F is nonnegative and row-stochastic, and for a connected and undirected graph has only one simple eigenvalue at unity and every other eigenvalue is inside the unit disk[59,68,71,72]. Furthermore, for unsigned graphs F is also column-stochastic. Note that row-stochasticity of F simply means that $\sum_{j=1}^{N} F_{ij} = 1$ for each $i \in \{1, \ldots, N\}$, whereas column-stochasticity yields $\sum_{i=1}^{N} F_{ij} = 1$ for each $j \in \{1, \ldots, N\}$. We will denote the eigenvalues of F as $\{\lambda_0, \ldots, \lambda_{N-1}\}$ with $\lambda_0 = 1$. With these properties of F in mind, a key limiting behavior is

$$\lim_{m \to \infty} F^m = \frac{1}{N} 1_N 1_N^T \tag{2.8}$$

Thus, for any vector $v \in \mathbb{R}^N$ we have

$$\lim_{m\to\infty} [F^m v]_i = \frac{1}{N}\sum_{i=1}^{N} v_i, \quad \text{for each } i \in \{1, \ldots, N\}. \tag{2.9}$$

In other words, if we take m as the number of consensus rounds, in the limit as $m \to \infty$ we see that each element of $F^m v$ will converge to the element-wise average value of v, which is precisely what we mean by achieving consensus. As an illustrative example, for a line graph with $N = 3$ nodes and taking $\varepsilon = 0.4$ we can construct a consensus matrix $F = \begin{bmatrix} 0.6 & 0.4 & 0 \\ 0.4 & 0.2 & 0.4 \\ 0 & 0.4 & 0.6 \end{bmatrix}$, and it can be seen that $F^\infty = \begin{bmatrix} 1/3 & 1/3 & 1/3 \\ 1/3 & 1/3 & 1/3 \\ 1/3 & 1/3 & 1/3 \end{bmatrix}$ as expected. We now discuss one method in which consensus dynamics have been employed to tackle the aforementioned distributed estimation problem.

2.3 A Two-Stage Distributed Linear Filter

We will focus on one particular method credited to Carli *et al.*[56] which was put forth to address the estimation problem posed in Section 2.1. The authors propose a two-stage distributed algorithm consisting of a measurement and prediction stage followed by a consensus stage. We now summarize the algorithm.

2.3.1 Measurement and Prediction Stage

At discrete time k each sensing node $i \in \{1, \ldots, N\}$ computes a convex combination of its predictive estimate of the current state of the process $x(k)$ which uses observations up until time $k-1$, denoted as $\hat{x}_i(k|k-1)$, and its current observation $y_i(k)$ to generate its estimate of the process given observations until time k, denoted as $\hat{x}_i(k|k)$. We can write this as

$$\hat{x}(k|k) = (1-\ell)\hat{x}(k|k-1) + \ell y(k), \tag{2.10}$$

where $\hat{x}(k|k)$ and $\hat{x}(k|k-1)$ are vectors of $\hat{x}_i(k|k)$ and $\hat{x}_i(k|k-1)$, respectively, and $\ell \in [0,1]$ is the gain. It is important to note here that the gain ℓ is assumed constant, which differs from the optimal Kalman filter gain. Thus, the distributed filter resulting from constant choice of ℓ will not necessarily be optimal; however, the authors show in [56] that this two-stage distributed algorithm leads to estimation error with bounded variance. This implies that choosing a constant gain ℓ will in fact be stabilizing.

2.3.2 Consensus Stage

The second stage of the algorithm utilizes consensus dynamics. Between two consecutive discrete times k and $k+1$ there are m rounds of consensus dynamics which operate on the local estimates $\hat{x}_i(k|k)$. As mentioned previously, such consensus dynamics ensure that the local estimate $\hat{x}_i(k+1|k)$ of each node converges towards the average of the group $\frac{1}{N}\sum_{j=1}^{N} \hat{x}_j(k|k)$. We write this stage as

$$\hat{x}\Big(k + \frac{h}{m}\Big|k\Big) = F\hat{x}\Big(k + \frac{(h-1)}{m}\Big|k\Big), \quad h \in \{1, \ldots, m\}. \tag{2.11}$$

Note that for such a two-stage algorithm a timescale separation between the dynamics of the stochastic process $x(k)$ and the consensus dynamics is inherently assumed. In other words, it is assumed that the communication and consensus dynamics of the sensing nodes operate at a much faster rate than the process dynamics. This is necessary if the sensors are expected to perform m rounds of consensus in between taking measurements of the process at discrete times k and $k+1$. We see that after the consensus rounds are completed, the second stage (2.11) yields the estimate vector $\hat{x}(k+1|k)$ which is then used in the first stage (2.10) to generate the estimate $\hat{x}(k+1|k+1)$. We also note that this algorithm is initialized with $\hat{x}(0|-1) = x_0 1_N$ where x_0 is the mean of the Gaussian random variable used to initialize the process $x(k)$. The estimates at future times are then computed recursively using (2.10) and (2.11). After computing the covariance of the estimation error for this algorithm, Carli *et al.*[56] used it to find the optimal filter gain ℓ that minimizes the trace of the error covariance matrix in the limit as $k \to \infty$, otherwise known as the asymptotic error covariance matrix. Assuming that the consensus step (2.11) occurs noiselessly, the described algorithm is stabilizing and will lead to bounded error covariance. However, we believe that the notion of noiseless consensus is an optimistic one and propose a modification to the consensus step to account for communication noise between nodes.

2.3.3 Implementing Noisy Communication

We will now assume that each node receives noisy estimates of each of its neighbors' states each time consensus is performed. This models the expected communication noise which will be present during the information transfer. Since the modification occurs in the consensus steps, the first stage of the algorithm remains the same as that outlined in (2.10). However, we modify the second stage (2.11) to include noisy communication as

$$\hat{x}\Big(k+\frac{h}{m}\Big|k\Big) = F\hat{x}\Big(k+\frac{(h-1)}{m}\Big|k\Big) + \sigma_c u\Big(k+\frac{h}{m}\Big), \tag{2.12}$$

where $u(k+h/m)$ is the N-variate zero-mean Gaussian noise with covariance $\mathcal{I}_N$, for each $k \in \mathbb{Z}_{\geq 0}$ and $h \in \{1, \dots, m\}$, $u(k+h/m)$ are independent, and σ_c^2 is the communication noise variance. Note that we assume that the communication noise experienced by each node has the same variance σ_c^2. We can define the estimation error at each discrete time k as

$$\tilde{x}(k|k-1) = x(k)1_N - \hat{x}(k|k-1). \tag{2.13}$$

We now investigate how the modified algorithm comprised of (2.10) and (2.12) performs with respect to the sum of the error variances across all nodes.

2.3.3.1 Numerical Performance Under Noisy Communication

We consider $N = 3$ nodes communicating over an undirected line graph. As described earlier, we construct a consensus matrix as $F = \mathcal{I}_3 - \varepsilon\mathcal{L}$, where we choose $\varepsilon = 0.4$. We analyze ten discrete time instances of the stochastic process (2.1) with $a = 1$, i.e., $k \in \{0, ..., 9\}$, and between each consecutive pair of time instances we apply m consensus rounds. For this analysis we consider $m \in \{0, 1, 2, 3, 4, 5\}$ and assume that the process noise variance is $q = 1$ and the sensor noise variance is $r = 25$. Furthermore, we use filter gain $\ell = 0.25$. We perform 200,000 Monte Carlo simulations as a way to estimate the trace of the error covariance matrix. Figure (2.3) shows the trace of the error covariance matrix for $k = 4$, which can be represented as $\sum_{i=1}^{3} \mathrm{var}(\tilde{x}_i(4|3))$, as a function of the number of consensus rounds m for a range of values of communication noise variance σ_c.

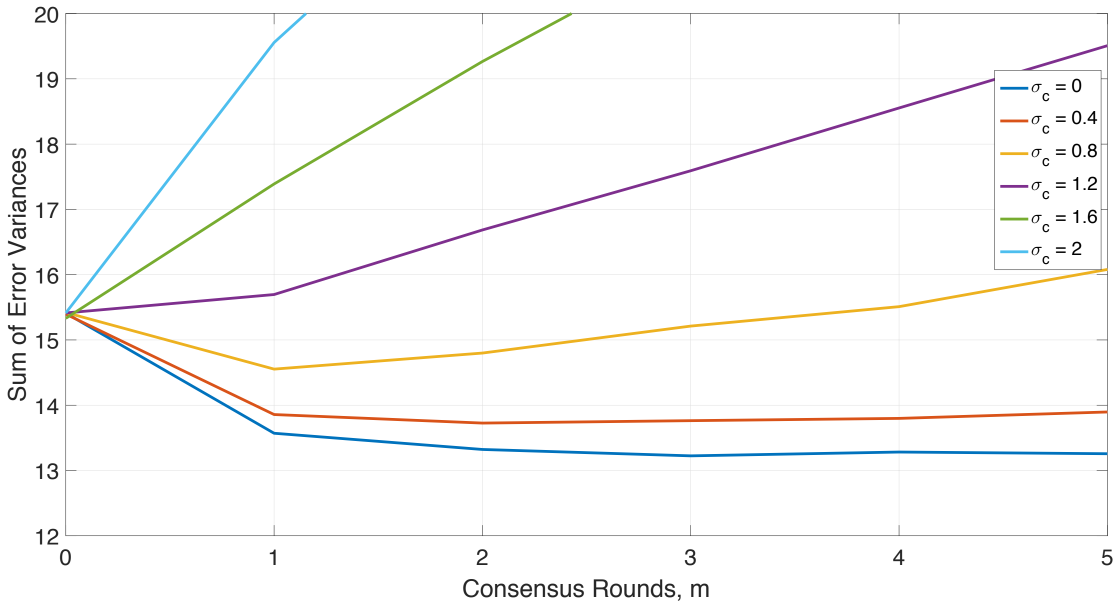


Figure 2.3: Influence of communication noise in consensus dynamics on error variance across 200,000 Monte Carlo runs for distributed filtering algorithm (2.10) and (2.12) with $N = 3$, $q = 1$, and $r = 25$ for an undirected line graph. We see that the error variance diverges with the number of consensus rounds. Repeated from Figure (8.1).

We can immediately see that for large enough values of σ_c the trace of the error covariance actually increases as more consensus rounds are performed, suggesting that the two-stage estimation algorithm is no longer stabilizing in the presence of communication noise. We observe that the trace of the error covariance diverges as the number of consensus rounds increases, which clearly is not a desired property of a distributed filtering algorithm which invokes consensus dynamics. We must note here that such destabilizing behavior is not unexpected since consensus dynamics inherently have one eigenvalue at unity as mentioned previously. This unit eigenvalue acts to integrate the communication noise across each consensus round, and the integrated noise has asymptotically infinite variance. Thus, it is clear that a distributed filtering strategy which has been deliberately designed as in [56] for noiseless communication does not and should not immediately extend to situations with communication noise. We will now introduce our novel two-stage distributed filter and rigorously show that it remains stabilizing even in the presence of noisy communication.

2.4 A Novel Two-Stage Distributed Linear Filter

We now describe our novel two-stage distributed filter which modifies and expands on the algorithm of Carli *et al.*[56] to account for and mitigate the effects of communication noise. The first stage of our algorithm remains the same as in (2.10), i.e.,

$$\hat{x}(k|k) = (1-\ell)\hat{x}(k|k-1) + \ell y(k), \tag{2.14}$$

with $\hat{x}(0|-1) = x_0 1_N$. However, we modify the consensus dynamics in the following way. We define $z(k|k) = \hat{x}(k|k)$ for each $k \in \mathbb{Z}_{\geq 0}$. We update z through m consensus rounds between consecutive time instances k and $k+1$ as

$$z\Big(k + \frac{h}{m}\Big|k\Big) = Fz\Big(k + \frac{(h-1)}{m}\Big|k\Big) + \sigma_c u\Big(k + \frac{h}{m}\Big) + \hat{x}(k|k) \tag{2.15}$$

for $h \in \{1, \dots, m\}$. Note that in (2.15), each node $i \in \{1, \dots, N\}$ now remembers its own estimate $\hat{x}_i(k|k)$ at discrete time k and re-injects it during each subsequent consensus round. To see the inspiration for such a modification, consider starting from a deterministic initial condition $z(k|k) = \hat{x}(k|k)$. After m rounds of consensus the dominating component of the variance of $z(k+1|k)$ will be $m\sigma_c^2$ (see Figure (2.3)). If we re-inject $\hat{x}(k|k)$ at each consensus step, we ensure that the dominating component of the expected value of $z_i(k+1|k)$ is $\frac{m+1}{N}\sum_{j=1}^{N}\hat{x}_j(k|k)$ for each $i \in \{1, \dots, N\}$. Finally, if we divide $z(k+1)$ by $(m+1)$, the resulting mean is $\frac{1}{N}\sum_{j=1}^{N}\hat{x}_j(k|k)$ and variance is $m\sigma_c^2/(m+1)^2$ which goes to 0 as $m \to +\infty$. Thus, for large m we recover the performance of the noise-free algorithm. An issue arises if m is small since the communication noise will still degrade the performance of the estimator. For this reason, we design the update $\hat{x}(k+1|k)$ as the convex sum of $\hat{x}(k|k)$ and $z(k+1|k)$ as

$$\hat{x}(k+1|k) = \zeta\hat{x}(k|k) + (1-\zeta)\frac{z(k+1|k)}{m+1}, \tag{2.16}$$

where $\zeta \in [0,1]$ is a constant. The parameter ζ is a way for us to trade off the variance of the two estimators $\hat{x}(k|k)$ and $z(k+1|k)$. As we intuitively explained earlier, for large m we can choose ζ close to 0 and for small m we can choose ζ close to 1. We note that the distributed filtering algorithm in[56] has only one tunable parameter ℓ, whereas our algorithm in contrast has two tunable parameters ℓ and ζ. In a similar way to[56] we can choose these two parameters such that the asymptotic error covariance of our estimator is minimized. With this in mind, we will now analyze the error covariance of this new algorithm.

2.5 Analysis of the Novel Two-Stage Distributed Linear Filter

We now analyze the properties of our novel distributed linear filter. We begin by deriving an expression for the asymptotic error covariance of our estimator.

2.5.1 Error Covariance of Estimator

We first define the predictive and posterior errors as

$$\begin{aligned} \tilde{x}(k+1|k) &= x(k+1)1_N - \hat{x}(k+1|k), \\ \text{and} \quad \tilde{x}(k+1|k+1) &= x(k+1)1_N - \hat{x}(k+1|k+1), \end{aligned} \tag{2.17}$$

respectively. Let

$$
\begin{aligned}
P(k+1|k) &= \mathbb{E}[\tilde{x}(k+1|k)\tilde{x}(k+1|k)^\top] \\
\text{and} \quad P(k+1|k+1) &= \mathbb{E}[\tilde{x}(k+1|k+1)\tilde{x}(k+1|k+1)^\top]
\end{aligned} \tag{2.18}
$$

be predictive and posterior error covariance matrices. With these definitions in place, we now state the main result.

Theorem 1 (***Asymptotic Error Covariance***[11]**).** *For the scalar linear stochastic dynamics* (2.1) *with $a = 1$ and the distributed linear filtering algorithm with noisy communication defined by* (2.14), (2.15) *and* (2.16), *the following statements hold:*

1. *The asymptotic error covariance is*

$$
\begin{aligned}
\lim_{k\to\infty} P(k|k-1) = \ell^2 r \sum_{i=0}^{\infty} (1-\ell)^{2i} F^{\dagger(i+1)} (F^{\dagger(i+1)})^\top + \frac{q}{1-(1-\ell)^2} 1_N 1_N^\top \\
+ \left(\frac{1-\zeta}{m+1}\right)^2 \sigma_c^2 \sum_{i=0}^{\infty} (1-\ell)^{2i} \sum_{j=0}^{m-1} F^{\dagger i} F^j (F^j)^\top (F^{\dagger i})^\top,
\end{aligned} \tag{2.19}
$$

where $F^\dagger = \zeta \mathcal{I}_N + \left(\frac{1-\zeta}{m+1}\right) \sum_{i=0}^{m} F^i$;

2. *The trace of the asymptotic covariance matrix is*

$$\operatorname{tr}\left(\lim_{k\to\infty} P(k|k-1)\right) = \frac{\ell^2 r + qN + \dfrac{(1-\zeta)^2\sigma_c^2 m}{(m+1)^2}}{1-(1-\ell)^2} + \ell^2 r \sum_{h=1}^{N-1} \frac{\left|\left(\dfrac{1-\zeta}{m+1}\right)\bar{\lambda}_h + \zeta\right|^2}{1-(1-\ell)^2\left|\left(\dfrac{1-\zeta}{m+1}\right)\bar{\lambda}_h + \zeta\right|^2} + \left(\frac{1-\zeta}{m+1}\right)^2 \sigma_c^2 \sum_{h=1}^{N-1} \frac{\left(\dfrac{1-|\lambda_h|^{2m}}{1-|\lambda_h|^2}\right)}{1-(1-\ell)^2\left|\left(\dfrac{1-\zeta}{m+1}\right)\bar{\lambda}_h + \zeta\right|^2}, \tag{2.20}$$

where $\bar{\lambda}_h = \sum_{n=0}^{m} \lambda_h^n$.

Importantly, note that the steady-state error covariance (2.20) is bounded and hence our distributed estimation algorithm is stabilizing in the mean squared sense. However, our algorithm is not necessarily optimal because we assume that the convex weights ℓ and ζ are both constant. Even with this fact, we can still select optimal parameters ℓ and ζ *with respect to our algorithm*.

2.5.2 Methodology to Tune Parameters

As previously mentioned, our novel distributed algorithm requires tuning of the two parameters ℓ and ζ. For a given graph structure with a fixed number of agents N and consensus matrix F, given process, measurement and communication variance q, r, and σ_c, and a given number of consensus rounds m, we choose these parameters to minimize the asymptotic error covariance (2.20). We define the relevant cost function as

$$J(\ell, \zeta) = \operatorname{tr}\left(\lim_{k\to\infty} P(k|k-1)\right). \tag{2.21}$$

Note the special case with no consensus ($m = 0$) when determining the optimal (ℓ, ζ) which minimize J. In this case, the modified algorithm will only involve (2.14) and (2.16). From (2.16) we see that the only appropriate formulation would have $\zeta = 1$. With $\zeta = 1$ and $m = 0$ we see that (2.20) simplifies to $J|_{m=0} = \frac{(\ell^2 r+q)N}{1-(1-\ell)^2}$. We then use `fmincon` in MATLAB to minimize $J|_{m=0}$ and solve for the optimal ℓ. Likewise, for $m > 0$ we can use `fmincon` to solve for the optimal ℓ and ζ which minimize J. The trends of optimal ℓ and ζ as a function of number of consensus rounds m and communication noise standard deviation σ_c are shown in Figure (2.4).

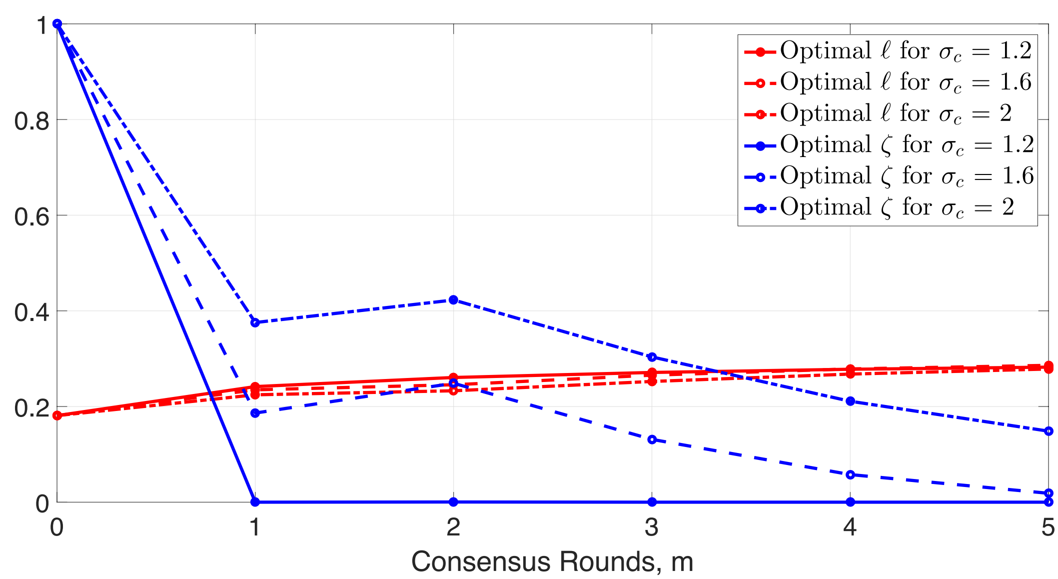


Figure 2.4: Optimal ℓ and ζ as a function of consensus rounds m and communication noise standard deviation σ_c with $N = 3$, $r = 25$, and $q = 1$ for an undirected line graph. Repeated from Figure (8.2).

We see that the optimal ℓ value increases with m, whereas the optimal ζ value does not display a monotonic trend with m. We attribute the initial trend of ζ for smaller values of m to the transient consensus dynamics. Furthermore, note that the optimal value of ζ goes toward zero as the number of consensus rounds m increases. This is precisely the behavior we would expect and aligns with the intuition we outlined earlier.

2.5.3 Numerical Simulations

We now numerically investigate the performance of our novel distributed linear filter. We again consider an undirected line graph with $N = 3$ nodes. We choose the same simulation parameters as in Figure (2.3) and again choose as our performance metric $\sum_{i=1}^{3} \mathrm{var}(\tilde{x}_i(4|3))$. For each pertinent combination of m and σ_c values we use the optimal ℓ and ζ in the algorithm as determined in Section 2.5.2. The summed error variance metric versus the number of consensus rounds m for our novel estimation algorithm is shown in Figure (2.5) for various values of the communication noise standard deviation σ_c.

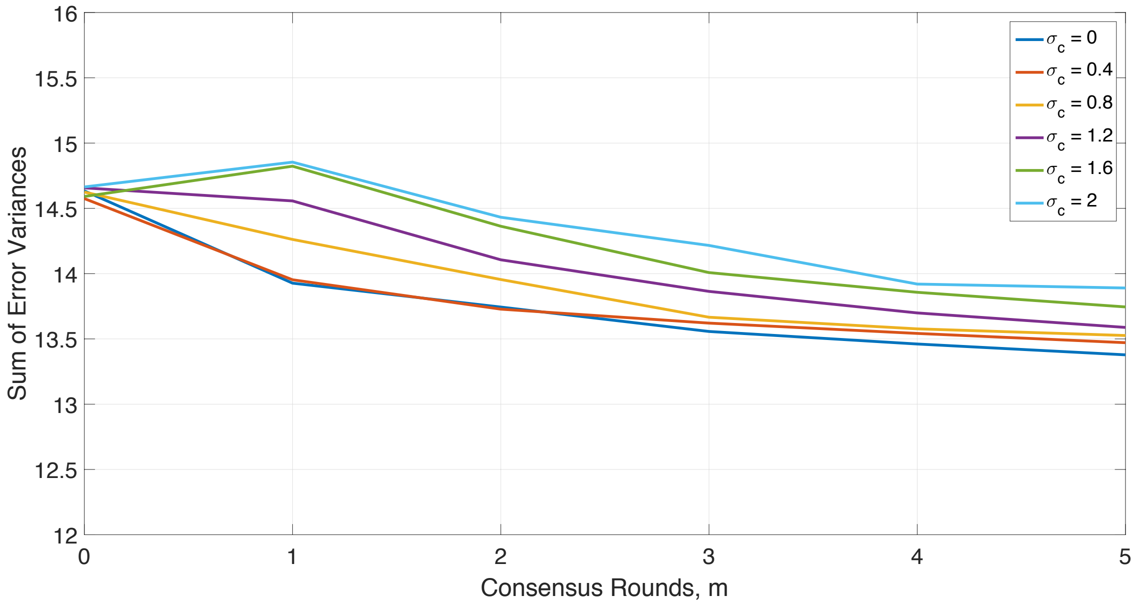


Figure 2.5: Influence of communication noise in consensus dynamics on error variance across 200,000 Monte Carlo runs for our novel distributed filtering algorithm with $N = 3$, $r = 25$, and $q = 1$ for an undirected line graph. Even as σ_c increases the error variance no longer diverges as more consensus rounds are performed. Repeated from Figure (8.3).

Note that there is a slight difference in scale between the vertical axes in Figures (2.3) and (2.5). We see that with our novel algorithm the error variance does not increase in an unbounded way as more consensus rounds are performed, which contrasts with what we observed in Figure (2.3). Instead, we observe that the trend in error variance versus consensus rounds is closer to the monotonically decreasing ideal that one would expect from a distributed filter without communication noise. This holds true even for the larger values of σ_c that we simulated. We see that our novel algorithm performs effectively despite the inclusion of communication noise in the dynamics.

2.6 Individual Node Differences for Novel Two-Stage Distributed Linear Filter

We are also interested in how the asymptotic error variances of individual nodes within a graph differ from one another as opposed to just looking at the trace of the asymptotic error covariance matrix. Table 2.1 serves as a reminder of the relevant nomenclature.

Parameter	Definition
ℓ	Filter gain
r	Sensor noise variance
q	Process noise variance
N	Total number of nodes
ζ	Algorithm parameter used to form convex combination of estimators
σ_c^2	Communication noise variance
m	Rounds of consensus dynamics
λ	Eigenvalue of consensus matrix, F

Table 2.1: Nomenclature relevant to our novel distributed estimation algorithm.

To determine individual node error variances for our novel distributed estimation algorithm defined by (2.14), (2.15) and (2.16), we first let the diagonalization of F be given by

$$F = U\Lambda U^* \tag{2.22}$$

where

$$\Lambda = \begin{bmatrix} \lambda_0 & 0 & \dots & 0 \\ 0 & \lambda_1 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \lambda_{N-1} \end{bmatrix} \tag{2.23}$$

and the column vectors of U are the right eigenvectors of F where U^* denotes the complex conjugate transpose of the matrix U. Given that F is a normal matrix, we have that U is a unitary matrix such that $UU^* = U^*U = \mathcal{I}_N$. Specifically, we choose the eigenvectors of F such that U forms an orthonormal basis of $\mathbb{C}^N$. Having done this, we let

$$U = \begin{bmatrix} \mathbf{u}_0 & \mathbf{u}_1 & \dots & \mathbf{u}_{N-1} \end{bmatrix} \tag{2.24}$$

where $\mathbf{u}_0$ through $\mathbf{u}_{N-1}$ are the appropriate eigenvectors described above. With this formulation, the asymptotic error variance of node i is given by

$$Var_i = \sum_{h=0}^{N-1} \frac{|\mathbf{u}_h(i)|^2 \left\{ \ell^2 r \left| \left(\frac{1-\zeta}{m+1} \right) \bar{\lambda}_h + \zeta \right|^2 + \left(\frac{1-\zeta}{m+1} \right)^2 \sigma_c^2 \left(\frac{1-|\lambda_h|^{2m}}{1-|\lambda_h|^2} \right) \right\}}{1-(1-\ell)^2 \left| \left(\frac{1-\zeta}{m+1} \right) \bar{\lambda}_h + \zeta \right|^2} + \frac{q}{1-(1-\ell)^2} \tag{2.25}$$

where $\bar{\lambda}_h = \sum_{n=0}^{m} \lambda_h^n$. Thus, the difference in asymptotic error variance between nodes is determined by the respective elements of each eigenvector of the consensus matrix F. As an example, consider an undirected line graph with $N = 3$ nodes. We denote nodes 1 and 3 as the periphery nodes, respectively, and node 2 as the middle node. For a process noise variance of $q = 1$, sensor noise variance of $r = 25$, communication noise variance of $\sigma_c^2 = 1$, and $\zeta = 0.1$, asymptotic error variances for each individual node in the line graph as a function of ℓ are shown for $m \in \{1, 2, 3, 4\}$ in Figure (2.6).

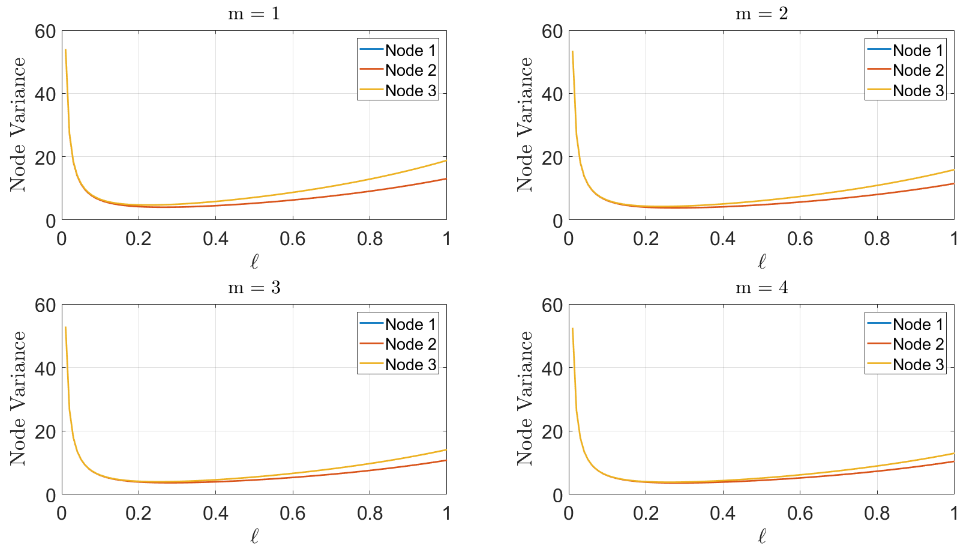


Figure 2.6: Asymptotic error variances across individual nodes for our novel distributed filtering algorithm with $N = 3$, $r = 25$, $q = 1$, $\sigma_c^2 = 1$, and $\zeta = 0.1$ for an undirected line graph.

There are some things to highlight. First we see that the individual node error variances for nodes 1 and 3 are identical, which is expected as these nodes are indistinguishable with regards to the estimation problem. We also observe that the error variance for node 2, the middle node, is the lowest across the ℓ and m values studied. This again aligns with our expectation since this node has the most immediate access to information within the network due to the way the information flows given the consensus mechanism at play. Furthermore, we see that as the number of consensus rounds m increases the node variances for the periphery nodes begin to converge to that for the middle node. This is a visualization of the consensus dynamics at work. We now provide a brief examination of the convexity of the trace of the asymptotic error covariance matrix with respect to algorithm parameters.

2.7 Convexity of Trace of Asymptotic Error Covariance Matrix

In Section 2.5.2 we described how we utilize `fmincon` in MATLAB to solve for the optimal ℓ and ζ which minimize the cost function $J(\ell, \zeta) = \mathrm{tr}\Big(\lim_{k\to\infty} P(k|k-1)\Big)$. However, it is important to be able to characterize the convexity of $J(\ell, \zeta)$ with respect to ℓ and ζ. If the cost function is not convex in these parameters then it is possible that `fmincon` would return a local minimum and not a global minimum. While finding a local minimum would not render our novel algorithm ineffective, it is obvious that we would like to select ℓ and ζ such that the cost function is as small as possible. We define $L := [0, 1]$ and $Z := [0, 1]$. Let $\ell \in L$ and $\zeta \in Z$ and let $A \subset L \times Z$. A well-known result is that a given function $J(\ell, \zeta) : A \to \mathbb{R}$ is convex in ℓ and ζ on A if and only if its Hessian matrix of second partial derivatives is positive semidefinite on A[50]. Thus, we require

$$\begin{bmatrix} \frac{\partial^2 J}{\partial \ell^2} & \frac{\partial^2 J}{\partial \ell \partial \zeta} \\ \frac{\partial^2 J}{\partial \zeta \partial \ell} & \frac{\partial^2 J}{\partial \zeta^2} \end{bmatrix} \succeq 0 \tag{2.26}$$

on A. By observing (2.20) we note that the Hessian will be a relatively complicated function of ℓ and ζ. To expedite our analysis we use Mathematica to calculate the Hessian. For a simple line graph with $N = 3$ nodes and with process noise variance $q = 1$, sensor noise variance $r = 25$, and communication noise variance $\sigma_c^2 = 1$, Figures (2.7) - (2.11) plot the real parts of any negative eigenvalues in the operating (ℓ, ζ) space for consensus rounds from $m = 1$ to $m = 5$.

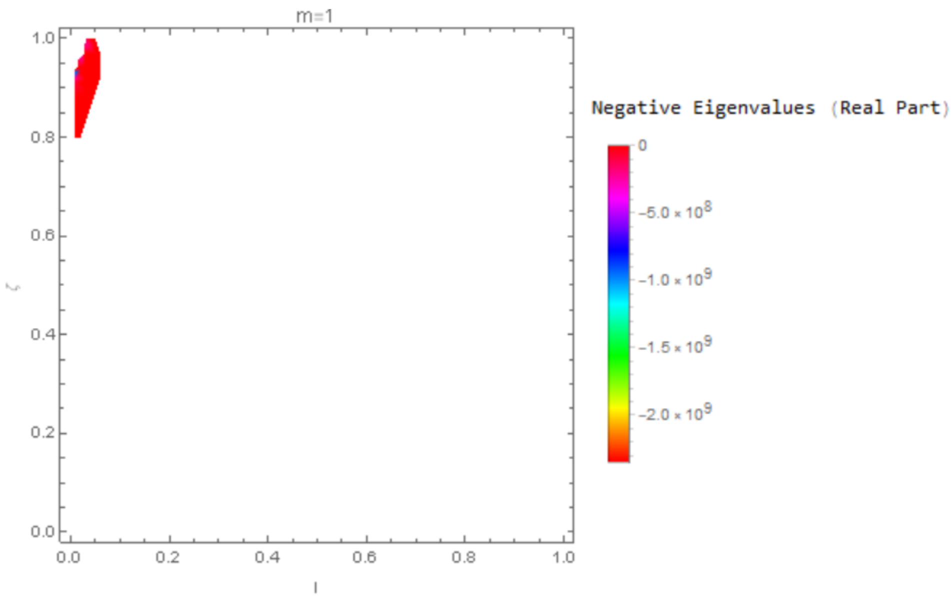


Figure 2.7: Negative eigenvalues of the cost function Hessian for our novel distributed filtering algorithm with $N = 3$, $r = 25, q = 1, \sigma_c^2 = 1$, and consensus rounds $m = 1$ for an undirected line graph.

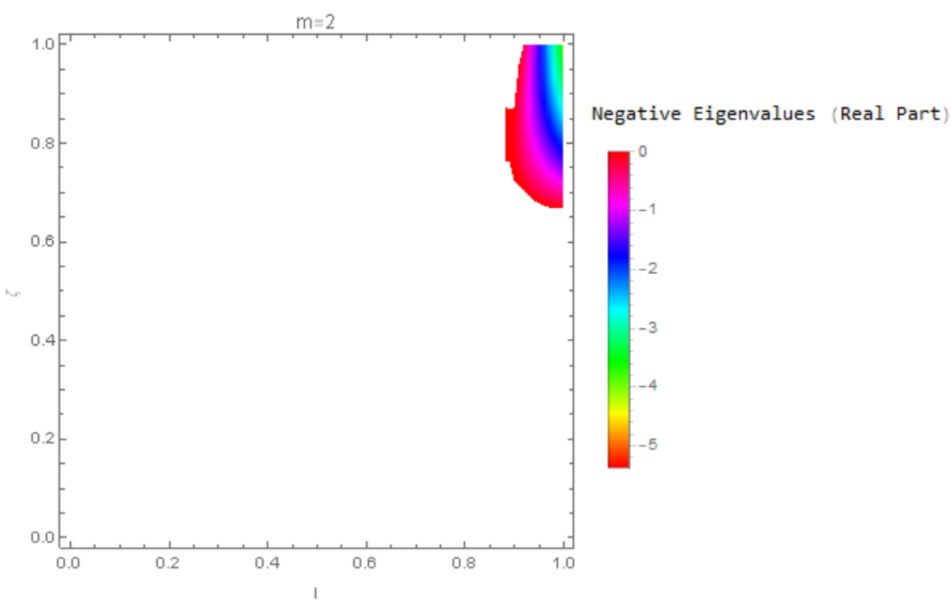


Figure 2.8: Negative eigenvalues of the cost function Hessian for our novel distributed filtering algorithm with $N = 3$, $r = 25, q = 1, \sigma_c^2 = 1$, and consensus rounds $m = 2$ for an undirected line graph.

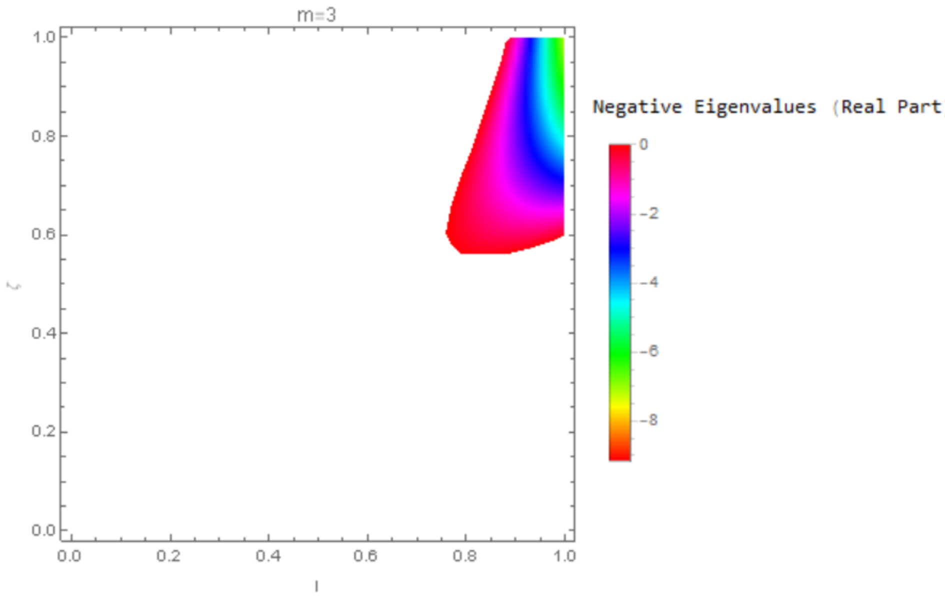


Figure 2.9: Negative eigenvalues of the cost function Hessian for our novel distributed filtering algorithm with $N = 3$, $r = 25, q = 1, \sigma_c^2 = 1$, and consensus rounds $m = 3$ for an undirected line graph.

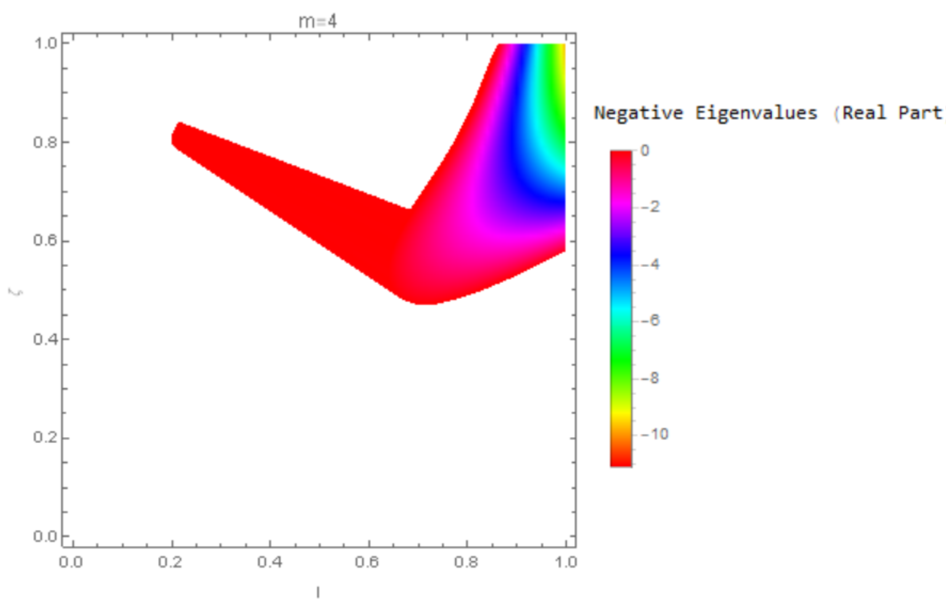


Figure 2.10: Negative eigenvalues of the cost function Hessian for our novel distributed filtering algorithm with $N = 3$, $r = 25, q = 1, \sigma_c^2 = 1$, and consensus rounds $m = 4$ for an undirected line graph.

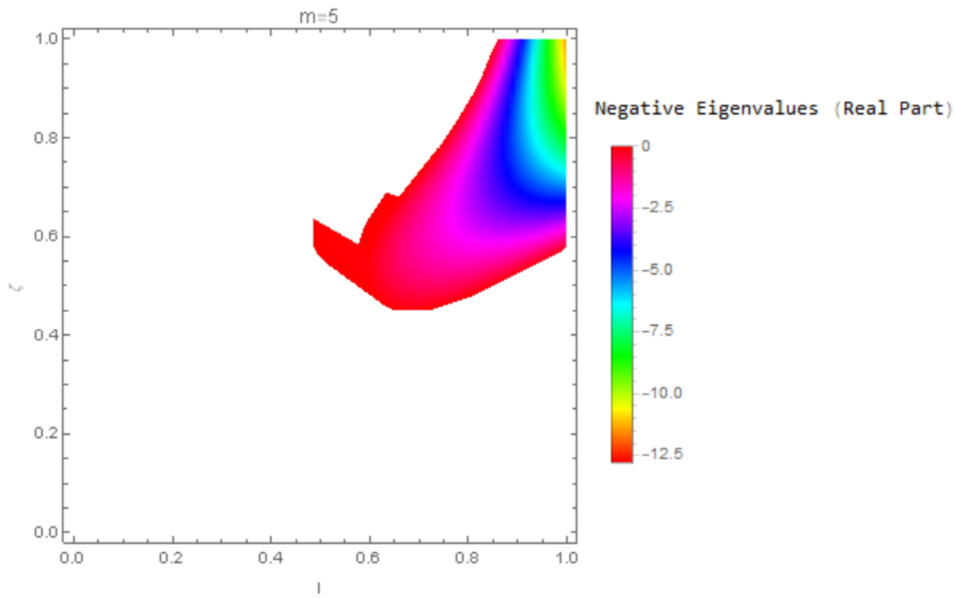


Figure 2.11: Negative eigenvalues of the cost function Hessian for our novel distributed filtering algorithm with $N = 3$, $r = 25, q = 1, \sigma_c^2 = 1$, and consensus rounds $m = 5$ for an undirected line graph.

The existence of negative eigenvalues means that the Hessian is not positive semidefinite on A and indicates that the trace of the asymptotic error covariance matrix is not convex with respect to the parameters ℓ and ζ. However, we can further investigate if it may be biconvex in the parameters of interest. The function $J(\ell, \zeta) : A \to \mathbb{R}$ is a biconvex function if, when fixing ℓ, the function $J_\ell(\zeta) = J(\ell, \zeta)$ is convex over Z and similarly, when fixing ζ, the function $J_\zeta(\ell) = J(\ell, \zeta)$ is convex over L. Figures (2.12) - (2.14) show how the second derivative $\frac{d^2 J}{d\ell^2}$ varies over L for fixed values of ζ and for $m = 5$ consensus rounds.

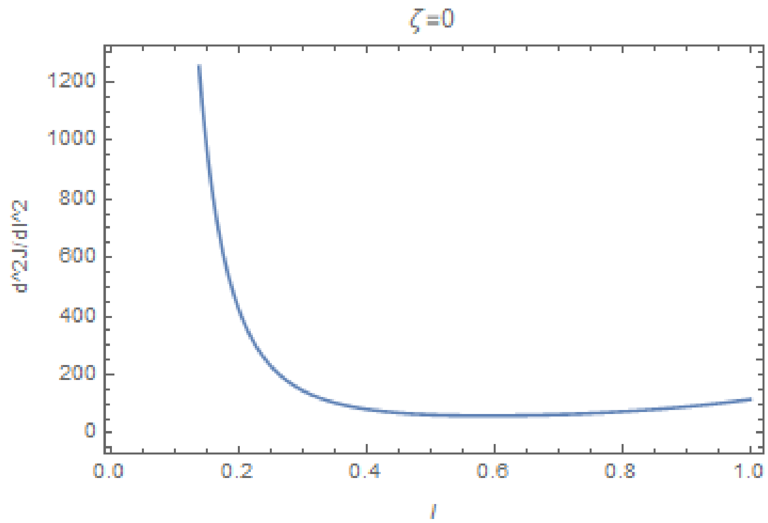


Figure 2.12: Second derivative $\frac{d^2J}{d\ell^2}$ for our novel distributed filtering algorithm with $N = 3, r = 25, q = 1, \sigma_c^2 = 1$, consensus rounds $m = 5$, and $\zeta = 0$ for an undirected line graph.

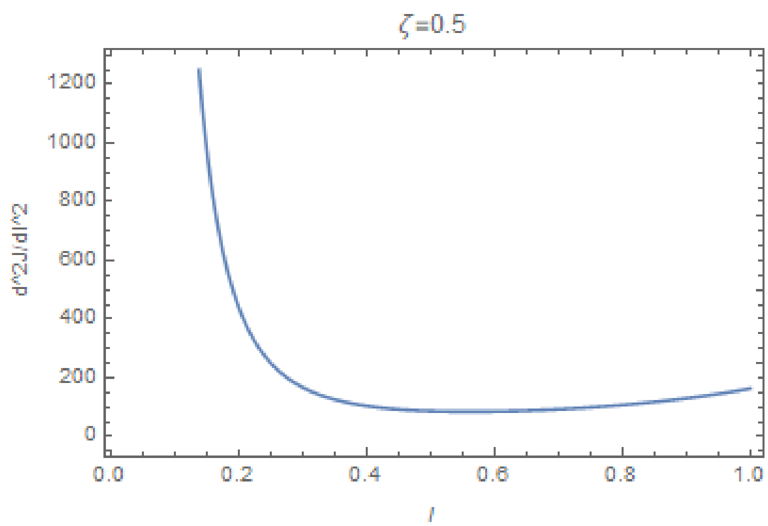


Figure 2.13: Second derivative $\frac{d^2J}{d\ell^2}$ for our novel distributed filtering algorithm with $N = 3, r = 25, q = 1, \sigma_c^2 = 1$, consensus rounds $m = 5$, and $\zeta = 0.5$ for an undirected line graph.

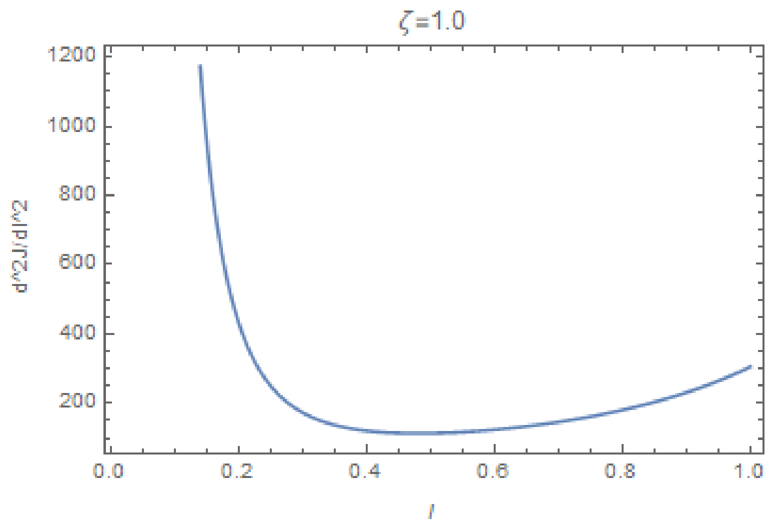


Figure 2.14: Second derivative $\frac{d^2J}{d\ell^2}$ for our novel distributed filtering algorithm with $N = 3, r = 25, q = 1, \sigma_c^2 = 1$, consensus rounds $m = 5$, and $\zeta = 1$ for an undirected line graph.

Similarly, Figures (2.15) - (2.17) show how the second derivative $\frac{d^2J}{d\zeta^2}$ varies over Z for fixed values of ℓ and for $m = 5$ consensus rounds.

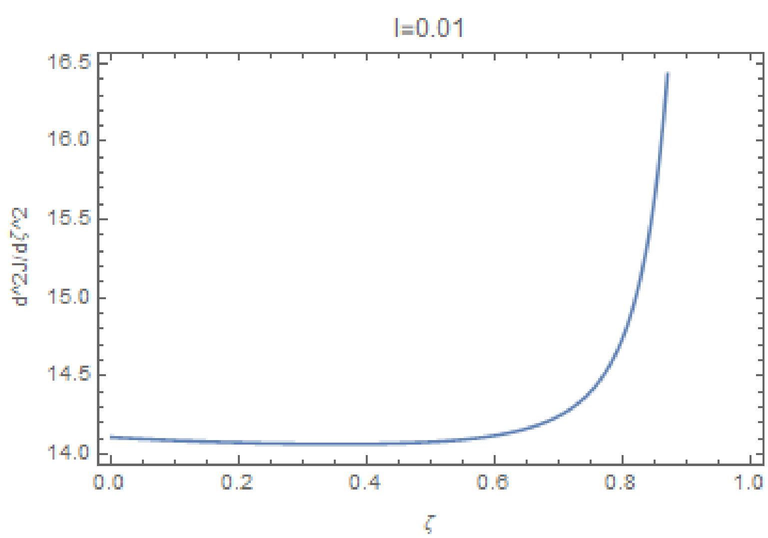


Figure 2.15: Second derivative $\frac{d^2J}{d\zeta^2}$ for our novel distributed filtering algorithm with $N = 3, r = 25, q = 1, \sigma_c^2 = 1$, consensus rounds $m = 5$, and $\ell = 0.01$ for an undirected line graph.

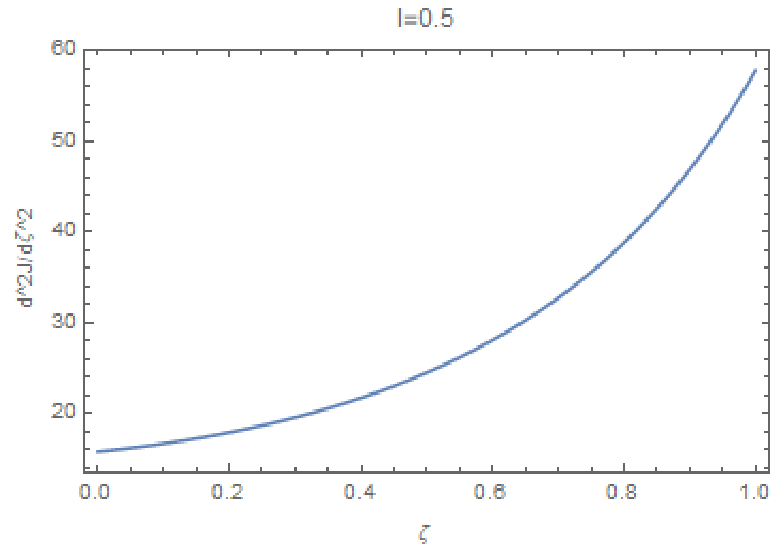


Figure 2.16: Second derivative $\frac{d^2J}{d\zeta^2}$ for our novel distributed filtering algorithm with $N = 3, r = 25, q = 1, \sigma_c^2 = 1$, consensus rounds $m = 5$, and $\ell = 0.5$ for an undirected line graph.

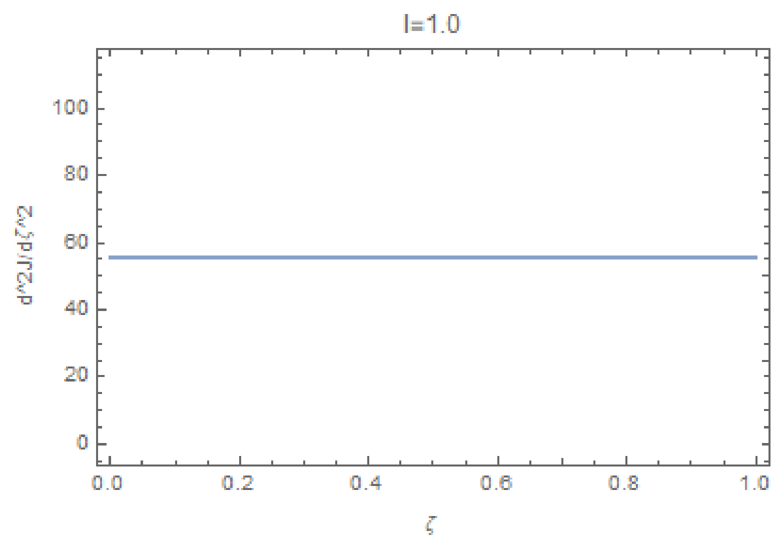


Figure 2.17: Second derivative $\frac{d^2J}{d\zeta^2}$ for our novel distributed filtering algorithm with $N = 3, r = 25, q = 1, \sigma_c^2 = 1$, consensus rounds $m = 5$, and $\ell = 1$ for an undirected line graph.

We observe that for the tested cases, the cost function $J(\ell, \zeta)$ appears to be biconvex in ℓ and ζ.

We must stress that this is a very preliminary analysis of the biconvexity of the cost function which only looks at a simple line graph with three nodes. Further analysis is needed to prove biconvexity over a larger operating regime involving more general networks. Nevertheless, assuming that biconvexity can be shown, there are methods to obtain the global minimum[73].

3

Distributed Control and Estimation of a Linear Time-invariant System

We now investigate the problem of distributed estimation *and* control in a networked multiagent system of a multidimensional linear process. In this chapter we summarize results presented in Part II: Chapter 9 which appears as Savas, Park, Poor, and Leonard[24]. In Section 3.1 we formulate our main problem and state conditions under which we are able to design pertinent estimation and control parameters separately from one another. In Section 3.2 we propose key LMI formulations which solve our problem and also analyze how the frequency of information exchange through linear consensus affects the feasibility of the LMI. In Section 3.3 we validate our main results through simulations of a multi-vehicle platooning example.

In Section 3.4 we introduce new results, not published in [24] of Chapter 9, which provide a cursory look at the role of the network structure on the performance of our algorithm. This notion of how the network affects our distributed control algorithm, along with how an actuator or sensor's location in the network influences the performance of the network as a whole, is explored more in depth in Chapters 4 and 5.

3.1 Problem Description

Consider a discrete-time LTI system given by

$$x(k+1)=Ax(k) + \textstyle\sum_{i=1}^{N} B_i u_i(k),\ x(0) \in \mathbb{R}^n \tag{3.1a}$$

$$y_i(k) = C_i x(k),\ i \in \{1, \cdots, N\} \tag{3.1b}$$

where $x(k) \in \mathbb{R}^n$ is the state, $u_i(k) \in \mathbb{R}^{q_i}$ is the i-th input, and $y_i(k) \in \mathbb{R}^{r_i}$ is the i-th output of the system. The network is comprised of N agents with each agent i having output $y_i(k)$ and control input $u_i(k)$. The agents can also communicate over a fixed directed graph $\mathcal{G} = (\mathbb{V}, \mathbb{E})$. Each vertex i in $\mathbb{V} = \{1, \cdots, N\}$ represents agent i and each edge $(j, i) \in \mathbb{E}$ indicates that agent j can transmit information to agent i. The neighborhood set $\mathbb{N}_i = \{j \in$

$\mathbb{V} \mid (j, i) \in \mathbb{E}\}$ specifies a subset of agents that can transmit information to agent i.

We assume that $\mathcal{G}$ is strongly connected, meaning that there is an edge path from each agent to every other agent. We further assume that (3.1) is jointly controllable and observable, meaning that both pairs (A, B) and (C, A) are controllable and observable, respectively, where $B = (B_1, \cdots, B_N)$ and $C = (C_1^T, \cdots, C_N^T)^T$. To make clear what we mean by *controllable* and *observable* we leverage the following definitions.

Definition 1 (Controllable subspace,[49]). *Given two times $k_1 > k_0 \geq 0$, the controllable subspace $\mathcal{C}[k_0, k_1]$ consists of all states x_0 for which there exists an input $u : [k_0, k_1] \to \mathbb{R}^q$ that transfers the state from $x(k_0) = x_0$ to $x(k_1) = 0$.*

Definition 2 (Controllable system,[49]). *Given two times $k_1 > k_0 \geq 0$, the pair (A, B) is controllable on $[k_0, k_1]$ if $\mathcal{C}[k_0, k_1] = \mathbb{R}^n$, i.e., if every state can be transferred to the origin.*

Furthermore, there is a *duality* between controllability and observability. A pair (C, A) is observable if and only if the dual pair (A^T, C^T) is controllable[49]. In a more descriptive sense, a system is observable if one can reconstruct an initial state $x(k_0)$ given future input and output data $u(k)$ and $y(k)$ for $k \in [k_0, k_1]$ for some k_0 and k_1. However, we note that in our problem setting individual agents do not necessarily have full controllability or observability of the system. This means that for every i in $\mathbb{V}$, pairs (A, B_i) and (C_i, A) may not be controllable and observable, respectively. Hence, without communication with others, each agent can neither estimate the full state of the system nor stabilize it. Below we provide an example of (3.1) and $\mathcal{G}$.

Example 1. *Consider a system of N vehicles moving on the plane where each vehicle has control over its own motion and can observe its own position. This is an example of a jointly controllable* *and jointly observable system. The state and parameters of the system's model* (3.1) *are given as*

$$x = (p_1^T, v_1^T, \cdots, p_N^T, v_N^T)^T \tag{3.2a}$$

$$A = I_N \otimes \begin{pmatrix} I_2 & 0.5 I_2 \\ 0 & I_2 \end{pmatrix} \tag{3.2b}$$

$$B_i = e_i \otimes \begin{pmatrix} 0 \\ I_2 \end{pmatrix}, \ C_i = e_i^T \otimes \begin{pmatrix} I_2 & 0 \end{pmatrix}, \ i \in \{1, \cdots, N\} \tag{3.2c}$$

where $p_i, v_i \in \mathbb{R}^2$ *are, respectively, the position and velocity of the i-th vehicle and* e_i *is a canonical basis in* $\mathbb{R}^N$ *whose elements are all zero except the i-th element, which is* 1*. In Section 3.3 we use this example to illustrate our main results. In particular, we consider line and ring graphs and observe how the graph structure affects key algorithm parameters. We then apply our framework to a multi-vehicle formation control problem utilizing a nonlinear control law.*

Parameter	**Definition**
n	the order of the linear time-invariant system
q_i, r_i	the dimensions of the i-th input and output, respectively
N	the number of agents in the network
m	the number of rounds of linear consensus
e_i	the canonical basis with i-th entry equal to 1, rest equal to 0
$\otimes$	the Kronecker product

Table 3.1: List of basic notation.

A summary of basic notation is provided in Table 3.1. Each agent i computes a state estimate $\hat{x}_i(k)$, uses local communication to exchange and fuse the estimate with those of its

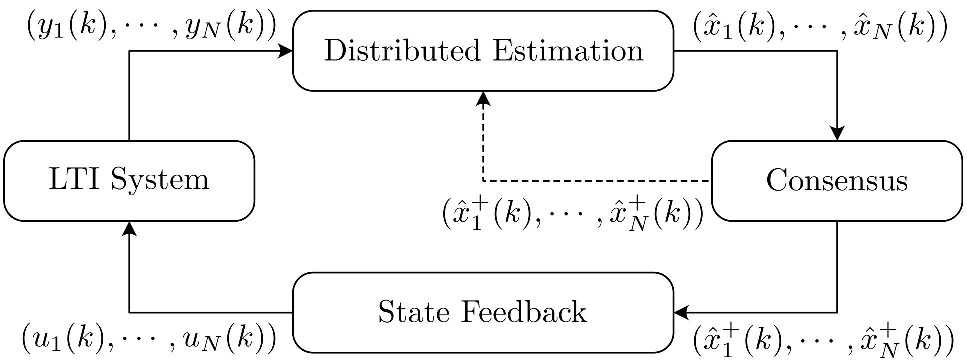


Figure 3.1: A diagram illustrating the closed loop consisting of the LTI system, distributed estimation, (linear) consensus, and state feedback. Repeated from Figure (9.1).

neighbors, and then uses the fused state estimate to compute a control input $u_i(k)$. The goal is to design the network of agents such that the control actions stabilize the system. To achieve this, we implement state feedback, distributed estimation, and linear consensus at each agent (see Figure 3.1 for an illustration of the closed loop consisting of the three components and the LTI system).

State feedback Let $\hat{x}_i(k)$ be the state estimate of agent i. The agent computes its control input $u_i(k)$ according to

$$u_i(k) = K_i \hat{x}_i(k). \tag{3.3}$$

In this framework, the agent needs to utilize only its own state estimate to compute $u_i(k)$. We further assume that $\{K_i\}_{i \in \mathbb{V}}$ satisfy that $A + \sum_{i \in \mathbb{V}} B_i K_i$ is Schur stable with eigenvalues inside the unit circle in the complex plane.

Distributed estimation Agent i computes its state estimate $\hat{x}_i(k)$ by recursively updating it according to

$$\begin{aligned}\hat{x}_i(k+1) &= A\hat{x}_i(k) + L_i\left(y_i(k) - C_i\hat{x}_i(k)\right) \\ &\quad + \textstyle\sum_{j\in\mathbb{V}} B_jK_j\hat{x}_i(k) + \sum_{j\in\mathbb{N}_i} W_{ij}\left(\hat{x}_j(k) - \hat{x}_i(k)\right), \end{aligned} \quad (3.4)$$

where K_i is the control gain matrix in (3.3) and $W_{ij} \in \mathbb{R}^{n\times n}$, $L_i \in \mathbb{R}^{n\times r_i}$ are the parameters to be designed. The formulation (3.4), which is motivated by the existing distributed estimation approaches proposed, for instance, in [53,75,76,77], utilizes the partial output $y_i(k)$, the state estimates $\{\hat{x}_j(k)\}_{j\in\mathbb{N}_i}$ from the agent's neighbors, and the estimate $\sum_{j\in\mathbb{V}} B_jK_j\hat{x}_i(k)$ of the control input applied to (3.1) in order to update an agent's state estimate $\hat{x}_i(k)$. Importantly, each agent must estimate the control input applied to (3.1) using only locally available information. According to (3.3), the control input applied to (3.1) is given by $\sum_{j\in\mathbb{V}} B_jK_j\hat{x}_j(k)$ which depends on every agent j's state estimate. However, an individual agent does not have access to the estimates of the other agents in the network. Due to this, the agent uses only its own state estimate in lieu of those of all other agents. With a well-designed algorithm, the estimate errors will converge to zero, validating such a framework.

m-round linear consensus This framework allows additional information exchange between agents in the form of m-round linear consensus. The agents can utilize consensus to exchange and fuse their state estimates with neighbors. Letting $\{\hat{x}_j(k)\}_{j\in\mathbb{V}}$ be the state estimates of the agents at the beginning of the linear consensus, the output $\hat{x}_i^+(k)$ at each agent i is determined as follows:

$$\hat{x}_i^+(k) = \sum_{j \in \mathbb{V}} \bar{P}_{ij} \hat{x}_j(k). \tag{3.5}$$

P is a stochastic matrix that conforms with $\mathcal{G} = (\mathbb{V}, \mathbb{E})$. In other words, if an edge $(j, i) \notin \mathbb{E}$, then $P_{ij} = 0$. We then define $\bar{P} = P^m \in \mathbb{R}^{N \times N}$, where m is a non-negative integer. Note that the parameter $\bar{P}_{ij} \geq 0$ for each $i, j \in \mathbb{V}$. We see that $\hat{x}_i^+(k)$ is thus agent i's updated state estimate after applying the m *consensus rounds* with *consensus matrix* P. Each agent's state estimate is then updated as $\hat{x}_i(k) = \hat{x}_i^+(k)$ and subsequently fed into (3.3) and (3.4). This is illustrated in Figure 3.1.

As previously mentioned, in order for this algorithm to be well-designed we require the state estimates $\hat{x}_i(k)$ to converge to the true state $x(k)$ for each agent i in $\mathbb{V}$. In the absence of state feedback, the work of[83,75,76] presents technical conditions on the system (3.1) and the graph $\mathcal{G}$ under which there are parameters $\{W_{ij}\}_{i,j \in \mathbb{V}}$, $\{L_i\}_{i \in \mathbb{V}}$ that ensure the convergence of that state estimates of each agent to the true state. When also considering state feedback, more recent work in[77] describes how to jointly compute $\{K_i\}_{i \in \mathbb{V}}, \{W_{ij}\}_{i,j \in \mathbb{V}}, and \{L_i\}_{i \in \mathbb{V}}$ to stabilize (3.1) using (3.3) and (3.4).

Similarly to the problem studied in[77], we investigate how to design the state feedback and distributed estimation to achieve system stabilization. However, our work is distinct from[77]. We specifically consider the case in which the gain matrices $\{K_i\}_{i \in \mathbb{V}}$ are designed *independently* of the parameters $\{W_{ij}\}_{i,j \in \mathbb{V}}$ and $\{L_i\}_{i \in \mathbb{V}}$. We provide technical conditions under which the network of agents stabilizes (3.1) and utilize a framework of two decoupled numerical methods to compute $\{K_i\}_{i \in \mathbb{V}}$ and $\{W_{ij}\}_{i,j \in \mathbb{V}}$, $\{L_i\}_{i \in \mathbb{V}}$ which achieves this stabilization. Being able to separately design the system parameters is a nontrivial result. This allows for more freedom in design such that updating a parameter in the control scheme, for instance, does not require one to then redesign the estimation scheme. This is very important due to the nature of the numerical methods utilized. As we describe in Section 3.3.1, finding the parameters for (3.4) involves finding a solution to a large-size linear matrix inequality, which can be computationally expensive. For this reason, whenever possible, it is preferred not to re-compute the parameters of (3.4) when the state feedback is revised. Our results can also be applied to scenarios in which nonlinear state feedback is adopted, as discussed in Section 3.2.3. In such cases, analysis such as in[77] cannot be directly applied.

We also highlight that our framework considers communication between agents at discrete-times, with further information exchange allowed through utilization of m-round linear consensus. Our methodology and main results can thus be applied to varied engineering problems in which agents might only be able to exchange information a limited number of times over finite time intervals. This makes our problem setting distinct from that investigated in[82,80].

We formalize our main problem as follows.

Problem 1. *For fixed* $m \geq 0$*, compute the parameters* $\{W_{ij}\}_{i,j\in\mathbb{V}}$*,* $\{L_i\}_{i\in\mathbb{V}}$ *and identify the set of state feedback gains* $\{K_i\}_{i\in\mathbb{V}}$ *for which the control inputs determined by* (3.3) *and* (3.4) *stabilize the system* (3.1)*.*

3.2 Parameter Design for State Feedback and Distributed Estimation

As mentioned previously, our goal is to compute the parameters of (3.4) that result in the stability of (3.1). We begin by considering the case with no linear consensus ($m = 0$), and provide analysis regarding the general case, with $m \geq 0$, in Section 3.2.2.

We define the estimation error for each agent as $\tilde{x}_i(k) = x(k) - \hat{x}_i(k)$. Using (3.1), (3.3)-(3.5), the state equation for the error $\tilde{x}_i(k)$ can be derived as follows:

$$\begin{aligned}\tilde{x}_i(k+1) = (A - L_i C_i)\, \tilde{x}_i(k) + \textstyle\sum_{j\in\mathbb{V}} B_j K_j \left(\tilde{x}_i(k) - \tilde{x}_j(k)\right) \\ + \textstyle\sum_{j\in\mathbb{N}_i} W_{ij} \left(\tilde{x}_j(k) - \tilde{x}_i(k)\right). \end{aligned} \quad (3.6)$$

We seek to provide a framework in which we can design pertinent control and estimation parameters independently. As a means to achieve this, we cast (3.6) as a feedback interconnection of two components – the *control component* (3.7) and the *estimation component* (3.8) which are defined below. We then find sufficient conditions for the feedback interconnection of the two components to attain the convergence $\lim_{k\to\infty} \|\tilde{x}_i(k)\|_2 = 0$ for each agent $i \in \mathbb{V}$. This ensures that each agent's estimation error converges to zero. We then use this result to address Problem 1 and, given a set of state feedback gains, find the parameters $\{W_{ij}\}_{i,j\in\mathbb{V}}$ and $\{L_i\}_{i\in\mathbb{V}}$ which stabilize (3.1).

We first split up the parameters W_{ij} into two parts, one of which will be utilized in the control component and the other which will be utilized by the estimation component. Let $W_{ij} = W_{ij}^{\mathrm{E}} + W_{ij}^{\mathrm{C}}$ and define

$$\tilde{v}_i(k) = \textstyle\sum_{j\in\mathbb{V}} B_j K_j(\tilde{x}_i(k) - \tilde{x}_j(k)) - \sum_{j\in\mathbb{N}_i} W_{ij}^{\mathrm{C}}(\tilde{x}_i(k) - \tilde{x}_j(k)) \quad (3.7)$$

$$\tilde{x}_i(k+1) = (A - L_i C_i)\, \tilde{x}_i(k) - \textstyle\sum_{j\in\mathbb{N}_i} W_{ij}^{\mathrm{E}}(\tilde{x}_i(k) - \tilde{x}_j(k)) + \tilde{v}_i(k). \quad (3.8)$$

The first thing to note is that the feedback interconnection of (3.7) and (3.8) is equivalent to (3.6). We have just used $\tilde{v}_i(k)$ as a means to contain all pertinent parameters related to the control aspect of our problem. Let us first consider (3.7). The first term $\sum_{j\in\mathbb{V}} B_j K_j(\tilde{x}_i(k) - \tilde{x}_j(k))$ is an indication of the error in agent i's estimate of the control action applied to the system. We then use $\sum_{j\in\mathbb{N}_i} W_{ij}^{\mathrm{C}}(\tilde{x}_i(k) - \tilde{x}_j(k))$ as a means to counteract this error. At first glance it might seem ideal to select $W_{ij}^{\mathrm{C}} = B_j K_j$. However, such choice of W_{ij}^{C} turns out to not be optimal in our formulation. We explore what we mean by this more in Section 3.4. In Section 3.3, we use a linear matrix inequality (LMI) to construct an appropriate optimization problem and solve for the most advantageous W_{ij}^{C}. Now considering (3.8), we see that it is equivalent to (3.6) except that we are now representing the control input estimation error term by $\tilde{v}_i(k)$ and instead adopt $\{W_{ij}^{\mathrm{E}}\}_{i,j\in\mathbb{V}}$ in place of $\{W_{ij}\}_{i,j\in\mathbb{V}}$.

The key to being able to separately design the control and estimation parameters is through the use of the small-gain theorem [84, Chapter 5.4] to specify conditions on the parameter selection that ensure the convergence in (3.6). To achieve this, we represent the estimation component (3.8) as an LTI system with state $\tilde{x}(k) = (\tilde{x}_1(k), \cdots, \tilde{x}_N(k)) \in \mathbb{R}^{nN}$ and input $\tilde{v}(k) = (\tilde{v}_1(k), \cdots, \tilde{v}_N(k)) \in \mathbb{R}^{nN}$ as follows:

$$\tilde{x}(k+1) = \tilde{A}\tilde{x}(k) + \tilde{v}(k), \tag{3.9}$$

where $\tilde{A} \in \mathbb{R}^{nN\times nN}$ is defined as

$$\tilde{A} = \mathrm{diag}(A - L_1 C_1, \cdots, A - L_N C_N) + W^{\mathrm{E}} \tag{3.10}$$

and W^{E} is a block matrix whose i,j-th block element is

$$[W^{\mathrm{E}}]_{ij} = \begin{cases} W^{\mathrm{E}}_{ij} & \text{if } j \in \mathbb{N}_i \backslash \{i\} \\ -\sum_{l \in \mathbb{N}_i \backslash \{i\}} W^{\mathrm{E}}_{il} & \text{if } j = i \\ 0_n & \text{otherwise.} \end{cases} \tag{3.11}$$

Also, we rewrite the control component (3.7) as

$$\tilde{v}(k) = \tilde{D}\tilde{x}(k), \tag{3.12}$$

where $\tilde{D}$ is a block matrix whose i,j-th block element is

$$[\tilde{D}]_{ij} = \begin{cases} -B_j K_j + W^{\mathrm{C}}_{ij} & \text{if } j \in \mathbb{N}_i \backslash \{i\} \\ \sum_{l \in \mathbb{V} \backslash \{i\}} B_l K_l - \sum_{l \in \mathbb{N}_i \backslash \{i\}} W^{\mathrm{C}}_{il} & \text{if } j = i \\ -B_j K_j & \text{otherwise.} \end{cases}$$

3.2.1 LMI Formulation for Parameter Design

Let $\tilde{G}$ be the (input-to-state) transfer function matrix of (3.9). As an application of the small-gain theorem [84, Chapter 5.4], the feedback interconnection of the estimation component (3.9) and the control component (3.12) is L_2-stable if it holds that $\|\tilde{G}\|_{H_\infty} \|\tilde{D}\|_2 < 1$.

Let us consider a more general condition $\|\tilde{G}\|_{H_\infty} < \gamma$ for some positive real γ. It is well-known that the bounded real lemma for discrete-time LTI systems [85,86] can be used to establish the following equivalence:

$$\|\tilde{G}\|_{H_\infty} < \gamma \Leftrightarrow \begin{pmatrix} -X & X\tilde{A} & I_{nN} & 0_{nN} \\ \tilde{A}^T X & -X & 0_{nN} & X \\ I_{nN} & 0_{nN} & -\gamma I_{nN} & 0_{nN} \\ 0_{nN} & X & 0_{nN} & -\gamma I_{nN} \end{pmatrix} \prec 0 \tag{3.13}$$

where γ is a positive real number and $X \in \mathbb{R}^{nN\times nN}$ is a symmetric and positive-definite matrix. In the following lemma, we provide a sufficient condition under which a solution X, γ exists for (3.13).

Lemma 1 ([24]). *Suppose that there are $\left\{W_{ij}^E\right\}_{i,j\in\mathbb{V}}$, $\{L_i\}_{i\in\mathbb{V}}$ for which $\tilde{A}$ given in* (3.10) *is Schur stable. Then, a solution $X = X^T \succ 0, \gamma > 0$ exists for* (3.13).

Remark 1. *In Lemma 1 we operate under the assumption that parameters $\left\{W_{ij}^E\right\}_{i,j\in\mathbb{V}}$, $\{L_i\}_{i\in\mathbb{V}}$ exist for which $\tilde{A}$ is Schur stable. However, this assumption, given the specifications of our problem, is justified. The results of[76,83,75] from the distributed estimation literature address the existence of the parameters $\left\{W_{ij}^E\right\}_{i,j\in\mathbb{V}}$, $\{L_i\}_{i\in\mathbb{V}}$ for which $\tilde{A}$ is Schur stable when the system* (3.1) *is jointly observable and the graph $\mathcal{G}$ is strongly connected. The result of[76,83] is based on state augmentation and that of[75] leverages the system model structure* (3.1) *and connectivity of the underlying graph. With these results, Lemma 1 implies that the joint observability of* (3.1) *and the strong connectivity of $\mathcal{G}$ are sufficient for the LMI* (3.13) *to have a solution.*

We now use Lemma 1 as a basis to address our main problem of finding estimation and control parameters to stabilize the system (3.1). First, given $\gamma_2 > 0$, we define

$$\mathbb{K}_{\gamma_2} = \{\{K_i\}_{i\in\mathbb{V}} \,|\, A + \textstyle\sum_{i\in\mathbb{V}} B_i K_i \text{ is Schur stable}, \min_{\{W^C_{ij}\}_{i,j\in\mathbb{V}}} \|\tilde{D}\|_2 < \gamma_2\}.$$

The set $\mathbb{K}_{\gamma_2}$ refers to the control gains $\{K_i\}_{i\in\mathbb{V}}$ for which $\|\tilde{D}\|_2 < \gamma_2$, where the design parameters $\{W^C_{ij}\}_{i,j\in\mathbb{V}}$ are selected to minimize the norm. In addition, these control gains must also satisfy $A + \sum_{i\in\mathbb{V}} B_i K_i$ being Schur stable. Given this definition, we have the following Theorem.

Theorem 2 ([24]). *Suppose given parameters $\{W^E_{ij}\}_{i,j\in\mathbb{V}}$, $\{L_i\}_{i\in\mathbb{V}}$ of the estimation component $\tilde{G}$ satisfy $\|\tilde{G}\|_{H_\infty} < \gamma$, e.g.,* (3.13) *has a solution with the same γ. Assuming that $\mathbb{K}_{\gamma^{-1}}$ is nonempty, for any state feedback gains $\{K_i\}_{i\in\mathbb{V}}$ belonging to $\mathbb{K}_{\gamma^{-1}}$ there is $\{W^C_{ij}\}_{i,j\in\mathbb{V}}$ such that the control inputs determined by* (3.3) *and* (3.4) *stabilize the system* (3.1)*.*

Note that it is possible that $\mathbb{K}_{\gamma^{-1}}$ is an empty set if γ in the statement of Theorem 2 is too large. In other words, when the H_∞-norm of the estimation component is too large, there is no control gain that stabilizes (3.1) while satisfying the inequality $\min_{\{W^C_{ij}\}_{i,j\in\mathbb{V}}} \|\tilde{D}\|_2 < \gamma^{-1}$ for the small-gain theorem to hold. This is where utilizing m-round linear consensus becomes critical. In Section 3.2.2, we show that with more frequent information exchange the set $\mathbb{K}_{\gamma^{-1}}$ becomes larger. In other words, the agents have more options to select state feedback gains that stabilize (3.1) and satisfy the inequality condition for the small-gain theorem to hold. In effect, linear consensus is used to extend the operating regime of our algorithm.

3.2.2 Effect of Linear Consensus on Stability

We now consider the case in which agents can utilize a consensus matrix P to fuse their state estimates with estimates of the neighbors using m-round linear consensus (3.5). Without loss of generality, we assume that m is an even number, $\mathcal{G}$ is undirected, and P is symmetric. We define $\tilde{x}'(k) = Q^{m/2}\tilde{x}(k)$ and $\tilde{v}'(k) = Q^{m/2}\tilde{v}(k)$, where $Q = P \otimes I_n$. Note that we can handle the case of odd m by setting $\tilde{x}'(k) = Q^{\lfloor m/2 \rfloor}\tilde{x}(k)$ and $\tilde{v}'(k) = Q^{\lfloor m/2 \rfloor}\tilde{v}(k)$ where the floor function $\lfloor \cdot \rfloor$ is given by $\lfloor x \rfloor = \max\{y \in \mathbb{Z} \mid y \le x\}$. By using (3.1), (3.3)-(3.5) and following similar steps to obtain (3.9) and (3.12) in Section 3.2, we can derive the state equations for the control and estimation components as follows:

$$\tilde{v}'(k) = Q^{m/2}\tilde{D}Q^{m/2}\tilde{x}'(k) \tag{3.14}$$

$$\tilde{x}'(k+1) = Q^{m/2}\tilde{A}Q^{m/2}\tilde{x}'(k) + \tilde{v}'(k) \tag{3.15}$$

We can now refine the definition of $\mathbb{K}_{\gamma_2}$ under the paradigm of linear consensus. Given $\gamma_2 > 0$,

$$\mathbb{K}_{\gamma_2,m} = \{\{K_i\}_{i\in\mathbb{V}} \mid A + \textstyle\sum_{i\in\mathbb{V}} B_i K_i \text{ is Schur stable, } \min_{\{W_{ij}^{\mathrm{C}}\}_{i,j\in\mathbb{V}}} \|Q^{m/2}\tilde{D}Q^{m/2}\|_2 < \gamma_2\}.$$

Similarly, we can extend Theorem 2 to the case where agents are able to employ m-rounds of linear consensus.

Theorem 3 ([24]). *For any given $\gamma_2 > 0$, there is $m^* \ge 0$ for which $\mathbb{K}_{\gamma_2,m}$ is non-empty for $m \ge m^*$. For sufficiently large m, we can design parameters $\{W_{ij}\}_{i,j\in\mathbb{V}}$, $\{L_i\}_{i\in\mathbb{V}}$ for which the LTI system (3.1) is stable with any $\{K_i\}_{i\in\mathbb{V}}$ belonging to $\mathbb{K}_{\gamma^{-1},m}$, where γ is the H_∞-norm of*

(3.15).

This is a powerful result regarding the applicability of our algorithm. If a sufficiently large number of consensus rounds m are able to be performed, Theorem 3 implies that $\mathbb{K}_{\gamma^{-1},m}$ will be nonempty and we can always find gains $\{K_i\}_{i\in\mathbb{V}}$ to stabilize the system. This also implies that by increasing m we can design the parameters $\{W_{ij}\}_{i,j\in\mathbb{V}}$, $\{L_i\}_{i\in\mathbb{V}}$ to allow the agents to adopt arbitrarily large state feedback gains $\{K_i\}_{i\in\mathbb{V}}$.

We now detail how our algorithm can be extended to employ nonlinear control.

3.2.3 Extension

Suppose that (3.3) consists of linear and nonlinear parts:

$$u_i(k) = K_i\hat{x}_i(k) + \mu_i\left(\hat{x}_i(k)\right). \tag{3.16}$$

This type of a control law could be utilized to maneuver multiple vehicles as in Example 1. The linear part $K_i\hat{x}_i(k)$ can be used for vehicle formation control. The nonlinear part $\mu_i(\hat{x}_i(k))$ can then be used for obstacle avoidance as the nonlinearities allow for a more reactive action. With the nonlinear feedback, we can represent (3.4) by the following two components:

$$\tilde{v}_i(k) \;=\; \textstyle\sum_{j\in\mathbb{V}} B_j(\mu_j(\hat{x}_i(k)) \;-\; \mu_j(\hat{x}_j(k))) \;-\; \sum_{j\in\mathbb{N}_i} B_j(\mu_j(\hat{x}_i(k)) \;-\; \mu_j(\hat{x}_j(k))) \tag{3.17}$$

where we assume that there is a constant γ_{C} for which $\|\tilde{v}(k)\|_2 \leq \gamma_{\mathrm{C}}\|\tilde{x}(k)\|_2$ holds and

$$\tilde{x}_i(k+1) = (A - L_iC_i)\,\tilde{x}_i(k) + \textstyle\sum_{j\in\mathbb{V}} B_jK_j(\tilde{x}_i(k) - \tilde{x}_j(k)) - \sum_{j\in\mathbb{N}_i} W^{\mathrm{E}}_{ij}(\tilde{x}_i(k) - \tilde{x}_j(k)) + \tilde{v}_i(k). \tag{3.18}$$

There are important distinctions with this formulation and that discussed in Section 3.2. Here, we substitute $W_{ij}(\hat{x}_j(k) - \hat{x}_i(k))$ in (3.4) with the following nonlinear function: $W_{ij}(\hat{x}_j(k), \hat{x}_i(k)) = W_{ij}^{\mathrm{E}}(\hat{x}_j(k) - \hat{x}_i(k)) + B_j(\mu_j(\hat{x}_i(k)) - \mu_j(\hat{x}_j(k)))$. This means that there are no longer design parameters W_{ij}^{C}, so there is no optimization being performed for the $\tilde{v}_i(k)$ component. Also, the design of the parameters $\{W_{ij}^{\mathrm{E}}\}_{i,j\mathbb{V}}$, $\{L_i\}_{i\in\mathbb{V}}$ for the estimation component will depend on the *linear part* $\{K_i\}_{i\in\mathbb{V}}$ of (3.16). In this case, the small-gain theorem can be used to establish the stability results as in Theorems 2 and 3 if it holds that $\|\tilde{G}\|_{H_\infty} < \gamma_{\mathrm{C}}^{-1}$, where $\tilde{G}$ is the transfer function of (3.18).

3.3 Simulations

We now consider the LTI system from Example 1 over line and ring graphs along with a multi-vehicle formation control problem to validate our analytical results.

3.3.1 Parameter Design

Recall that the control gains $\{K_i\}_{i\in\mathbb{V}}$ are assumed to be given. As such, the key design parameters in our framework are P, $\left\{W_{ij}^{E}\right\}_{i,j\in\mathbb{V}}$, $\left\{W_{ij}^{C}\right\}_{i,j\in\mathbb{V}}$, and $\{L_i\}_{i\in\mathbb{V}}$. We select P to be a stochastic matrix with smallest second eigenvalue which conforms with $\mathcal{G}$. This allows the agents to fuse the estimates with those of their neighbors as fast as possible. In addition, we can simplify our analysis by choosing $W_{ij}^{E} = P_{ij}A$ as motivated by the approach of[76].

We are then left with the design parameters $\{L_i\}_{i\in\mathbb{V}}$ and $\{W_{ij}^{C}\}_{i,j\in\mathbb{V}}$. Since we want to afford ourselves the largest safety factor when satisfying the small-gain theorem, we seek to compute $\{L_i\}_{i\in\mathbb{V}}$ and $\{W_{ij}^{C}\}_{i,j\in\mathbb{V}}$ that minimize the H_∞-norm of (3.15) and the 2-norm of (3.14), respectively. Utilizing the bounded-real lemma condition for the estimation component, we can minimize the H_∞-norm of (3.15) as

$$\text{minimize}_{\gamma,X,\{L_i\}_{i\in\mathbb{V}}}\gamma \tag{3.19}$$

$$\text{subject to} \begin{pmatrix} -X & X\left(Q^{m/2}\tilde{A}Q^{m/2}\right) & I_{nN} & 0_{nN} \\ \left(Q^{m/2}\tilde{A}Q^{m/2}\right)^T X & -X & 0_{nN} & X \\ I_{nN} & 0_{nN} & -\gamma I_{nN} & 0_{nN} \\ 0_{nN} & X & 0_{nN} & -\gamma I_{nN} \end{pmatrix} \prec 0$$

where $\gamma > 0$, $X = X^T \succ 0$, and $Q = P \otimes I_n$. Note that (3.19) is a non-convex optimization. Similarly, when minimizing the norm of the control component we have that $\|Q^{m/2}\tilde{D}Q^{m/2}\|_2 = \sigma_{max}[Q^{m/2}\tilde{D}Q^{m/2}]$ where $\sigma_{max}[M]$ is the maximum singular value of matrix M. The constraint $\sigma_{max}[Q^{m/2}\tilde{D}Q^{m/2}] < \gamma'$ for positive and real γ' can be written as the convex LMI[50]

$$\begin{pmatrix} \gamma' I_{nN} & Q^{m/2}\tilde{D}Q^{m/2} \\ \left(Q^{m/2}\tilde{D}Q^{m/2}\right)^T & \gamma' I_{nN} \end{pmatrix} \succ 0. \tag{3.20}$$

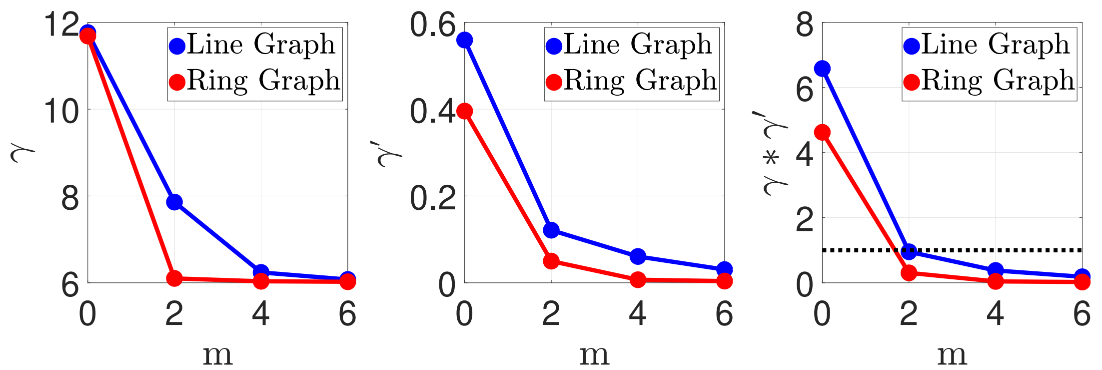
Thus, we can minimize the 2-norm of (3.14) as

$$\text{minimize}_{\gamma',\left\{W_{ij}^C\right\}_{i,j\in\mathbb{V}}}\gamma' \tag{3.21}$$

$$\text{subject to (3.20).}$$

3.3.2 Simulation Results

Consider Example 1 with $N = 4$ agents. In particular, we consider undirected line and ring graphs as possible underlying communication structures. For each graph, we compute the



(a) (b) (c)

Figure 3.2: Plots of the optimal value of (a) γ for the estimation component design (3.19), (b) γ' for the control component design (3.21), and (c) the product $\gamma * \gamma'$, where the critical value of 1 is drawn as a dotted line. Repeated from Figure (9.2).

parameters of (3.4) by solving (3.19) and (3.21) for $m = 0, 2, 4, 6$, where the state feedback gain, motivated by the centralized LQR controller, is given by

$$K_i = e_i^T \otimes \begin{pmatrix} -0.192 & 0 & -0.284 & 0 \\ 0 & -0.192 & 0 & -0.284 \end{pmatrix} \tag{3.22}$$

Here the vector e_i is a canonical basis of dimension 4.

Figure 3.2 depicts the minimal costs γ and γ' obtained in the optimizations (3.19) and (3.21) as the number of consensus rounds m increases over both line and ring graphs. Note that both the estimation and control component norms decrease with increasing consensus rounds. This illustrates how linear consensus can be used to extend the operating regime of our algorithm. If agents are able to communicate more frequently, then a larger safety factor can be established within which the small-gain theorem is satisfied. This affords advantages in parameter design. Given an estimation strategy, with more rounds of consensus m agents have greater flexibility to select state feedback gains which achieve system stabilization. We also note that for a given value of m, the norm values γ and γ' are smaller for the ring graph than for the line graph. Intuitively this makes sense as the ring graph has one additional edge compared to the line graph. This additional edge increases information flow for estimate sharing between agents and results in a faster rate of consensus. Another observation is that the small-gain theorem is not satisfied for all simulated values of m. Recall that we require $\gamma * \gamma' < 1$ for the small-gain theorem to be satisfied. Figure 3.2(c) shows that $m \geq 2$ consensus rounds are needed to meet this criterion for the control gain we select in (3.22). Given these control gains, we see that linear consensus is *necessary* for our algorithm to be stabilizing.

To illustrate the performance of our algorithm using the nonlinear control scheme outlined in Section 3.2.3, we consider a formation control problem. Again, we employ the LTI system from Example 1 with $N = 4$ vehicles. The goal is for the vehicles to move along the x-axis direction at a speed of $0.2\ m/s$ while maintaining a line formation with an inter-vehicle spacing of $2\ m$. We further assume that vehicle 1 is the leader in that it will control for its speed as well as maneuver for obstacle avoidance while the rest of the vehicles control to remain in formation. The linear part of the control law keeps all of the vehicles in the desired formation, while the nonlinear control is only employed by the leader whenever it detects an obstacle. To formalize this, we adopt (3.16) to with a different notation. We define

$u_i(k) = u_i^{\text{linear}}(k) + \mu_i(\hat{x}_i(k))$ with $u_i^{\text{linear}}(k) = (u_i^{\text{linear},x}(k), u_i^{\text{linear},y}(k))$ as

$$u_1^{\text{linear},x}(k) = -\sum_{j\in\mathbb{V}}(k_p(\hat{p}_1^x(k) - \hat{p}_j^x(k) + d_{1j}) + k_v(\hat{v}_1^x(k) - \hat{v}_j^x(k))) + k_v(\hat{v}_1^x(k) - 0.2))$$
$$u_1^{\text{linear},y}(k) = -\sum_{j\in\mathbb{V}}(k_p(\hat{p}_1^y(k) - \hat{p}_j^y(k)) + k_v(\hat{v}_1^y(k) - \hat{v}_j^y(k))) + k_v(\hat{v}_1^y(k)))$$
$$u_i^{\text{linear},x}(k) = -\sum_{j\in\mathbb{V}}(k_p(\hat{p}_i^x(k) - \hat{p}_j^x(k) + d_{ij}) + k_v(\hat{v}_i^x(k) - \hat{v}_j^x(k)))$$
$$u_i^{\text{linear},y}(k) = -\sum_{j\in\mathbb{V}}(k_p(\hat{p}_i^y(k) - \hat{p}_j^y(k)) + k_v(\hat{v}_i^y(k) - \hat{v}_j^y(k)))$$

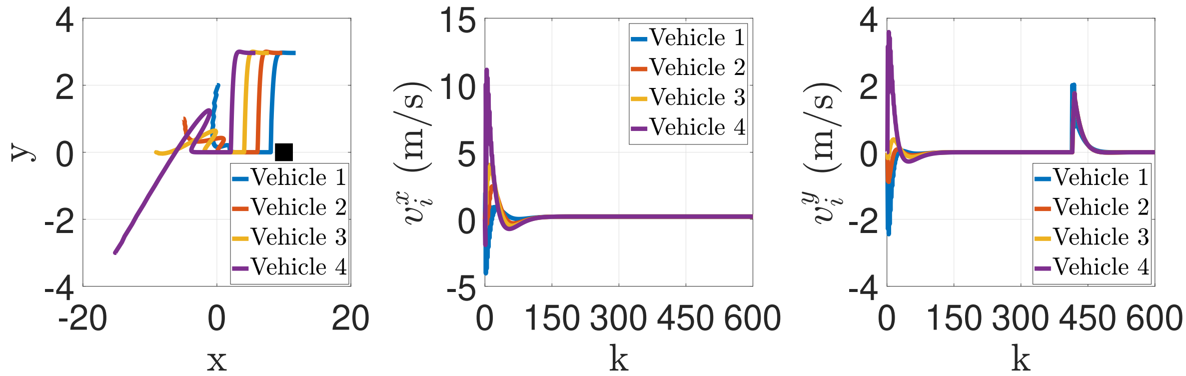
for $i \in \{2, 3, 4\}$, where $d_{ij} = 2(i - j)$. We select $k_p = 0.16$ and $k_v = 0.3$. This setup was motivated by the approach in[88] to achieve the desired formation control. Note that such a control scheme will drive the leaders x-velocity to 0.2 m/s and y-velocity to 0 m/s. Additionally, the inter-vehicle distances in the x-direction will be driven to 2 m, whereas the inter-vehicle distances in the y-direction will be driven to 0 m. The velocities of the vehicles in both the x-direction and y-direction will also be driven to be identical. Thus, we see that the linear part of the control law is responsible for the formation control aspect of the problem.

To avoid an obstacle, the leader uses the nonlinear control $\mu_i = (\mu_i^x, \mu_i^y)$ defined below.

$$\mu_1^x(\hat{x}(k)) = 0, \tag{3.23a}$$

$$\mu_1^y(\hat{x}(k)) = \begin{cases} \frac{\xi}{|\xi|} & \text{if } \|\hat{p}_1(k) - p_o(k)\| \leq 1 \\ -\xi + 2\frac{\xi}{|\xi|} & \text{if } 1 < \|\hat{p}_1(k) - p_o(k)\| \leq 2 \\ 0 & \text{otherwise} \end{cases} \tag{3.23b}$$

where $\xi = \hat{p}_1^y(k) - p_o^y(k)$ and $\mu_i(\hat{x}(k)) = (0, 0),\ i \in \{2, 3, 4\}$. Note that we assume that



(a) (b) (c)

Figure 3.3: Formation control with $m = 4$ rounds of linear consensus showing (a) trajectories in the xy-plane, (b) v_i^x, and (c) v_i^y of all 4 vehicles as they maintain a line formation while avoiding a stationary obstacle at $(10, 0)$. Repeated from Figure (9.3).

the vehicles can measure the location $p_o = (p_o^x, p_o^y)$ of the obstacle. Each vehicle estimates the state of the system based on (3.18), which includes the linear part of the state feedback in computing its parameters.

Figure 3.3 illustrates the simulation results on the formation control. Figure 3.3(a) illustrates the vehicle trajectories in the xy-plane across the simulation. An obstacle is located at $(x, y) = (10, 0)$. We see the vehicles move from their initial positions down to the x-axis, at which point they enter into the desired line formation. When the leader nears the vicinity of the obstacle, it employs the nonlinear part of the control law (3.23) and successfully avoids the obstacle. The remaining vehicles move in formation with the leader, and the formation then continues traveling along the x-direction. We see in Figure 3.3(b) that after initial transients the vehicles maintain the desired velocity of 0.2 m/s along the x-axis. Furthermore, we see in Figure 3.3(c) that after initial transients the vehicles begin to maintain the desired velocity of 0 m/s in the y-direction. However, once the leader detects that is is near the obstacle and employs the nonlinear control, we see the y-velocities of the vehicles spike during the avoidance maneuver before once again settling down to 0 m/s as desired.

We now provide a brief analysis of the role of the network structure on the performance of our algorithm.

3.4 Role of Network Structure on Performance of Distributed Algorithm

Our algorithm formulation allows us to investigate the role of the network structure on the norms of the estimation and control components. We can also get a sense of how the distribution of control authority and the distribution of sensing across the agents each affects these norms. We use as a comparison metric the quantity $\xi = \gamma * \gamma'$. A smaller ξ can be thought of as requiring less or even no consensus to satisfy the small-gain theorem. So, we can compare different graph structures and different distributions of control and sensing authority using ξ.

We present simulation results where we consider a state $x(k) \in \mathbb{R}^3$ with state transition matrix $A = I_3$ and actuation and sensing being performed by $N = 4$ agents. Each case is jointly controllable and jointly observable. We consider three graph classes: line graphs, ring graphs, and star graphs. For each graph class we consider unique distributions of control and sensing authority among the agents.

Recall that each agent i actuates q_i states and senses r_i states. For the actuation, we consider all unique distributions of only one agent controlling all three states, one agent controlling two states and one agent controlling the remaining one state, and three agents controlling one unique state each. Similarly for sensing, we consider all possible distributions of only one agent sensing all three states, one agent sensing two states and one agent sensing the remaining one state, and three agents sensing one unique state each.

For simplicity, we consider identical actuation effort among actuator agents, meaning that if we denote column j of the matrix B_iK_i as $[b_ik_i]_j$ for $j \in \{1, 2, 3\}$, then $[b_ik_i]_j = -0.1e_j$ if agent i controls state j where e_j is the j-th basis vector for $\mathbb{R}^3$, and $[b_ik_i]_j = 0$ otherwise where 0 is the 3×1 vector of zeros. Further, the rows of the matrix C_i equal e_j^T if agent i senses state j (note that each C_i has r_i rows, so we simply ensure that each row contains a unique e_j^T for each sensed state j). Furthermore, we consider cases with no additional consensus, i.e., $m = 0$.

Recall that we use the bounded real lemma for discrete-time LTI systems to assert that the condition $\|\tilde{G}\|_{H_\infty} < \gamma$ holds given that the LMI (3.13) holds. However, we noted that the optimization minimizing γ is not convex. In order to simplify the following analysis we can perform a change of variables to transform the problem into a convex one. Let $Q = X\tilde{L}$ and $R = XW^E$ where $\tilde{L} = -diag(L_1, \ldots, L_N)$. Defining $\tilde{C} = diag(C_1, \ldots, C_N)$, we have $\|\tilde{G}\|_{H_\infty} < \gamma$ holds given that there exists a symmetric and diagonal $X \in \mathbb{R}^{nN \times nN}$ where $X \succ 0$ such that

$$\begin{pmatrix} -X & X\tilde{A} + Q\tilde{C} + R & I_{nN} & 0_{nN} \\ * & -X & 0_{nN} & X \\ * & * & -\gamma I_{nN} & 0_{nN} \\ * & * & * & -\gamma I_{nN} \end{pmatrix} \prec 0 \tag{3.24}$$

where repeated blocks within the symmetric matrix have been omitted for brevity. Note that we have restricted X to be diagonal. While the conditions for the bounded real lemma are still satisfied, we have restricted the dimensionality of the problem and may now only be able to find suboptimal solutions. However, such a restriction along with the change in variables turns the problem into a convex one, greatly simplifying the computation.

3.4.1 Line Graphs

Consider the unique ways to distribute actuation or sensing across four agents in a line graph:

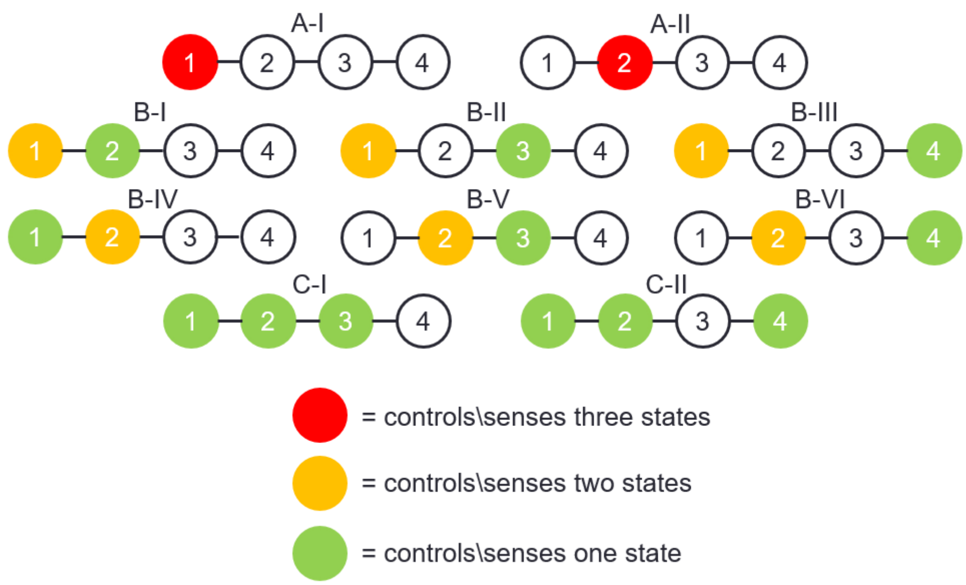


Figure 3.4: Distributions of actuation or sensing across a line graph with $N = 4$ agents. Class A includes one agent controlling all three states, class B includes one agent controlling two states and one agent controlling the remaining one state, and class C includes three agents controlling one unique state each. The class definitions are the same for sensing.

There are ten unique distributions shown in Figure 3.4. Thus, we can consider the 100 unique combinations of actuation and sensing which are possible for line graphs of $N = 4$ agents. We find that for actuation distributions A-II and B-V, when no peripheral agents have any control authority, we get the smallest gain $\gamma' = 0.1$, whereas for the other eight distributions we get $\gamma' = 0.1414$. Similarly, sensing distributions A-II and B-V, when no peripheral agents do any sensing, yield the smallest gain $\gamma = 3.054$, whereas the other distributions all yield $\gamma = 4.036$. Thus, there are only four combinations of actuation and sensing which produce distinct values of ξ, and all are stabilizing without using consensus.

3.4.2 Ring Graphs

The unique ways to distribute actuation or sensing across four agents in a ring graph are shown in Figure 3.5.

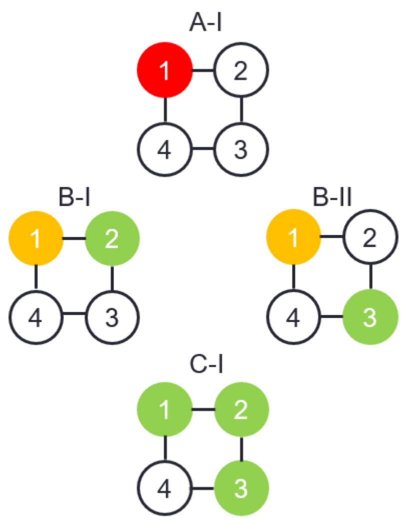


Figure 3.5: Distributions of actuation or sensing across a ring graph with $N = 4$ agents. The class definitions and color scheme are as in Figure 3.4.

While there are 16 unique combinations of actuation and sensing which are possible for ring graphs of $N = 4$ agents, all combinations yield the same value of ξ, with $\gamma = 2.332$ and $\gamma' = 0.071$, and thus all are stabilizing without using consensus.

3.4.3 Star Graphs

The unique ways to distribute actuation or sensing across four agents in a star graph are shown in Figure 3.6.

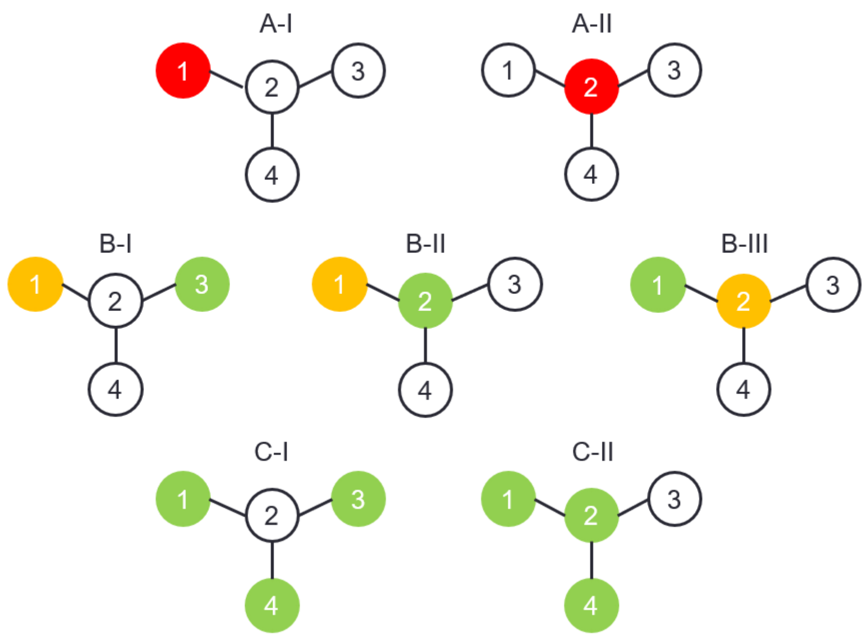


Figure 3.6: Distributions of actuation or sensing across a star graph with $N = 4$ agents. The class definitions and color scheme are as in Figure 3.4.

There are 49 unique combinations of actuation and sensing distribution for star graphs of $N = 4$ agents. For actuation distribution A-II, where the center agent does all the control, we get $\gamma' = 0$, whereas for the other six cases we get $\gamma' = 0.1414$. Similarly, for sensing distribution A-II, where the center agent does all the sensing, we find $\gamma = 1$, whereas we have $\gamma = 4.036$ for the other six distributions. The star alone is best for the stability margin, but all combinations are stabilizing without consensus.

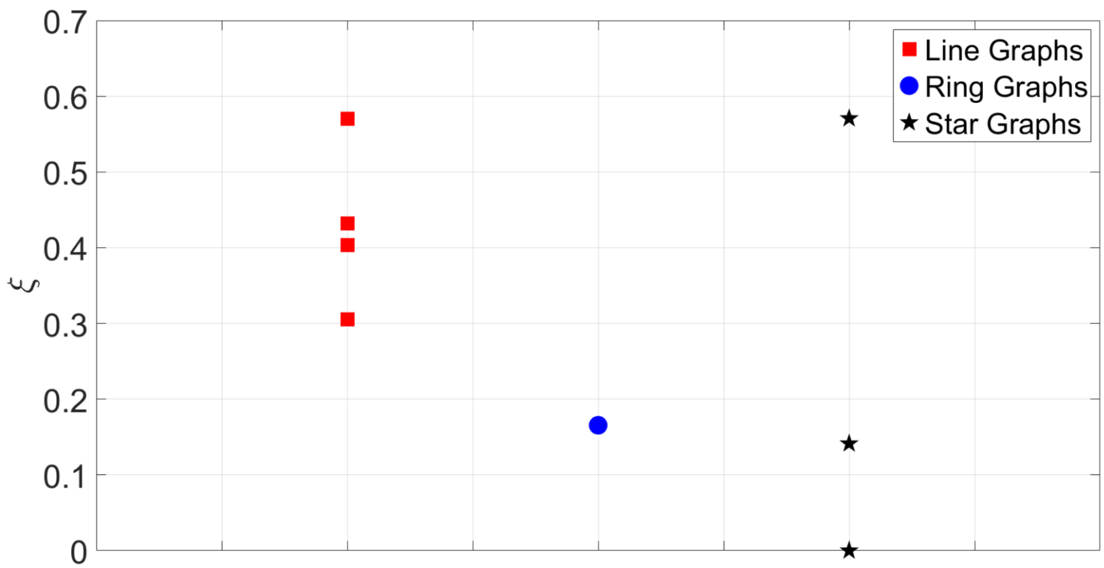


Figure 3.7: Unique values of ξ for line, ring, and star graphs with $N = 4$ agents.

A comparison of unique values of ξ for the three graph classes is shown in Figure 3.7. The best-performing star graph combination outperforms the best-performing ring graph combination, which in turn outperforms the best-performing line graph combination. We classify the best-performing combinations as those which provide the largest stability margin with regard to satisfying the small-gain theorem. It is easy to observe from Figure 3.7 that all of the combinations considered here stabilize the LTI system (3.1) without needing consensus as the small-gain theorem is satisfied for $m = 0$.

We also point out a result related to the optimal values of W_{ij}^C. Take actuation distribution B-II for the star graph in Figure 3.6. Solving for the W_{ij}^C parameters which minimize γ', we find that $W_{32}^C = B_1K_1 + B_2K_2$. In other words, agent 3 is weighting its communication with neighbor agent 2 is such a way as to diminish the influence on the norm of the control component from both agent 1, which is not a neighbor of agent 3, and agent 2. It is thus optimal in certain scenarios to weight communication with a neighbor to cancel out the influence of n-hop neighbors where $n > 1$.

3.5 Conclusions

We investigated the design of a network of agents for the estimation and control of LTI systems. Since the separation principle does not hold, the estimation and control strategies need to be jointly designed, which involves finding a solution to a large-scale optimization. This could be a disadvantage if the agents need to change their control strategies without re-solving the optimization. We have presented LMI formulations to characterize the conditions under which the design of estimation and control can be decoupled, and shown how the frequency of information exchange between agents affects the establishment of the conditions. Our simulation results illustrate an application of our framework to multi-vehicle platooning. We also briefly investigated the influence of the network structure on algorithm performance across various actuation and sensing distributions.

4

Actuator and Sensor Allocation for Distributed Control and Estimation

We now explore a methodology for allocating actuators and sensors across a given network of nodes, focusing heavily on the actuation side of the problem. We consider the distributed estimation and control framework summarized in Chapter 3 and described in detail in Chapter

9 and use as a comparison metric the quantity $\xi = \gamma \|\tilde{D}\|_2$ where $\tilde{D}$ is a block matrix whose i, j-th block satisfies

$$[\tilde{D}]_{ij} = \begin{cases} -B_j K_j + W^C_{ij} & \text{if } j \in N_i \backslash \{i\} \\ \sum_{\ell \in \mathbb{V} \backslash \{i\}} B_\ell K_\ell - \sum_{\ell \in N_i \backslash \{i\}} W^C_{i\ell} & \text{if } j = i \\ -B_j K_j & \text{otherwise.} \end{cases} \tag{4.1}$$

Here γ is an upper bound on the norm of the transfer function of the estimation component, which is related to the graph structure through an associated LMI. Thus, we can take ξ to represent the worst case product of the norms. A smaller ξ can be thought of as requiring less or even no consensus to satisfy the small-gain theorem constraint. Thus, we seek to minimize the metric ξ. Note that for this analysis we have assumed that there are no consensus rounds performed in the algorithm ($m = 0$).

In Section 4.1 we introduce our methodology for actuator selection and formulate the main problem. In Section 4.2 we build intuition using simple graphs. Sections 4.3-4.11 describe the simulation study as well as the statistical techniques used to answer our main question. We also review pertinent centrality measures and specify how we will compare different measures against one another as heuristics for actuator selection. Sections 4.12-4.15 illustrate our statistical results across our test suite. In Section 4.16 we consider the analogous sensor selection problem. Finally, we summarize key results and conclude in Section 4.17.

4.1 Actuator Allocation

We would like to identify "important" nodes to select as actuators. This concept is illustrated by scenarios in which we have a fundamental limit on how much actuation capacity we can sustain. For instance, if we have a limited amount of battery power, we would like to distributed our available actuation effort to those agents that can contributed the most to minimizing ξ and, thereby, providing us with a larger stability margin. To get at this question, we consider convexly weighting actuators to minimize $\|\tilde{D}\|_2$. For ease of discussion, consider a scalar process x with state transition matrix $A = 1$. Let $B_i = 1$ with actuator weights $K_i = -0.1\alpha_i$ for each $i \in [1, \ldots, N]$ with

$$\sum_{i=1}^{N} \alpha_i = 1 \tag{4.2}$$

This scheme allows us to control if an actuator is active or inactive through the allocation of the α_i terms. If $\alpha_i = 0$ then node i is inactive in an actuation sense. Note that a requirement of our algorithm is that we design K_i such that the quantity $A + \sum_{i=1}^{N} B_i K_i$ is Schur stable. With $A = 1$ our convex weighting scheme yields

$$\begin{aligned} A + \sum_{i=1}^{N} B_i K_i &= 1 + \sum_{i=1}^{N} (1)(-0.1\alpha_i) \\ &= 1 - 0.1 \sum_{i=1}^{N} \alpha_i \\ &= 1 - 0.1(1) \\ &= 0.9 \end{aligned} \tag{4.3}$$

which is Schur stable. Given a predefined set of B_i values, we can use a semidefinite program to find feasible W_{ij}^C and α_i for $i, j \in [1, \ldots, N]$ which minimize $\|\tilde{D}\|_2$. Recall that $\|\tilde{D}\|_2$ is just the maximum singular value of $\tilde{D}$. The constraint

$$\sigma_{max}[\tilde{D}] \leq \Sigma \tag{4.4}$$

for $\Sigma \in \mathbb{R}_{\geq 0}$ can be written as the convex LMI [50]

$$\begin{bmatrix} \Sigma \mathcal{I}_{nN} & \tilde{D} \\ \tilde{D}^T & \Sigma \mathcal{I}_{nN} \end{bmatrix} \succeq 0. \tag{4.5}$$

Thus, a semidefinite program can be formulated to find feasible W_{ij}^C and α_i which satisfy (4.5) while minimizing Σ while also satisfying the convexity constraint (4.2). We first analyze some simple graphs to get intuition.

4.2 Building Intuition

We first look at a line graph with $N = 5$ nodes. Minimizing $\|\tilde{D}\|_2$ yields the optimal distribution of actuation weights seen in Figure (4.1).

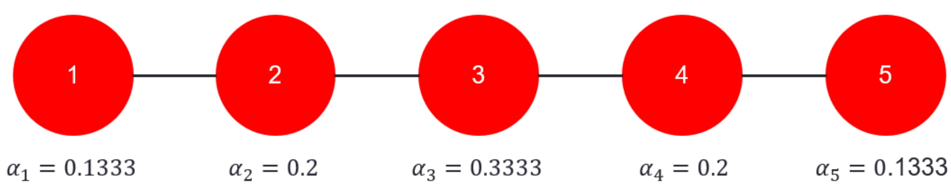


Figure 4.1: Distribution of optimal actuation weights across a line graph with $N = 5$ nodes.

The optimal selection of active actuators giving the most weight to node $i = 3$ naturally leads us to wonder if node centrality could provide a guide. We note that the distribution of actuation for this simple case trends with the distribution of betweenness centrality (BC) as seen in Figure (4.2).

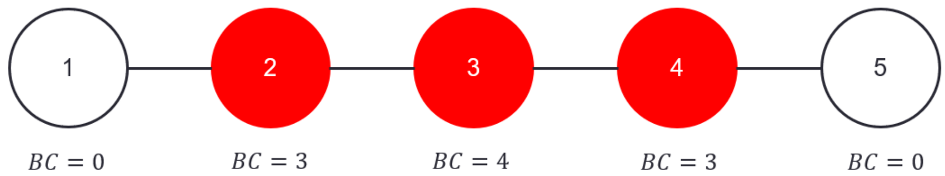


Figure 4.2: Distribution of betweenness centrality (BC) across a line graph with $N = 5$ nodes.

4.2.1 Betweenness Centrality

Betweenness centrality is a measure of how many shortest paths between pairs of nodes a particular node lies on[33]. For a graph $G := (V, E)$ the betweenness centrality of a node $i \in V$ is given by

$$BC(i) = \sum_{j \neq i \neq k \in V} \frac{\sigma_{jk}(i)}{\sigma_{jk}} \tag{4.6}$$

where σ_{jk} represents the total number of shortest paths between a given node j to another node k and $\sigma_{jk}(i)$ is the total number of shortest paths between these nodes which also pass through node i.

4.2.2 Impact of Multiple Actuators

We next investigate a simple graph with $N = 5$ nodes. We see in Figure (4.3) the distribution of actuation weights which minimizes $||\tilde{D}||_2$, where the betweenness centrality has also been included for ease of comparison.

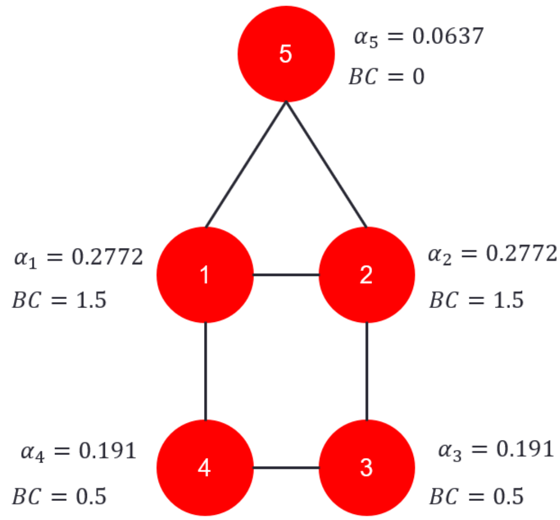


Figure 4.3: Distribution of optimal actuation weights and betweenness centrality (BC) across a graph with $N = 5$ nodes.

Again we see that the trend observed between optimal actuation weighting and betweenness centrality holds for this simple case. We also note that allowing for multiple actuators improves upon the case with only a single actuation node. In the multiple actuation case shown in Figure (4.3) we achieve $\|\tilde{D}\|_2 = 0.0427$; however, if we restrict ourselves to only allowing for one agent to actuate, i.e., $\alpha_i = 1$ for some node i, we obtain the optimal results shown in Figure (4.4).

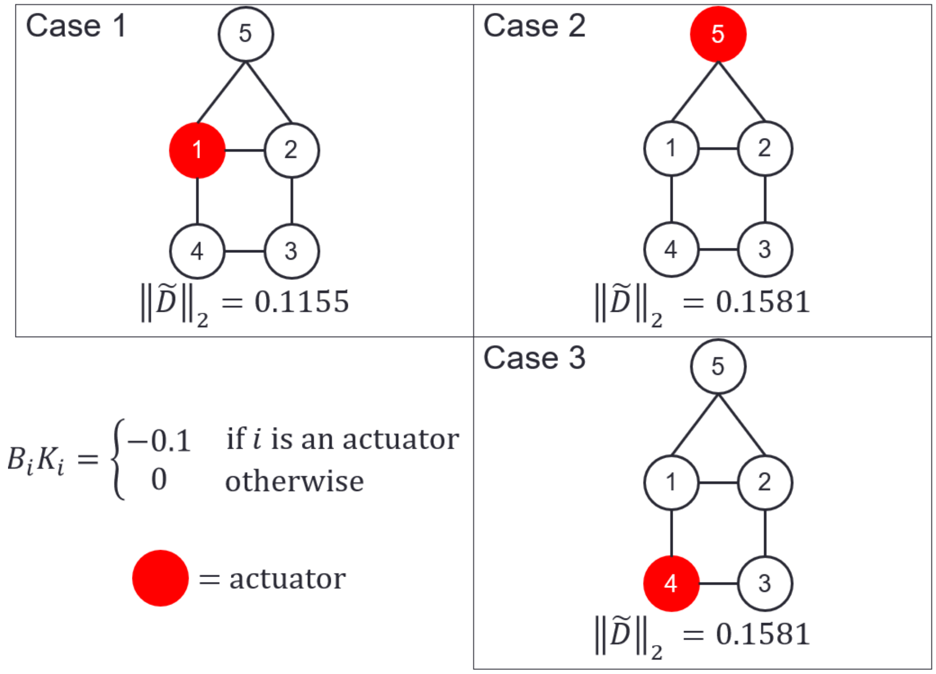


Figure 4.4: Distribution of norms for unique selections of individual actuation agents across a graph with $N = 5$ nodes.

Note that the smallest we can make $\|\tilde{D}\|_2$ is 0.1155 assuming a single actuating node i with $B_iK_i = -0.1$, which was readily improved upon by allowing for heterogeneous actuation and multiple actuators in the optimization.

We can also look at more irregular graph structures. Results from repeating the analysis for a different graph with $N = 5$ nodes are shown in Figure (4.5).

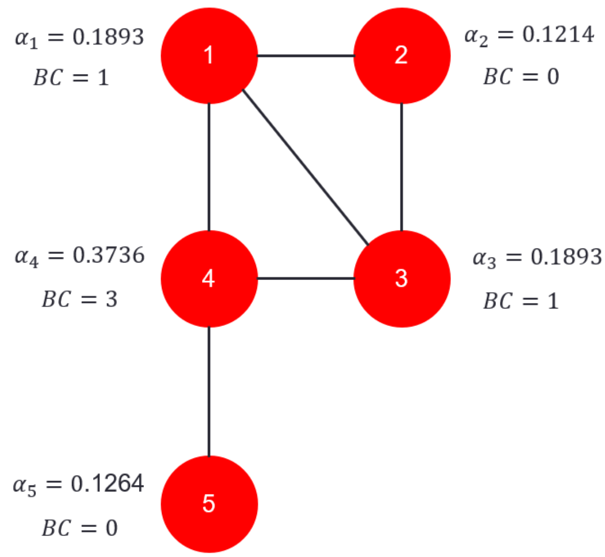


Figure 4.5: Distribution of optimal actuation weights and betweenness centrality (BC) across an irregular graph with $N = 5$ nodes.

We once again see the trend between optimal heterogeneous actuation weights and betweenness centrality. Similar to the graphs analyzed in Figures (4.3)-(4.4), we also see an improvement by allowing for multiple heterogeneous actuators compared to the cases with only a single actuator. In the heterogeneous case shown in Figure (4.5) we achieve $||\tilde{D}||_2 = 0.0645$. If we only allow for a single active actuator the optimal norms for each unique single actuator placement are shown in Figure (4.6).

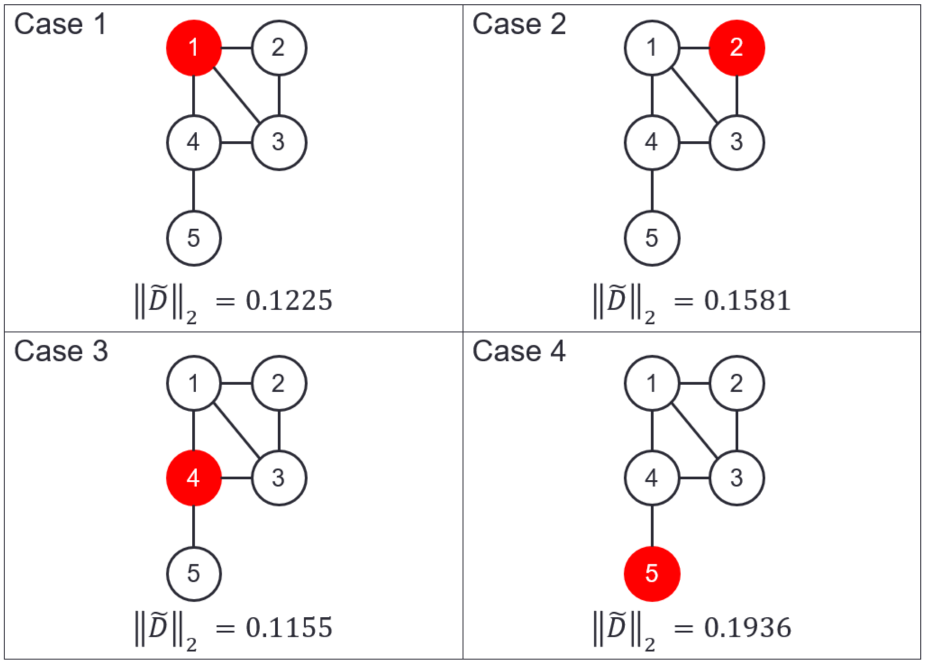


Figure 4.6: Distribution of norms for unique selections of individual actuators across a graph with $N = 5$ nodes.

We see that allowing for multiple heterogeneous actuators improves upon the minimum norm of 0.1155 in the single actuation cases. While it would be ideal to determine an analytical expression which can be used to ascertain which qualities of a node's centrality in the graph might influence its optimality in an actuation sense, we nevertheless can resort to statistical simulation techniques to determine probabilities of certain centrality measures being better indicators of actuation node selection than others. This will be the topic of the following section.

4.3 Simulation Study

One method to inform if certain centrality measures are good indicators of which nodes to use as actuators is to conduct a statistical simulation analysis. There is precedent for obtaining statistics from simulations in the graph theory literature[34,35,36]. Specifically, we will make use of null hypothesis testing to further understand how different centrality measures could aid in actuator selection. We can consider selecting the heterogeneous actuation weights as functions of respective centrality measures. This implies that the actuation effort of an agent would be selected proportional to its centrality measure, with more central agents exerting larger actuation magnitudes. Before describing the logistics of the simulation study we will first introduce some important concepts from statistical hypothesis testing.

4.4 Statistical Hypothesis Testing

Statistical hypothesis testing involves comparing a null hypothesis, H_0, with an alternative hypothesis, H_1[37]. The comparison of these two hypotheses is considered to be statistically significant if, according to some threshold probability, the observed data would be unlikely to occur given the truth of the null hypothesis. If this predetermined condition is met, one chooses to reject the null hypothesis in favor of the alternative hypothesis. Note that specifying this threshold probability is a means to control for the risk of incorrectly rejecting the null hypothesis when it is in fact true[37].

4.5 Two-sample testing

The idea of two-sample testing is where we observe two independent random samples $\mathbf{z} = (z_1, z_2, \ldots, z_n)$ and $\mathbf{y} = (y_1, y_2, \ldots, y_m)$ which are drawn from possibly different probability distributions. After observing these samples, we wish to test the validity of our null hypothesis which can be stated as

H_0: There is no difference between the means $\bar{z}$ of $\mathbf{z}$ and $\bar{y}$ of $\mathbf{y}$.

The hypothesis test begins with determining a test statistic $\hat{\theta}$ which measures the effect we are looking for. For example, if we expect that $\bar{z} > \bar{y}$, then we can choose as an alternative hypothesis

H_1: The mean $\bar{z}$ of $\mathbf{z}$ is greater than the mean $\bar{y}$ of $\mathbf{y}$.

To put this into perspective, $\mathbf{z}$ could represent the test scores of students who studied and $\mathbf{y}$ could represent the test scores of students who didn't study. Our alternative hypothesis would then be a sensible one, stating that the test scores of studious students should on average be higher than those who do not study. In this case, we could choose as our test statistic $\hat{\theta} = \bar{z} - \bar{y}$, which we would expect to be positive if studying works. In regards to the hypothesis testing, if we cannot decisively reject the truth of H_0 then we cannot demonstrate the superiority of studying over not studying in relation to test scores. The hypothesis test is just a formal way to determine whether or not we should reject H_0.
Going back to the test-taking example, if the null hypothesis H_0 is not true then we would expect the test statistic $\hat{\theta}$ to be larger than if H_0 is true. One nice thing about hypothesis testing is we do not need to quantify what we mean here by "larger", only that larger values of $\hat{\theta}$ imply more evidence to reject H_0. The next important concept is the achieved significance level (*ASL*) of the test. The *ASL* is defined to be the probability of observing at least as large

of a value of the test statistic $\hat{\theta}$ when the null hypothesis H_0 is true. Mathematically we have

$$ASL = Prob_{H_0}\{\hat{\theta}^* \geq \hat{\theta}\} \tag{4.7}$$

A smaller ASL implies stronger evidence against H_0. Note that $\hat{\theta}^*$ is a hypothetical random variable with the H_0 distribution. In other words, it is the distribution of $\hat{\theta}$ if H_0 is true. The hypothesis test would then be to compute the ASL and make a decision on whether or not to reject H_0 according to a conventional threshold. One thing to immediately note is that the computation of the ASL requires knowledge of the underlying distribution of H_0. However, there need not be one single distribution to describe H_0. Thankfully, there are simulation techniques which we can use to get estimates of the statistics for our null hypothesis of interest.

4.6 Centrality Measures of Interest

To apply the machinery of hypothesis testing as just described to our framework, we need a way to determine relevant statistics for the null hypothesis. Before getting into the methodology for achieving that, we will first describe the various centrality measures, in addition to betweenness centrality, which will be considered as possible heuristics for actuator selection. For all definitions we assume a graph $G := (V, E)$ with node set V and edge set E.

4.6.1 Closeness Centrality

The closeness centrality (CC) of a node is classically defined as the inverse of the sum of the distances between that node and all other nodes in the graph[38]. The closeness centrality of a node $i \in V$ is

$$CC(i) = \frac{1}{\sum_{j \in V} d(i,j)} \tag{4.8}$$

where $d(i,j)$ is the distance between nodes i and j.

4.6.2 Degree Centrality

The degree centrality (DC) of a node is defined simply as

$$DC(i) = deg(i) \tag{4.9}$$

where $deg(i)$ represents the total number of edges incident to node i. Note that for directed graphs we have two notions of degree centrality, *in-degree* and *out-degree*. In-degree is a count of how many edges direct toward an agent (signalling that the agent can transmit information to others) and out-degree is a count of how many edges direct away from an agent (signalling that the agent can receive information from others). In the cases which follow we consider graphs in which the in-degree and out-degree are equal and just consider the degree.

4.6.3 Eigenvector Centrality

Eigenvector centrality (EC) assigns relative scores to nodes in a graph under the assumption that connections to high-scoring nodes are weighted more heavily in a node's score than connections to low-scoring nodes. Given the adjacency matrix $\mathcal{A}$ of the graph we can define the relative score of a node i as

$$x_i = \frac{1}{\lambda} \sum_{j \in V} \mathcal{A}_{ij} x_j \tag{4.10}$$

which in vector notation is the eigenvalue equation $\mathcal{A}\mathbf{x} = \lambda\mathbf{x}$[39]. The eigenvector centrality $EC(i)$ of a node i is then the i-th component of the eigenvector associated with the unique largest eigenvalue for this given eigenvalue equation.

4.6.4 Information Centrality

Information centrality (IC) is related to how much information is contained in a path between any two nodes[40,41]. The information in a path between two nodes i and j is defined as the inverse of the path length between these two nodes. The total information between nodes i and j, I_{ij}^{tot}, is the sum of the information contained in every path which connects nodes i and j and is given by

$$I_{ij}^{tot} = (L_{ii}^{\dagger} + L_{jj}^{\dagger} - 2L_{ij}^{\dagger})^{-1} \tag{4.11}$$

where L is the graph Laplacian. We can then define the information centrality of node i as

$$IC(i) = \left(\frac{1}{N} \sum_{j \in V} \frac{1}{I_{ij}^{tot}} \right)^{-1} \tag{4.12}$$

which is just the harmonic average of the total information between node i and all other nodes in the graph.

4.7 Monte Carlo Simulations

To show how we can incorporate our problem statement within the framework of hypothesis testing, consider the following null and alternative hypotheses.

H_0: Selecting actuators by weighting as a function of betweenness centrality provides no benefit over selecting actuators by weighting as a function of degree centrality.

H_1: Selecting actuators by weighting as a function of betweenness centrality is better than selecting actuators by weighting as a function of degree centrality.

Weighting an actuator i as a function of a specific centrality measure is accomplished by selecting

$$\alpha_i = \frac{CM(i)}{\sum_{j \in V} CM(j)} \tag{4.13}$$

where $CM(\cdot)$ is the centrality measure of interest. For exposition purposes let's consider comparing H_0 and H_1.

4.7.1 Erdos-Renyi Graphs

To generate the samples of interest we employ Erdos-Renyi random graphs[42,43]. Such graphs allow us to easily explore the actuator selection problem as a function of network density. The version of the model we use constructs a graph by connecting pairs of nodes according to a given connection probability. Let $G(N, \rho)$ be an Erdos-Renyi graph with N nodes and connection probability ρ, meaning that each edge between a pair of nodes is included in the graph with probability ρ independently of every other edge. Thus, ρ is a parameter which we can tune to generate graphs with varying edge densities. Example graphs for $G(10, 0.3)$ and $G(10, 0.7)$ are shown in Figures (4.7) and (4.8), respectively.

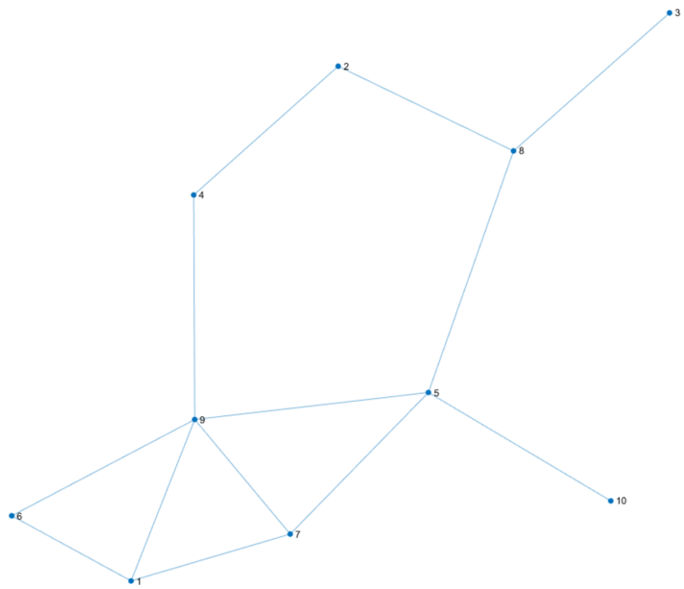


Figure 4.7: Erdos-Renyi graph with $N = 10$ nodes and connection probability $\rho = 0.3$.

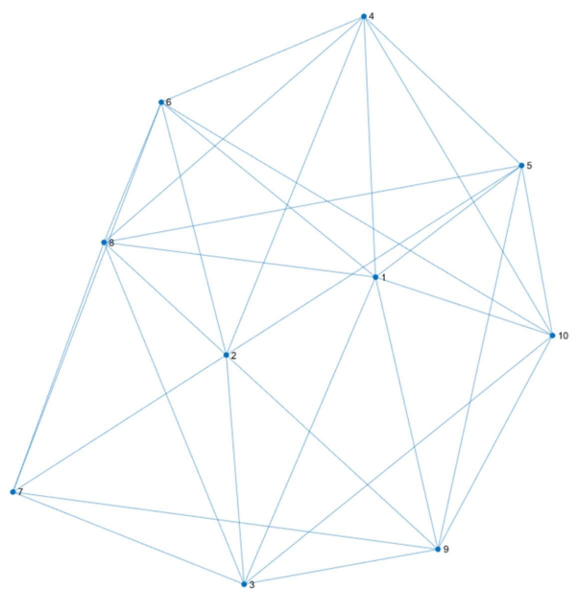


Figure 4.8: Erdos-Renyi graph with $N = 10$ nodes and connection probability $\rho = 0.7$.

Note that for our analysis we only consider connected graphs.

4.8 Generation of Samples

To compare the hypotheses H_0 and H_1 we generate MC Erdos-Renyi random graphs where MC is the number of Monte Carlo rounds. The first actuation weighting scheme generates weights as

$$\alpha_i = \frac{DC(i)}{\sum_{j \in V} DC(j)} \tag{4.14}$$

where $DC(i)$ is the degree centrality of node i. This will give us the samples $\mathbf{DC} = (dc_1, dc_2, \ldots, dc_{MC})$ where $dc_i = [\|\tilde{D}\|_2]_i$ for $i \in [1, \ldots, MC]$. In other words, the samples are the values of $\|\tilde{D}\|_2$ for each simulated case where an associated LMI is solved to find feasible W_{ij}^C terms to minimize the norm given the α actuation weights calculated using (4.14). Similarly, for the same MC Erdos-Renyi random graphs the second actuation weighting scheme assigns actuation weights as

$$\alpha_i = \frac{BC(i)}{\sum_{j \in V} BC(j)} \tag{4.15}$$

where $BC(i)$ is the betweenness centrality of node i. This will give samples $\mathbf{BC} = (bc_1, bc_2, \ldots, bc_{MC})$ where $bc_i = [\|\tilde{D}\|_2]_i$ for $i \in [1, \ldots, MC]$ where the norm is again minimized by solving a new LMI which selects appropriate W_{ij}^C terms.

4.9 Hypothesis Testing

Given the samples $\mathbf{DC}$ and $\mathbf{BC}$, we also calculate the sample mean and sample variance of each as $\bar{DC}$, σ^2_{DC} and $\bar{BC}$, σ^2_{BC}, respectively. We now define the test statistic as

$$\hat{\theta} = \frac{\bar{DC} - \bar{BC}}{\sqrt{\sigma^2_{DC}/MC + \sigma^2_{BC}/MC}} \tag{4.16}$$

Note that if we expect the norms on average to be smaller when weighting actuators as a function of betweenness centrality (H_0 is not true), then we would expect larger values of $\hat{\theta}$ than if H_0 was true. Going back to the previous methodology, the next step in the hypothesis testing would be to calculate the ASL; however, we don't necessarily have access to the distribution statistics of the H_0 case. To solve this we can make use of bootstrapping methods.

4.10 Bootstrapping

Bootstrapping methods allow us to estimate the statistics of the null hypothesis under consideration [37]. We will construct a sampling distribution that the test statistic would have if the perceived effects of weighting according to betweenness centrality rather than degree centrality were <u>not</u> present in the population. Then, we can locate the test statistic on this new distribution which we have created and compare with our observed test statistic to construct the ASL. To construct a sampling distribution under the assumption of the null hypothesis H_0, which says that weighting according to betweenness centrality is no better than weighting according to degree centrality, we can construct two new data sets with the appropriate null hypothesis underlying them. Let

$$dc_i' = dc_i - \bar{DC} + \bar{a} \tag{4.17}$$

and

$$bc_i' = bc_i - \bar{BC} + \bar{a} \tag{4.18}$$

for $i \in [1, \ldots, MC]$ where $\bar{a}$ is the mean of the combined sample $\mathbf{DC} + \mathbf{BC}$. These new samples now operate under the pretense of the null hypothesis. We are now ready to implement bootstrapping. We draw a random sample (dc_i^*) of size MC *with replacement* from dc_i' as well as another random sample bc_i^* of size MC *with replacement* from bc_i'. We then calculate a new test statistic

$$\hat{\theta}^* = \frac{\bar{DC}^* - \bar{BC}^*}{\sqrt{\sigma_{DC}^{*2}/MC + \sigma_{BC}^{*2}/MC}} \tag{4.19}$$

We then repeat this bootstrapping B times for some large value of B (we use $B = 10000$). We now have B values of the test statistic $\hat{\theta}^*$. We can estimate the ASL, or the p-value, of the hypothesis test as

$$p = \frac{\sum_{i=1}^{B} I\{\hat{\theta}_i^* \geq \hat{\theta}\}}{B} \tag{4.20}$$

where I is the indicator function equaling 1 if its argument is true and 0 otherwise. We now have a statistical approach to compare our hypotheses against one another. If the calculated p-value is below a determined threshold we choose to reject the null hypothesis in favor of the alternative hypothesis. While this discussion compared H_0 with H_1, we can see how the machinery can be extended to comparing other hypotheses against one another.

4.11 Test Cases

We consider the following hypotheses:

H_{01}: Selecting actuators by weighting as a function of betweenness centrality provides no benefit over selecting actuators by weighting as a function of closeness centrality.

H_1: Selecting actuators by weighting as a function of betweenness centrality is better than selecting actuators by weighting as a function of closeness centrality.

H_{02}: Selecting actuators by weighting as a function of betweenness centrality provides no benefit over selecting actuators by weighting as a function of degree centrality.

H_2: Selecting actuators by weighting as a function of betweenness centrality is better than selecting actuators by weighting as a function of degree centrality.

H_{03}: Selecting actuators by weighting as a function of betweenness centrality provides no benefit over selecting actuators by weighting as a function of eigenvector centrality.

H_3: Selecting actuators by weighting as a function of betweenness centrality is better than selecting actuators by weighting as a function of eigenvector centrality.

H_{04}: Selecting actuators by weighting as a function of betweenness centrality provides no benefit over selecting actuators by weighting as a function of information centrality.

H_4: Selecting actuators by weighting as a function of betweenness centrality is better than selecting actuators by weighting as a function of information centrality.

We perform hypothesis testing to compare each pair of hypotheses $\{H_{01}, H_1\}$, $\{H_{02}, H_2\}$, $\{H_{03}, H_3\}$, and $\{H_{04}, H_4\}$ and generate appropriate p-values given by (4.20). Our test suite consists of Erdos-Renyi graphs $G(N, \rho)$ with $N \in [6, 7, 8, 9, 10, 11, 12, 13]$ and $\rho \in [0.3, 0.35, 0.4, 0.45, 0.5, 0.55, 0.6, 0.65, 0.7]$. This allows us to investigate graphs with disparate numbers of nodes and varying edge densities.

4.12 Comparing Betweenness Centrality with Closeness Centrality

Comparing the pair of hypotheses $\{H_{01}, H_1\}$ allows us to investigate if weighting actuators as a function of betweenness centrality provides any tangible benefits (in the sense of minimizing $\|\tilde{D}\|_2$) over weighting actuators as a function of closeness centrality. A plot of appropriate p-values vs. ρ is shown in Figure (4.9).

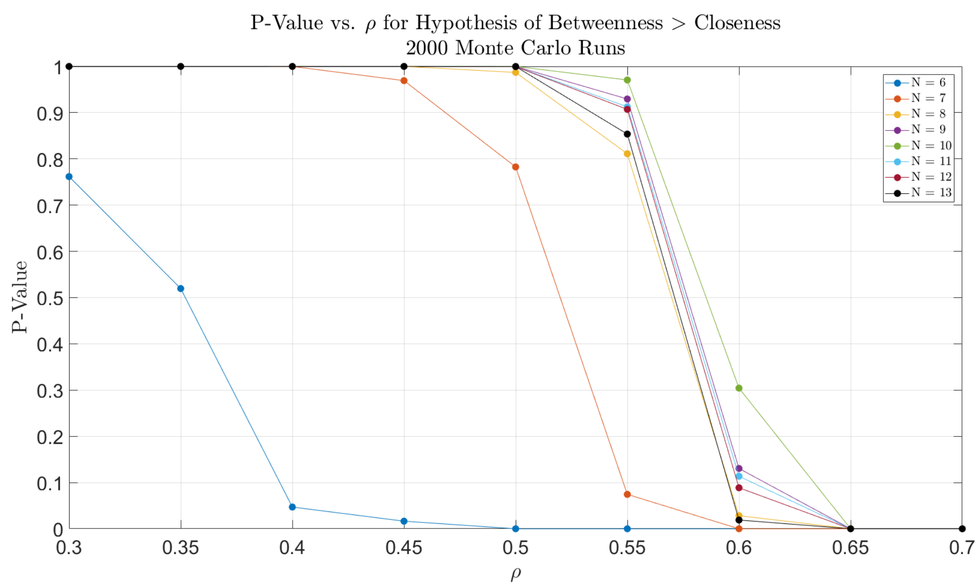


Figure 4.9: Comparison of hypotheses $\{H_{01}, H_1\}$ across Erdos-Renyi graphs. Each data point is calculated using 2000 Monte Carlo runs.

An accepted practice is for p-values < 0.05 we can claim that the results provide statistically significant evidence for accepting hypothesis H_1 over H_{01}. Table 4.1 highlights the statistically significant cases in green.

		ρ								
		0.3	**0.35**	**0.4**	**0.45**	**0.5**	**0.55**	**0.6**	**0.65**	**0.7**
	6	0.7617	0.5196	0.047	0.0164	0	0	0	0	0
	7	1	1	0.9999	0.9693	0.7826	0.0744	0	0	0
	8	1	1	1	1	0.987	0.8111	0.0283	0	0
N	**9**	1	1	1	1	0.9999	0.9296	0.1304	0	0
	10	1	1	1	1	1	0.9707	0.3042	0	0
	11	1	1	1	1	0.9995	0.9127	0.1136	0	0
	12	1	1	1	1	0.9995	0.9071	0.0887	0	0
	13	1	1	1	1	0.9997	0.8537	0.0189	0	0

Table 4.1: Comparison of hypotheses $\{H_{01}, H_1\}$ across Erdos-Renyi graphs. Each data point is calculated using 3000 Monte Carlo runs and statistically significant cases ($p < 0.05$) are highlighted in green.

We see that for many cases with $\rho \geq 0.6$ (excluding those for $9 \leq N \leq 12$) the hypothesis testing informs us that there is significant evidence that weighting actuators as a function of betweenness centrality provides a marked benefit over weighting actuators as a function of closeness centrality. Note that for larger values of ρ the generated Erdos-Renyi graphs have proportionally higher average edge densities where the edge density for an undirected graph $G := (V, E)$ is defined as

$$EdgeDensity = \frac{|E|}{n(n-1)/2} \tag{4.21}$$

where $|E|$ is the cardinality of the edge set and $n(n-1)/2$ is the total number of possible edges. Note that in the limit as the number of generated Erdos-Renyi graphs goes to ∞ the average edge density goes to ρ.

4.12.1 Impact of Fully Connected Nodes

A major driving factor behind the finding that p-values tend to become statistically significant as ρ increases is the emergence of fully connected nodes. We have the following definition of a fully connected node:

Definition: A fully connected node for an undirected graph $G := (V, E)$ is a node $i \in V$ such that $\sum_{j \in V} \mathcal{A}(i,j) = N - 1$ where $\mathcal{A}$ is the adjacency matrix of graph G and $|V| = N$.

Recall that to minimize $\|\tilde{D}\|_2$ we use a semidefinite program to find feasible W_{ij}^C and α_i for $i, j \in V$ which satisfy the relevant LMI while minimizing the maximum singular value of $\tilde{D}$.

Conjecture: Let M be the set of all fully connected nodes in V with $M \subseteq V$. It is optimal in the norm-minimizing sense to select $\alpha_i = \frac{1}{|M|}$ for all $i \in M$ and $\alpha_j = 0$ for all $j \notin M$.

In other words, the optimal distribution of actuation for a graph with fully connected nodes is to proportionally weight those nodes and not weight any other nodes, meaning that only the fully connected nodes end up actively actuating. Weighting in this way yields $\|\tilde{D}\|_2 = 0$. This has been observed in all numerical simulations considered. In order to develop this idea further, we now consider the optimal weighting scheme along with weighting using the centrality measures as heuristics. An example Erdos-Renyi graph $G(6, 0.7)$ with two fully connected nodes is shown in Figure (4.10), and the distribution of normalized α_i values as a function of node i for various weighting schemes along with the optimal weighting scheme on this graph is shown in Figure (4.11).

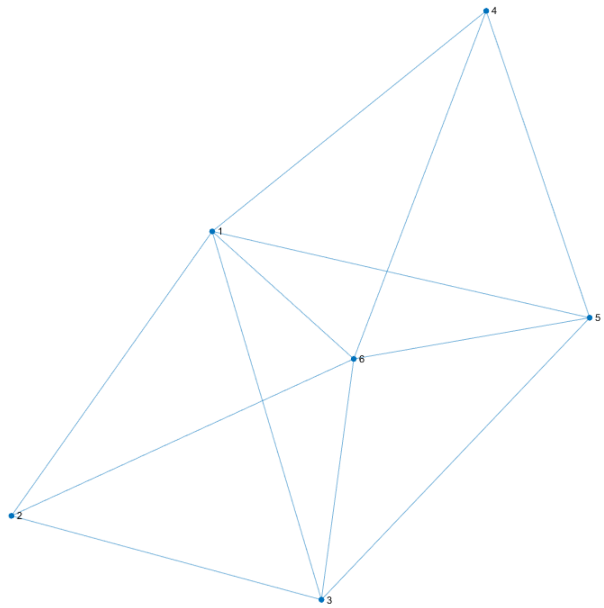


Figure 4.10: Erdos-Renyi graph with $N = 6$ and edge connection probability $\rho = 0.7$. Nodes 1 and 6 are fully connected.

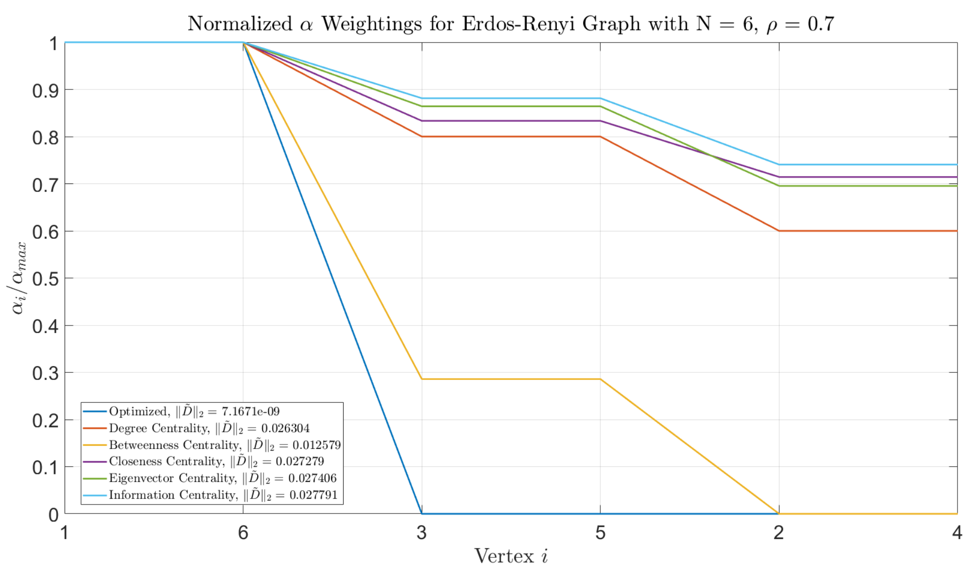


Figure 4.11: Normalized α weightings for an Erdos-Renyi graph with $N = 6$ and edge connection probability $\rho = 0.7$. Nodes 1 and 6 are fully connected.

Immediately note how actuation weighting as a function of betweenness centrality yields a minimum norm which is closest to the optimal case that is achieved by only weighting fully connected nodes 1 and 6. In all cases, each weighting scheme puts the most weight on the fully connected nodes, with the optimal scheme weighting no other nodes and betweenness centrality placing relatively little actuation effort on nodes 3 and 5. In contrast, the other centrality measures all put nonzero weights on the remaining nodes, resulting in higher overall norms. Looking at the distribution of graphs which comprise the data in Figure (4.9) and Table 4.1 we see that for a given N as ρ increases there is a larger proportion of graphs which have at least one fully connected node. Graph distributions for $N = 6$ and $N = 13$ are shown in Figure (4.12) and Figure (4.13), respectively, to illustrate this point.

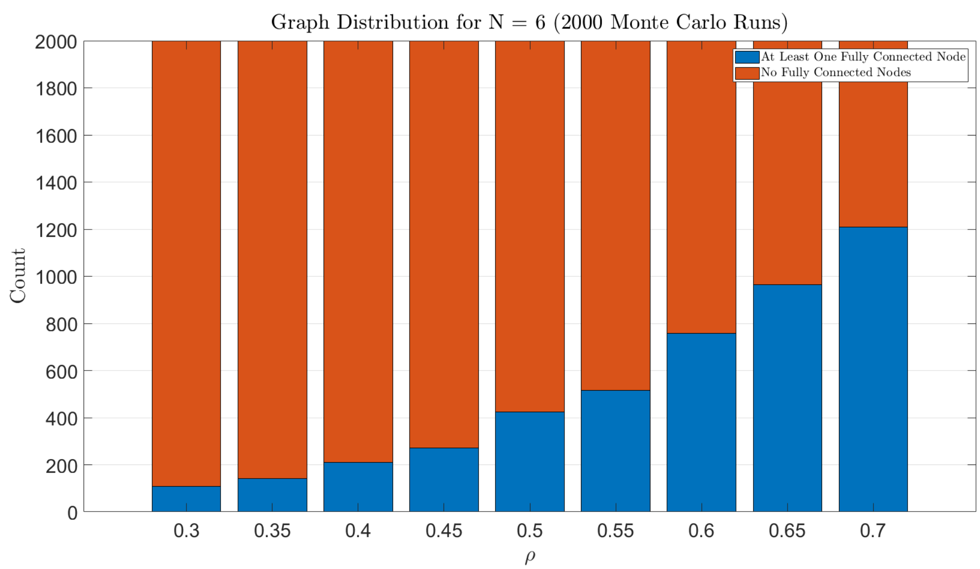


Figure 4.12: Distribution of graph types used for comparison of hypotheses $\{H_{01}, H_1\}$ for Erdos-Renyi graphs with $N = 6$ nodes.

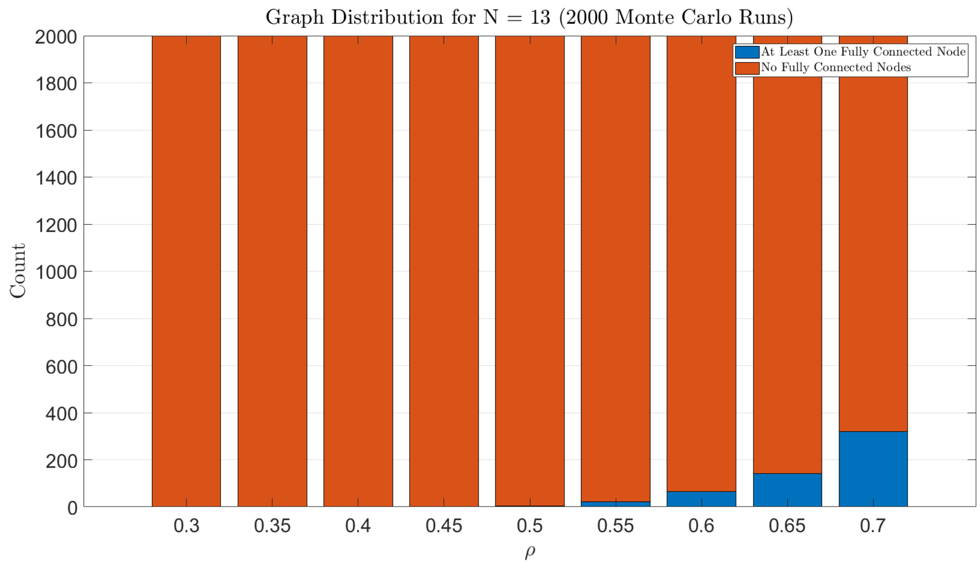


Figure 4.13: Distribution of graph types used for comparison of hypotheses $\{H_{01}, H_1\}$ for Erdos-Renyi graphs with $N = 13$ nodes.

More evidence that the presence of fully connected nodes provides a situation in which weighting as a function of betweenness centrality tends to outperform weighting as a function of closeness centrality can be seen by looking at the difference in norm values under these two weighting schemes as a function of distance from a complete graph. The assumption is that as ρ increases and graphs get closer to a complete graph in the sense of the Frobenius norm, the prevalence of graphs with at least one fully connected node will increase and yield lower overall norms when weighting as a function of betweenness centrality compared to weighting as a function of closeness centrality. Figure (4.14) and Figure (4.15) illustrate findings in this regard for Erdos-Renyi graphs $G(10, 0.4)$ and $G(10, 0.7)$, respectively.

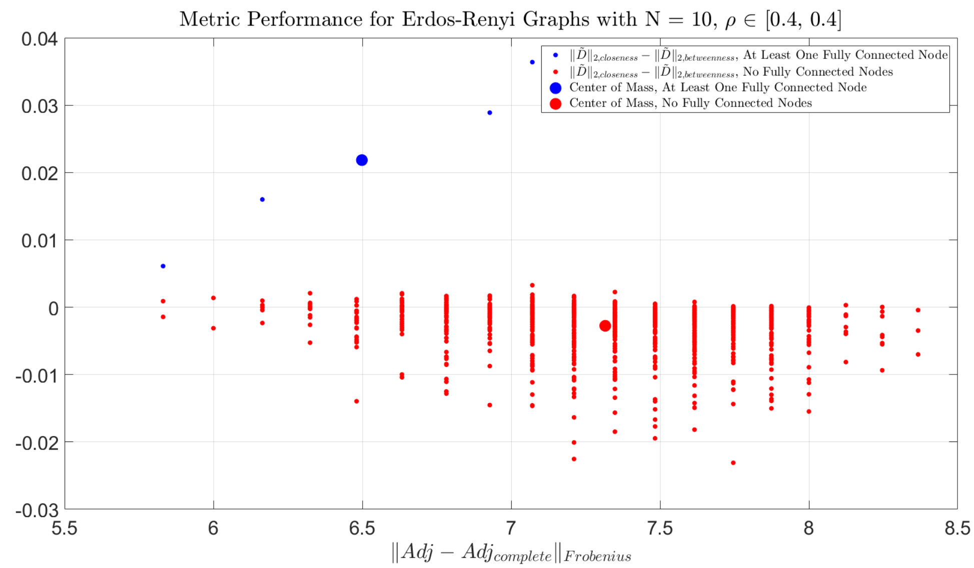


Figure 4.14: Difference between norms with actuation weighting as a function of closeness centrality and betweenness centrality vs. distance from a complete graph for Erdos-Renyi graphs with $N = 10$ and $\rho = 0.4$. 1000 data points comprise this figure.

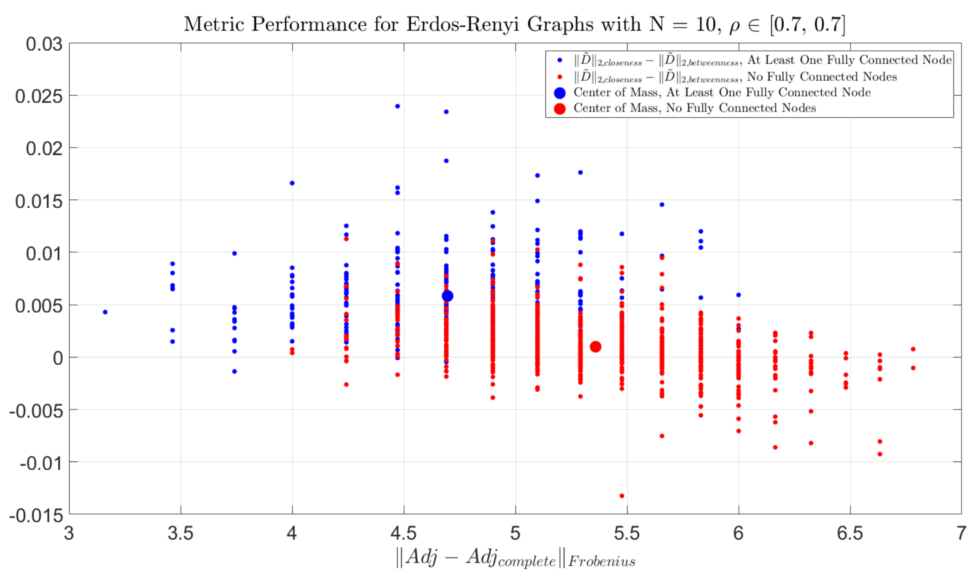


Figure 4.15: Difference between norms with actuation weighting as a function of closeness centrality and betweenness centrality vs. distance from a complete graph for Erdos-Renyi graphs with $N = 10$ and $\rho = 0.7$. 1000 data points comprise this figure.

We can see that when there is a high proportion of graphs with at least one fully connected node the average difference in norm values in more positive than the case when there are mostly graphs with no fully connected nodes. This implies that on average $\|\tilde{D}\|_2$ will be lower when weighting actuators as a function of betweenness centrality than it would when weighting actuators as a function of closeness centrality when there are larger proportions of graphs with at least one fully connected node.

Since there is evidence that the presence of fully connected nodes provides a situation in which weighting as a function of betweenness centrality tends to outperform weighting as a function of other centrality measures, it is illuminating to look at the statistics only considering those graphs which have no fully connected nodes. These tend to be the graph types in which the benefits of weighting by betweenness centrality are not as statistically significant. Hypothesis testing comparing the hypotheses $\{H_{01}, H_1\}$ only considering the graphs with no fully connected nodes is shown in Figure (4.16) and statistically significant cases are highlighted in green in Table 4.2.

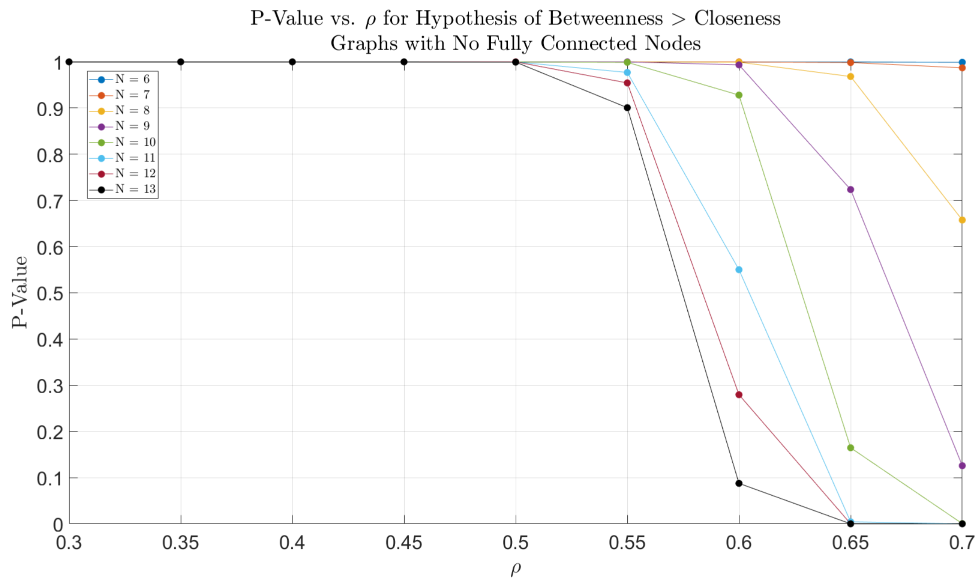


Figure 4.16: Comparison of hypotheses $\{H_{01}, H_1\}$ across Erdos-Renyi graphs. The statistics are generated only using graphs with no fully connected nodes.

	ρ									
		0.3	**0.35**	**0.4**	**0.45**	**0.5**	**0.55**	**0.6**	**0.65**	**0.7**
N	**6**	1	1	1	1	1	1	0.9999	0.9999	0.9991
	7	1	1	1	1	1	1	1	0.9985	0.9871
	8	1	1	1	1	1	1	0.9993	0.9683	0.6576
	9	1	1	1	1	1	1	0.9935	0.7236	0.1258
	10	1	1	1	1	1	0.999	0.9281	0.1646	0.0004
	11	1	1	1	1	0.9998	0.9771	0.5501	0.0043	0
	12	1	1	1	1	1	0.9544	0.2795	0	0
	13	1	1	1	1	0.9994	0.9008	0.0876	0	0

Table 4.2: Comparison of hypotheses $\{H_{01}, H_1\}$ across Erdos-Renyi graphs. The statistics are generated only using graphs with no fully connected nodes and statistically significant cases ($p < 0.05$) are highlighted in green.

A significant result is that there are still statistically significant cases which provide evidence in support of the hypothesis that weighting as a function of betweenness centrality provides benefit over weighting as a function of closeness centrality *even when only considering graphs with no fully connected nodes*. In particular, these statistically significant cases tend to be for larger N as well as larger ρ. This is an important results as it provides us with more evidence toward a more systematic methodology to select actuators. We are seeking a heuristic which can be applied to general networks in which there may or may not be fully connected nodes. For robustness reasons, if there is only one node which is fully connected it may not be desirable to put all of the actuation on that node (even if it is optimal in the sense of norm minimization). Furthermore, one link failure could change whether a node is fully connected, motivating the desire to use a heuristic with more built-in redundancy. These statistical results imply that betweenness centrality would be a good heuristic to use that still allows us to get close to optimal performance without sacrificing resilience.

4.13 Comparing Betweenness Centrality with Degree Centrality

We can similarly perform hypothesis testing for the other pertinent pairs of hypotheses we laid out previously. Comparing the pair of hypotheses $\{H_{02}, H_2\}$ allows us to investigate if weighting actuators as a function of betweenness centrality provides any tangible benefits over weighting actuators as a function of degree centrality. A plot of appropriate p-values vs. ρ is shown in Figure (4.17) and statistically significant cases are highlighted in Table 4.3.

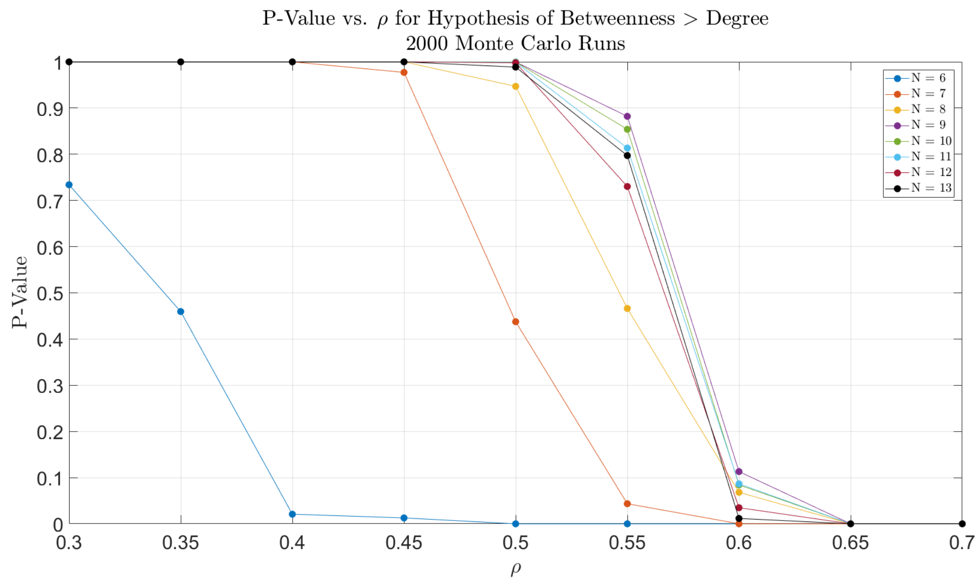


Figure 4.17: Comparison of hypotheses $\{H_{02}, H_2\}$ across Erdos-Renyi graphs. Each data point is calculated using 2000 Monte Carlo runs.

		ρ								
		0.3	**0.35**	**0.4**	**0.45**	**0.5**	**0.55**	**0.6**	**0.65**	**0.7**
N	**6**	0.734	0.4595	0.0208	0.0128	0	0	0	0	0
	7	1	1	1	0.9772	0.4375	0.0435	0.0002	0	0
	8	1	1	1	1	0.947	0.4662	0.0682	0	0
	9	1	1	1	1	0.9997	0.8821	0.1131	0	0
	10	1	1	1	1	0.9997	0.854	0.0842	0	0
	11	1	1	1	1	0.999	0.8135	0.0865	0	0
	12	1	1	1	1	0.998	0.7302	0.0349	0	0
	13	1	1	1	1	0.9887	0.7972	0.0116	0	0

Table 4.3: Comparison of hypotheses $\{H_{02}, H_2\}$ across Erdos-Renyi graphs. Each data point is calculated using 2000 Monte Carlo runs and statistically significant cases ($p < 0.05$) are highlighted in green.

We again provide data only considering graphs with no fully connected nodes in Figure (4.18) and Table 4.4.

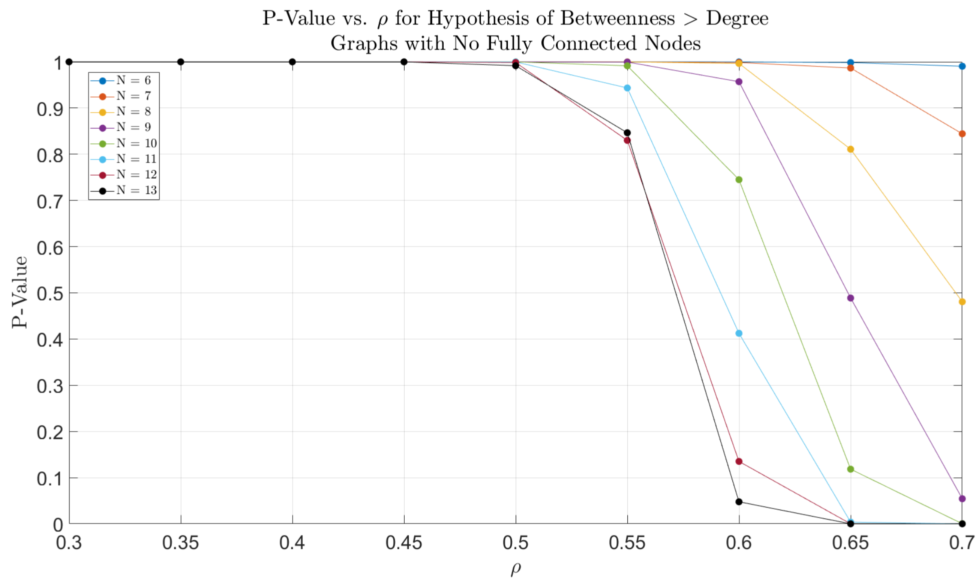


Figure 4.18: Comparison of hypotheses $\{H_{02}, H_2\}$ across Erdos-Renyi graphs. The statistics are generated only using graphs with no fully connected nodes.

		ρ								
		0.3	**0.35**	**0.4**	**0.45**	**0.5**	**0.55**	**0.6**	**0.65**	**0.7**
N	**6**	1	1	1	1	1	1	0.9999	0.9985	0.9905
	7	1	1	1	1	1	1	0.9986	0.9867	0.8443
	8	1	1	1	1	1	1	0.9966	0.8109	0.4807
	9	1	1	1	1	1	0.9996	0.9571	0.4888	0.0543
	10	1	1	1	1	1	0.9915	0.745	0.1183	0.0002
	11	1	1	1	1	0.9998	0.9435	0.4124	0.0037	0
	12	1	1	1	1	0.9986	0.83	0.1351	0	0
	13	1	1	1	1	0.9917	0.8465	0.0477	0	0

Table 4.4: Comparison of hypotheses $\{H_{02}, H_2\}$ across Erdos-Renyi graphs. The statistics are generated only using graphs with no fully connected nodes and statistically significant cases ($p < 0.05$) are highlighted in green.

There is again strong statistical evidence to support that for larger N and larger ρ weighting actuators as a function of betweenness centrality provides improvement over weighting actuators as a function of degree centrality.

Comparing the pair of hypotheses $\{H_{03}, H_3\}$ allows us to investigate if weighting actuators as a function of betweenness centrality improves over weighting actuators as a function of eigenvector centrality. A plot of appropriate p-values vs. ρ is shown in Figure (4.19) and statistically significant cases are highlighted in Table 4.5.

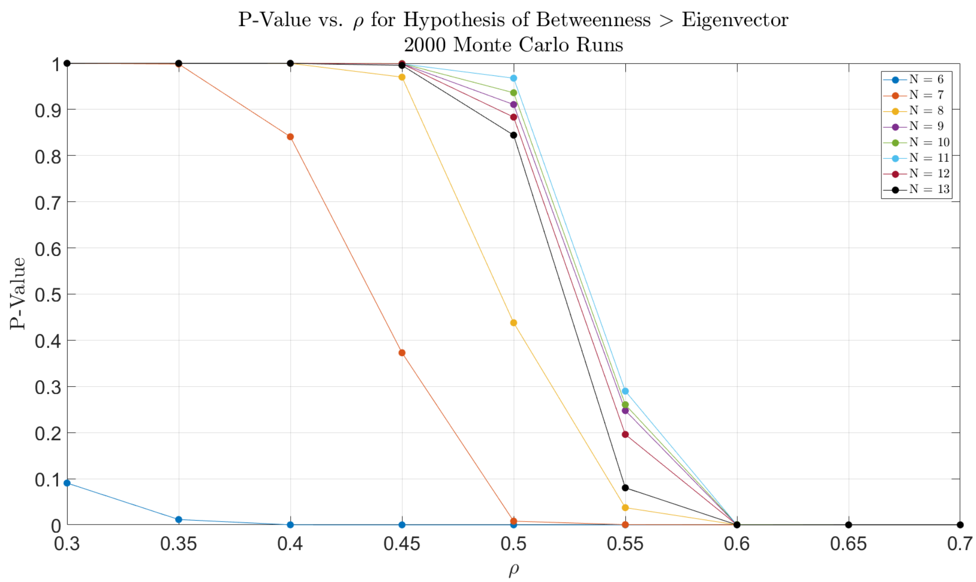


Figure 4.19: Comparison of hypotheses $\{H_{03}, H_3\}$ across Erdos-Renyi graphs. Each data point is calculated using 2000 Monte Carlo runs.

N \ ρ	0.3	0.35	0.4	0.45	0.5	0.55	0.6	0.65	0.7
6	0.0906	0.0116	0.0001	0	0	0	0	0	0
7	1	0.9985	0.8409	0.3729	0.008	0.0005	0	0	0
8	1	1	0.9999	0.9701	0.4379	0.0373	0.0001	0	0
9	1	1	1	0.999	0.9107	0.2476	0	0	0
10	1	1	1	0.9995	0.9362	0.2604	0.0006	0	0
11	1	1	1	0.9987	0.9678	0.2898	0	0	0
12	1	1	1	0.9989	0.8834	0.1959	0	0	0
13	1	1	1	0.9955	0.8443	0.0803	0	0	0

Table 4.5: Comparison of hypotheses $\{H_{03}, H_3\}$ across Erdos-Renyi graphs. Each data point is calculated using 2000 Monte Carlo runs and statistically significant cases ($p < 0.05$) are highlighted in green.

We again provide data only considering graphs with no fully connected nodes in Figure (4.20) and Table 4.6.

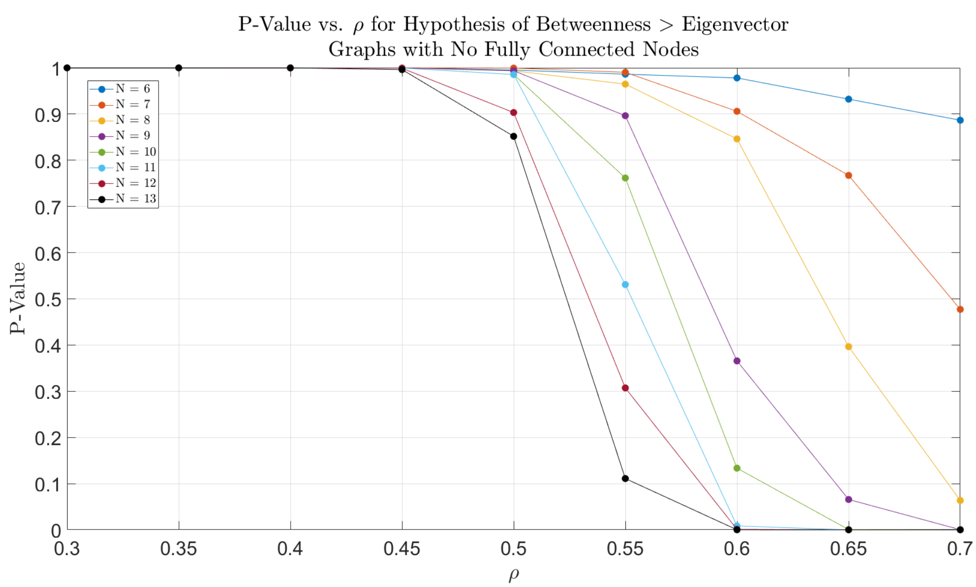


Figure 4.20: Comparison of hypotheses $\{H_{03}, H_3\}$ across Erdos-Renyi graphs. The statistics are generated only using graphs with no fully connected nodes.

		ρ								
		0.3	**0.35**	**0.4**	**0.45**	**0.5**	**0.55**	**0.6**	**0.65**	**0.7**
	6	1	1	1	1	0.9952	0.9863	0.9782	0.9323	0.8867
	7	1	1	1	1	0.9995	0.9904	0.906	0.7672	0.4772
	8	1	1	1	0.9999	0.9936	0.9648	0.8462	0.3965	0.0635
N	**9**	1	1	1	1	0.9938	0.8964	0.3657	0.0657	0.0003
	10	1	1	1	0.9998	0.9856	0.7616	0.1335	0.0005	0
	11	1	1	1	0.9996	0.9857	0.5309	0.0084	0	0
	12	1	1	0.9998	0.9991	0.9033	0.3071	0.0011	0	0
	13	1	1	1	0.9963	0.8518	0.111	0.0001	0	0

Table 4.6: Comparison of hypotheses $\{H_{03}, H_3\}$ across Erdos-Renyi graphs. The statistics are generated only using graphs with no fully connected nodes and statistically significant cases ($p < 0.05$) are highlighted in green.

Once again networks with larger N and larger ρ exhibit strong statistical evidence that weighting actuators as a function of betweenness centrality provides a benefit over weighting actuators as a function of eigenvector centrality. Even when only considering graphs with no fully connected nodes these trends remain.

4.15 Comparing Betweenness Centrality with Information Centrality

Finally we compare the pair of hypotheses $\{H_{04}, H_4\}$, allowing us to investigate whether weighting actuators as a function of betweenness centrality provides any benefit over weighting actuators as a function of information centrality. A plot of appropriate p-values vs. ρ is shown in Figure (4.21) and statistically significant cases are highlighted in Table 4.7.

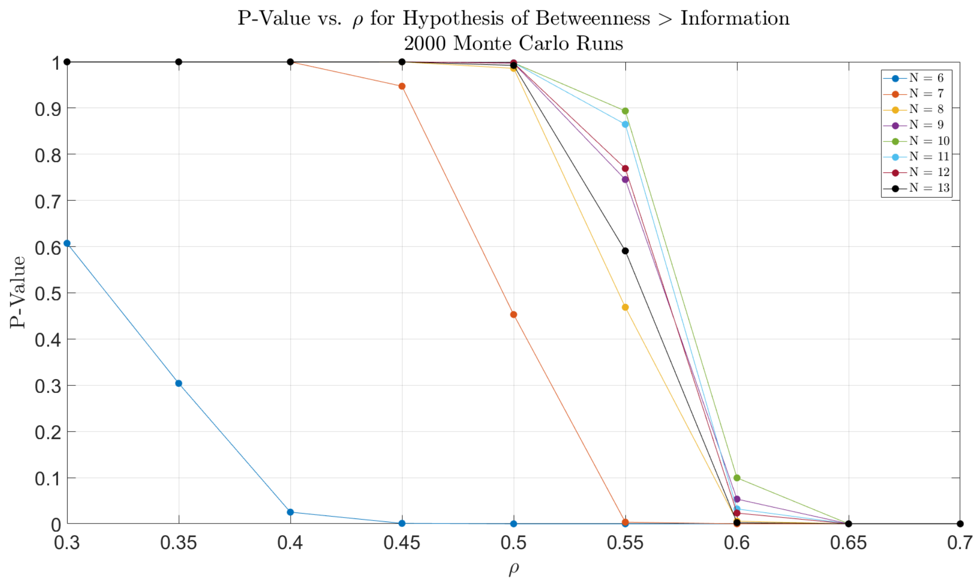


Figure 4.21: Comparison of hypotheses $\{H_{04}, H_4\}$ across Erdos-Renyi graphs. Each data point is calculated using 2000 Monte Carlo runs.

	ρ									
		0.3	**0.35**	**0.4**	**0.45**	**0.5**	**0.55**	**0.6**	**0.65**	**0.7**
N	**6**	0.6073	0.3041	0.0253	0.001	0	0	0	0	0
	7	1	1	1	0.9473	0.453	0.0036	0	0	0
	8	1	1	1	0.9999	0.986	0.4688	0.0057	0	0
	9	1	1	1	1	0.9971	0.7453	0.0535	0	0
	10	1	1	1	1	0.9978	0.8937	0.0995	0	0
	11	1	1	1	1	0.998	0.8648	0.0323	0	0
	12	1	1	1	1	0.9976	0.7692	0.0233	0	0
	13	1	1	1	1	0.9923	0.5908	0.0026	0	0

Table 4.7: Comparison of hypotheses $\{H_{04}, H_4\}$ across Erdos-Renyi graphs. Each data point is calculated using 2000 Monte Carlo runs and statistically significant cases ($p < 0.05$) are highlighted in green.

We again provide data only considering graphs with no fully connected nodes in Figure (4.22) and Table 4.8.

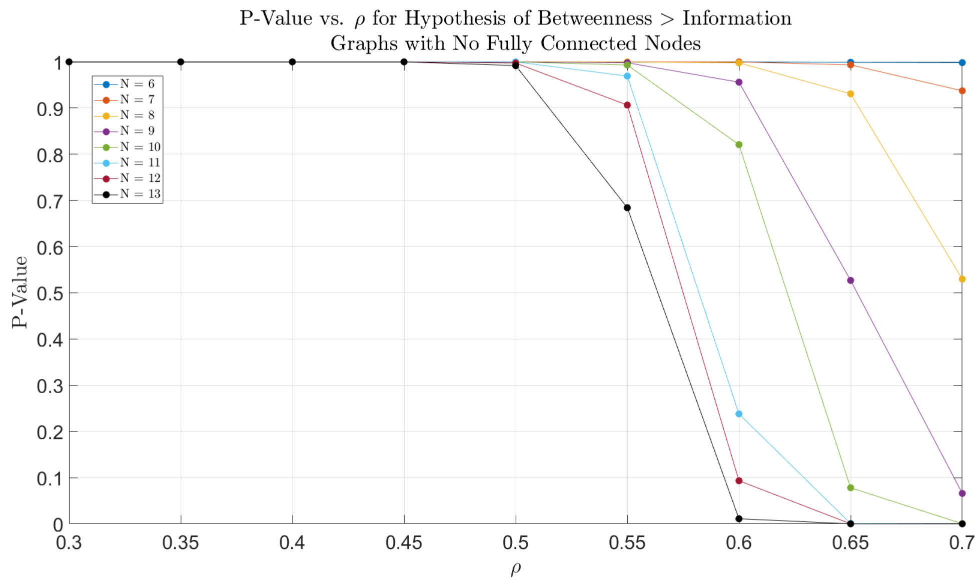


Figure 4.22: Comparison of hypotheses $\{H_{04}, H_4\}$ across Erdos-Renyi graphs. The statistics are generated only using graphs with no fully connected nodes.

N \ ρ	0.3	0.35	0.4	0.45	0.5	0.55	0.6	0.65	0.7
6	1	1	1	1	1	1	1	0.9991	0.9983
7	1	1	1	1	1	1	0.9999	0.9935	0.9374
8	1	1	1	1	1	1	0.9979	0.9309	0.5298
9	1	1	1	1	1	0.9981	0.956	0.5269	0.0658
10	1	1	1	1	1	0.9932	0.8208	0.078	0.0003
11	1	1	1	1	1	0.9692	0.238	0.0007	0
12	1	1	1	1	0.9974	0.9065	0.0933	0	0
13	1	1	1	1	0.992	0.6842	0.011	0	0

Table 4.8: Comparison of hypotheses $\{H_{04}, H_4\}$ across Erdos-Renyi graphs. The statistics are generated only using graphs with no fully connected nodes and statistically significant cases ($p < 0.05$) are highlighted in green.

We again see statistical evidence which shows that for larger N and larger ρ betweenness centrality may provide benefits over information centrality when weighting actuators. These results also hold when only considering graphs with no fully connected nodes.

4.16 Sensor Allocation

Up until this point we have focused on studying the actuator assignment problem. Now we seek to gain insight into how similar techniques can be utilized to understand the analogous sensor assignment problem. This necessitates investigating the estimation component of our distributed control and estimation framework. Denoting the transfer function matrix of the estimation component as $\tilde{G}$, recall that in Chapter 3 we considered the general condition $\|\tilde{G}\|_{H_\infty} < \gamma$ for $\gamma \in \mathbb{R}_{>0}$. We have encountered γ in the metric $\xi = \gamma * \gamma'$ which we seek to minimize. We now set out to identify important nodes to select as sensors and consider convexly weighting sensors to minimize γ. For ease of discussion we consider a scalar process x with state transition matrix $A = 1$. Let $C_i = 1$ with observer gains $L_i = \alpha_i$ for each $i \in [1, \ldots, N]$ with

$$\sum_{i=1}^{N} \alpha_i = 1 \tag{4.22}$$

This scheme allows us to control if a sensor is active or inactive through the allocation of the α_i terms. If $\alpha_i = 0$ then node i is inactive in a sensing sense. The same reasons that apply for using only a subset of available actuators extend to the sensing case. We are operating under the assumptions of power limitations or a hierarchy of sensing tasks where we may only allocate a subset of sensors to a specific task for a limited amount of time. Here the relative magnitude of α_i values correspond to the magnitude of observer gains used in the estimation process. To compare across our centrality measures of interest, we weight a sensor i as a function of a specific centrality measure by selecting

$$\alpha_i = \frac{CM(i)}{\sum_{j \in V} CM(j)} \tag{4.23}$$

where $CM(\cdot)$ is the centrality measure of interest. Recall that in Section 3.4 of Chapter 3 we used a convex version of the estimation component LMI given in (3.24) to greatly simplify the analysis. We again employ this formulation for the following simulations. Recall that the LMI (3.24) is a function of X, A, L_i, C_i, and W^E_{ij} for $i, j \in [1, \ldots, N]$. Thus, given a predefined set of C_i values and with observer weights given by (4.23), we can use a semidefinite program to find feasible $X \succ 0$ and W^E_{ij} for $i, j \in [1, \ldots, N]$ which satisfy (3.24) while minimizing γ.

4.16.1 Test Cases

We consider the following hypotheses:

H'_{01}: Selecting sensors by weighting as a function of betweenness centrality provides no benefit over selecting sensors by weighting as a function of closeness centrality.

H'_1: Selecting sensors by weighting as a function of betweenness centrality is better than selecting sensors by weighting as a function of closeness centrality.

H'_{02}: Selecting sensors by weighting as a function of betweenness centrality provides no benefit over selecting sensors by weighting as a function of degree centrality.

H'_2: Selecting sensors by weighting as a function of betweenness centrality is better than selecting sensors by weighting as a function of degree centrality.

H'_{03}: Selecting sensors by weighting as a function of betweenness centrality provides no benefit over selecting sensors by weighting as a function of eigenvector centrality.

H'_3: Selecting sensors by weighting as a function of betweenness centrality is better than selecting sensors by weighting as a function of eigenvector centrality.

H'_{04}: Selecting sensors by weighting as a function of betweenness centrality provides no benefit over selecting sensors by weighting as a function of information centrality.

H'_4: Selecting sensors by weighting as a function of betweenness centrality is better than selecting sensors by weighting as a function of information centrality.

We perform hypothesis testing to compare the pair of hypotheses $\{H'_{01}, H'_1\}$, $\{H'_{02}, H'_2\}$, $\{H'_{03}, H'_3\}$, and $\{H'_{04}, H'_4\}$ and generate appropriate p-values given by (4.20). Our test suite consists of Erdos-Renyi graphs $G(N, \rho)$ with $N \in [6, 8, 10, 12, 14]$ and $\rho \in [0.3, 0.4, 0.5, 0.6, 0.7]$.

4.16.2 Comparing Betweenness Centrality with Closeness Centrality

Comparing the pair of hypotheses $\{H'_{01}, H'_1\}$ allows us to investigate if weighting sensors as a function of betweenness centrality provides any tangible benefits over weighting sensors as a function of closeness centrality. A plot of appropriate p-values vs. ρ is shown in Figure (4.23) and a table highlighting statistically significant cases is shown in Table 4.9.

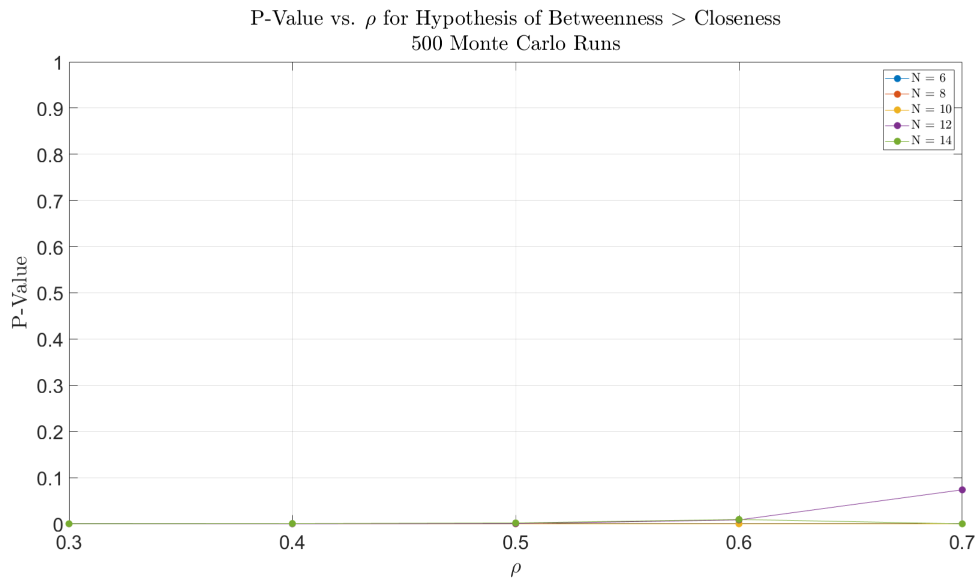


Figure 4.23: Comparison of hypotheses $\{H'_{01}, H'_1\}$ across Erdos-Renyi graphs. Each data point is calculated using 500 Monte Carlo runs.

N \ ρ	0.3	0.4	0.5	0.6	0.7
6	0	0	0	0	0
8	0	0	0	0	0
10	0	0	0	0	0
12	0	0	0.0004	0.0084	0.0736
14	0.0001	0.0006	0.0019	0.0094	0

Table 4.9: Comparison of hypotheses $\{H'_{01}, H'_1\}$ across Erdos-Renyi graphs. Each data point is calculated using 500 Monte Carlo runs and statistically significant cases ($p < 0.05$) are highlighted in green.

We see that for all simulated cases there is extremely strong statistical evidence that weighting sensors as a function of betweenness centrality provides a benefit to weighting sensors as a function of closeness centrality, save for the case with $N = 12$ and $\rho = 0.7$. This p-value, while not below our defined threshold for statistical significance, is noted to still be relatively small in magnitude.

4.16.3 Comparing Betweenness Centrality with Degree Centrality

Comparing the pair of hypotheses $\{H'_{02}, H'_2\}$ allows us to investigate if weighting sensors as a function of betweenness centrality provides any tangible benefits over weighting sensors as a function of degree centrality. A plot of appropriate p-values vs. ρ is shown in Figure (4.24) and a table highlighting statistically significant cases is shown in Table 4.10.

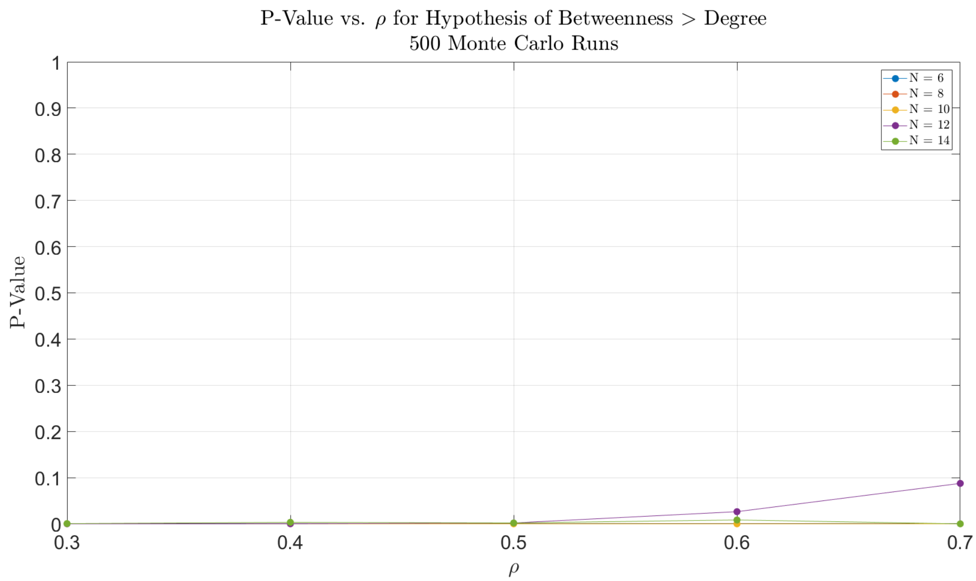


Figure 4.24: Comparison of hypotheses $\{H'_{02}, H'_2\}$ across Erdos-Renyi graphs. Each data point is calculated using 500 Monte Carlo runs.

		ρ				
		0.3	**0.4**	**0.5**	**0.6**	**0.7**
N	**6**	0	0	0	0	0
	8	0	0	0	0	0
	10	0	0	0	0	0
	12	0	0.0001	0.0019	0.0263	0.0875
	14	0.0005	0.0033	0.0019	0.0083	0

Table 4.10: Comparison of hypotheses $\{H'_{02}, H'_2\}$ across Erdos-Renyi graphs. Each data point is calculated using 500 Monte Carlo runs and statistically significant cases ($p < 0.05$) are highlighted in green.

We see that for all simulated cases there is extremely strong statistical evidence that weighting sensors as a function of betweenness centrality provides a benefit to weighting sensors as a function of degree centrality, save for the case with $N = 12$ and $\rho = 0.7$. This p-value, while not below our defined threshold for statistical significance, is noted to still be relatively small in magnitude.

4.16.4 Comparing Betweenness Centrality with Eigenvector Centrality

Comparing the pair of hypotheses $\{H'_{03}, H'_3\}$ allows us to investigate if weighting sensors as a function of betweenness centrality provides any tangible benefits over weighting sensors as a function of eigenvector centrality. A plot of appropriate p-values vs. ρ is shown in Figure (4.25) and a table highlighting statistically significant cases is shown in Table 4.11.

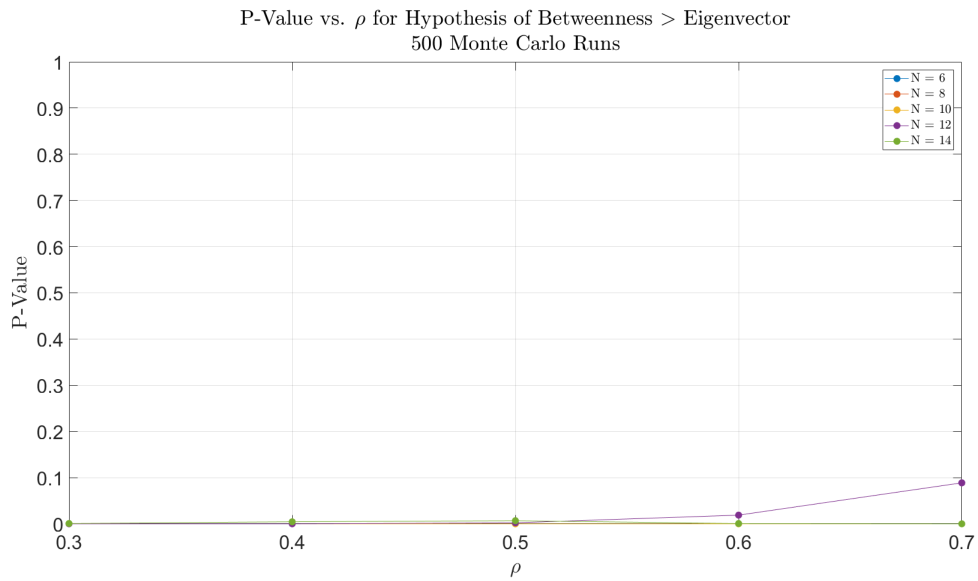


Figure 4.25: Comparison of hypotheses $\{H'_{03}, H'_3\}$ across Erdos-Renyi graphs. Each data point is calculated using 500 Monte Carlo runs.

		ρ				
		0.3	**0.4**	**0.5**	**0.6**	**0.7**
N	**6**	0	0	0	0	0
	8	0	0	0	0	0
	10	0	0	0	0	0
	12	0	0	0.0022	0.0188	0.0887
	14	0.0007	0.0044	0.0067	0.0006	0

Table 4.11: Comparison of hypotheses $\{H'_{03}, H'_3\}$ across Erdos-Renyi graphs. Each data point is calculated using 500 Monte Carlo runs and statistically significant cases ($p < 0.05$) are highlighted in green.

We see that for all simulated cases there is extremely strong statistical evidence that weighting sensors as a function of betweenness centrality provides a benefit to weighting sensors as a function of eigenvector centrality, save for the case with $N = 12$ and $\rho = 0.7$. This p-value, while not below our defined threshold for statistical significance, is noted to still be relatively small in magnitude.

4.16.5 Comparing Betweenness Centrality with Information Centrality

Comparing the pair of hypotheses $\{H'_{04}, H'_4\}$ allows us to investigate if weighting sensors as a function of betweenness centrality provides any tangible benefits over weighting sensors as a function of information centrality. A plot of appropriate p-values vs. ρ is shown in Figure (4.26) and a table highlighting statistically significant cases is shown in Table 4.12.

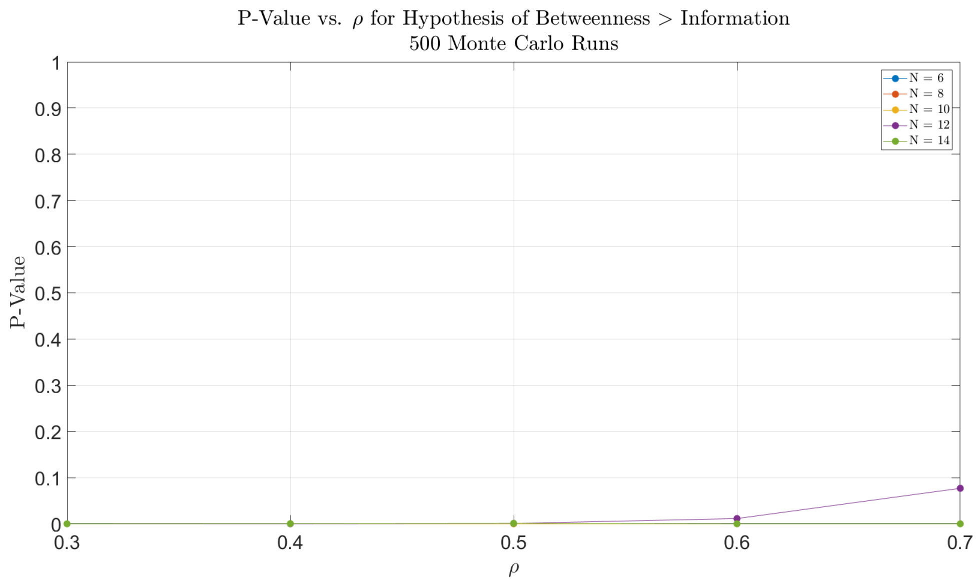


Figure 4.26: Comparison of hypotheses $\{H'_{04}, H'_4\}$ across Erdos-Renyi graphs. Each data point is calculated using 500 Monte Carlo runs.

		ρ				
		0.3	**0.4**	**0.5**	**0.6**	**0.7**
	6	0	0	0	0	0
	8	0	0	0	0	0
N	**10**	0	0	0	0	0
	12	0	0	0.0008	0.0114	0.0769
	14	0	0.0004	0.0011	0	0

Table 4.12: Comparison of hypotheses $\{H'_{04}, H'_4\}$ across Erdos-Renyi graphs. Each data point is calculated using 500 Monte Carlo runs and statistically significant cases ($p < 0.05$) are highlighted in green.

We see that for all simulated cases there is extremely strong statistical evidence that weighting sensors as a function of betweenness centrality provides a benefit to weighting sensors as a function of information centrality, save for the case with $N = 12$ and $\rho = 0.7$. This p-value, while not below our defined threshold for statistical significance, is noted to still be relatively small in magnitude.

4.16.6 Typical Values for Upper Bound on Estimation Block Norm

For the Erdos-Renyi graph $G(7, 0.5)$ shown in Figure (4.27), the minimum γ^* values achievable when weighting sensors according to the centrality measures of interest as in (4.23) are displayed in Figure (4.28).

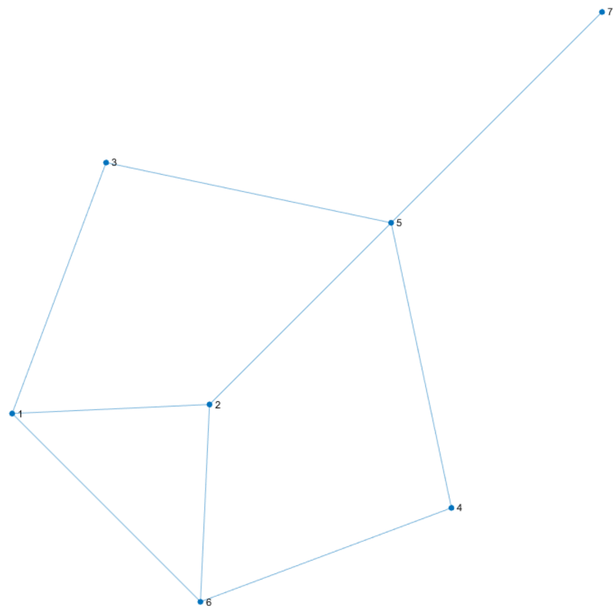


Figure 4.27: Erdos-Renyi graph with $N = 7$ and connection probability $\rho = 0.5$.

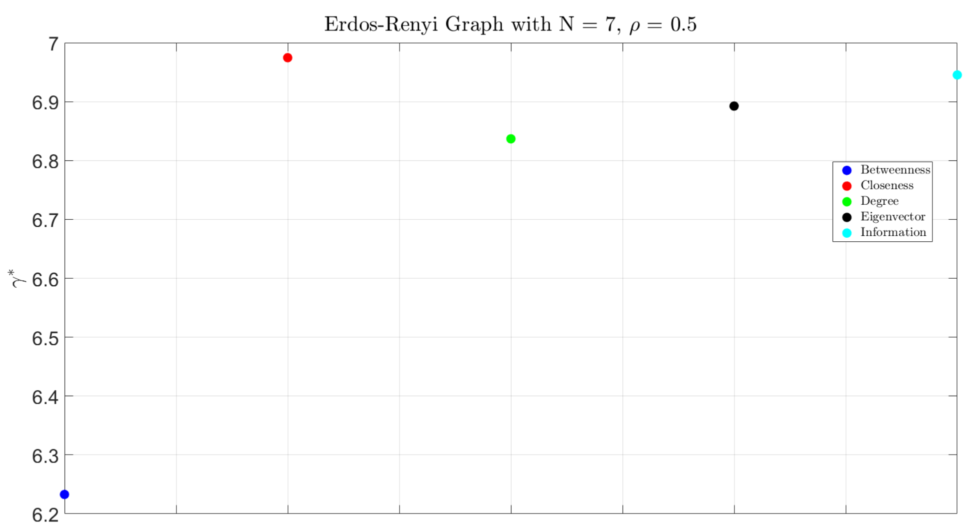


Figure 4.28: Minimum γ^* values when weighting sensors by the various centrality measures of interest for an Erdos-Renyi graph with $N = 7$ and $\rho = 0.5$.

We see that when weighting sensors using betweenness centrality, we achieve a minimum $\gamma^* = 6.233$. Weighting sensors according to the other centrality measures of interest result in larger minimum achievable values of γ^*, with the largest occurring when weighting sensors by closeness centrality which yields $\gamma^* = 6.975$. This is but one example, but it gives a good idea of typical values of γ^* which are achievable for the Erdos-Renyi graphs under consideration.

4.17 Conclusions

We have considered the problem of actuator and sensor allocation for the distributed estimation and control framework detailed in Chapter 3. We considered a set of centrality measures as heuristics for actuator and sensor allocation. Specifically, we considered betweenness, closeness, degree, and information centrality measures. We outlined a methodology to use such centrality measures as weighting schemes for agent selection. With the goal of minimizing a key norm, we compared the results across of suite of Erdos-Renyi random graphs. We showed that for a given connected graph, it is optimal to select fully-connected agents as actuators. We further showed that among the centrality measures considered, in cases both with and without the presence of fully connected agents, betweenness centrality statistically has the most evidence suggesting that it is the best heuristic to use for actuator selection given a sufficient edge density. We also showed that these results extend to the problem of sensor selection, although in this case seemingly without the edge density restriction.

5

Robustness and Scalability for Distributed Control and Estimation

We are interested in assessing how our statistical conclusions regarding actuator selection extend from both a robustness and a scalability standpoint. The areas of robustness we will consider include performance under graph generation methods other than Erdos-Renyi graphs, under irrecoverable node failure, and under sparsely available control authority.

In Section 5.1 we consider different graph generation methods in addition to Erdos-Renyi graphs. Specifically, we look at Barabasi-Albert graphs as well as Watts-Strogatz graphs. In Section 5.2 we consider the impact of irrecoverable actuator failures. We consider cases in which one or two actuators fail, respectively. In Section 5.3 we analyze the impact of sparsely available control authority on our statistical results. These are cases in which individual agents do not have full controllability of the system. Rather, the network must rely on the joint controllability amongst all of the agents. We show that this paradigm change does result in different statistical conclusions being drawn; specifically, degree centrality shows strong statistical evidence as being the best actuator selection heuristic in these cases rather than betweenness centrality for networks with sufficient edge density. In Section 5.4 we consider how our statistical results generalize to larger networks. Finally, in Section 5.5 we provide a summary of key conclusions and takeaways.

5.1 Robustness to Graph Generation Method

We previously considered graphs generated using the Erdos-Renyi model. While this model is extremely useful in facilitating general conclusions on graphs in a random way, it does have certain limitations when comparing Erdos-Renyi-generated graphs to some graphs typically encountered in design. We will highlight some of these limitations and discuss additional graph generation methods which specifically address these concerns.

5.1.1 Barabasi-Albert Model

One belief among researchers is that many real-world and technological networks approximately exhibit power-law degree distributions where the fraction $P(k)$ of nodes in the network having degree k is given for large k by

$$P(k) \sim k^{-\gamma} \tag{5.1}$$

where typically $2 < \gamma < 3$[44]. Such networks are called scale-free and exhibit relatively few hub nodes with high degrees of connectivity. Examples of studied networks with purported power-law degree distributions include the movie actor collaboration network (where nodes are actors and edges indicate that connected actors have worked in a movie together) as well as phone call networks (where nodes are phone numbers and edges indicate a completed phone call directed from the caller to the receiver)[45]. Note that this contrasts with the Erdos-Renyi model which produces networks with degree distributions that converge to a Poisson distribution[45].

Barabasi-Albert networks have also been shown to have applications to robotic networks, with one example being drone swarms foraging in dynamic environments[46]. Such scale-free networks work to accelerate the collective response of the swarm to help cope with environmental changes, but at the cost of a less coherent collective design. Since relatively few hubs exert a large degree of influence on the network, collective decision-making can become fickle in these types of networks.

The Barabasi-Albert model begins with an initial connected seed network consisting of n_0 nodes. New nodes are then sequentially added to the network, with each additional node connecting to $n \leq n_0$ existing nodes with a probability that is proportional to the current nodes' existing degrees. The probability p_i that this new node will connect with existing node i is

$$p_i = \frac{k_i}{\sum_j k_j} \tag{5.2}$$

where k_i is the degree of node i and index j sums over all of the currently existing nodes. It is clear that hub nodes which have high degrees will accumulate new nodes over time in a mechanism known as preferential attachment, with new nodes "preferring" to attach to more well-connected nodes over their less well-connected peers. Given a three node line graph as the seed network, Figures (5.1) - (5.3) show networks with $N = 20$ nodes generated using the Barabasi-Albert model with $n = 1$, $n = 2$, and $n = 3$, respectively. Note that the $n = 1$ case corresponds to tree structures since each additional node connects to only one other node and, correspondingly, there are no loops in these networks. As such, one can expect the results for the $n = 1$ cases to have pertinent differences to the $n = 2$ and $n = 3$ cases due to the inherent differences in graph structure.

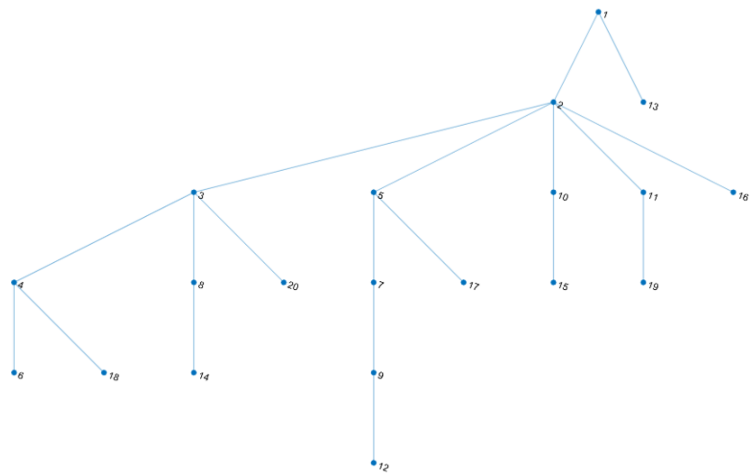


Figure 5.1: Barabasi-Albert graph with $N = 20$ nodes. This graph was seeded with a three node line graph and each sequentially added node connected to $n = 1$ existing nodes.

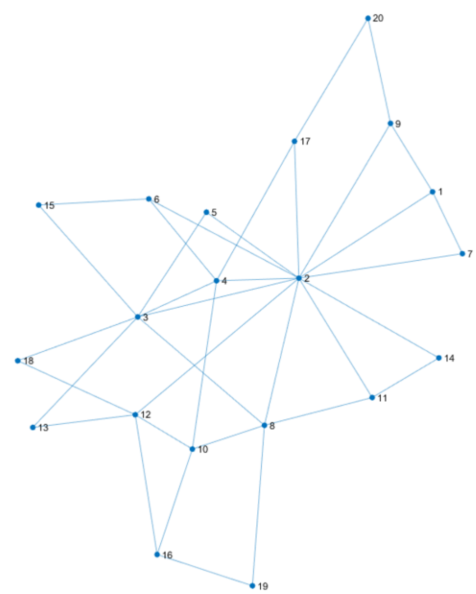


Figure 5.2: Barabasi-Albert graph with $N = 20$ nodes. This graph was seeded with a three node line graph and each sequentially added node connected to $n = 2$ existing nodes.

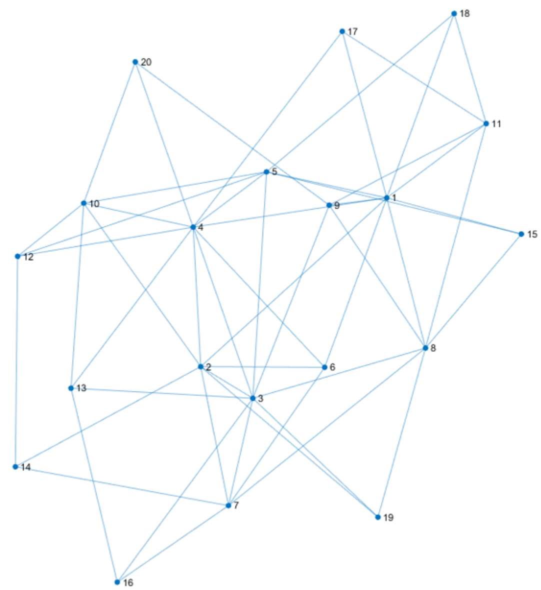


Figure 5.3: Barabasi-Albert graph with $N = 20$ nodes. This graph was seeded with a three node line graph and each sequentially added node connected to $n = 3$ existing nodes.

5.1.1.1 Test Cases

As in Chapter 3, we perform hypothesis testing to compare each pair of hypotheses $\{H_{01}, H_1\}$, $\{H_{02}, H_2\}$, $\{H_{03}, H_3\}$, and $\{H_{04}, H_4\}$ and generate appropriate p-values given by (4.20). Our test suite consists of Barabasi-Albert graphs with $N \in [10, 20, 30]$ and $n \in [1, 2, 3]$ where the seed networks are three node line graphs.

5.1.1.2 Comparing Betweenness Centrality with Closeness Centrality

Comparing the pair of hypotheses $\{H_{01}, H_1\}$ allows us to investigate if weighting actuators as a function of betweenness centrality is an improvement over weighting actuators as a function of closeness centrality. A plot of p-values vs. n is shown in Figure (5.4) and a table highlighting statistically significant cases is shown in Table 5.1.

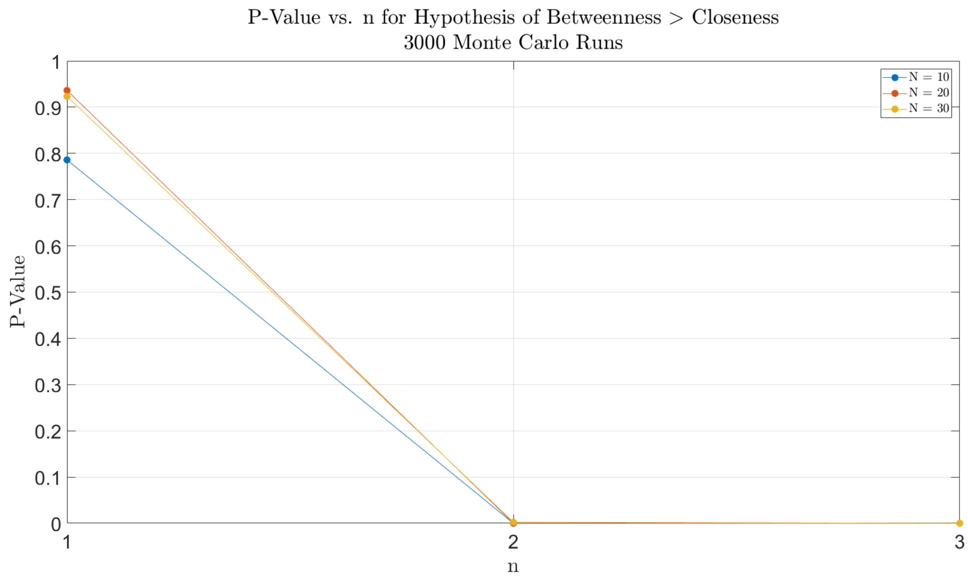


Figure 5.4: Comparison of hypotheses $\{H_{01}, H_1\}$ across Barabasi-Albert graphs. Each data point is calculated using 3000 Monte Carlo runs.

		n		
		1	**2**	**3**
N	**10**	0.7859	0	0
	20	0.9361	0.0002	0
	30	0.9233	0.0025	0

Table 5.1: Comparison of hypotheses $\{H_{01}, H_1\}$ across Barabasi-Albert graphs. Each data point is calculated using 3000 Monte Carlo runs and statistically significant cases ($p < 0.05$) are highlighted in green.

We again provide data only considering graphs with no fully connected nodes in Figure (5.5) and Table 5.2.

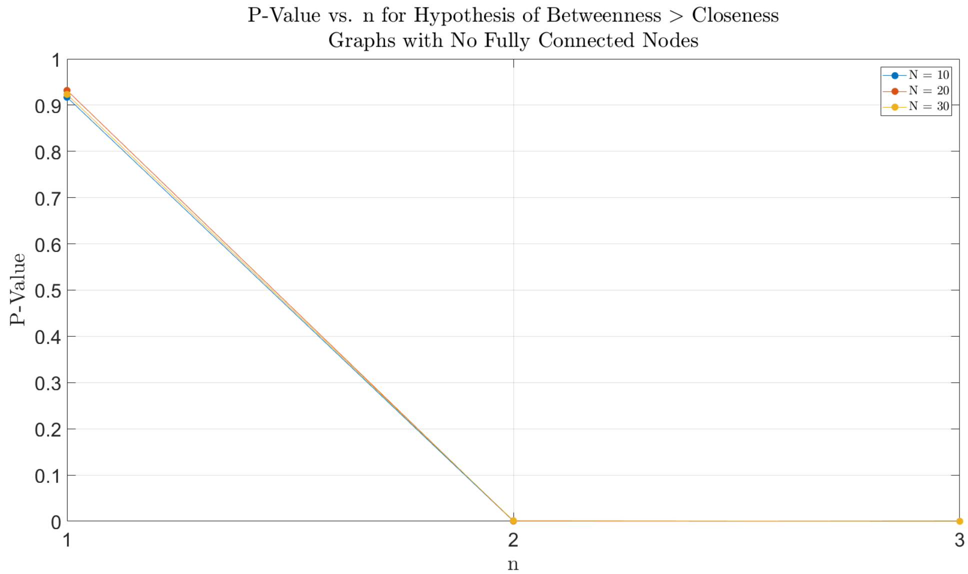


Figure 5.5: Comparison of hypotheses $\{H_{01}, H_1\}$ across Barabasi-Albert graphs. The statistics are generated only using graphs with no fully connected nodes.

		n: 1	n: 2	n: 3
N	10	0.9171	0.0004	0
	20	0.9319	0.0005	0
	30	0.9232	0.0014	0

Table 5.2: Comparison of hypotheses $\{H_{01}, H_1\}$ across Barabasi-Albert graphs. The statistics are generated only using graphs with no fully connected nodes and statistically significant cases ($p < 0.05$) are highlighted in green.

There is statistical evidence that in all cases considered with $n = 2$ and $n = 3$ betweenness centrality provides benefits over closeness centrality when weighting actuators. We see that there is not strong statistical evidence for betweenness centrality in the cases with $n = 1$ and tree graph structure. We note that the tree graphs are not very dense compared to the $n = 2$ and $n = 3$ cases for a given N. As we have seen evidence that betweenness centrality works well for denser graph structures, this result aligns with our intuition. These results also apply when only considering graphs with no fully connected nodes.

5.1.1.3 Comparing Betweenness Centrality with Degree Centrality

Comparing the pair of hypotheses $\{H_{02}, H_2\}$ allows us to investigate if weighting actuators as a function of betweenness centrality is an improvement over weighting actuators as a function of degree centrality. A plot of p-values vs. n is shown in Figure (5.6) and a table highlighting statistically significant cases is shown in Table 5.3.

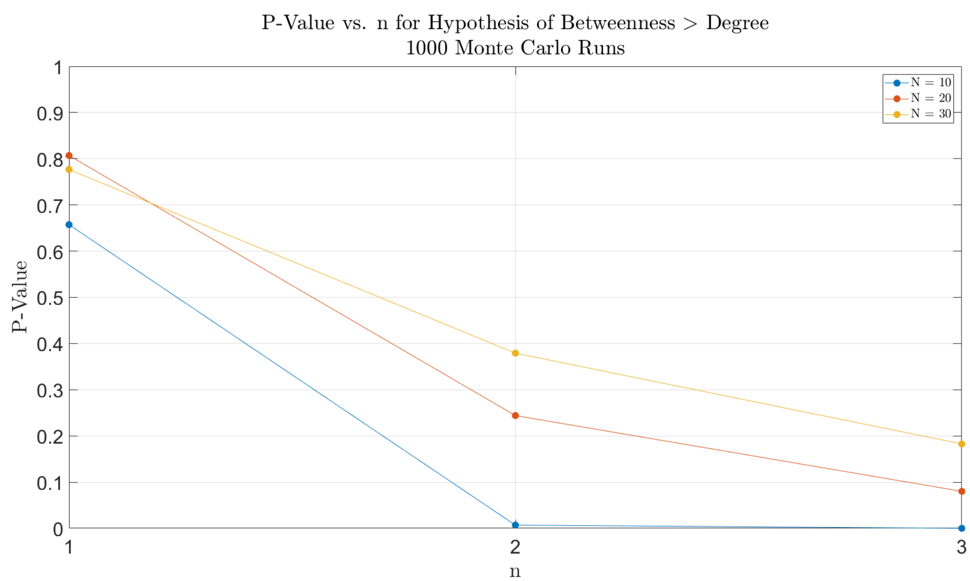


Figure 5.6: Comparison of hypotheses $\{H_{02}, H_2\}$ across Barabasi-Albert graphs. Each data point is calculated using 1000 Monte Carlo runs.

		n		
		1	**2**	**3**
N	**10**	0.6573	0.0069	0
	20	0.807	0.2441	0.0801
	30	0.7767	0.3792	0.1827

Table 5.3: Comparison of hypotheses $\{H_{02}, H_2\}$ across Barabasi-Albert graphs. Each data point is calculated using 1000 Monte Carlo runs and statistically significant cases ($p < 0.05$) are highlighted in green.

We again provide data only considering graphs with no fully connected nodes in Figure (5.7) and Table 5.4.

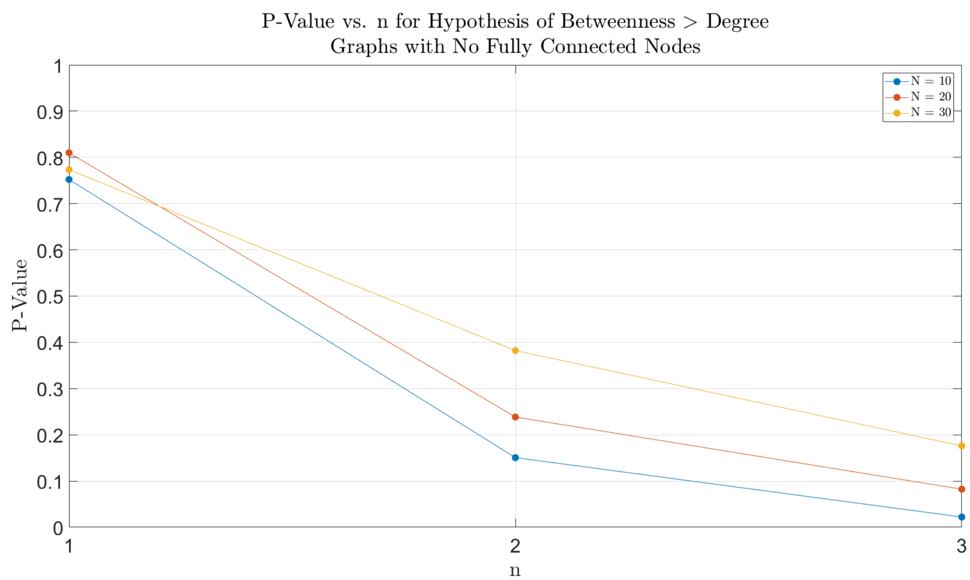


Figure 5.7: Comparison of hypotheses $\{H_{02}, H_2\}$ across Barabasi-Albert graphs. The statistics are generated only using graphs with no fully connected nodes.

		n		
		1	**2**	**3**
	10	0.7522	0.1506	0.0222
N	**20**	0.81	0.2384	0.0823
	30	0.7734	0.3822	0.1761

Table 5.4: Comparison of hypotheses $\{H_{02}, H_2\}$ across Barabasi-Albert graphs. The statistics are generated only using graphs with no fully connected nodes and statistically significant cases ($p < 0.05$) are highlighted in green.

There is statistical evidence that for Barabasi-Albert graphs with $N = 10$ nodes and $n = 2, 3$ that weighting actuators as a function of betweenness centrality provides a benefit over weighting as a function of degree centrality. When only considering graphs that have no fully connected nodes, the statistically significant case is only for $N = 10$ nodes and $n = 3$. We see that betweenness centrality is not as clear-cut an improvement over degree centrality for certain Barabasi-Albert graphs, but as n increases for the cases considered here there is more and more evidence toward rejecting the null hypothesis H_{02} in favor of betweenness centrality providing an edge over degree centrality for actuator weighting. Note that for a given n, as we increase N the overall density of the graph will decrease. Thus, we expect for each n that the statistical evidence favoring betweenness centrality will decrease as N increases, and this trend is generally observed. Likewise, for a given N as we increase n the overall density of the graph will increase. As expected, we observe that the statistical evidence favoring betweenness centrality increases for a given N as n increases.

5.1.1.4 Comparing Betweenness Centrality with Eigenvector Centrality

Comparing the pair of hypotheses $\{H_{03}, H_3\}$ allows us to investigate if weighting actuators as a function of betweenness centrality is an improvement over weighting actuators as a function of eigenvector centrality. A plot of p-values vs. n is shown in Figure (5.8) and a table highlighting statistically significant cases is shown in Table 5.5.

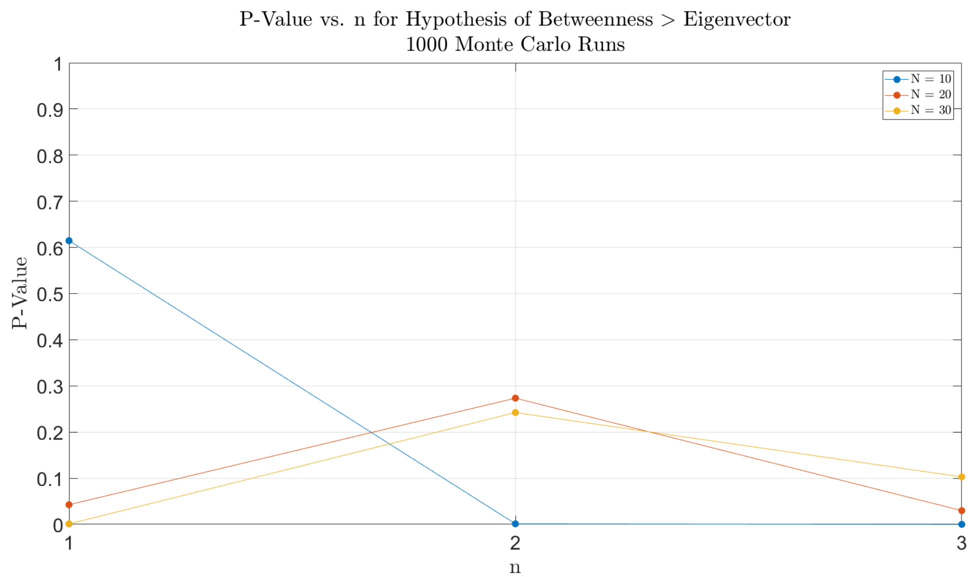


Figure 5.8: Comparison of hypotheses $\{H_{03}, H_3\}$ across Barabasi-Albert graphs. Each data point is calculated using 1000 Monte Carlo runs.

		n		
		1	**2**	**3**
N	**10**	0.6147	0.0012	0
	20	0.0426	0.2735	0.0298
	30	0.001	0.2423	0.1028

Table 5.5: Comparison of hypotheses $\{H_{03}, H_3\}$ across Barabasi-Albert graphs. Each data point is calculated using 1000 Monte Carlo runs and statistically significant cases ($p < 0.05$) are highlighted in green.

We again provide data only considering graphs with no fully connected nodes in Figure (5.9) and Table 5.6.

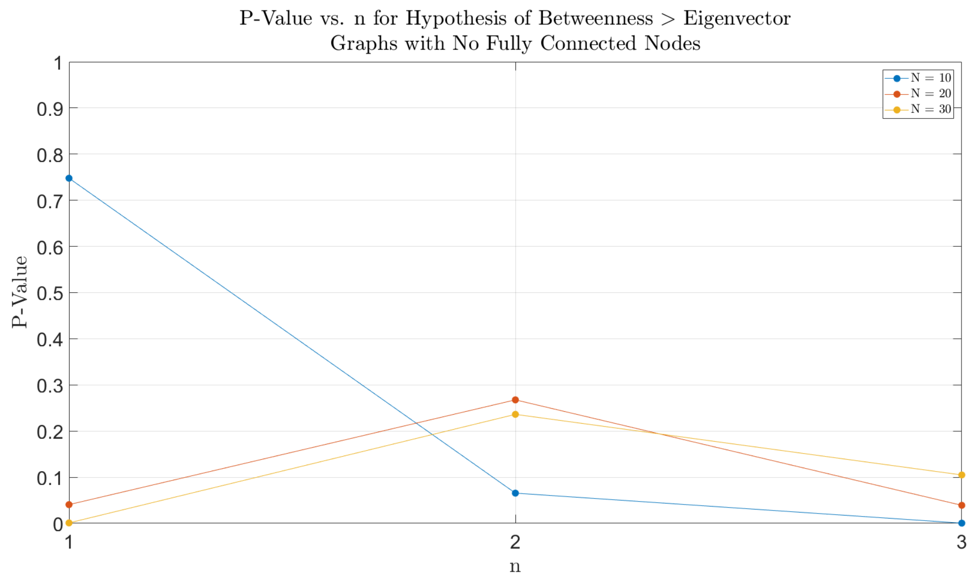


Figure 5.9: Comparison of hypotheses $\{H_{03}, H_3\}$ across Barabasi-Albert graphs. The statistics are generated only using graphs with no fully connected nodes.

		n		
		1	**2**	**3**
N	**10**	0.7478	0.0655	0.0005
	20	0.0406	0.2676	0.0391
	30	0.001	0.2362	0.1049

Table 5.6: Comparison of hypotheses $\{H_{03}, H_3\}$ across Barabasi-Albert graphs. The statistics are generated only using graphs with no fully connected nodes and statistically significant cases ($p < 0.05$) are highlighted in green.

We again see strong statistical evidence for scenarios where weighting actuators as a function of betweenness centrality would provide improvements over weighting as a function of eigenvector centrality. Note the qualitatively different result here compared to previous cases, however. Now we observe that even for tree graph structures with $n = 1$ that there is strong statistical evidence for cases with $N = 20$ and $N = 30$ favoring betweenness centrality. We conjecture that this is evidence that for our framework eigenvector centrality is not an appropriate centrality measure in acyclic graphs.

5.1.1.5 Comparing Betweenness Centrality with Information Centrality

Comparing the pair of hypotheses $\{H_{04}, H_4\}$ allows us to investigate if weighting actuators as a function of betweenness centrality is an improvement over weighting actuators as a function of information centrality. A plot of p-values vs. n is shown in Figure (5.10) and a table highlighting statistically significant cases is shown in Table 5.7.

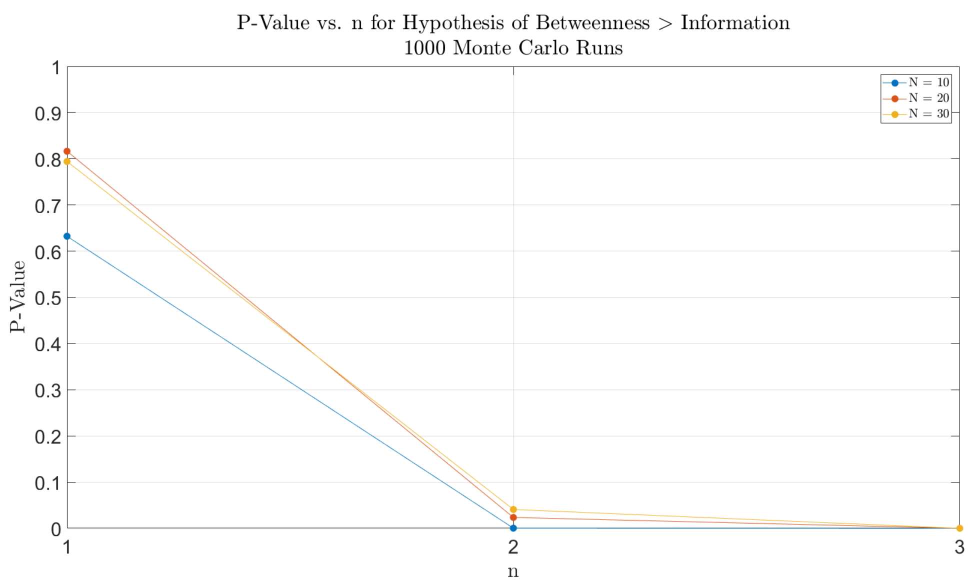


Figure 5.10: Comparison of hypotheses $\{H_{04}, H_4\}$ across Barabasi-Albert graphs. Each data point is calculated using 1000 Monte Carlo runs.

		n		
		1	**2**	**3**
N	**10**	0.6322	0.0003	0
	20	0.8163	0.0235	0
	30	0.7941	0.0408	0

Table 5.7: Comparison of hypotheses $\{H_{04}, H_4\}$ across Barabasi-Albert graphs. Each data point is calculated using 1000 Monte Carlo runs and statistically significant cases ($p < 0.05$) are highlighted in green.

We again provide data only considering graphs with no fully connected nodes in Figure (5.11) and Table 5.8.

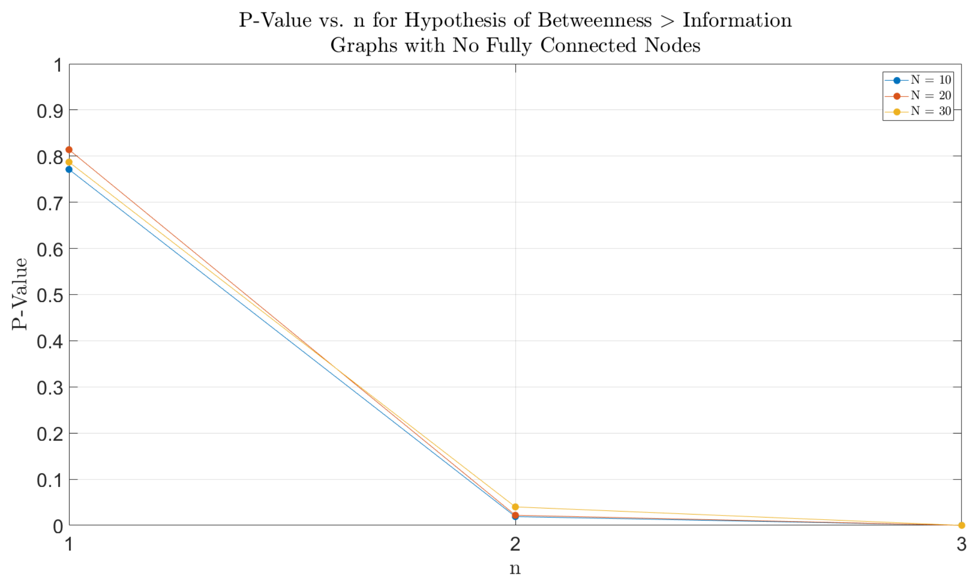


Figure 5.11: Comparison of hypotheses $\{H_{04}, H_4\}$ across Barabasi-Albert graphs. The statistics are generated only using graphs with no fully connected nodes.

		n		
		1	**2**	**3**
N	**10**	0.7712	0.019	0.0001
	20	0.814	0.0219	0
	30	0.7871	0.04	0

Table 5.8: Comparison of hypotheses $\{H_{04}, H_4\}$ across Barabasi-Albert graphs. The statistics are generated only using graphs with no fully connected nodes and statistically significant cases ($p < 0.05$) are highlighted in green.

For the Barabasi-Albert graphs studied we see for those with $n = 2, 3$ there is strong statistical evidence that betweenness centrality would provide benefit over information centrality when weighting actuators. We again see that there is not strong statistical evidence for betweenness centrality in the cases with $n = 1$ and tree graph structure. Note that these results trend with those seen when comparing to closeness centrality. In fact, information centrality reduces to closeness centrality for tree graphs with $n = 1$, so the similarities in statistics compared to our comparisons with closeness centrality is to be expected here.

5.1.1.6 Summary of Results

Our statistical analysis over select Barabasi-Albert graphs reveals that there is strong statistical evidence in favor of betweenness centrality being an appropriate heuristic for actuator selection, especially as the overall density of the graph increases. We note that for tree graphs structures with $n = 1$ that betweenness centrality does not perform well except when compared to eigenvector centrality; however, this could very well be due to the fact that eigenvector centrality is not a useful measure for our framework when the graphs are acyclic.

5.1.2 Watts-Strogatz Model

Besides not accounting for the formation of hubs, another shortcoming of the Erdos-Renyi model is it produces graphs with low clustering. Since the Erdos-Renyi model employs a constant probability to connect nodes, we can intuitively see that local clustering or cliques of nodes will not arise. The Watts-Strogatz model was developed as a simple way to randomly generate graphs which naturally have high clustering, an attractive property since social networks as well as power grid networks have been shown to have clustering properties[47]. Another property of Watts-Strogatz graphs is that they have short average path lengths similar to Erdos-Renyi graphs, where the average path length is just the average number of hops along the shortest paths between each pair of nodes in a graph. These two properties, a high degree of clustering along with short average path lengths, give rise to what are known as small-world networks.

Small-world networks logically have certain advantages for the design of large mobile robotic teams where both local and global collaboration are required[48]. Specifically, the clustering inherent to small-world networks is beneficial for local interactions, whereas the short average path lengths are conducive to global interactions.

The Watts-Strogatz model generates undirected graphs with N nodes and $\frac{NK}{2}$ edges where K is the mean degree (assumed to be an even integer). First a regular ring lattice is constructed having N nodes each connected to K total neighbors with $\frac{K}{2}$ neighbors on each side. For every node i, each edge connecting node i to its $\frac{K}{2}$ rightmost neighbors is rewired with probability $0 \leq \beta \leq 1$. Rewiring is done by replacing an edge with another edge selected uniformly at random from the remaining possible edges after excluding self-loops and edge duplication. Figures (5.12) - (5.14) show Watts-Strogatz networks with $N = 10$ nodes, mean degree

$K = 4$, and $\beta = 0$, $\beta = 0.2$, and $\beta = 1$, respectively.

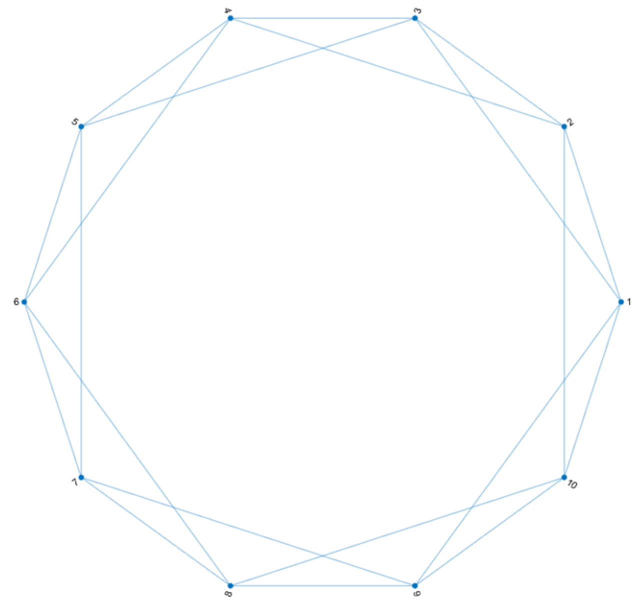


Figure 5.12: Watts-Strogatz graph with $N = 10$ nodes, mean degree $K = 4$, and rewiring probability $\beta = 0$.

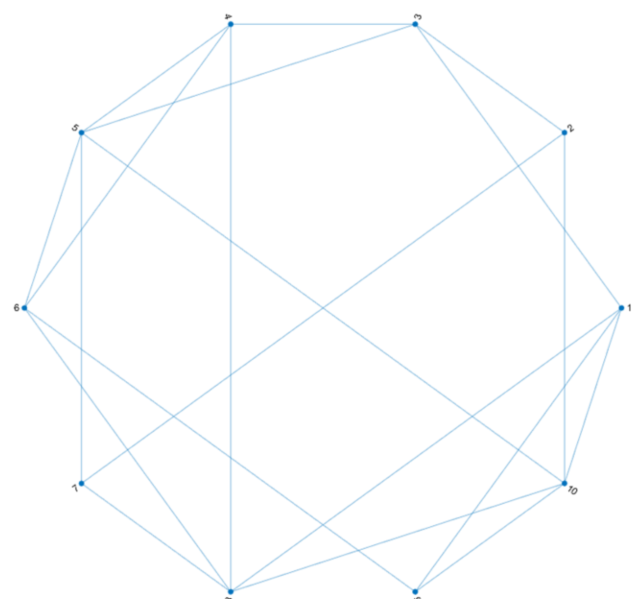


Figure 5.13: Watts-Strogatz graph with $N = 10$ nodes, mean degree $K = 4$, and rewiring probability $\beta = 0.2$.

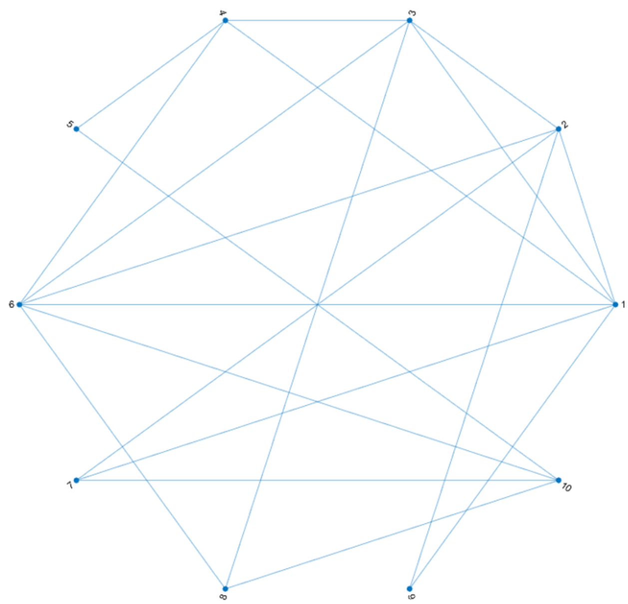


Figure 5.14: Watts-Strogatz graph with $N = 10$ nodes, mean degree $K = 4$, and rewiring probability $\beta = 1$.

5.1.2.1 Test Cases

As in Chapter 3, we perform hypothesis testing to compare each pair of hypotheses $\{H_{01}, H_1\}$, $\{H_{02}, H_2\}$, $\{H_{03}, H_3\}$, and $\{H_{04}, H_4\}$ and generate appropriate p-values given by (4.20). Our test suite consists of Watts-Strogatz graphs with $N \in [10, 20, 30]$ and $K \in [4, 6]$ for $\beta = 0.2$, as well as $N \in [10, 20, 30]$ and $\beta \in [0.1, 0.3, 0.5]$ for $K = 4$.

5.1.2.2 Comparing Betweenness Centrality with Closeness Centrality

Comparing the pair of hypotheses $\{H_{01}, H_1\}$ allows us to investigate whether weighting actuators as a function of betweenness centrality provides an improvement over weighting actuators as a function of closeness centrality. A plot of p-values vs. K for $\beta = 0.2$ is shown in Figure (5.15) and a table highlighting statistically significant cases is shown in Table 5.9.

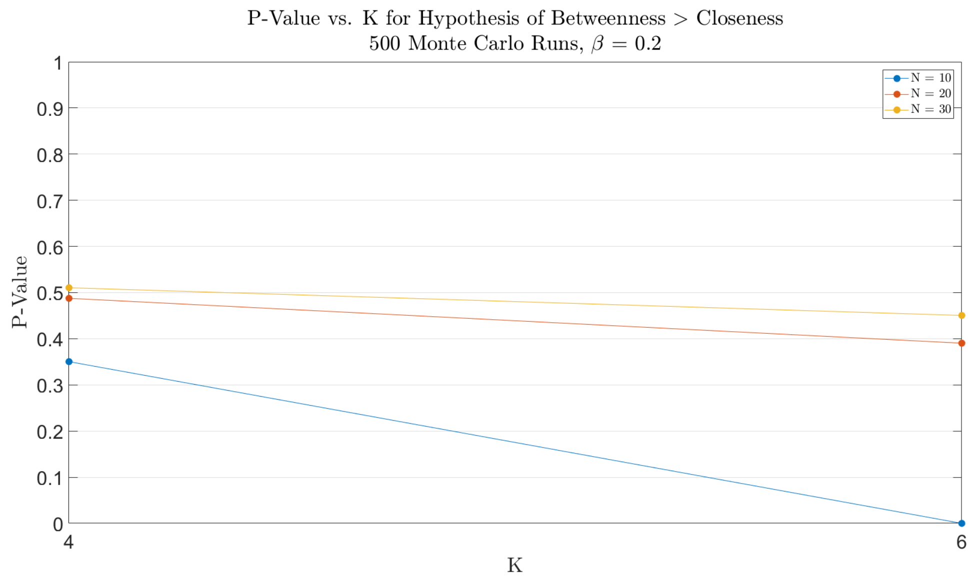


Figure 5.15: Comparison of hypotheses $\{H_{01}, H_1\}$ across Watts-Strogatz graphs. Each data point is calculated using 500 Monte Carlo runs.

		K: 4	K: 6
N	10	0.3504	0
N	20	0.4876	0.3904
N	30	0.5105	0.4505

Table 5.9: Comparison of hypotheses $\{H_{01}, H_1\}$ across Watts-Strogatz graphs with $\beta = 0.2$. Each data point is calculated using 500 Monte Carlo runs and statistically significant cases ($p < 0.05$) are highlighted in green.

We see that there is strong statistical evidence that weighting actuators as a function of betweenness centrality improves over weighting as a function of closeness centrality for the cases with $N = 10$ nodes and mean degree $K = 6$. This corresponds to the densest selection of graphs in the test suite. We will now compare the hypotheses as β varies. A plot of p-values vs. β for $K = 4$ is shown in Figure (5.16) and a table highlighting statistically significant cases is shown in Table 5.10.

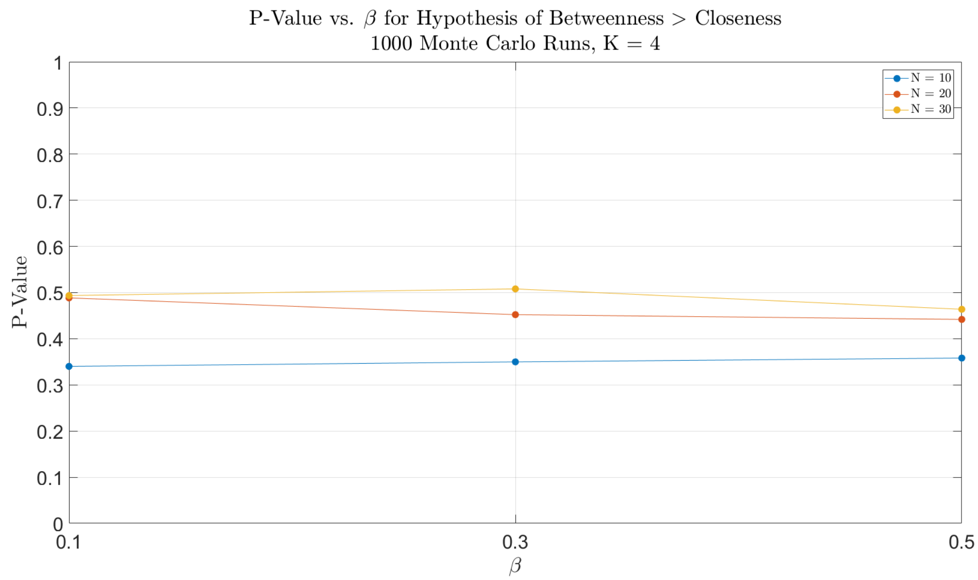


Figure 5.16: Comparison of hypotheses $\{H_{01}, H_1\}$ across Watts-Strogatz graphs. Each data point is calculated using 1000 Monte Carlo runs.

		β		
		0.1	**0.3**	**0.5**
N	**10**	0.3401	0.3499	0.3582
	20	0.4885	0.4522	0.4419
	30	0.4936	0.508	0.4638

Table 5.10: Comparison of hypotheses $\{H_{01}, H_1\}$ across Watts-Strogatz graphs with $K = 4$. Each data point is calculated using 1000 Monte Carlo runs and statistically significant cases ($p < 0.05$) are highlighted in green.

We see in Figure (5.16) that the p-values do not appreciably change as β is varied. Intuitively this makes sense since the overall density of the graph will not be altered by changing β. We have seen that graph density is tied to the performance of betweenness centrality as a heuristic to weight actuators, so this falls in line with our observations here.

5.1.2.3 Comparing Betweenness Centrality with Degree Centrality

Comparing the pair of hypotheses $\{H_{02}, H_2\}$ allows us to investigate whether weighting actuators as a function of betweenness centrality provides an improvement over weighting actuators as a function of degree centrality. A plot of p-values vs. K for $\beta = 0.2$ is shown in Figure (5.17) and a table highlighting statistically significant cases is shown in Table 5.11.

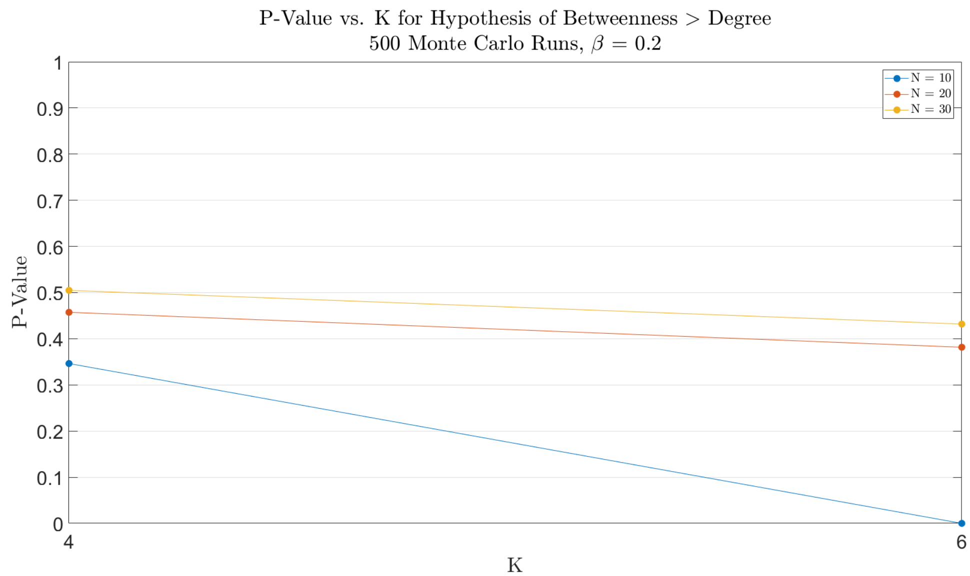


Figure 5.17: Comparison of hypotheses $\{H_{02}, H_2\}$ across Watts-Strogatz graphs. Each data point is calculated using 500 Monte Carlo runs.

		K	
		4	**6**
N	**10**	0.3463	0
	20	0.4573	0.3816
	30	0.5048	0.4318

Table 5.11: Comparison of hypotheses $\{H_{02}, H_2\}$ across Watts-Strogatz graphs with $\beta = 0.2$. Each data point is calculated using 500 Monte Carlo runs and statistically significant cases ($p < 0.05$) are highlighted in green.

We again see that there is strong statistical evidence that weighting actuators as a function of betweenness centrality improves over weighting as a function of degree centrality for the cases with $N = 10$ nodes and mean degree $K = 6$. This corresponds to the densest selection of graphs in the test suite. We will now compare the hypotheses as β varies. A plot of p-values vs. β for $K = 4$ is shown in Figure (5.18) and a table highlighting statistically significant cases is shown in Table 5.12.

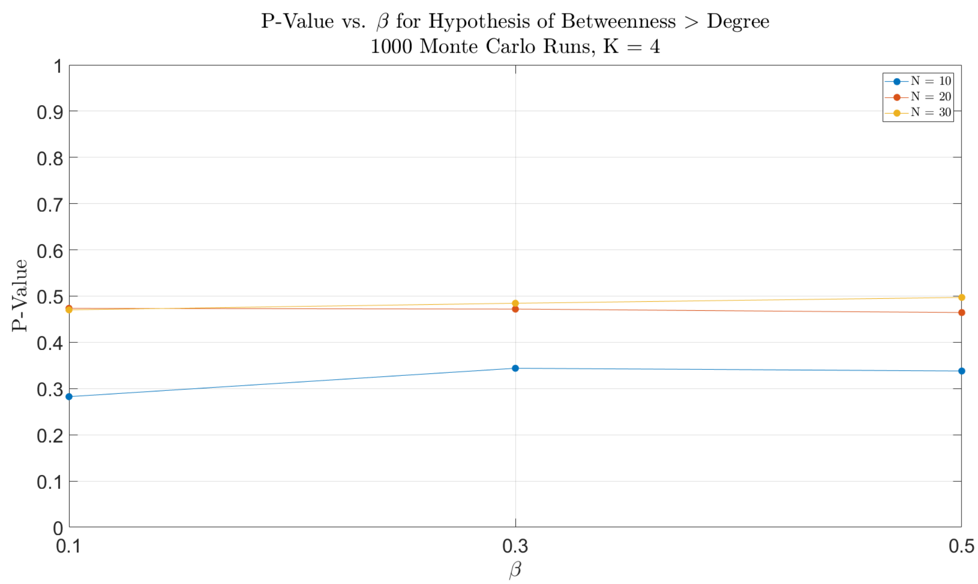


Figure 5.18: Comparison of hypotheses $\{H_{02}, H_2\}$ across Watts-Strogatz graphs. Each data point is calculated using 1000 Monte Carlo runs.

		β		
		0.1	**0.3**	**0.5**
N	**10**	0.2825	0.3439	0.338
	20	0.4735	0.4719	0.4644
	30	0.4702	0.4845	0.4971

Table 5.12: Comparison of hypotheses $\{H_{02}, H_2\}$ across Watts-Strogatz graphs with $K = 4$. Each data point is calculated using 1000 Monte Carlo runs and statistically significant cases ($p < 0.05$) are highlighted in green.

We again see in Figure (5.18) that the p-values do not appreciably change as β is varied.

5.1.2.4 Comparing Betweenness Centrality with Eigenvector Centrality

Comparing the pair of hypotheses $\{H_{03}, H_3\}$ allows us to investigate whether weighting actuators as a function of betweenness centrality provides an improvement over weighting actuators as a function of eigenvector centrality. A plot of p-values vs. K for $\beta = 0.2$ is shown in Figure (5.19) and a table highlighting statistically significant cases is shown in Table 5.13.

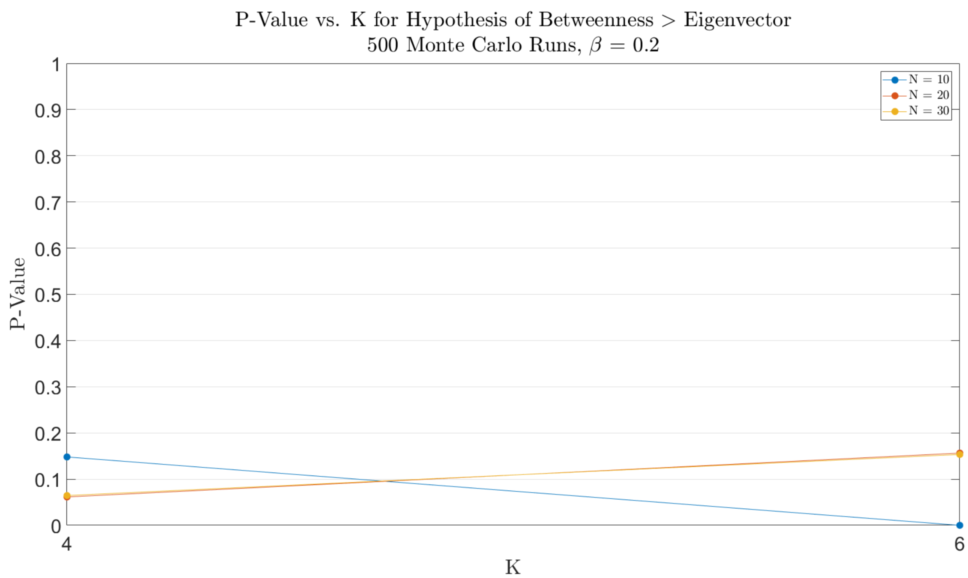


Figure 5.19: Comparison of hypotheses $\{H_{03}, H_3\}$ across Watts-Strogatz graphs. Each data point is calculated using 500 Monte Carlo runs.

		K	
		4	6
N	10	0.148	0
	20	0.0614	0.1564
	30	0.0645	0.1534

Table 5.13: Comparison of hypotheses $\{H_{03}, H_3\}$ across Watts-Strogatz graphs with $\beta = 0.2$. Each data point is calculated using 500 Monte Carlo runs and statistically significant cases ($p < 0.05$) are highlighted in green.

There is strong statistical evidence that weighting actuators as a function of betweenness centrality improves over weighting as a function of eigenvector centrality for the cases with $N = 10$ nodes and mean degree $K = 6$. We do observe that the observed p-values are lower in magnitude than comparable cases seen in previous hypotheses tests involving Watts-Strogatz graphs. We will now compare the hypotheses as β varies. A plot of p-values vs. β for $K = 4$ is shown in Figure (5.20) and a table highlighting statistically significant cases is shown in Table 5.14.

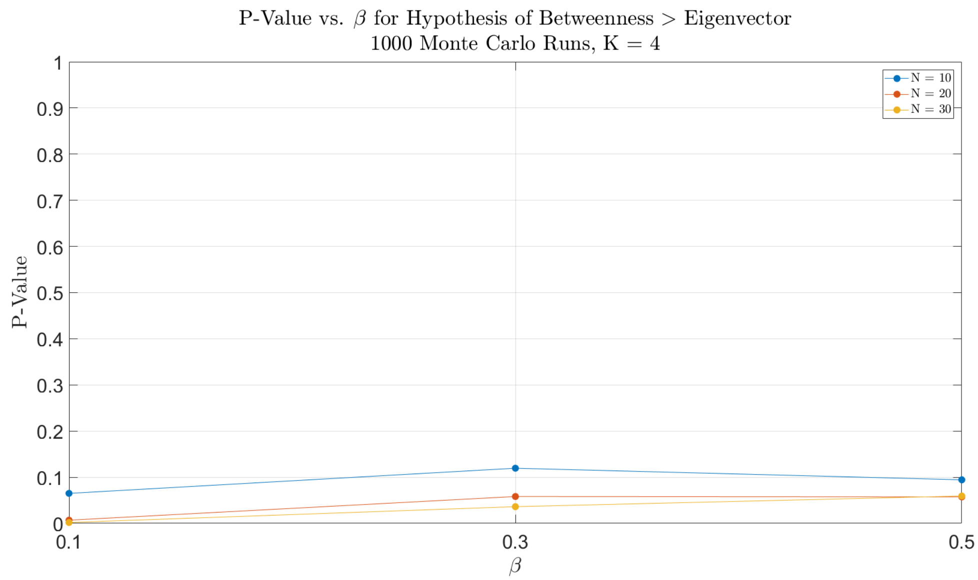


Figure 5.20: Comparison of hypotheses $\{H_{03}, H_3\}$ across Watts-Strogatz graphs. Each data point is calculated using 1000 Monte Carlo runs.

		β		
		0.1	**0.3**	**0.5**
N	**10**	0.065	0.1195	0.0944
	20	0.0069	0.0582	0.0573
	30	0.0019	0.0362	0.0593

Table 5.14: Comparison of hypotheses $\{H_{03}, H_3\}$ across Watts-Strogatz graphs with $K = 4$. Each data point is calculated using 1000 Monte Carlo runs and statistically significant cases ($p < 0.05$) are highlighted in green.

There is evidence to reject the null hypothesis H_{03} in favor of the hypothesis H_3 that weighting actuators as a function of betweenness centrality is beneficial to weighting them as a function of eigenvector centrality for some of the tested cases as seen in Table 5.14.

5.1.2.5 Comparing Betweenness Centrality with Information Centrality

Comparing the pair of hypotheses $\{H_{04}, H_4\}$ allows us to investigate whether weighting actuators as a function of betweenness centrality provides an improvement over weighting actuators as a function of information centrality. A plot of p-values vs. K for $\beta = 0.2$ is shown in Figure (5.21) and a table highlighting statistically significant cases is shown in Table 5.15.

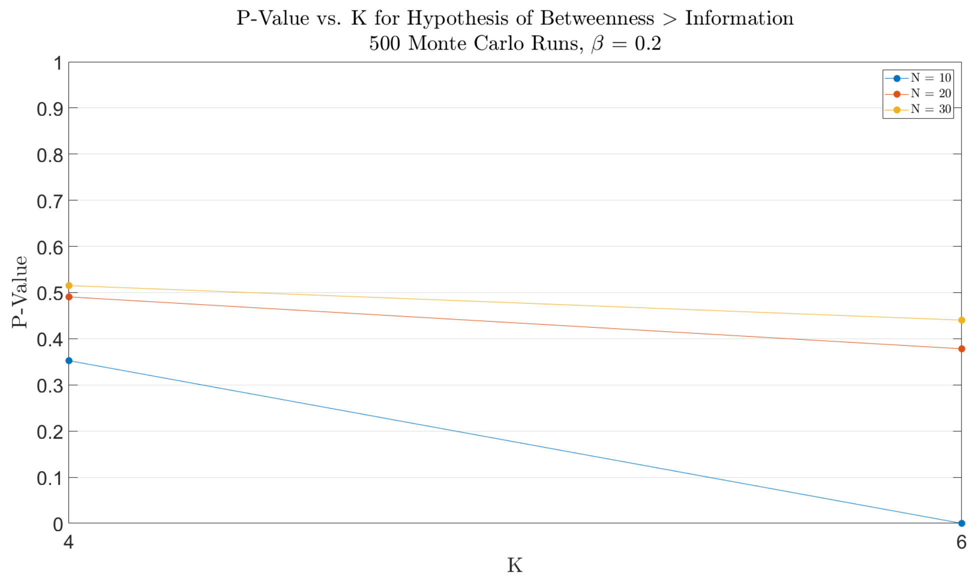


Figure 5.21: Comparison of hypotheses $\{H_{04}, H_4\}$ across Watts-Strogatz graphs. Each data point is calculated using 500 Monte Carlo runs.

		K	
		4	**6**
N	**10**	0.3525	0
	20	0.4906	0.3782
	30	0.5152	0.4403

Table 5.15: Comparison of hypotheses $\{H_{04}, H_4\}$ across Watts-Strogatz graphs with $\beta = 0.2$. Each data point is calculated using 500 Monte Carlo runs and statistically significant cases ($p < 0.05$) are highlighted in green.

As in all the other tested cases for Watts-Strogatz graphs there is strong statistical evidence that weighting actuators as a function of betweenness centrality improves over weighting as a function of information centrality for graphs with $N = 10$ nodes and mean degree $K = 6$. We will now compare the hypotheses as β varies. A plot of p-values vs. β for $K = 4$ is shown in Figure (5.22) and a table highlighting statistically significant cases is shown in Table 5.16.

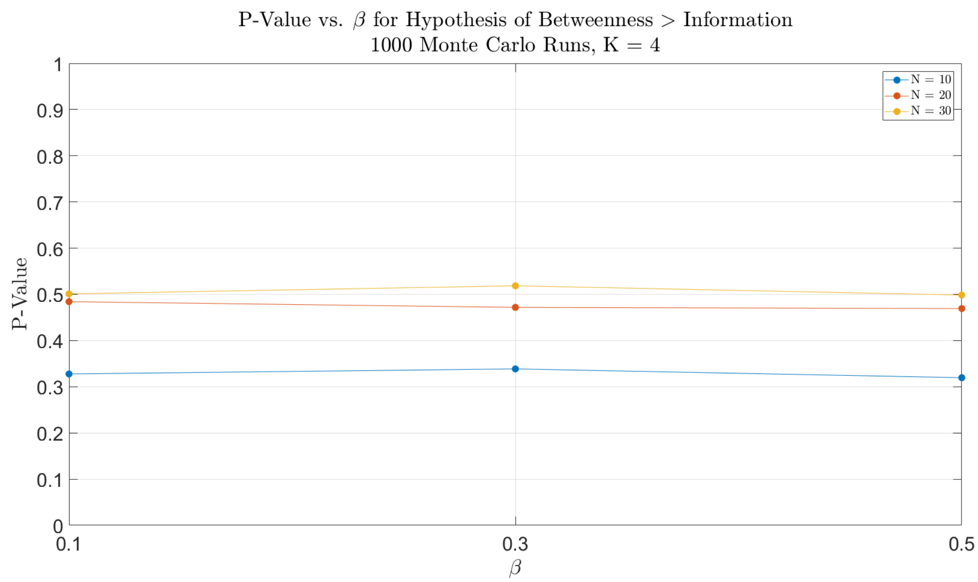


Figure 5.22: Comparison of hypotheses $\{H_{04}, H_4\}$ across Watts-Strogatz graphs. Each data point is calculated using 1000 Monte Carlo runs.

		β		
		0.1	**0.3**	**0.5**
N	**10**	0.3277	0.3386	0.3196
	20	0.4842	0.4718	0.4693
	30	0.5011	0.5185	0.4985

Table 5.16: Comparison of hypotheses $\{H_{04}, H_4\}$ across Watts-Strogatz graphs with $K = 4$. Each data point is calculated using 1000 Monte Carlo runs and statistically significant cases ($p < 0.05$) are highlighted in green.

There are no strong statistical conclusions that can me made from these test cases. We again see that the statistics do not dramatically change for a given (N, K) as β varies.

5.1.2.6 Summary of Results

Our statistical analysis over select Watts-Strogatz graphs shows the importance of graph density on the performance of betweenness centrality as a heuristic. As we vary β for a select value of K we see that the statistics do no appreciably change, which is expected as varying β will not change the overall graph density for a given N. Likewise, for a given value of β as we increase K the overall density of the graph will increase for a given N. Indeed, we see that in general the p-values decrease for a given N as K increases, implying that there is more and more statistical evidence favoring betweenness centrality. The only scenarios in which we do not observe this trend are when comparing betweenness centrality with eigenvector centrality for $N = 20$ and $N = 30$ nodes. The observed increases in p-values are relatively small in magnitude, however. We also note that for the comparisons between betweenness and eigenvector centralities that a relatively modest 500 Monte Carlo runs were performed. It is possible that a larger data set would provide more converged statistics.

5.2 Robustness to Actuator Failure

We now investigate how the conclusions of our hypothesis testing may change under actuator failure. We implement failure in a very simple way. If an actuator i fails we set $\alpha_i = 0$, otherwise we weight the actuator i as a function of a specific centrality measure as before by selecting $\alpha_i = \frac{CM(i)}{\sum_{j \in V} CM(j)}$ where $CM(\cdot)$ is the centrality measure of interest. Note that if an actuator fails it is unable to provide actuation effort for the entirety of the simulation and has no hope of being recovered. A failed actuator is however still able to communicate information to its neighbors. It is important to note that in the case of actuator failure we no longer satisfy the condition $\sum_{i=1}^{N} \alpha_i = 1$. We will have $0.9 < A + \sum_{i=1}^{N} B_i K_i < 1$, which is still Schur stable.

5.2.1 One Actuator Fails

We first consider the case where one actuator fails. Within each simulation only a single actuator loses its actuation capability, where the failed actuator is chosen at random within each network.

5.2.1.1 Test Cases

We again perform hypothesis testing to compare each pair of hypotheses $\{H_{01}, H_1\}$, $\{H_{02}, H_2\}$, $\{H_{03}, H_3\}$, and $\{H_{04}, H_4\}$ and generate appropriate p-values given by (4.20). Our test suite consists of Erdos-Renyi graphs with $N \in [6, 7, 8, 9, 10, 11, 12, 13]$ and $\rho \in [0.3, 0.4, 0.5, 0.6, 0.7]$.

5.2.1.2 Comparing Betweenness Centrality with Closeness Centrality

Comparing the pair of hypotheses $\{H_{01}, H_1\}$ allows us to investigate if weighting actuators as a function of betweenness centrality provides any tangible benefits over weighting actuators as a function of closeness centrality. A plot of appropriate p-values vs. ρ is shown in Figure (5.23) and a table highlighting statistically significant cases is shown in Table 5.17.

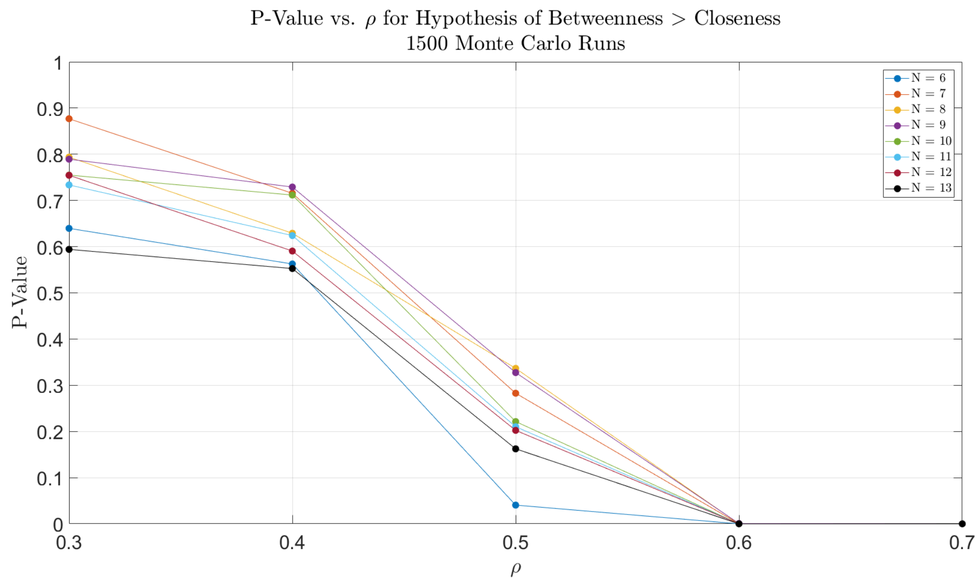


Figure 5.23: Comparison of hypotheses $\{H_{01}, H_1\}$ across Erdos-Renyi graphs with one actuator failing. Each data point is calculated using 1500 Monte Carlo runs.

		ρ				
		0.3	**0.4**	**0.5**	**0.6**	**0.7**
N	**6**	0.6396	0.5625	0.0405	0	0
	7	0.8769	0.7155	0.2827	0.0001	0
	8	0.7938	0.629	0.3365	0	0
	9	0.7893	0.7291	0.3274	0.0007	0
	10	0.7549	0.7117	0.2214	0	0
	11	0.7336	0.624	0.2102	0	0
	12	0.7545	0.5905	0.2023	0	0
	13	0.5941	0.5526	0.1623	0	0

Table 5.17: Comparison of hypotheses $\{H_{01}, H_1\}$ across Erdos-Renyi graphs with one actuator failing. Each data point is calculated using 1500 Monte Carlo runs and statistically significant cases ($p < 0.05$) are highlighted in green.

We see that for $N \in [6, \ldots, 13]$ and $\rho \in [0.6, 0.7]$ there is strong statistical evidence to accept the hypothesis H_1 that weighting actuators as a function of betweenness centrality provides a marked benefit over weighting as a function of closeness centrality. If we compare this to Figure (4.9) and Table 4.1 we see that there is indeed degraded performance compared to when there were no node failures; however, these results suggest that if we are in the regime $0.6 \leq \rho \leq 0.7$ that our statistical conclusions do not change even with a single node failure. We again provide data only considering graphs with no fully connected nodes in Figure (5.24) and Table 5.18.

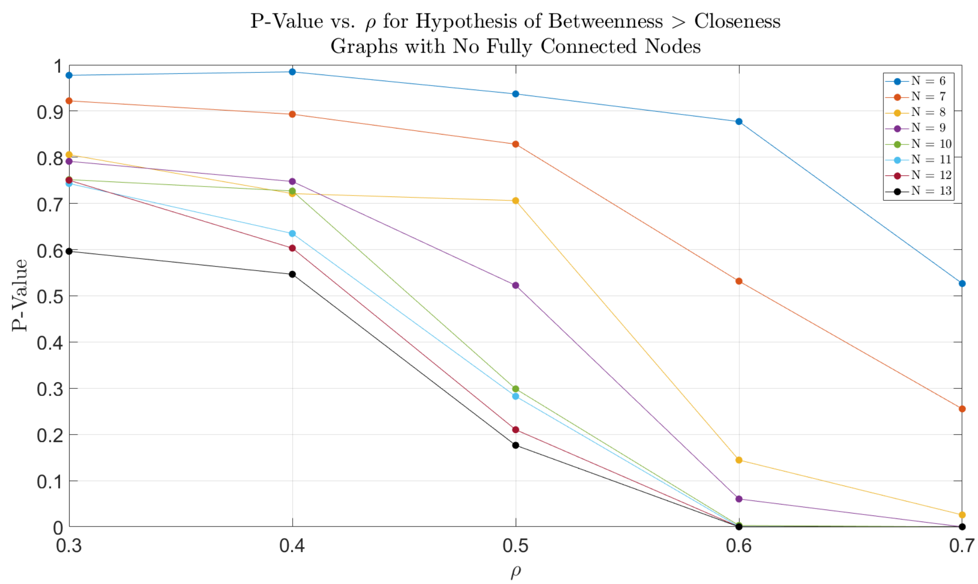


Figure 5.24: Comparison of hypotheses $\{H_{01}, H_1\}$ across Erdos-Renyi graphs with one actuator failing. The statistics are generated only using graphs with no fully connected nodes.

		ρ				
		0.3	**0.4**	**0.5**	**0.6**	**0.7**
	6	0.9774	0.9849	0.9371	0.8773	0.5266
	7	0.9221	0.8931	0.8284	0.5318	0.2553
	8	0.8056	0.7213	0.7061	0.1445	0.0258
N	**9**	0.7912	0.7476	0.5227	0.0603	0
	10	0.7516	0.727	0.2986	0.0029	0
	11	0.7432	0.6349	0.2825	0.0001	0
	12	0.7503	0.6033	0.2102	0	0
	13	0.5964	0.5467	0.1765	0	0

Table 5.18: Comparison of hypotheses $\{H_{01}, H_1\}$ across Erdos-Renyi graphs with one actuator failing. The statistics are generated only using graphs with no fully connected nodes and statistically significant cases ($p < 0.05$) are highlighted in green.

Even when only considering networks with no fully connected nodes we still see cases in which there is statistical evidence that weighting actuators as a function of betweenness centrality provides benefits over weighting as a function of closeness centrality, particularly when $10 \leq N \leq 13$ and $0.6 \leq \rho \leq 0.7$. In other words, there is evidence that betweenness centrality remains a good heuristic for denser graphs and, likewise, for graphs with more overall nodes. If we contrast this with the analogous results with no actuation failures and networks with no fully connected nodes in Figure (4.16) and Table 4.2 we see that indeed when there is a single actuation failure there are fewer tested cases with strong evidence to reject H_{01} in favor of H_1. However, it must be noted that statistically significant cases still remain.

5.2.1.3 Comparing Betweenness Centrality with Degree Centrality

We can similarly perform hypothesis testing for the other pertinent pairs of hypotheses we laid out previously. Comparing the pair of hypotheses $\{H_{02}, H_2\}$ allows us to investigate if weighting actuators as a function of betweenness centrality provides any tangible benefits over weighting actuators as a function of degree centrality. A plot of appropriate p-values vs. ρ is shown in Figure (5.25) and a table highlighting statistically significant cases is shown in Table 5.19.

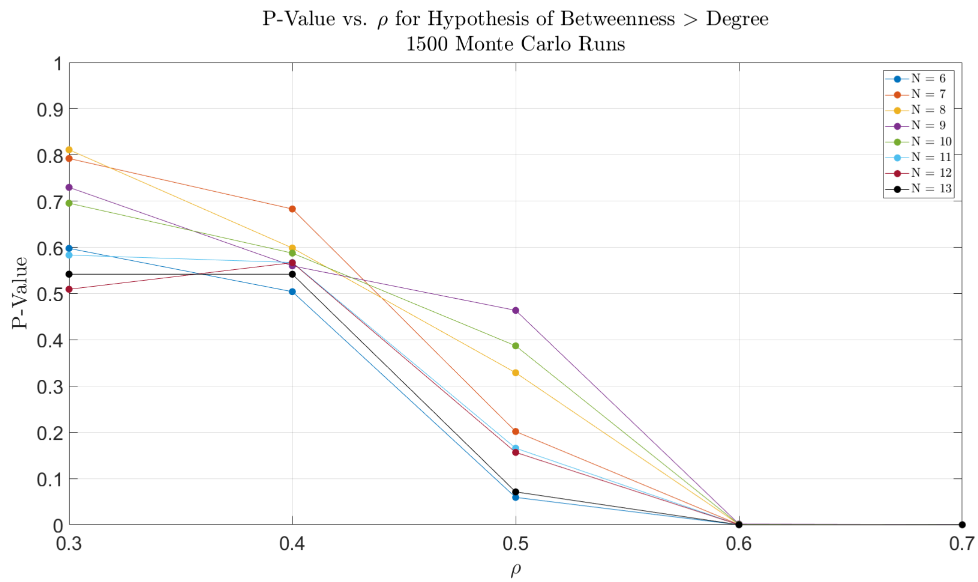


Figure 5.25: Comparison of hypotheses $\{H_{02}, H_2\}$ across Erdos-Renyi graphs with one actuator failing. Each data point is calculated using 1500 Monte Carlo runs.

		ρ				
		0.3	**0.4**	**0.5**	**0.6**	**0.7**
N	**6**	0.5978	0.5041	0.0592	0	0
	7	0.7922	0.683	0.2017	0.0003	0
	8	0.8111	0.5986	0.3288	0.0009	0
	9	0.7298	0.56	0.4637	0.0016	0
	10	0.6955	0.5876	0.387	0.0007	0
	11	0.5833	0.5673	0.1656	0	0
	12	0.5094	0.5668	0.1565	0	0
	13	0.542	0.5419	0.0711	0	0

Table 5.19: Comparison of hypotheses $\{H_{02}, H_2\}$ across Erdos-Renyi graphs with one actuator failing. Each data point is calculated using 1500 Monte Carlo runs and statistically significant cases ($p < 0.05$) are highlighted in green.

We once more see that for simulated cases with $0.6 \leq \rho \leq 0.7$ there is strong statistical evidence to accept the hypothesis H_2 that weighting actuators as a function of betweenness centrality is beneficial compared to weighting them as a function of degree centrality even with a single actuator failure. We again provide data only considering graphs with no fully connected nodes in Figure (5.26) and Table 5.20.

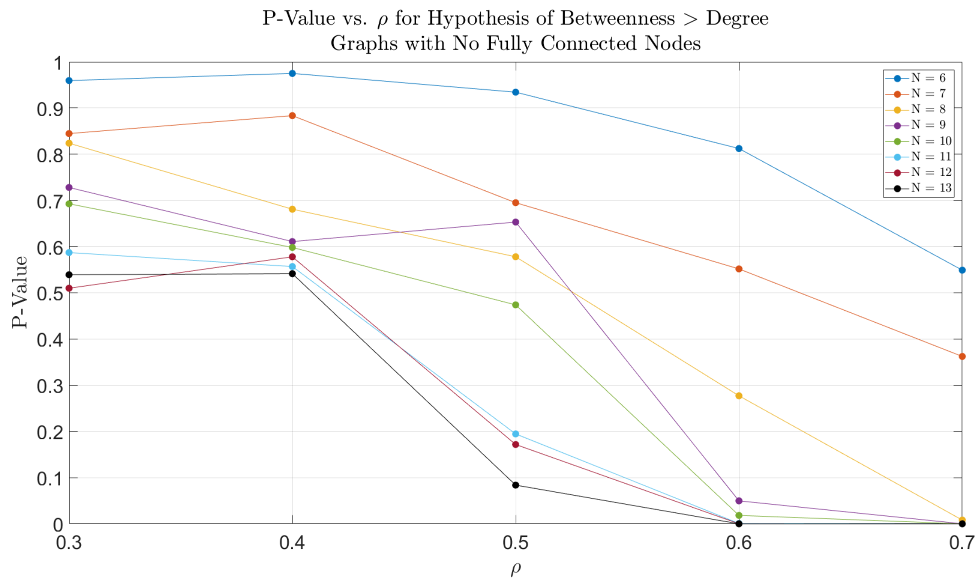


Figure 5.26: Comparison of hypotheses $\{H_{02}, H_2\}$ across Erdos-Renyi graphs with one actuator failing. The statistics are generated only using graphs with no fully connected nodes.

		ρ				
		0.3	**0.4**	**0.5**	**0.6**	**0.7**
N	**6**	0.9595	0.975	0.9344	0.8124	0.5491
	7	0.8448	0.8836	0.6951	0.5518	0.3623
	8	0.824	0.681	0.5781	0.2771	0.0081
	9	0.7281	0.611	0.6532	0.0497	0.0001
	10	0.6927	0.5982	0.474	0.0182	0
	11	0.5872	0.5569	0.1946	0.0008	0
	12	0.51	0.578	0.1717	0.0001	0
	13	0.5392	0.5415	0.0838	0	0

Table 5.20: Comparison of hypotheses $\{H_{02}, H_2\}$ across Erdos-Renyi graphs with one actuator failing. The statistics are generated only using graphs with no fully connected nodes and statistically significant cases ($p < 0.05$) are highlighted in green.

We observe that while certain simulated cases, particularly those with $6 \leq N \leq 8$, no longer yield statistically significant conclusions when compared with the simulations including networks with fully connected nodes, we still see that in general for $9 \leq N \leq 13$ and $0.6 \leq \rho \leq 0.7$ that we still see strong statistical evidence that weighting actuators as a function of betweenness centrality provides benefit over weighting as a function of degree centrality. Again, betweenness centrality shows itself to be a good heuristic for denser graphs and for graphs with more overall nodes.

5.2.1.4 Comparing Betweenness Centrality with Eigenvector Centrality

Comparing the pair of hypotheses $\{H_{03}, H_3\}$ allows us to investigate if weighting actuators as a function of betweenness centrality provides any tangible benefits over weighting actuators as a function of eigenvector centrality. A plot of appropriate p-values vs. ρ is shown in Figure (5.27) and a table highlighting statistically significant cases is shown in Table 5.21.

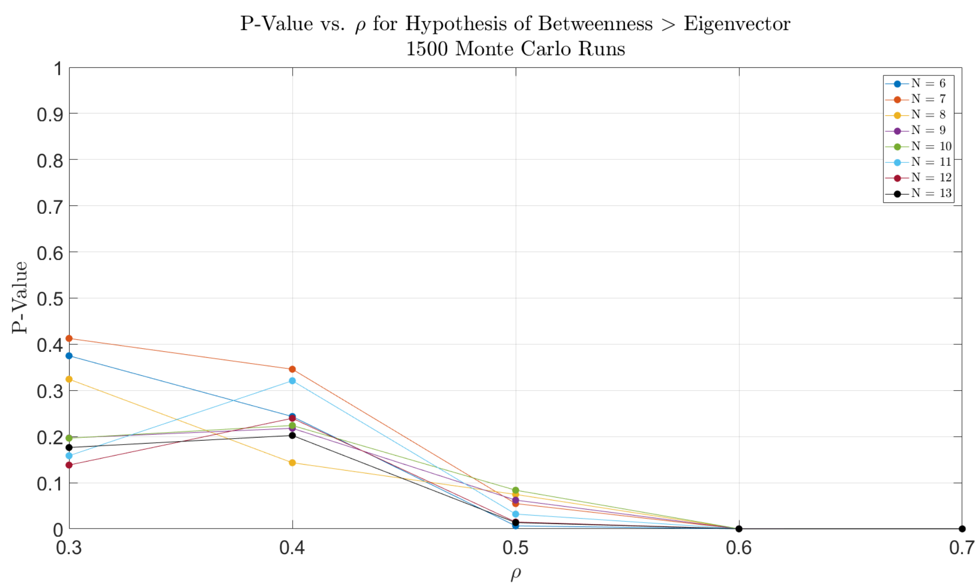


Figure 5.27: Comparison of hypotheses $\{H_{03}, H_3\}$ across Erdos-Renyi graphs with one actuator failing. Each data point is calculated using 1500 Monte Carlo runs.

		ρ				
		0.3	**0.4**	**0.5**	**0.6**	**0.7**
	6	0.375	0.2434	0.0065	0	0
	7	0.4126	0.3459	0.0548	0	0
	8	0.3243	0.1433	0.0746	0	0
N	**9**	0.1969	0.2179	0.0623	0.0001	0
	10	0.1965	0.224	0.0838	0	0
	11	0.1585	0.3209	0.032	0	0
	12	0.1383	0.2396	0.0148	0	0
	13	0.1763	0.2024	0.0136	0	0

Table 5.21: Comparison of hypotheses $\{H_{03}, H_3\}$ across Erdos-Renyi graphs with one actuator failing. Each data point is calculated using 1500 Monte Carlo runs and statistically significant cases ($p < 0.05$) are highlighted in green.

We see that once again for $0.6 \leq \rho \leq 0.7$ there is strong statistical evidence in favor of accepting the hypothesis H_3 that actuation weighting by betweenness centrality is beneficial to actuation weighting by eigenvector centrality. We also note that there are also cases with $\rho = 0.5$, particularly with $N = 6, 11, 12, 13$, which provide strong statistical evidence to accept H_3. This suggests that there is evidence for betweenness centrality outperforming eigenvector centrality over a larger range of graph densities than in the cases when comparing with other centrality measures. This aligns with our previous results over Erdos-Renyi graphs suggesting that eigenvector centrality is outperformed by betweenness centrality over a larger range of simulated cases. We again provide data only considering graphs with no fully connected nodes in Figure (5.28) and Table 5.22.

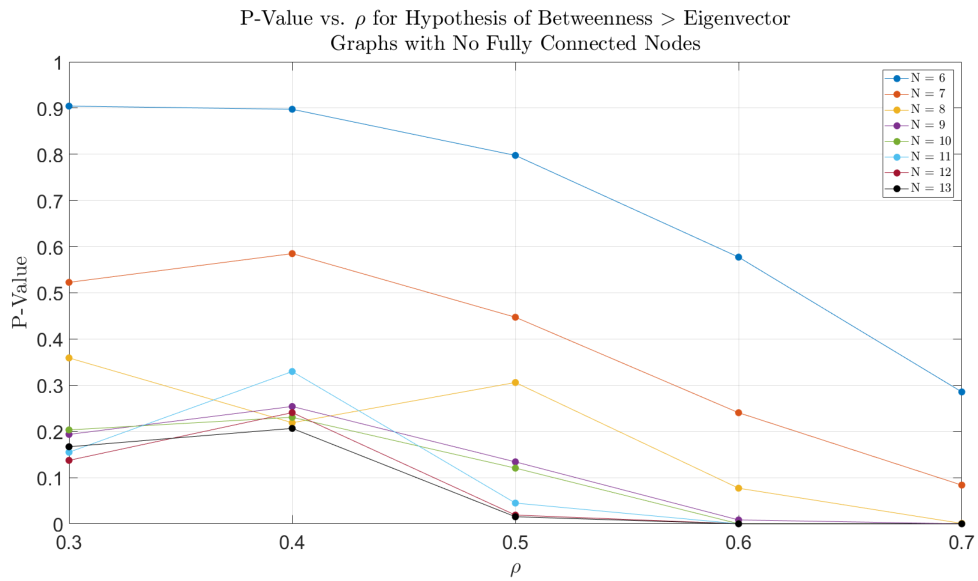


Figure 5.28: Comparison of hypotheses $\{H_{03}, H_3\}$ across Erdos-Renyi graphs with one actuator failing. The statistics are generated only using graphs with no fully connected nodes.

		ρ				
		0.3	**0.4**	**0.5**	**0.6**	**0.7**
N	**6**	0.9042	0.8974	0.7975	0.5773	0.2855
	7	0.5227	0.5849	0.4472	0.2405	0.0838
	8	0.3591	0.2188	0.3059	0.0771	0.0009
	9	0.1936	0.254	0.1341	0.0084	0
	10	0.2033	0.2306	0.1206	0.0002	0
	11	0.155	0.3297	0.0449	0	0
	12	0.1374	0.2407	0.0189	0	0
	13	0.1668	0.2066	0.0153	0	0

Table 5.22: Comparison of hypotheses $\{H_{03}, H_3\}$ across Erdos-Renyi graphs with one actuator failing. The statistics are generated only using graphs with no fully connected nodes and statistically significant cases ($p < 0.05$) are highlighted in green.

Again, even when considering graphs with no fully connected nodes we see cases with strong statistical evidence to accept H_3, particularly for $8 \leq N \leq 13$ and $0.5 \leq \rho \leq 0.7$.

5.2.1.5 Comparing Betweenness Centrality with Information Centrality

Finally we compare the pair of hypotheses $\{H_{04}, H_4\}$, allowing us to investigate whether weighting actuators as a function of betweenness centrality provides any benefit over weighting actuators as a function of information centrality. A plot of appropriate p-values vs. ρ is shown in Figure (5.29) and a table highlighting statistically significant cases is shown in Table 5.23.

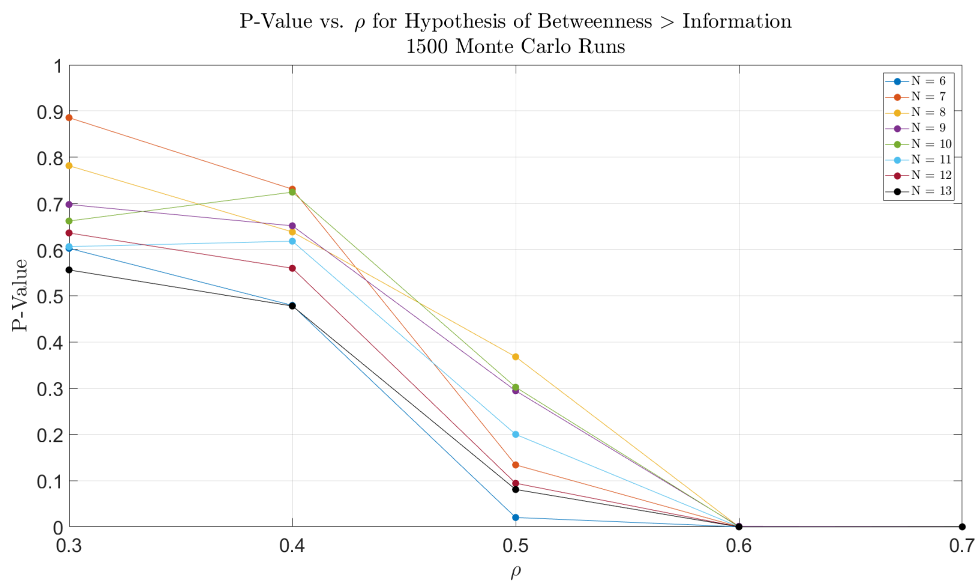


Figure 5.29: Comparison of hypotheses $\{H_{04}, H_4\}$ across Erdos-Renyi graphs with one actuator failing. Each data point is calculated using 1500 Monte Carlo runs.

		ρ				
		0.3	**0.4**	**0.5**	**0.6**	**0.7**
N	**6**	0.6028	0.4793	0.0202	0	0
	7	0.8857	0.7309	0.134	0.0001	0
	8	0.7817	0.6383	0.368	0.0001	0
	9	0.6977	0.6516	0.2945	0.001	0
	10	0.662	0.7245	0.3022	0	0
	11	0.6065	0.6183	0.2001	0	0
	12	0.6361	0.5597	0.0943	0	0
	13	0.5561	0.478	0.0808	0	0

Table 5.23: Comparison of hypotheses $\{H_{04}, H_4\}$ across Erdos-Renyi graphs with one actuator failing. Each data point is calculated using 1500 Monte Carlo runs and statistically significant cases ($p < 0.05$) are highlighted in green.

We again provide data only considering graphs with no fully connected nodes in Figure (5.30) and Table 5.24.

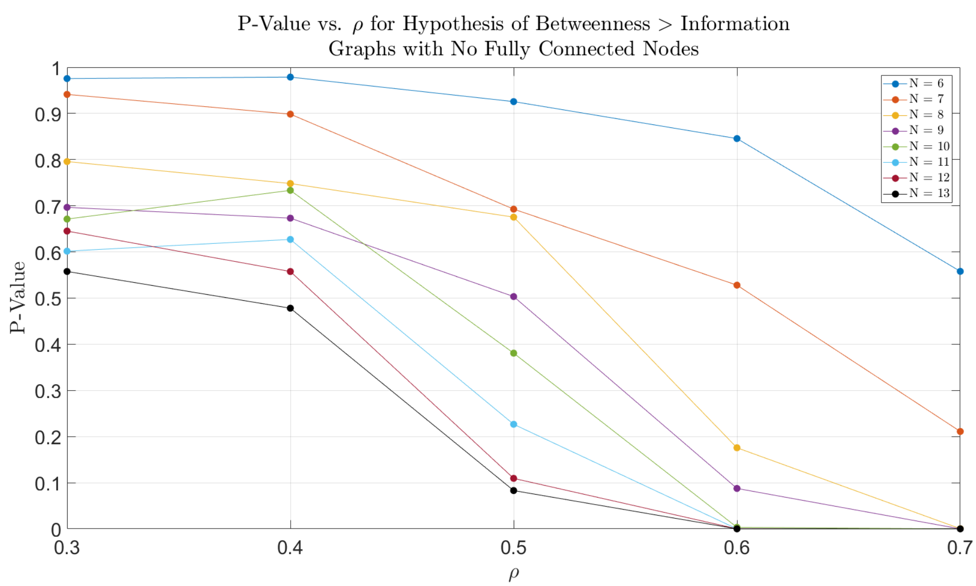


Figure 5.30: Comparison of hypotheses $\{H_{04}, H_4\}$ across Erdos-Renyi graphs with one actuator failing. The statistics are generated only using graphs with no fully connected nodes.

		ρ				
		0.3	**0.4**	**0.5**	**0.6**	**0.7**
N	**6**	0.9756	0.9788	0.9257	0.8456	0.5579
	7	0.9413	0.8985	0.6927	0.528	0.211
	8	0.7958	0.7482	0.6754	0.1757	0.0004
	9	0.6966	0.6732	0.5033	0.0876	0
	10	0.6712	0.7334	0.3807	0.0034	0
	11	0.6017	0.6271	0.2266	0	0
	12	0.6452	0.5576	0.1094	0.0001	0
	13	0.5577	0.4781	0.083	0	0

Table 5.24: Comparison of hypotheses $\{H_{04}, H_4\}$ across Erdos-Renyi graphs with one actuator failing. The statistics are generated only using graphs with no fully connected nodes and statistically significant cases ($p < 0.05$) are highlighted in green.

For simulated cases with $0.6 \leq \rho \leq 0.7$ we see strong statistical evidence to accept the hypothesis H_4 that weighting actuators as a function of betweenness centrality provides benefit to weighting actuators as a function of information centrality. This is true both when considering the full set of Erdos-Renyi graphs as well as the subset with no fully connected nodes. We do note that for graphs with no fully connected nodes only for $8 \leq N \leq 13$ do we see strong statistical evidence to accept H_4. This again suggest that betweenness centrality performs well for denser graphs as well as for graphs with more overall nodes.

5.2.1.6 Summary of Results

For all simulated cases we again see strong statistical evidence favoring betweenness centrality over the other centrality measures as the overall density of the graphs increase as well as for graphs with more overall nodes. The results trend with our previous results over Erdos-Renyi graphs with no actuator failures. We note that when one actuator fails, the overall graph density required for statistical significance in support of betweenness centrality increases. Intuitively, this suggests that the degradation in performance due to the actuator failure must be offset by more density in the network.

5.2.2 Two Actuators Fail

We now consider the case where two actuators fail. Within each simulation the two failed actuators are chosen at random within each network.

5.2.2.1 Test Cases

We again perform hypothesis testing to compare each pair of hypotheses $\{H_{01}, H_1\}$, $\{H_{02}, H_2\}$, $\{H_{03}, H_3\}$, and $\{H_{04}, H_4\}$ and generate appropriate p-values given by (4.20). Our test suite consists of Erdos-Renyi graphs with $N \in [6, 7, 8, 9, 10, 11, 12, 13]$ and $\rho \in [0.3, 0.4, 0.5, 0.6, 0.7]$. Combined conclusions can be found after the presentation of the data.

5.2.2.2 Comparing Betweenness Centrality with Closeness Centrality

Comparing the pair of hypotheses $\{H_{01}, H_1\}$ allows us to investigate if weighting actuators as a function of betweenness centrality provides any tangible benefits over weighting actuators as a function of closeness centrality. A plot of appropriate p-values vs. ρ is shown in Figure (5.31) and a table highlighting statistically significant cases is shown in Table 5.25.

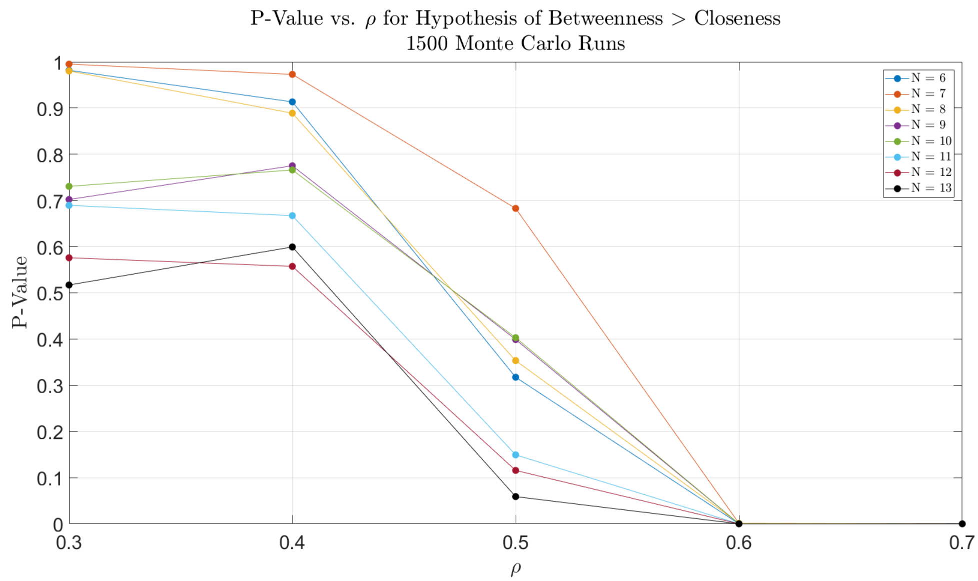


Figure 5.31: Comparison of hypotheses $\{H_{01}, H_1\}$ across Erdos-Renyi graphs with two actuators failing. Each data point is calculated using 1500 Monte Carlo runs.

N \ ρ	0.3	0.4	0.5	0.6	0.7
6	0.9817	0.9134	0.3173	0	0
7	0.9951	0.9729	0.6828	0.0012	0
8	0.9803	0.8889	0.3534	0.0011	0
9	0.7021	0.7751	0.3989	0.0002	0
10	0.7305	0.766	0.403	0.0003	0
11	0.6894	0.6671	0.1493	0	0
12	0.5759	0.5573	0.1156	0	0
13	0.517	0.5993	0.0589	0	0

Table 5.25: Comparison of hypotheses $\{H_{01}, H_1\}$ across Erdos-Renyi graphs with two actuators failing. Each data point is calculated using 1500 Monte Carlo runs and statistically significant cases ($p < 0.05$) are highlighted in green.

We again provide data only considering graphs with no fully connected nodes in Figure (5.32) and Table 5.26.

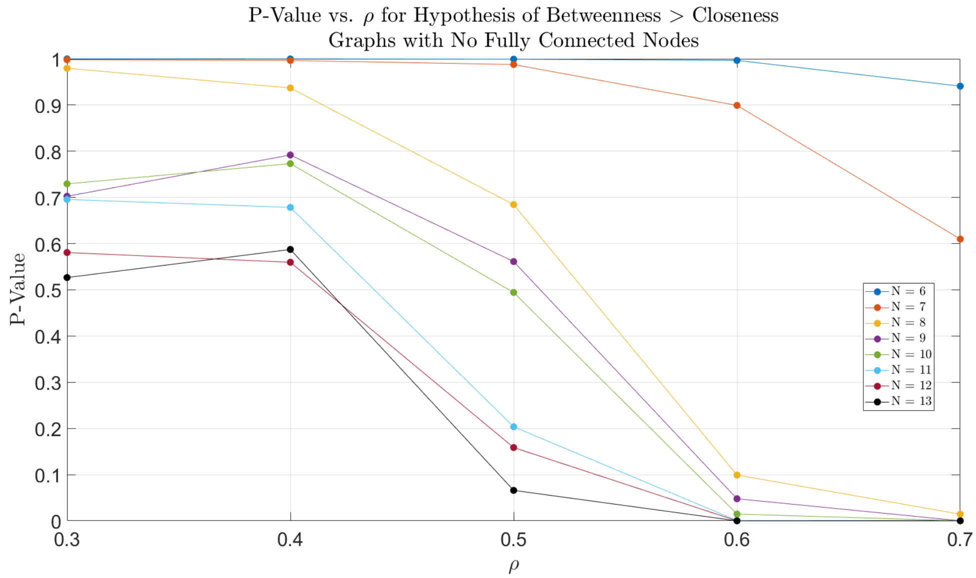


Figure 5.32: Comparison of hypotheses $\{H_{01}, H_1\}$ across Erdos-Renyi graphs with two actuators failing. The statistics are generated only using graphs with no fully connected nodes.

		ρ				
		0.3	**0.4**	**0.5**	**0.6**	**0.7**
N	**6**	1	0.9999	0.9993	0.9967	0.9409
	7	0.9981	0.9965	0.9877	0.8993	0.6099
	8	0.9797	0.937	0.6845	0.0991	0.0143
	9	0.7025	0.7919	0.5612	0.0478	0
	10	0.7294	0.7732	0.4944	0.0146	0
	11	0.6957	0.6784	0.2036	0.0006	0
	12	0.5807	0.5597	0.1586	0	0
	13	0.5266	0.5874	0.0659	0	0

Table 5.26: Comparison of hypotheses $\{H_{01}, H_1\}$ across Erdos-Renyi graphs with two actuators failing. The statistics are generated only using graphs with no fully connected nodes and statistically significant cases ($p < 0.05$) are highlighted in green.

5.2.2.3 Comparing Betweenness Centrality with Degree Centrality

Comparing the pair of hypotheses $\{H_{02}, H_2\}$ allows us to investigate if weighting actuators as a function of betweenness centrality provides any tangible benefits over weighting actuators as a function of degree centrality. A plot of appropriate p-values vs. ρ is shown in Figure (5.33) and a table highlighting statistically significant cases is shown in Table 5.27.

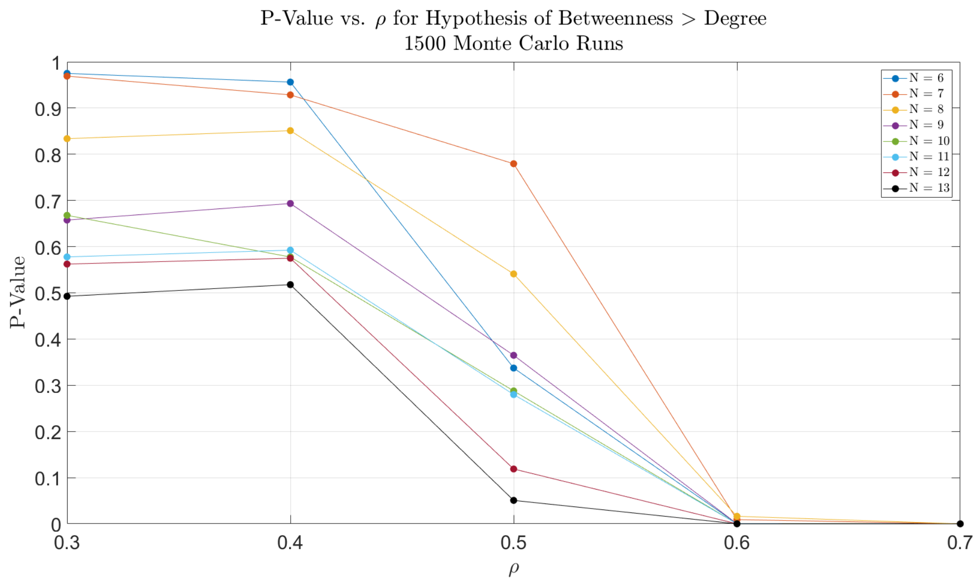


Figure 5.33: Comparison of hypotheses $\{H_{02}, H_2\}$ across Erdos-Renyi graphs with two actuators failing. Each data point is calculated using 1500 Monte Carlo runs.

		ρ				
		0.3	**0.4**	**0.5**	**0.6**	**0.7**
N	**6**	0.975	0.9563	0.3372	0.0003	0
	7	0.9691	0.9285	0.7795	0.0091	0
	8	0.8338	0.8511	0.5409	0.0163	0
	9	0.6575	0.6934	0.3648	0.0002	0
	10	0.6678	0.5776	0.2877	0.0002	0
	11	0.5778	0.5924	0.2798	0.0001	0
	12	0.5623	0.5749	0.1187	0	0
	13	0.4926	0.5177	0.0507	0	0

Table 5.27: Comparison of hypotheses $\{H_{02}, H_2\}$ across Erdos-Renyi graphs with two actuators failing. Each data point is calculated using 1500 Monte Carlo runs and statistically significant cases ($p < 0.05$) are highlighted in green.

We again provide data only considering graphs with no fully connected nodes in Figure (5.34) and Table 5.28.

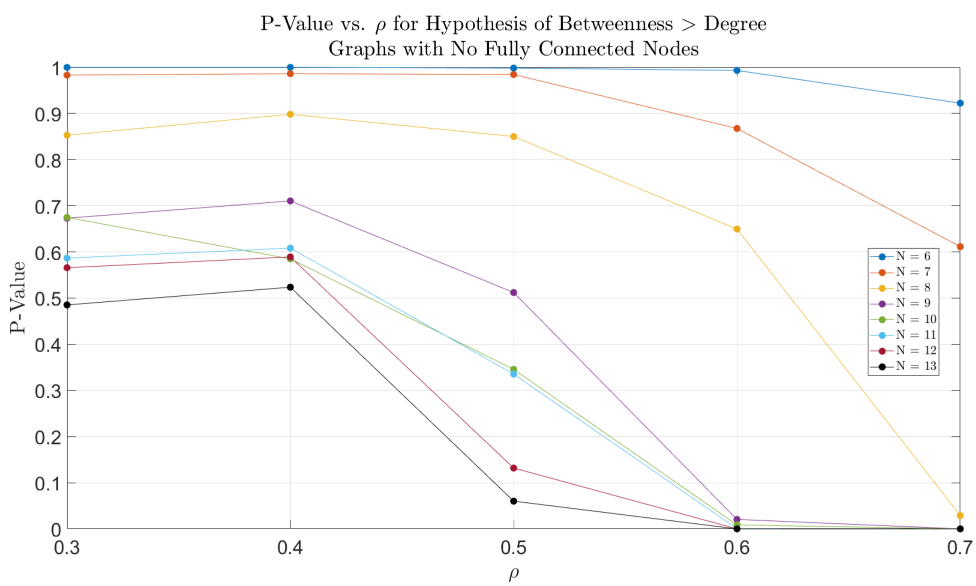


Figure 5.34: Comparison of hypotheses $\{H_{02}, H_2\}$ across Erdos-Renyi graphs with two actuators failing. The statistics are generated only using graphs with no fully connected nodes.

		ρ				
		0.3	**0.4**	**0.5**	**0.6**	**0.7**
N	**6**	0.9999	0.9999	0.9983	0.9933	0.9224
	7	0.9831	0.9862	0.9844	0.8675	0.6114
	8	0.8529	0.8983	0.8501	0.6496	0.0289
	9	0.6731	0.7106	0.5121	0.0205	0
	10	0.6749	0.585	0.3457	0.0086	0
	11	0.5866	0.6084	0.3353	0.0003	0
	12	0.5658	0.5892	0.1318	0	0
	13	0.4852	0.5236	0.0601	0	0

Table 5.28: Comparison of hypotheses $\{H_{02}, H_2\}$ across Erdos-Renyi graphs with two actuators failing. The statistics are generated only using graphs with no fully connected nodes and statistically significant cases ($p < 0.05$) are highlighted in green.

5.2.2.4 Comparing Betweenness Centrality with Eigenvector Centrality

Comparing the pair of hypotheses $\{H_{03}, H_3\}$ allows us to investigate if weighting actuators as a function of betweenness centrality provides any tangible benefits over weighting actuators as a function of eigenvector centrality. A plot of appropriate p-values vs. ρ is shown in Figure (5.35) and a table highlighting statistically significant cases is shown in Table 5.29.

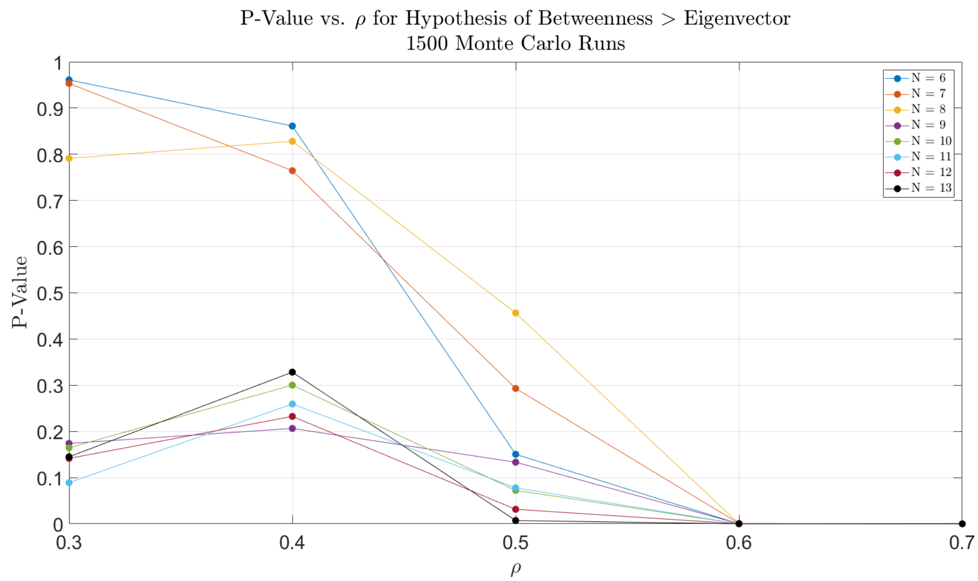


Figure 5.35: Comparison of hypotheses $\{H_{03}, H_3\}$ across Erdos-Renyi graphs with two actuators failing. Each data point is calculated using 1500 Monte Carlo runs.

		ρ				
		0.3	**0.4**	**0.5**	**0.6**	**0.7**
N	**6**	0.961	0.8611	0.1505	0	0
	7	0.9533	0.7644	0.2931	0.0009	0
	8	0.7911	0.8281	0.4564	0.0001	0
	9	0.1741	0.2065	0.1334	0.0003	0
	10	0.1646	0.3003	0.0724	0	0
	11	0.089	0.2594	0.0779	0	0
	12	0.1416	0.2325	0.0314	0	0
	13	0.1451	0.3283	0.0071	0	0

Table 5.29: Comparison of hypotheses $\{H_{03}, H_3\}$ across Erdos-Renyi graphs with two actuators failing. Each data point is calculated using 1500 Monte Carlo runs and statistically significant cases ($p < 0.05$) are highlighted in green.

We again provide data only considering graphs with no fully connected nodes in Figure (5.36) and Table 5.30.

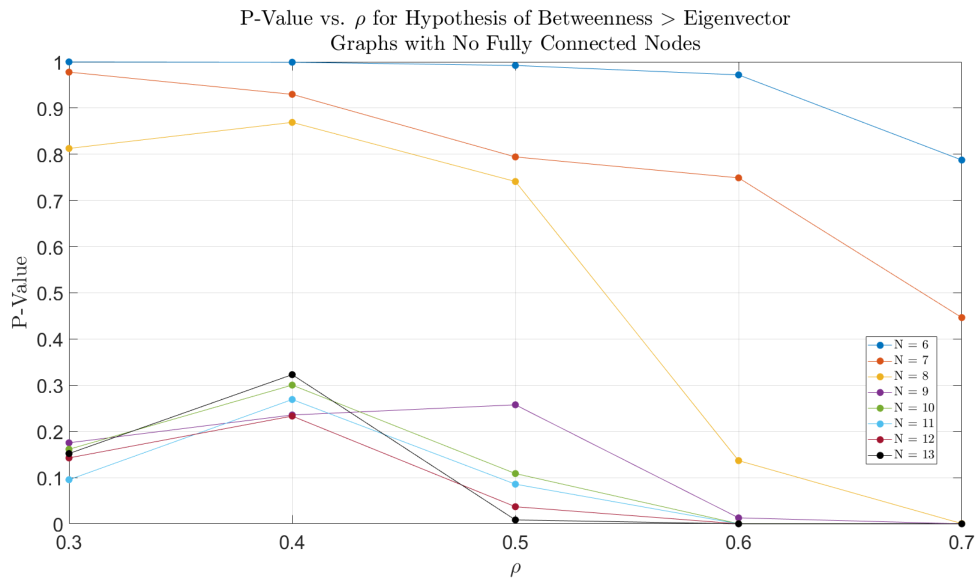


Figure 5.36: Comparison of hypotheses $\{H_{03}, H_3\}$ across Erdos-Renyi graphs with two actuators failing. The statistics are generated only using graphs with no fully connected nodes.

	ρ					
		0.3	**0.4**	**0.5**	**0.6**	**0.7**
N	**6**	0.9999	0.9992	0.9922	0.9718	0.7875
	7	0.9778	0.9297	0.7942	0.7489	0.4464
	8	0.8126	0.869	0.741	0.1368	0.0005
	9	0.1754	0.2356	0.2576	0.0129	0
	10	0.1614	0.3003	0.1088	0.0002	0
	11	0.0955	0.2691	0.0858	0	0
	12	0.1427	0.2332	0.037	0	0
	13	0.1521	0.3229	0.0084	0	0

Table 5.30: Comparison of hypotheses $\{H_{03}, H_3\}$ across Erdos-Renyi graphs with two actuators failing. The statistics are generated only using graphs with no fully connected nodes and statistically significant cases ($p < 0.05$) are highlighted in green.

5.2.2.5 Comparing Betweenness Centrality with Information Centrality

Finally we compare the pair of hypotheses $\{H_{04}, H_4\}$, allowing us to investigate whether weighting actuators as a function of betweenness centrality provides any benefit over weighting actuators as a function of information centrality. A plot of appropriate p-values vs. ρ is shown in Figure (5.37) and a table highlighting statistically significant cases is shown in Table 5.31.

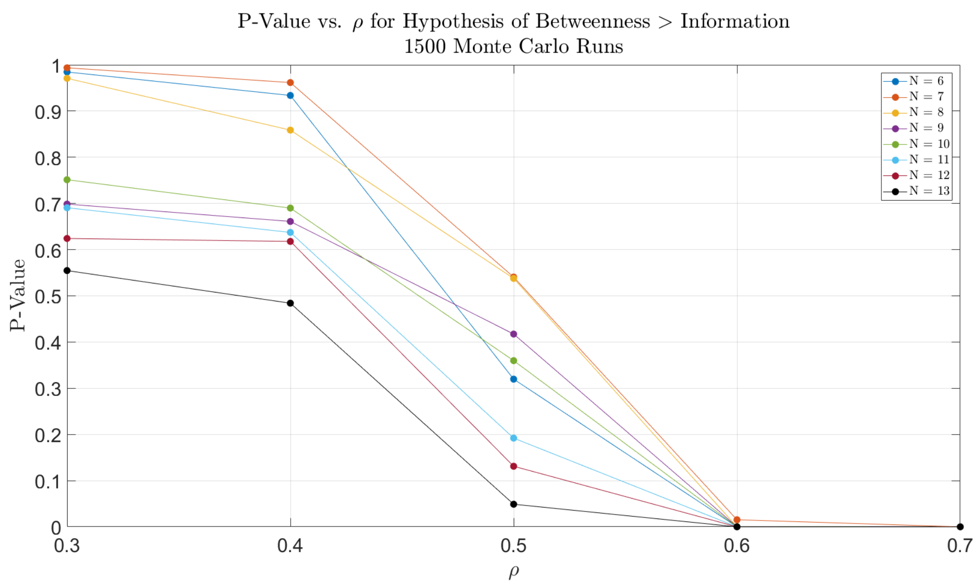


Figure 5.37: Comparison of hypotheses $\{H_{04}, H_4\}$ across Erdos-Renyi graphs with two actuators failing. Each data point is calculated using 1500 Monte Carlo runs.

		ρ				
		0.3	**0.4**	**0.5**	**0.6**	**0.7**
N	**6**	0.9846	0.9338	0.3197	0	0
	7	0.9936	0.9617	0.5406	0.0153	0
	8	0.9709	0.8588	0.5377	0	0
	9	0.6986	0.6611	0.4173	0.0003	0
	10	0.7514	0.69	0.3597	0.0002	0
	11	0.6907	0.6373	0.192	0	0
	12	0.6242	0.6178	0.1311	0	0
	13	0.5547	0.4841	0.0489	0	0

Table 5.31: Comparison of hypotheses $\{H_{04}, H_4\}$ across Erdos-Renyi graphs with two actuators failing. Each data point is calculated using 1500 Monte Carlo runs and statistically significant cases ($p < 0.05$) are highlighted in green.

We again provide data only considering graphs with no fully connected nodes in Figure (5.38) and Table 5.32.

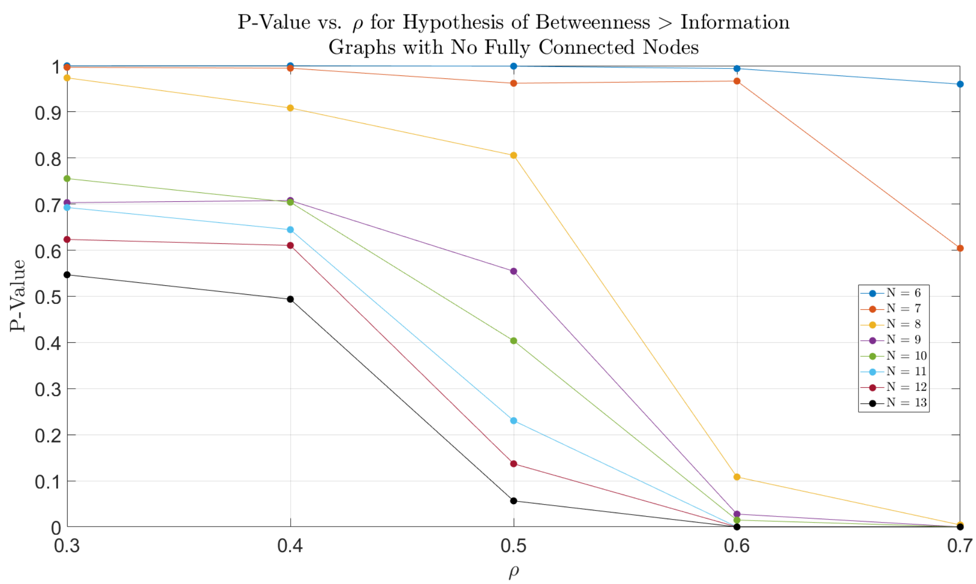


Figure 5.38: Comparison of hypotheses $\{H_{04}, H_4\}$ across Erdos-Renyi graphs with two actuators failing. The statistics are generated only using graphs with no fully connected nodes.

		ρ				
		0.3	**0.4**	**0.5**	**0.6**	**0.7**
	6	0.9999	1	0.9991	0.9939	0.9601
	7	0.9967	0.9947	0.9621	0.9668	0.6043
	8	0.9738	0.9085	0.8058	0.1083	0.0043
N	**9**	0.7032	0.7078	0.5545	0.0278	0
	10	0.7554	0.7041	0.4037	0.0148	0
	11	0.6927	0.6446	0.2304	0	0
	12	0.6234	0.6104	0.1369	0	0
	13	0.5468	0.4938	0.0565	0	0

Table 5.32: Comparison of hypotheses $\{H_{04}, H_4\}$ across Erdos-Renyi graphs with two actuators failing. The statistics are generated only using graphs with no fully connected nodes and statistically significant cases ($p < 0.05$) are highlighted in green.

5.2.2.6 Combined Conclusions

We see that in comparing the sets of hypotheses $\{H_{01}, H_1\}$, $\{H_{02}, H_2\}$, $\{H_{03}, H_3\}$, and $\{H_{04}, H_4\}$ that in all cases there is strong statistical evidence to reject the null hypotheses for $0.6 \leq \rho \leq 0.7$. This means that even with two actuators failing for Erdos-Renyi graphs with $6 \leq N \leq 13$ and $0.6 \leq \rho \leq 0.7$ there is evidence that weighting actuators as a function of betweenness centrality is beneficial to weighting actuators as functions of either closeness centrality, degree centrality, eigenvector centrality, or information centrality. When considering only graphs with no fully connected nodes the statistically significant cases suggesting to reject the null hypotheses remain in the regime $0.6 \leq \rho \leq 0.7$; however, in these cases we only have statistical significance if $8 \leq N \leq 13$. This again suggests that the degradation in performance due to actuator failure must be offset by increased density within the network or by having more overall nodes present to aid in robustness.

5.3 Robustness to Sparsely Available Control Authority

One important property of all the cases studied up until this point is that every node has full controllability of the system. This is trivially obvious since we have considered a one-dimensional process x with state transition matrix $A = 1$ and matrices $B_i = 1$ for each $i \in [1, \ldots, N]$. However, we now consider the case where each node doesn't individually have full control of the system and it is only at the level of the entire network that we have joint controllability.

5.3.1 Multi-dimensional Process with Joint Controllability

Consider a process $\mathbf{x} \in \mathbb{R}^3$ with state transition matrix $A = \mathcal{I}_3$. For ease of exposition consider a network with $N = 3$ nodes. Each node has three actuation channels, i.e. $\mathbf{u}_i \in \mathbb{R}^3$ for $i \in [1, 2, 3]$, and we let $B_1 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$, $B_2 = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$, and $B_3 = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}$. Note that each node individually is unable to control the process. This is readily seen by constructing the controllability matrices for each matrix pair (A, B_1), (A, B_2), and (A, B_3) and checking their row ranks. Let the controllability matrices for each matrix pair (A, B_i) be denoted $\mathcal{C}_i = \begin{bmatrix} B_i & AB_i & A^2B_i \end{bmatrix}$ for $i \in [1, 2, 3]$. It is well-known that the system is controllable by an individual node i if the row rank of $\mathcal{C}_i$ is equal to the dimension of the process $\mathbf{x}$[49], or 3 in this case. It is trivial to see that $rank(\mathcal{C}_i) = 1 < 3$ for each $i \in [1, 2, 3]$, showing the result that each node does not individually have full control of the process. Intuitively this is obvious from the structure of the B_i matrices since we see that each node i is only able to control a single dimension of the three-dimensional process $\mathbf{x}$. However, from the standpoint of the network of nodes as a whole we can intuit that the process is controllable at the network level. We can show this by constructing the joint controllability matrix $\mathcal{C}_{joint}$ for the matrix pair $\left(A, \begin{bmatrix} B_1 & B_2 & B_3 \end{bmatrix}\right)$. We see that $rank(\mathcal{C}_{joint}) = 3$, which shows that the process is indeed jointly controllable at the network level.

5.3.2 Methodology to Investigate Joint Controllability

We can expand on these ideas to easily investigate how our previous findings on the efficacy of certain centrality measures for actuator selection extend to scenarios in which individual nodes do not have full controllability of the process of interest but, rather, there is joint controllability at the level of the network. Consider a network with N nodes actuating a process $\mathbf{x} \in \mathbb{R}^n$ with state transition matrix $A = \mathcal{I}_n$. We restrict $n \geq 2$ so that we are in a realm where joint controllability without individual node controllability is possible. Let $\mathbf{e}_i \in \mathbb{R}^n$ be the standard unit vector with a 1 as the i-th element and zeros elsewhere. We assign $B_i \in \mathbb{R}^{n \times n}$ columnwise as

$$B_i(:, j) = \begin{cases} \mathbf{e}_i & \text{for } j = i \\ \mathbf{0}_{n\times 1} & \text{for } j \in [1, \ldots, n] \backslash \{i\} \end{cases} \tag{5.3}$$

for $i \in [1, \ldots, n]$ where $\mathbf{0}_{n\times 1}$ is the n-dimensional column vector of zeros. We further assign $B_{i+(k)(n)} \in \mathbb{R}^{n\times n}$ as

$$B_{i+(k)(n)} = B_i \tag{5.4}$$

$\forall k \in \mathbb{Z}_{>0}$ such that $i + (k)(n) \leq N$. This ensures that every node i is only actuating in one dimension. To ensure that joint controllability is possible we further require $N \geq n$, which plainly means that there are at least as many nodes as dimensions of the process $\mathbf{x}$. We can see that there will be n subsets of nodes where nodes within a subset all actuate the same singular dimension of the process $\mathbf{x}$. Given a node set V, we define these subsets $S_i \subset V$ for $i \in [1, \ldots, n]$ to contain all nodes which actuate the i-th dimension of the process $\mathbf{x}$.

5.3.2.1 Actuator Weights for Multi-dimensional Process

Previously we have only considered actuator weights α_i for each node i since each node was only actuating a scalar process x. However, we now have actuation in n dimensions, necessitating the use of n actuation weights so that we can tune the actuation in each dimension separately. Let's immediately specialize to a process $\mathbf{x} \in \mathbb{R}^3$ with state transition matrix $A = \mathcal{I}_3$ for ease of exposition. Since $n = 3$ we will have 3 subsets of nodes: S_1 which contains all nodes that actuate the first dimension of process $\mathbf{x}$; S_2 which contains all nodes that actuate the second dimension of process $\mathbf{x}$; and S_3 which contains all nodes that actuate the third dimension of process $\mathbf{x}$. From our previous definition (5.3) we see that $B_1 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$, $B_2 = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$, and $B_3 = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}$. For nodes i with $i \in [4, \ldots, N]$ we appropriately assign B_i matrices according to (5.4). We consider actuator gain matrices of the form $K_i = \begin{bmatrix} -0.1\alpha_i & 0 & 0 \\ 0 & -0.1\beta_i & 0 \\ 0 & 0 & -0.1\gamma_i \end{bmatrix}$ for $i \in [1, \ldots, N]$. This form for the gain matrices is general in that it allows for each node to actuate all three dimensions of the process $\mathbf{x}$, but it is important to note that because of the specialized assignment of B_i matrices in (5.3) and (5.4) any individual node only ends up actuating a single dimension of the process. Thus, we can think of the α parameters as actuation weights along the first dimension, the β parameters as actuation weights along the second dimension, and the γ parameters as actuation weights along the third dimension. Recall that a requirement of our algorithm is that we design K_i such that the quantity $A + \sum_{i=1}^{N} B_i K_i$ is Schur stable. We impose the constraints

$$\sum_{i \in S_1} \alpha_i = \sum_{i \in S_2} \beta_i = \sum_{i \in S_3} \gamma_i = 1 \tag{5.5}$$

This restricts all α, β, and γ weights to be convex within the subsets S_1, S_2, and S_3, respectively. We can easily see that the matrix sum becomes

$$\begin{aligned}
A + \sum_{i=1}^{N} B_i K_i &= A + \sum_{i \in S_1} B_i K_i + \sum_{i \in S_2} B_i K_i + \sum_{i \in S_3} B_i K_i \\
&= \mathcal{I}_3 + \sum_{i \in S_1} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} -0.1\alpha_i & 0 & 0 \\ 0 & -0.1\beta_i & 0 \\ 0 & 0 & -0.1\gamma_i \end{bmatrix} \\
&+ \sum_{i \in S_2} \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} -0.1\alpha_i & 0 & 0 \\ 0 & -0.1\beta_i & 0 \\ 0 & 0 & -0.1\gamma_i \end{bmatrix} \\
&+ \sum_{i \in S_3} \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} -0.1\alpha_i & 0 & 0 \\ 0 & -0.1\beta_i & 0 \\ 0 & 0 & -0.1\gamma_i \end{bmatrix} \\
&= \mathcal{I}_3 + \sum_{i \in S_1} \begin{bmatrix} -0.1\alpha_i & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} + \sum_{i \in S_2} \begin{bmatrix} 0 & 0 & 0 \\ 0 & -0.1\beta_i & 0 \\ 0 & 0 & 0 \end{bmatrix} \\
&+ \sum_{i \in S_3} \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & -0.1\gamma_i \end{bmatrix} \\
&= \begin{bmatrix} 0.9 & 0 & 0 \\ 0 & 0.9 & 0 \\ 0 & 0 & 0.9 \end{bmatrix}
\end{aligned} \tag{5.6}$$

which is Schur stable. Weighting an actuator i as a function of a specific centrality measure is accomplished by selecting

$$\alpha_i = \begin{cases} \frac{CM(i)}{\sum_{j \in S_1} CM(j)} & \text{for } i \in S_1 \\ 0 & \text{otherwise} \end{cases} \tag{5.7}$$

$$\beta_i = \begin{cases} \frac{CM(i)}{\sum_{j \in S_2} CM(j)} & \text{for } i \in S_2 \\ 0 & \text{otherwise} \end{cases} \tag{5.8}$$

$$\gamma_i = \begin{cases} \frac{CM(i)}{\sum_{j \in S_3} CM(j)} & \text{for } i \in S_3 \\ 0 & \text{otherwise} \end{cases} \tag{5.9}$$

where $CM(\cdot)$ is the centrality measure of interest. It is important to highlight here that in general there could be cases in which $\sum_{j \in S_i} BC(j) = 0$ for some $i \in [1, \ldots, n]$ where n is the dimension of the process $\mathbf{x}$ and $BC(j)$ is the betweenness centrality of node j. In these cases we set the actuator weights for nodes within the subset S_i of interest to be equal to $\frac{1}{|S_i|}$ where $|S_i|$ is the cardinality of the subset S_i. As in the one-dimensional cases analyzed previously, given a predefined set of $B_i K_i$ and with actuator weights α, β, and γ as defined in (5.7)-(5.9) we can use a semidefinite program to find feasible W_{ij}^C for $i, j \in [1, \ldots, N]$ which satisfy the LMI given in (4.5) while minimizing Σ.

5.3.3 Test Cases

We perform hypothesis testing to compare each pair of hypotheses $\{H_{01}, H_1\}$, $\{H_{02}, H_2\}$, $\{H_{03}, H_3\}$, and $\{H_{04}, H_4\}$ and generate appropriate p-values given by (4.20). Our test suite consists of Erdos-Renyi graphs $G(N, \rho)$ with $N \in [6, 7, 8, 9, 10, 11, 12, 13]$ and $\rho \in [0.3, 0.4, 0.5, 0.6, 0.7]$.

We note that here when we refer to centralities we are computing an individual node's centrality over the whole network, not within its given subset S_i. We do this in an effort to keep our procedure more systematic and robust to changes in node assignment between subsets.

5.3.4 Comparing Betweenness Centrality with Closeness Centrality

Comparing the pair of hypotheses $\{H_{01}, H_1\}$ allows us to investigate if weighting actuators as a function of betweenness centrality provides any tangible benefits over weighting actuators as a function of closeness centrality. A plot of appropriate p-values vs. ρ is shown in Figure (5.39) and a table highlighting statistically significant cases is shown in Table 5.33.

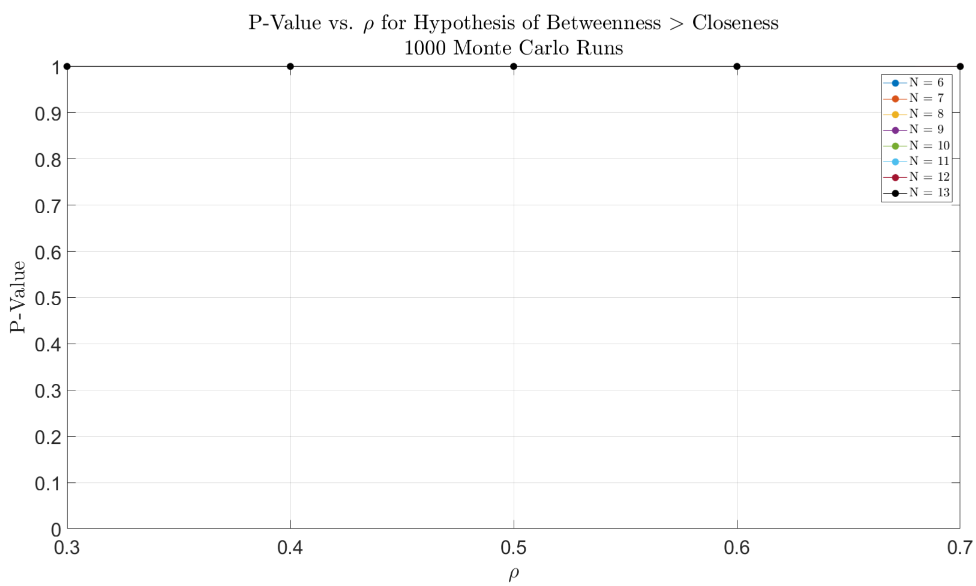


Figure 5.39: Comparison of hypotheses $\{H_{01}, H_1\}$ across Erdos-Renyi graphs where no individual node has full control of the process. Each data point is calculated using 1000 Monte Carlo runs.

		ρ				
		0.3	**0.4**	**0.5**	**0.6**	**0.7**
N	**6**	1	1	1	1	1
	7	1	1	1	1	1
	8	1	1	1	1	1
	9	1	1	1	1	1
	10	1	1	1	1	1
	11	1	1	1	1	1
	12	1	1	1	1	1
	13	1	1	1	1	0.9999

Table 5.33: Comparison of hypotheses $\{H_{01}, H_1\}$ across Erdos-Renyi graphs where no individual node has full control of the process. Each data point is calculated using 1000 Monte Carlo runs and statistically significant cases ($p < 0.05$) are highlighted in green.

We see that there is extremely strong evidence to accept the null hypothesis H_{01} that weighting actuators as a function of betweenness centrality provides no benefit over weighting them as a function of closeness centrality. This is a marked change from our previous results when all nodes individually had full control of the process. One reason for such a change in performance could be that we are still considering a node's centrality over the whole network when weighting actuation. However, in our formulation each subset of nodes is independently actuating a single dimension of the process. Thus, the betweenness centrality of a node in relation to the network as a whole seems to no longer provide as much useful information on its performance due to the partitioning of actuation arising from the problem setup.

5.3.5 Comparing Betweenness Centrality with Degree Centrality

We can similarly perform hypothesis testing for the other pertinent pairs of hypotheses we laid out previously. Comparing the pair of hypotheses $\{H_{02}, H_2\}$ allows us to investigate if weighting actuators as a function of betweenness centrality provides any tangible benefits over weighting actuators as a function of degree centrality. A plot of appropriate p-values vs. ρ is shown in Figure (5.40) and a table highlighting statistically significant cases is shown in Table 5.34.

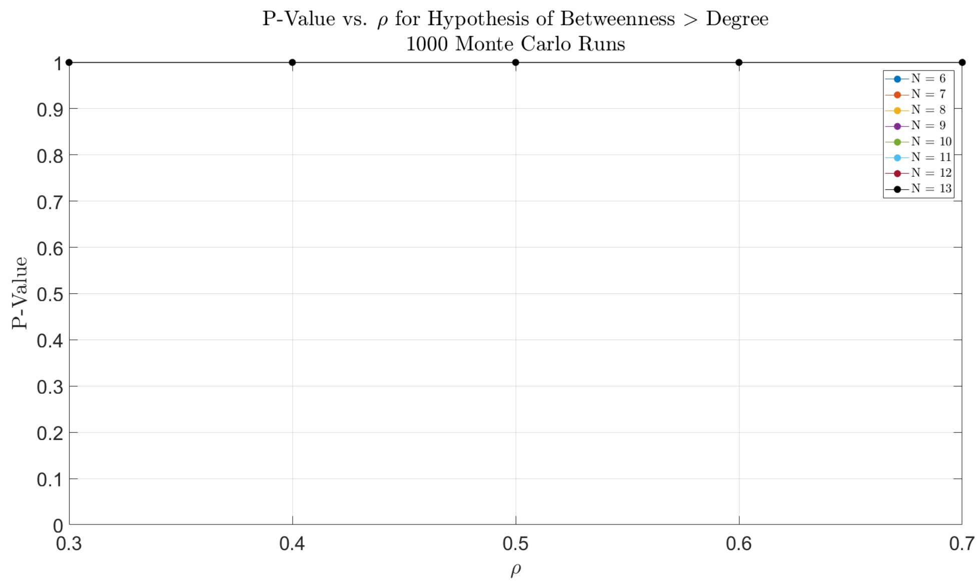


Figure 5.40: Comparison of hypotheses $\{H_{02}, H_2\}$ across Erdos-Renyi graphs where no individual node has full control of the process. Each data point is calculated using 1000 Monte Carlo runs.

		ρ				
		0.3	**0.4**	**0.5**	**0.6**	**0.7**
N	**6**	1	1	1	1	1
	7	1	1	1	1	1
	8	1	1	1	1	1
	9	1	1	1	1	1
	10	1	1	1	1	1
	11	1	1	1	1	1
	12	1	1	1	1	1
	13	1	1	1	1	1

Table 5.34: Comparison of hypotheses $\{H_{02}, H_2\}$ across Erdos-Renyi graphs where no individual node has full control of the process. Each data point is calculated using 1000 Monte Carlo runs and statistically significant cases ($p < 0.05$) are highlighted in green.

Again, we see strong evidence to accept the null hypothesis H_{02} that weighting actuators as a function of betweenness centrality provides no benefit over weighting them as a function of degree centrality.

5.3.6 Comparing Betweenness Centrality with Eigenvector Centrality

Comparing the pair of hypotheses $\{H_{03}, H_3\}$ allows us to investigate if weighting actuators as a function of betweenness centrality provides any tangible benefits over weighting actuators as a function of eigenvector centrality. A plot of appropriate p-values vs. ρ is shown in Figure (5.41) and a table highlighting statistically significant cases is shown in Table 5.35.

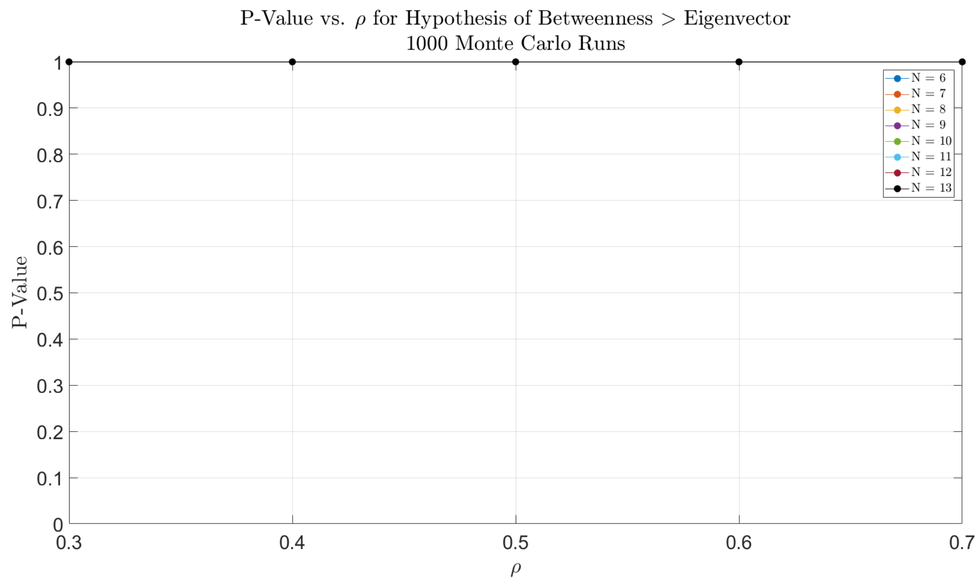


Figure 5.41: Comparison of hypotheses $\{H_{03}, H_3\}$ across Erdos-Renyi graphs where no individual node has full control of the process. Each data point is calculated using 1000 Monte Carlo runs.

		ρ				
		0.3	**0.4**	**0.5**	**0.6**	**0.7**
N	**6**	1	1	1	1	1
	7	1	1	1	1	1
	8	1	1	1	1	1
	9	1	1	1	1	1
	10	1	1	1	1	1
	11	1	1	1	1	1
	12	1	1	1	1	1
	13	1	1	1	1	1

Table 5.35: Comparison of hypotheses $\{H_{03}, H_3\}$ across Erdos-Renyi graphs where no individual node has full control of the process. Each data point is calculated using 1000 Monte Carlo runs and statistically significant cases ($p < 0.05$) are highlighted in green.

We see strong evidence to accept the null hypothesis H_{03} that weighting actuators as a function of betweenness centrality provides no benefit over weighting them as a function of eigenvector centrality.

5.3.7 Comparing Betweenness Centrality with Information Centrality

Finally we compare the pair of hypotheses $\{H_{04}, H_4\}$, allowing us to investigate whether weighting actuators as a function of betweenness centrality provides any benefit over weighting actuators as a function of information centrality. A plot of appropriate p-values vs. ρ is shown in Figure (5.42) and a table highlighting statistically significant cases is shown in Table 5.36.

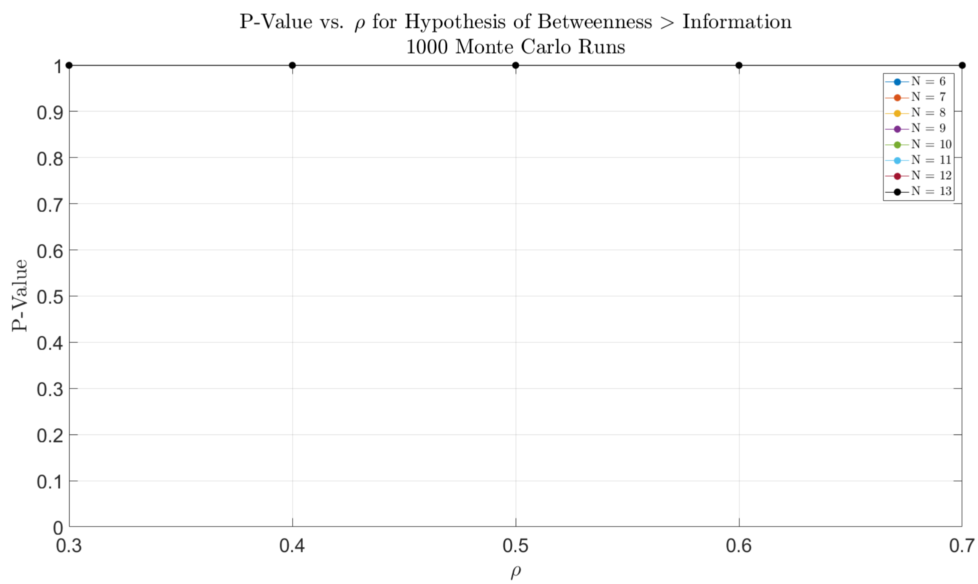


Figure 5.42: Comparison of hypotheses $\{H_{04}, H_4\}$ across Erdos-Renyi graphs where no individual node has full control of the process. Each data point is calculated using 1000 Monte Carlo runs.

		ρ				
		0.3	0.4	0.5	0.6	0.7
N	6	1	1	1	1	1
	7	1	1	1	1	1
	8	1	1	1	1	1
	9	1	1	1	1	1
	10	1	1	1	1	1
	11	1	1	1	1	1
	12	1	1	1	1	1
	13	1	1	1	1	1

Table 5.36: Comparison of hypotheses $\{H_{04}, H_4\}$ across Erdos-Renyi graphs where no individual node has full control of the process. Each data point is calculated using 1000 Monte Carlo runs and statistically significant cases ($p < 0.05$) are highlighted in green.

Finally, we see strong evidence to accept the null hypothesis H_{04} that weighting actuators as a function of betweenness centrality provides no benefit over weighting them as a function of information centrality. These results prompt us to consider additional hypothesis tests to help identify which centrality measure, if any, provides a marked benefit over the others as a weighting heuristic for actuation.

5.3.8 Additional Test Cases

Since there is extremely strong evidence that betweenness centrality is no longer the right centrality measure to leverage for cases in which no node individually has full control over the process, we will shift our attention to degree centrality and compare it against closeness centrality, eigenvector centrality, and information centrality. We formalize this as the following additional hypotheses:

H_{05}: Selecting actuators by weighting as a function of degree centrality provides no benefit over selecting actuators by weighting as a function of closeness centrality.

H_5: Selecting actuators by weighting as a function of degree centrality is better than selecting actuators by weighting as a function of closeness centrality.

H_{06}: Selecting actuators by weighting as a function of degree centrality provides no benefit over selecting actuators by weighting as a function of eigenvector centrality.

H_6: Selecting actuators by weighting as a function of degree centrality is better than selecting actuators by weighting as a function of eigenvector centrality.

H_{07}: Selecting actuators by weighting as a function of degree centrality provides no benefit over selecting actuators by weighting as a function of information centrality.

H_7: Selecting actuators by weighting as a function of degree centrality is better than selecting actuators by weighting as a function of information centrality.

We perform hypothesis testing to compare each pair of hypotheses $\{H_{05}, H_5\}$, $\{H_{06}, H_6\}$, and $\{H_{07}, H_7\}$ and generate appropriate p-values given by (4.20). Our test suite will again consist of Erdos-Renyi graphs $G(N, \rho)$ with $N \in [6, 7, 8, 9, 10, 11, 12, 13]$ and $\rho \in [0.3, 0.4, 0.5, 0.6, 0.7]$.

5.3.9 Comparing Degree Centrality with Closeness Centrality

Comparing the pair of hypotheses $\{H_{05}, H_5\}$ allows us to investigate if weighting actuators as a function of degree centrality provides any tangible benefits over weighting actuators as a function of closeness centrality. A plot of appropriate p-values vs. ρ is shown in Figure (5.43) and a table highlighting statistically significant cases is shown in Table 5.37.

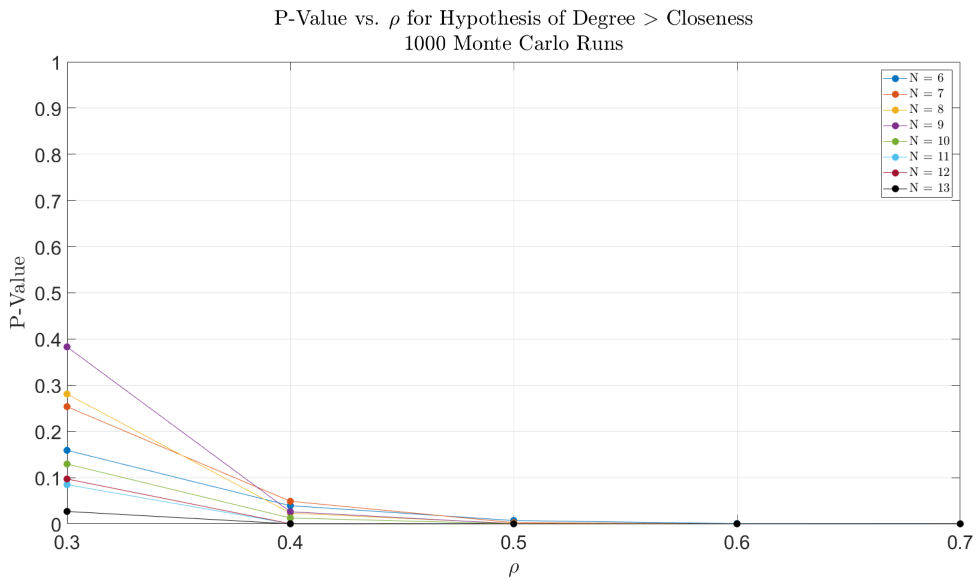


Figure 5.43: Comparison of hypotheses $\{H_{05}, H_5\}$ across Erdos-Renyi graphs where no individual node has full control of the process. Each data point is calculated using 1000 Monte Carlo runs.

		ρ				
		0.3	**0.4**	**0.5**	**0.6**	**0.7**
N	**6**	0.1591	0.0393	0.0071	0.0006	0
	7	0.2537	0.0488	0.0026	0	0
	8	0.281	0.0231	0.0009	0	0
	9	0.383	0.0261	0.0001	0	0
	10	0.1295	0.0126	0	0	0
	11	0.0848	0.0004	0	0	0
	12	0.0971	0.0002	0	0	0
	13	0.0268	0	0	0	0

Table 5.37: Comparison of hypotheses $\{H_{05}, H_5\}$ across Erdos-Renyi graphs where no individual node has full control of the process. Each data point is calculated using 1000 Monte Carlo runs and statistically significant cases ($p < 0.05$) are highlighted in green.

We see strong evidence to reject the null hypothesis H_{05} and accept the alternative hypothesis H_5 that weighting actuators as a function of degree centrality provides a benefit over weighting actuators as a function of closeness centrality. The statistically significant cases are in the regime $0.4 \leq \rho \leq 0.7$ and $6 \leq N \leq 13$, and we also see statistical significance if $\rho = 0.3$ and $N = 13$.

5.3.10 Comparing Degree Centrality with Eigenvector Centrality

Comparing the pair of hypotheses $\{H_{06}, H_6\}$ allows us to investigate if weighting actuators as a function of degree centrality provides any tangible benefits over weighting actuators as a function of eigenvector centrality. A plot of appropriate p-values vs. ρ is shown in Figure (5.44) and a table highlighting statistically significant cases is shown in Table 5.38.

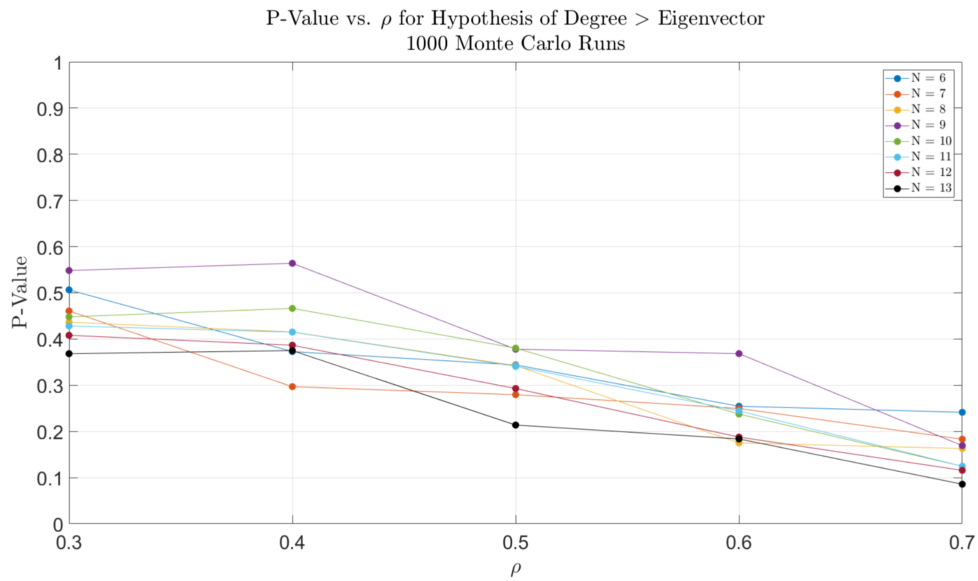


Figure 5.44: Comparison of hypotheses $\{H_{06}, H_6\}$ across Erdos-Renyi graphs where no individual node has full control of the process. Each data point is calculated using 1000 Monte Carlo runs.

		ρ				
		0.3	**0.4**	**0.5**	**0.6**	**0.7**
N	**6**	0.5065	0.3728	0.344	0.2545	0.2414
	7	0.4609	0.2969	0.2795	0.2497	0.1832
	8	0.4364	0.4153	0.3425	0.1751	0.163
	9	0.5483	0.564	0.378	0.3684	0.1693
	10	0.448	0.4663	0.3804	0.2375	0.1246
	11	0.4283	0.415	0.3409	0.2445	0.1246
	12	0.4081	0.3866	0.2929	0.1878	0.1158
	13	0.3683	0.375	0.2138	0.1837	0.0858

Table 5.38: Comparison of hypotheses $\{H_{06}, H_6\}$ across Erdos-Renyi graphs where no individual node has full control of the process. Each data point is calculated using 1000 Monte Carlo runs and statistically significant cases ($p < 0.05$) are highlighted in green.

We see that none of the simulated cases yield strong statistical significance to reject H_{06} in favor of H_6. This implies that there is not strong evidence that weighting actuators as a function of degree centrality provides benefits over weighting actuators as a function of eigenvector centrality. We do, however, see that as N and ρ increase respectively that the magnitude of the calculated p-values decrease toward the threshold of statistical significance. This suggests that if we were to simulate cases with $N \geq 13$ and/or $\rho \geq 0.7$ it is likely that we would begin to see cases emerge with statistical significance in favor of rejecting the null hypothesis H_{06}.

5.3.11 Comparing Degree Centrality with Information Centrality

Comparing the pair of hypotheses $\{H_{07}, H_7\}$ allows us to investigate if weighting actuators as a function of degree centrality provides any tangible benefits over weighting actuators as a function of information centrality. A plot of appropriate p-values vs. ρ is shown in Figure (5.45) and a table highlighting statistically significant cases is shown in Table 5.39.

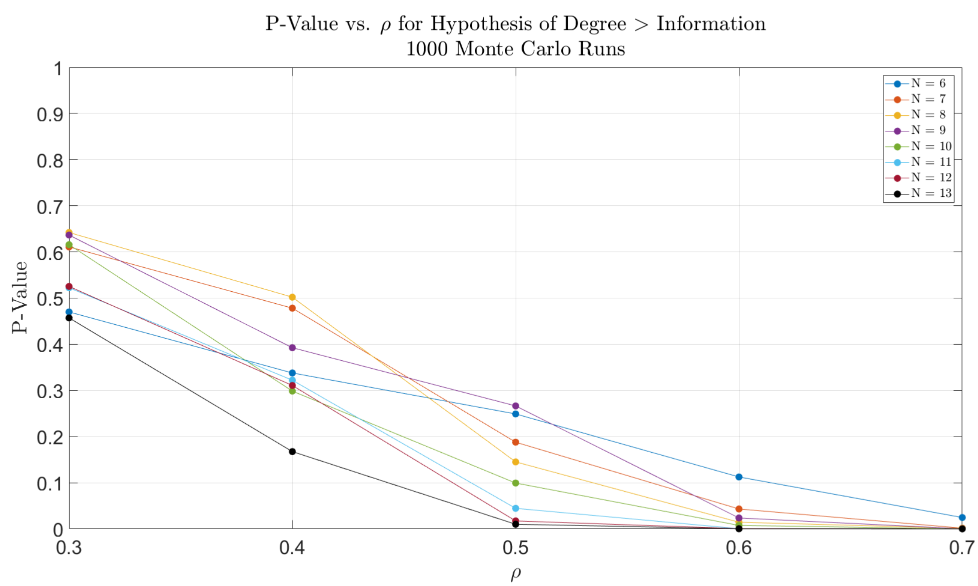


Figure 5.45: Comparison of hypotheses $\{H_{07}, H_7\}$ across Erdos-Renyi graphs where no individual node has full control of the process. Each data point is calculated using 1000 Monte Carlo runs.

		ρ				
		0.3	**0.4**	**0.5**	**0.6**	**0.7**
N	**6**	0.47	0.3378	0.249	0.1125	0.0246
	7	0.6109	0.4782	0.1878	0.0431	0.0015
	8	0.6421	0.502	0.1451	0.0144	0.0012
	9	0.6366	0.3925	0.2664	0.0236	0.0002
	10	0.6157	0.2985	0.0993	0.0075	0
	11	0.5224	0.3221	0.0445	0.0004	0
	12	0.5253	0.3107	0.0172	0.0005	0
	13	0.4571	0.1674	0.01	0.0002	0

Table 5.39: Comparison of hypotheses $\{H_{07}, H_7\}$ across Erdos-Renyi graphs where no individual node has full control of the process. Each data point is calculated using 1000 Monte Carlo runs and statistically significant cases ($p < 0.05$) are highlighted in green.

We see that for $0.6 \leq \rho \leq 0.7$ the simulated cases show strong statistical evidence that weighting actuators as a function of degree centrality is beneficial to weighting actuators as a function of information centrality. In addition, we also see statistically significant evidence for accepting the hypothesis H_7 for $\rho = 0.5$ and $11 \leq N \leq 13$.

5.3.12 Summary of Results

In cases in which no individual node has full control of the process, we find that betweenness centrality over the whole network is no longer a useful heuristic for actuator selection. In situations in which subsets of nodes actuate given dimensions of a process, there is statistical evidence that degree centrality is the most appropriate heuristic out of the centrality measures considered. This implies that the number of connections a node has to other nodes is an important metric in this case. Intuitively, if nodes within a certain actuation subset S_i are separated within the network, having more connections provides a node with more information pathways through which to get its information out to other nodes within its subset. It is akin to sending out an email with many people carbon copied and hoping that one of those people has a clear line of communication with the desired recipients in your given subset. We conjecture that this is why eigenvector centrality performs well in this case, as we see that there is the least amount of statistical evidence in favor of degree centrality when comparing with eigenvector centrality. This makes sense according to the previous line of reasoning as eigenevctor centrality gives more weight to a node if it is in turn connected with high ranking nodes. However, we must note that for the cases considered, while no statistical conclusion can be drawn when comparing degree and eigenvector centrality, we do see the p-values decreasing with increasing ρ, implying that as the networks become more dense there is growing statistical evidence in favor of degree centrality being a more appropriate heuristic than eigenvector centrality. We note that in all cases when we compute centrality measures we compute them over the whole network, not just within a node's designated subset.

5.4 Robustness to Scale

We will now investigate how our statistical conclusions extend to larger networks. For our purposes we will consider an Erdos-Renyi network with $N = 50$ nodes to be "large" as it is a roughly 4x increase over the largest network previously considered at $N = 13$.

5.4.1 Test Cases

We perform hypothesis testing to compare each pair of hypotheses $\{H_{01}, H_1\}$, $\{H_{02}, H_2\}$, $\{H_{03}, H_3\}$, and $\{H_{04}, H_4\}$ and generate appropriate p-values given by (4.20). Our test suite consists of Erdos-Renyi graphs $G(N, \rho)$ with $N = 50$ and $\rho \in [0.3, 0.4, 0.5]$. We note that due to excessive computational intensity and simulation times we only simulate 100 Monte Carlo runs for all of our test cases.

5.4.2 Comparing Betweenness Centrality with Closeness Centrality

Comparing the pair of hypotheses $\{H_{01}, H_1\}$ allows us to investigate if weighting actuators as a function of betweenness centrality provides any tangible benefits over weighting actuators as a function of closeness centrality. A plot of appropriate p-values vs. ρ is shown in Figure (5.46) and a table highlighting statistically significant cases is shown in Table 5.40.

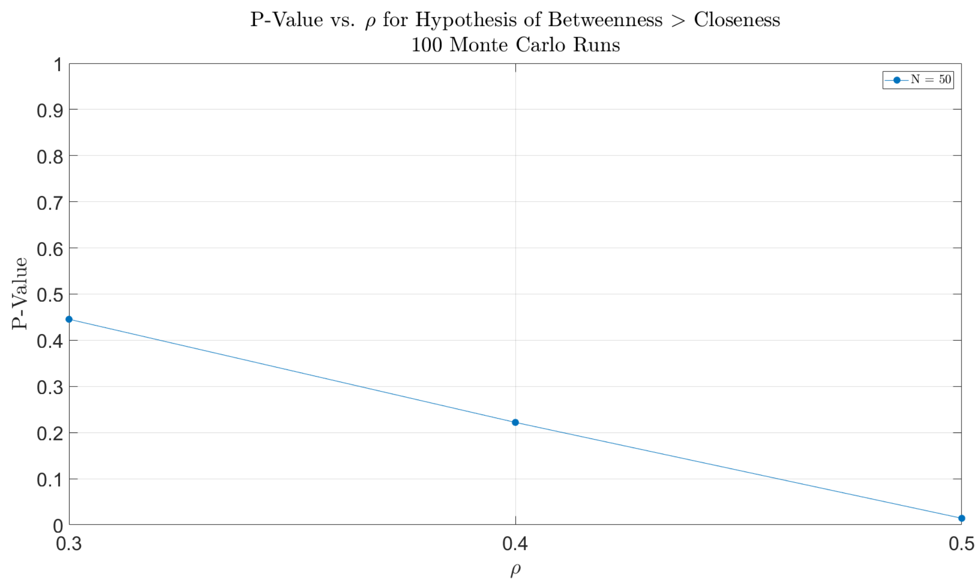


Figure 5.46: Comparison of hypotheses $\{H_{01}, H_1\}$ across Erdos-Renyi graphs with $N = 50$ nodes. Each data point is calculated using 100 Monte Carlo runs.

		ρ		
		0.3	**0.4**	**0.5**
N	**50**	0.4455	0.222	0.0143

Table 5.40: Comparison of hypotheses $\{H_{01}, H_1\}$ across Erdos-Renyi graphs with $N = 50$ nodes. Each data point is calculated using 100 Monte Carlo runs and statistically significant cases ($p < 0.05$) are highlighted in green.

We see that for $N = 50$ nodes and $\rho = 0.5$ that there is strong statistical evidence that weighting actuators as a function of betweenness centrality provides a benefit to weighting actuators as a function of closeness centrality. This suggests that for Erdos-Renyi graphs our results from simulating cases with $6 \leq N \leq 13$ seem to be scale-independent.

5.4.3 Comparing Betweenness Centrality with Degree Centrality

Comparing the pair of hypotheses $\{H_{02}, H_2\}$ allows us to investigate if weighting actuators as a function of betweenness centrality provides any tangible benefits over weighting actuators as a function of degree centrality. A plot of appropriate p-values vs. ρ is shown in Figure (5.47) and a table highlighting statistically significant cases is shown in Table 5.41.

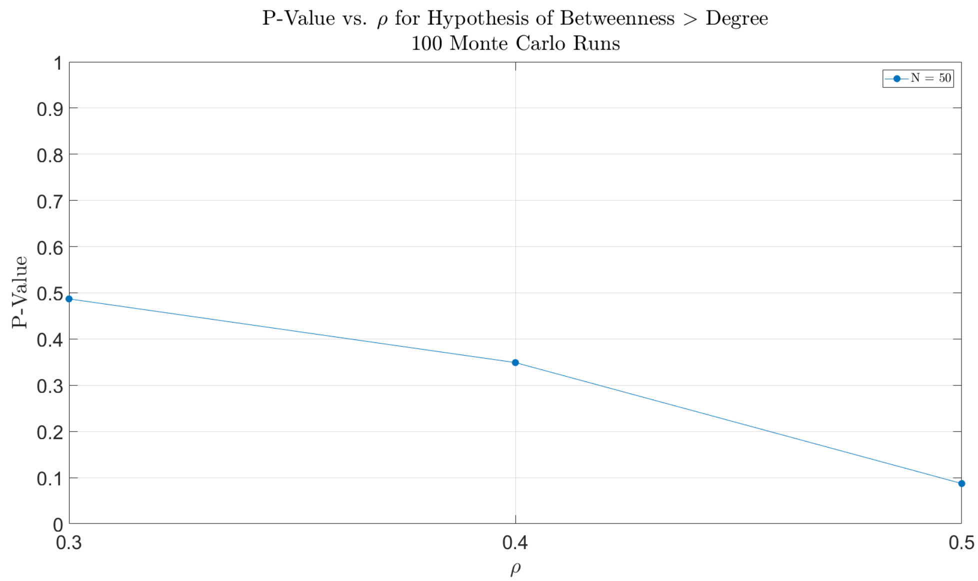


Figure 5.47: Comparison of hypotheses $\{H_{02}, H_2\}$ across Erdos-Renyi graphs with $N = 50$ nodes. Each data point is calculated using 100 Monte Carlo runs.

		ρ		
		0.3	**0.4**	**0.5**
N	**50**	0.4869	0.349	0.0872

Table 5.41: Comparison of hypotheses $\{H_{02}, H_2\}$ across Erdos-Renyi graphs with $N = 50$ nodes. Each data point is calculated using 100 Monte Carlo runs and statistically significant cases ($p < 0.05$) are highlighted in green.

We see that none of the simulated cases show statistically significant evidence to reject the null hypothesis H_{02} in favor of the alternative hypothesis H_2 that weighting actuators as a function of betweenness centrality provides a marked benefit over weighting actuators as a function of degree centrality. However, we note that for $N = 50$ and $\rho = 0.5$ that the calculated p-value is 0.0872. While we set the threshold for statistical significance at a p-value of 0.05, the p-values are observed to be decreasing in magnitude as ρ increases in Figure (5.47). This suggests that if we were to simulate Erdos-Renyi graphs with $N = 50$ and $\rho > 0.5$ we might find cases that yield statistically significant evidence to reject H_{02}.

5.4.4 Comparing Betweenness Centrality with Eigenvector Centrality

Comparing the pair of hypotheses $\{H_{03}, H_3\}$ allows us to investigate if weighting actuators as a function of betweenness centrality provides any tangible benefits over weighting actuators as a function of eigenvector centrality. A plot of appropriate p-values vs. ρ is shown in Figure (5.48) and a table highlighting statistically significant cases is shown in Table 5.42.

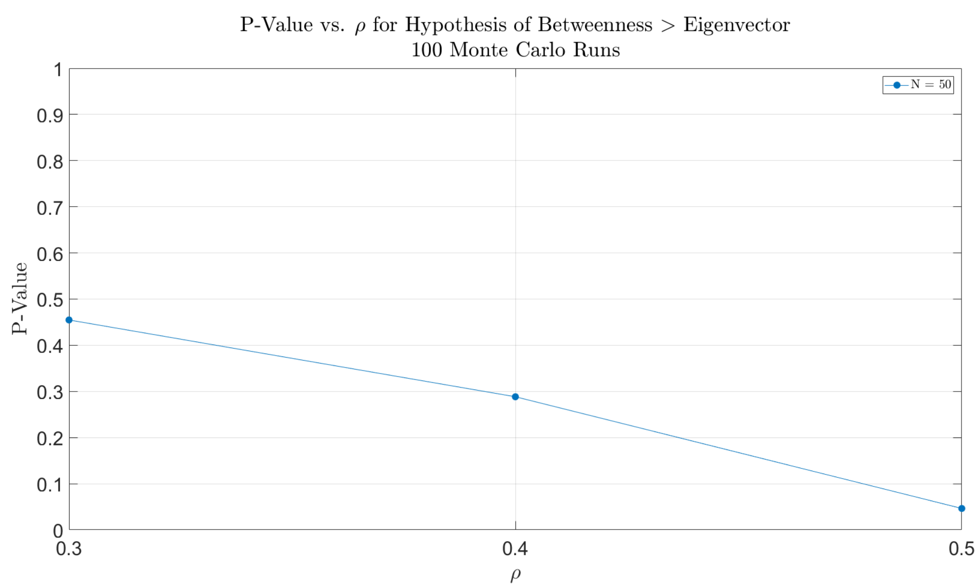


Figure 5.48: Comparison of hypotheses $\{H_{03}, H_3\}$ across Erdos-Renyi graphs with $N = 50$ nodes. Each data point is calculated using 100 Monte Carlo runs.

		ρ		
		0.3	**0.4**	**0.5**
N	**50**	0.4548	0.2884	0.0464

Table 5.42: Comparison of hypotheses $\{H_{03}, H_3\}$ across Erdos-Renyi graphs with $N = 50$ nodes. Each data point is calculated using 100 Monte Carlo runs and statistically significant cases ($p < 0.05$) are highlighted in green.

We see that for Erdos-Renyi graphs with $N = 50$ and $\rho = 0.5$ that there is strong statistical evidence that weighting actuators as a function of betweenness centrality is beneficial to weighting actuators as a function of eigenvector centrality.

5.4.5 Comparing Betweenness Centrality with Information Centrality

Finally, we compare the pair of hypotheses $\{H_{04}, H_4\}$ allowing us to investigate whether weighting actuators as a function of betweenness centrality provides any benefit over weighting actuators as a function of information centrality. A plot of appropriate p-values vs. ρ is shown in Figure (5.49) and a table highlighting statistically significant cases is shown in Table 5.43.

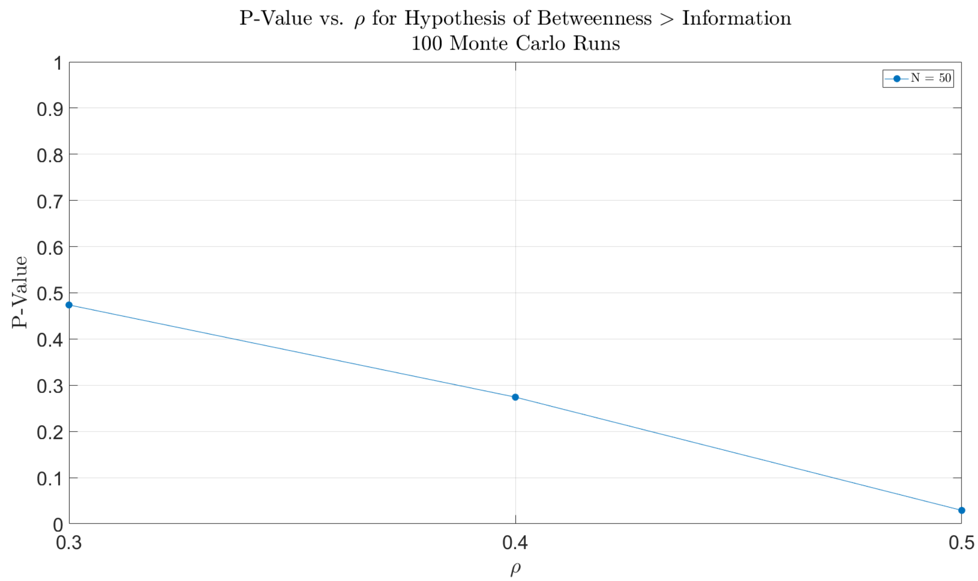


Figure 5.49: Comparison of hypotheses $\{H_{04}, H_4\}$ across Erdos-Renyi graphs with $N = 50$ nodes. Each data point is calculated using 100 Monte Carlo runs.

		ρ		
		0.3	**0.4**	**0.5**
N	**50**	0.4738	0.2742	0.0291

Table 5.43: Comparison of hypotheses $\{H_{04}, H_4\}$ across Erdos-Renyi graphs with $N = 50$ nodes. Each data point is calculated using 100 Monte Carlo runs and statistically significant cases ($p < 0.05$) are highlighted in green.

Again, we see that for $N = 50$ and $\rho = 0.5$ there is strong statistical evidence that weighting actuators as a function of betweenness centrality provides a benefit over weighting actuators as a function of information centrality.

5.4.6 Summary of Results

We see the repeated trend that as the density of the network increases, there is statistical evidence that betweenness centrality is a good heuristic for actuator selection when compared with closeness, degree, and eigenvector centrality. This implies that our previous results for $6 \leq N \leq 13$ are robust to network size. We note that as the number of nodes increases, the computational complexity of the associated LMI becomes greater. For this reason, statistics in this section are generated using only 100 Monte Carlo runs. Although the test suite consists of less data, the statistical trends are still evident.

5.5 Conclusions

We have analyzed how our initial statistical results over Erdos-Renyi graphs extend to other graph generation methods, to scenarios in which there are actuator failures, to cases with sparse control authority, as well as to larger networks. We saw that for Barabasi-Albert and Watts-Strogatz graphs, respectively, there continues to be strong statistical evidence that betweenness centrality is a good heuristic to use for actuator selection. We see that graph density remains a strong predictor of the efficacy of betweenness centrality as a selection heuristic. These results also held for instances in which one or two actuators irrecoverably fail during operation and for larger networks on the order of 50 agents. However, in scenarios in which we only have joint controllability at the network level but no agent individually has full controllability of the process, betweenness centrality no longer seems to be an appropriate heuristic. In this case, there is statistical evidence that, among the centrality measures we considered, degree centrality is a good heuristic to inform actuator selection.

6

Final Remarks

In this dissertation, we developed and analyzed algorithms for distributed control and estimation of networked multiagent systems. In Chapter 2 we considered the problem of distributed filtering of a scalar linear stochastic process where communication between agents is corrupted by Gaussian noise. We considered a two-stage consensus filter from the literature and showed how the introduction of communication noise results in destabilization. We noted that this is not unexpected as consensus dynamics are inherently not robust to noise and, in fact, act as integrators on such noise. We proposed a novel two-stage distributed filtering algorithm that is robust to communication noise and achieves a bounded asymptotic error covariance. We analyzed how to optimize two tunable parameters in our algorithm to minimize the asymptotic error covariance of the estimator. We showed how the asymptotic error variances of individual agents within a graph are determined by the respective elements of each eigenvector of the consensus matrix. We provided a preliminary analysis into the convexity of the trace of the asymptotic error covariance matrix.

In Chapter 3 we considered a framework for distributed control of linear time-invariant systems. We used the small-gain theorem to characterize linear matrix inequality conditions under which a weaker notion of the separation principle holds in the estimator and controller design. We showed how linear consensus dynamics can be applied to extend the regime of applicability of our algorithm. This is important as there could be situations in which, given a set of designed estimator parameters, there is no control gain that can stabilize the system while simultaneously satisfying the small-gain theorem. In these scenarios, multiple rounds of linear consensus can be utilized to allow agents more options in choosing state feedback gains which meet the desired criteria. We further showed how our framework can handle nonlinear control laws, which may be useful for applications such as collision avoidance.

In Chapter 4 we proposed a statistical hypothesis testing framework to aid in the problem of sensor and actuator selection for our distributed control methodology, focusing on the actuator selection part of the problem. We considered a set of centrality measures as heuristics for actuator selection with the goal being to minimize a key matrix norm related to the actuation. We considered Erdos-Renyi random graphs and showed that there is strong statistical evidence that betweenness centrality is the best performing heuristic for actuator selection given sufficient edge density.

In Chapter 5 we examined how our previous statistical conclusions extend to additional scenarios of interest. In particular, we extended our statistical analysis to additional graph generation methods, actuator failures, lack of controllability at an individual agent, as well as larger network scale. We founds strong statistical evidence that, given sufficient edge density, betweenness centrality remained the best heuristic for actuator selection in each of these domains except when considering sparser control authority. When each individual agent no longer had full controllability of the process and instead had to rely on joint controllability at the level of the network, we found strong statistical evidence that degree centrality became the best heuristic for actuator selection.

6.1 Future Work

We see many exciting future directions for this work. We mentioned in Chapter 3 that solving for the appropriate estimation and control parameters through optimizations (3.19) and (3.21) can be computationally expensive, especially when the networks under consideration grow in size. This is a main motivating factor behind developing a framework in which we can separately design these parameters so that if, for whatever reason, we need to update one set it doesn't necessitate a reoptimization to determine the other set. However, this does not change the fact that these optimizations may need a lot of computing resources in the first place. Thus, one exciting area of research we believe could be applicable to this class of problems is graph neural networks. Indeed, graph neural networks are being applied to the problem of distributed control[93,94]. The neural networks learn maps from the sensor inputs to the control outputs and can be trained using imitation learning of a set of chosen controllers. It is a possible area of exploration to use the estimator and controller designs from the optimizations (3.19) and (3.21) in Chapter 3 as training data for a graph neural network. This would allow for interpolation between different networks or being able to handle time-varying networks by using the controllers and estimators from Chapter 3 as a guide.

Another area of future work is analytically analyzing the influence of the network structure on the distributed control algorithm in Chapter 3. We resorted to statistical methods in Chapters 4 and 5 to begin exploring the role of the network structure, and in particular how an agent's location in the network influences its performance on actuation and sensing. However, given the scope of the problem we were only able to look at a chosen set of centrality measures as heuristics. If analytical results could be obtained which exploit the network structure within the relevant LMIs, it could then inform further statistical analyses and perhaps introduce more applicable centrality measures of interest.

Part II

Published Work

7
Outline and Contributions

Part II of this dissertation contains one chapter covering work that has been published in a peer-reviewed conference and a second chapter covering work that has been submitted for publication.

7.1 Outline

In Chapter 8 we present a novel algorithm for distributed filtering of a scalar linear stochastic process which shows mitigated degradation in performance when communication between agents is noisy [11]. We show that a two-stage distributed filter proposed in [56] consisting of a measurement and prediction stage followed by a consensus stage suffers in performance when allowing for communication between agents to be corrupted by Gaussian noise. In particular, we show that under certain communication noise conditions the trace of the error covariance matrix actually increases as more rounds of consensus between agents are performed, which is clearly not a desirable property of a distributed filter which utilizes consensus. We note that such behavior is not totally unexpected as consensus dynamics are inherently non-robust to noise and act as an integrator on the noise. After highlighting how such distributed filters fail in the presence of communication noise, we propose and rigorously analyze an innovative filtering algorithm which retains stabilizing properties even when exposed to communication noise.

In Chapter 9 we investigate a weaker notion of the separation principle for distributed control of LTI systems [24]. We use the small-gain theorem along with the bounded real lemma for discrete time LTI systems to characterize linear matrix inequality conditions under which agents in the network are able to change their control inputs without needing to change their estimation strategies and still achieve a stabilizing design. We additionally utilize linear consensus dynamics and show how we can extend the operating regime of our algorithm by tuning the frequency of information exchange between agents during consensus.

7.2 Contributions

In Chapter 8, Vaibhav Srivastava and I framed the main questions. I developed the algorithm along with Vaibhav Srivastava. I performed all analysis and wrote all of the simulations. Naomi Leonard and Vaibhav Srivastava revised and edited the manuscript. Chapter 8 appeared in the Proceedings of the 2017 American Control Conference [11], where I presented the work.

In Chapter 9, Shinkyu Park and I framed the main questions with guidance from Naomi Leonard and H. Vincent Poor. I developed the framework along with Shinkyu Park. Shinkyu Park and I collaboratively performed the analysis and wrote the simulations. Naomi Leonard, H. Vincent Poor, and Shinkyu Park revised and edited the manuscript. Chapter 9 has been submitted for publication.

8

On Distributed Linear Filtering with Noisy Communication

* We consider distributed filtering of a scalar linear stochastic process under communication corrupted by Gaussian noise. We investigate how communication noise degrades the per-

*This chapter was published as Savas, Srivastava, and Leonard in the American Control Conference on May 24, 2017[11].

formance of an existing distributed algorithm and develop a novel algorithm that mitigates these problems. We rigorously investigate the properties of the new distributed estimator and discuss optimal tuning of (fixed) gains that minimize the asymptotic error covariance. We demonstrate the effectiveness of our algorithm through numerical simulations.

8.1 Introduction

Distributed estimation is a problem of fundamental interest in a variety of problems ranging from robotic networks, transportation networks, power networks, and synthetic biological networks. With increasing deployment of networked multiagent systems the algorithms for distributed estimation are of increasing importance. Some of the desired features of these algorithms include scalability, adaptability, and resilience.

In this paper, we investigate the problem of distributed filtering in a networked multiagent system of a scalar linear stochastic process under communication corrupted by Gaussian noise. There is a significant and growing literature on distributed filtering in networked systems [51,52,53,54,55,56,57,58]. Typically, however, these works assume no communication noise. We design algorithms that are robust to the communication noise in such networks.

Distributed filtering in a networked multiagent system is designed to allow each individual agent to improve its estimate of the state of a dynamical system by sharing measurements or estimates through a communication network. In consensus-based distributed filtering, agents update their estimates with measurements or estimates communicated from others using linear consensus dynamics [59,60]. Olfati-Saber [53] considered distributed linear filtering with two consensus dynamics: one for weighted measurements and one for precision matrices, see[61] for related work. Distributed linear filtering in continuous time was examined in[54]. Spanos *et al.*[62] investigated the distributed least-squares estimation problem using consensus dynamics. Speranzon *et al.*[63] studied distributed linear filtering of a noisy time-varying signal using adaptive time-varying consensus.

In the context of robotic networks, cooperative Kalman filtering techniques have been used to explore noisy scalar fields in the plane[58]. Lynch *et al.*[64] studied the problem of information maximization in a scalar uncertain field using optimal filtering and consensus techniques.

We investigate the problem of consensus-based distributed filtering under *noisy communication*. The robustness of consensus dynamics under noisy communication has been studied in[57,65] and in the context of decision-making[66,67].

The consensus algorithm has its root in the sociology literature and is the same as the famous DeGroot model[68]. The modification to the consensus protocols that we propose to mitigate effects of communication noise has similarities with the DeGroot-Friedkin model in sociology[69]. The analysis in this paper suggests that the DeGroot-Friedkin model may have superior robustness properties under noisy communication.

To address the problem of distributed filtering of a scalar linear stochastic process under noisy communication, we build upon the algorithm proposed by Carli *et al.* in[56]. They propose an algorithm comprising discrete-time sampling of the noisy process and a fixed number of consensus rounds between sampling instances. We develop a new algorithm that mitigates the effect of communication noise on the performance of the distributed filter. The major challenge consensus-based strategies face under noisy communication is the presence of integrator dynamics in consensus protocols which aggregate noise over time leading to large variances and poor estimation performance. Here, we design novel consensus dynamics that alleviate this problem.

The major contributions of this paper are threefold. First, we examine the algorithm proposed in [56] for distributed filtering of a scalar linear stochastic process and show how the performance of this algorithm degrades under noisy communication. Second, we build upon [56] to develop a novel algorithm that mitigates the effects of noisy communication. Third, we rigorously analyze the new algorithm and discuss methods to tune its parameters to optimize performance.

The remainder of the paper is organized as follows. In §8.2, we formally pose the distributed linear filtering problem. In §8.3, we recall the distributed filtering algorithm from [56] and study its performance under noisy communication. In §8.4, we develop a novel algorithm to provide robustness to communication noise. We analyze this algorithm and illustrate in §8.5, and we conclude in §8.6.

8.2 Problem Setup

Consider the following scalar linear stochastic process

$$x(k+1) = ax(k) + w(k), \quad x(0) = X_0, \tag{8.1}$$

for each $k \in \mathbb{Z}_{\geq 0}$, where $a \in \mathbb{R}$ is a constant, $\{w(k)\}_{k\in\mathbb{Z}_{\geq 0}}$ is a sequence of i.i.d. zero-mean Gaussian noise with variance $q \in \mathbb{R}_{>0}$, and X_0 is a Gaussian random variable with mean x_0 and variance σ. Suppose a sensor samples this process at each time k to obtain a noisy measurement

$$y(k) = x(k) + n(k), \quad \text{for each } k \in \mathbb{Z}_{\geq 0}, \tag{8.2}$$

where $\{n(k)\}_{k \in \mathbb{Z}_{\geq 0}}$ is a sequence of i.i.d. zero-mean Gaussian noise with variance $r \in \mathbb{R}_{>0}$. The estimation of state $x(k)$ in (8.1) using measurements $y(k)$ in (8.2) is the standard scalar Kalman filtering problem[70].

We consider the problem of distributed estimation of the state $x(k)$ using multiple communicating agents. For simplicity, we assume $a = 1$ but our analysis is generalizable to the case $a \neq 1$. Specifically, we consider the estimation of the white noise process

$$x(k+1) = x(k) + w(k), \quad x(0) = X_0. \tag{8.3}$$

We consider a multiagent network in which agents can communicate over a fixed graph $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$, where $\mathcal{V} = \{1, \dots, N\}$ is the vertex set, $\mathcal{E} \in \mathcal{V} \times \mathcal{V}$ is the edge set, and N is the total number of agents. We assume that the graph is undirected and connected in the sense that there exists a path from each node to every other node. We assume that each agent $i \in \{1, \dots, N\}$ samples the process (8.3) at times k and collects a noisy measurement $y_i(k)$ of the process $x(k)$ defined by

$$y_i(k) = x(k) + n_i(k), \quad \text{for each } i \in \{1, \dots, N\}, \tag{8.4}$$

where $\{n_i(k)\}_{k \in \mathbb{Z}_{\geq 0}}$ are i.i.d. zero-mean Gaussian noises with variance r. We further assume the noise sequences $n_i(k)$ are independent for different $i \in \{1, \dots, N\}$. We can write (8.4) in vector form as

$$y(k) = x(k)1_N + n(k), \tag{8.5}$$

where $y(k)$ and $n(k)$ are the N-column vectors of $y_i(k)$'s and $n_i(k)$'s, respectively, and 1_N is the N-column vector of all ones.

We focus on consensus-based dynamics for distributed estimation[59,60,71]. However, in contrast to standard approaches to this problem we assume that the communication among agents is noisy. We recall that in consensus dynamics each agent at each (discrete) time averages its state with its neighbors in the communication graph[59,68,72]. Here we assume that each agent receives a noisy estimate of the state of each of its neighbors and it uses these noisy estimates in the consensus dynamics. Let Q be the consensus matrix, i.e., the matrix of the (convex) weights an agent i assigns to its neighbor j. Then the consensus dynamics with noisy communication are

$$z(l+1) = Qz(l) + \sigma_c u(l), \tag{8.6}$$

where $z(l)$ is the vector of states of agents at times $l \in \mathbb{Z}_{\geq 0}$, σ_c^2 is the variance of the communication noise, and $\{u(l)\}_{l\in\mathbb{Z}_{\geq 0}}$ is the sequence of i.i.d. N-variate zero-mean Gaussian random vectors with covariance $\mathcal{I}_N$, where $\mathcal{I}_N$ is the identity matrix of order N. Here for simplicity, we have assumed that the communication noise for each agent has the same variance σ_c^2.

It is well-known that the matrix Q is row stochastic and for a connected undirected graph is irreducible, i.e., the matrix Q has only one simple eigenvalue at unity and every other eigenvalue is inside the unit disk[59,68,71,72]. Moreover, the eigenvalue at unity corresponds to the eigenvector $\frac{1}{\sqrt{N}}1_N$. We assume that Q is doubly stochastic and denote its eigenvalues as $\{\lambda_0, \ldots, \lambda_{N-1}\}$ with $\lambda_0 = 1$.

8.3 A state-of-the-art distributed linear filter

The estimation problem posed in §8.2 was studied by Carli *et al.* [56] and they proposed a two-stage algorithm that we summarize in this section. We then apply their algorithm to the setting of noisy communication and observe that it is no longer stabilizing. This is to be expected since consensus dynamics have one eigenvalue at unity and consequently integrate noise. The integrated noise has asymptotically infinite variance. So any strategy that is designed for noise-free communication doesn't immediately extend to noisy communication.

8.3.1 A two-stage distributed linear filter under noise-free communication

In this section we recall the two-stage distributed linear filter proposed in [56]. During the first stage, at time k each agent i computes the estimate of process $x(k)$ given measurements until time k, i.e., $\hat{x}_i(k|k)$, by computing a convex combination of the predictive estimate of the current state using observations until time $k-1$, i.e., $\hat{x}_i(k|k-1)$ and the current observation $y_i(k)$. Formally, the first stage updates the state as

$$\hat{x}(k|k) = (1-\ell)\hat{x}(k|k-1) + \ell y(k), \tag{8.7}$$

where $\hat{x}(k|k)$ and $\hat{x}(k|k-1)$ are vectors of $\hat{x}_i(k|k)$ and $\hat{x}_i(k|k-1)$, respectively, and $\ell \in [0,1]$ is the gain. Note that, unlike the optimal Kalman filter, here the gain ℓ is assumed constant. This means that the resulting filter is not necessarily optimal. However, as shown in [56] this leads to a bounded variance of estimation error and, hence, the choice of constant ℓ is stabilizing.

The second stage comprises m rounds of the consensus dynamics (8.6) between two consecutive time instances k and $k+1$ using local estimates $\hat{x}_i(k|k)$. The consensus dynamics ensure that the local estimate $\hat{x}_i(k+1|k)$ of each agent converges towards the average of the group $\frac{1}{N}\sum_{j=1}^{N}\hat{x}_j(k|k)$. Formally, the second stage is

$$\hat{x}\Big(k+\frac{h}{m}\Big|k\Big) = Q\hat{x}\Big(k+\frac{(h-1)}{m}\Big|k\Big), \quad h \in \{1,\dots,m\}, \tag{8.8}$$

Here, a timescale separation between process dynamics and consensus dynamics is assumed, i.e., the communication and consensus dynamics are much faster than the process dynamics. Note that at the end of the consensus rounds, update (8.8) yields $\hat{x}(k+1|k)$ that can be used with update (8.7) to compute $\hat{x}(k+1|k+1)$. The distributed linear filtering algorithm is initialized with $\hat{x}(0|-1) = x_0 1_N$ and the estimates at future times are computed recursively using (8.7) and (8.8).

Carli *et al.*[56] computed the covariance of the estimation error for the above algorithm and used it to find the optimal ℓ that minimizes the trace of the asymptotic error covariance matrix.

The above algorithm is easy to implement and under noise-free communication is stabilizing, i.e., always leads to bounded error covariance. In the next section, we investigate the performance of this algorithm under noisy communication.

8.3.2 Performance under noisy communication

We now consider the two-stage distributed linear filtering algorithm in §8.3.1 with noisy communication in the consensus dynamics. The first stage of the algorithm remains identical to update (8.7). In the second stage the update (8.8) is replaced by noisy consensus dynamics

$$\hat{x}\Big(k+\frac{h}{m}\Big|k\Big) = Q\hat{x}\Big(k+\frac{(h-1)}{m}\Big|k\Big) + \sigma_c u\Big(k+\frac{h}{m}\Big), \tag{8.9}$$

where $u(k+h/m)$ is the N-variate zero-mean Gaussian noise with covariance $\mathcal{I}_N$, for each $k \in \mathbb{Z}_{\geq 0}$ and $h \in \{1,\ldots,m\}$, $u(k+h/m)$ are independent, and σ_c^2 is the communication noise variance. The estimation error at time k is defined by

$$\tilde{x}(k|k-1) = x(k)1_N - \hat{x}(k|k-1). \tag{8.10}$$

We numerically investigate the performance of the two-stage algorithm described in §8.3.1 under noisy communication. Consider a set of three agents $\{1,2,3\}$ communicating over an undirected line graph. Let $Q = \mathcal{I}_3 - \varepsilon L$, where $\mathcal{I}_3$ is the identity matrix and $\varepsilon = 0.4$ is a constant.

We examine ten time instances of the process (8.3), i.e., $k \in \{0, \ldots, 9\}$, and between each consecutive pair of time instances we apply m consensus rounds. We illustrate performance for values of m from 0 to 5. We assume the process noise variance is $q = 1$ and the measurement noise variance is $r = 25$.

We employ the distributed filtering algorithm (8.7) and (8.9) with convexity parameter $\ell = 0.25$. We performed $200,000$ Monte-Carlo simulations to estimate the trace of the error covariance matrix. Fig. 8.1 shows the trace of the error covariance matrix for $k = 4$, which can be represented as $\sum_{i=1}^{3} \mathrm{var}(\tilde{x}_i(4|3))$, as a function of the number of consensus rounds m for a range of values of σ_c. It can be seen that for large enough values of σ_c the trace of the error covariance actually increases as more consensus rounds are performed, suggesting that the two-stage estimation algorithm in §8.3.1 is not stabilizing, i.e., the trace of the error covariance diverges as the number of consensus rounds are increased.

This is not totally unexpected because the consensus dynamics are inherently non-robust due to the presence of an eigenvalue of unity. This eigenvalue at unity acts as an integrator and integrates noise. As we integrate more and more noise the covariance of the system diverges.

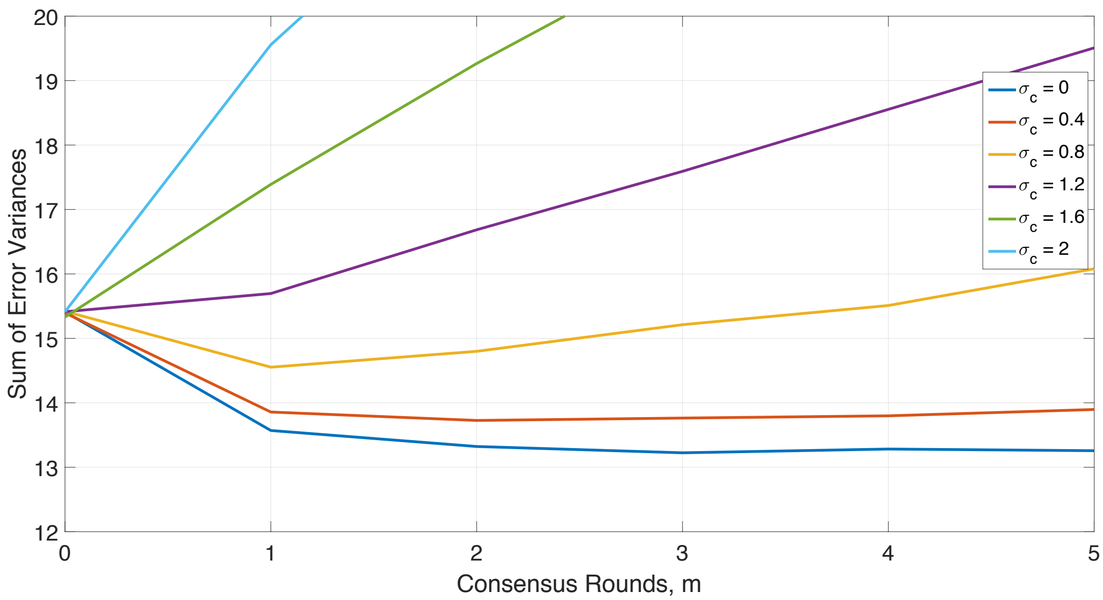


Figure 8.1: Influence of communication noise in consensus dynamics on error variance across 200,000 Monte Carlo runs for distributed filtering algorithm (8.7) and (8.9) with $N = 3$, $r = 25$, and $q = 1$ for an undirected line graph. We see that the error variance diverges with the number of consensus rounds.

8.4 A novel two-stage distributed linear filter

As discussed in the previous section, the filter in §8.3.1 suffers under noisy communication. In this section, we modify the algorithm of Carli *et al.* [56] to mitigate the effects of noisy communication.

We keep the update in the first stage of the algorithm the same as in (8.7), i.e.,

$$\hat{x}(k|k) = (1-\ell)\hat{x}(k|k-1) + \ell y(k), \tag{8.11}$$

with $\hat{x}(0|-1) = x_0 1_N$.

We modify the second stage, i.e., the consensus dynamics, in the following way. We define $z(k|k) = \hat{x}(k|k)$ for each $k \in \mathbb{Z}_{\geq 0}$. We update z through m consensus rounds between consecutive time instances k and $k+1$ as follows:

$$z\Big(k+\frac{h}{m}\Big|k\Big) = Qz\Big(k+\frac{(h-1)}{m}\Big|k\Big) + \sigma_c u\Big(k+\frac{h}{m}\Big) + \hat{x}(k|k). \tag{8.12}$$

In (8.12), each agent $i \in \{1,\dots,N\}$ remembers its estimate $\hat{x}_i(k|k)$ at time k and re-injects it at each consensus round. Loosely speaking, the intuition for such an update is that starting from a deterministic initial condition $z(k|k) = \hat{x}(k|k)$ and after m rounds of consensus the dominating component of the variance of $z(k+1|k)$ is $m\sigma_c^2$ (see Fig. 8.1). By re-injecting $\hat{x}(k|k)$ at each step, we ensure that the dominating component of the expected value of $z_i(k+1|k)$ is $\frac{m+1}{N}\sum_{j=1}^{N}\hat{x}_j(k|k)$, for each $i \in \{1,\dots,N\}$. Finally, if we divide $z(k+1)$ by $(m+1)$, the resulting mean is $\frac{1}{N}\sum_{j=1}^{N}\hat{x}_j(k|k)$ and variance is $m\sigma_c^2/(m+1)^2$ which goes to 0 as $m \to +\infty$. Thus, for large m we recover the performance of the noise-free algorithm. However, if m is small noise still degrades performance, so we set the update $\hat{x}(k+1|k)$ as the convex sum of $\hat{x}(k|k)$ and $z(k+1|k)$ as below

$$\hat{x}(k+1|k) = \zeta\hat{x}(k|k) + (1-\zeta)\frac{z(k+1|k)}{m+1}, \tag{8.13}$$

where $\zeta \in [0,1]$ is a constant. ζ trades off the variance of the two estimators $\hat{x}(k|k)$ and $z(k+1|k)$. Thus, for large m we can choose ζ close to 0 and for small m we can choose ζ close to 1.

In contrast to the distributed filtering algorithm in [56] which has only one tunable parameter ℓ, our algorithm has two tunable parameters ℓ and ζ. Similar to [56], these parameters can be chosen to minimize asymptotic error covariance of the estimator. Towards this end, we analyze the error covariance of the new algorithm in the next section.

8.5 Analysis of the novel two-stage distributed linear filter

In this section we analyze the properties of the novel distributed linear filter proposed in §8.4. We first derive an expression for the asymptotic error covariance and then analyze its properties. Our analysis follows similarly to [56].

8.5.1 Error covariance of the estimator

We define the predictive and posterior errors as

$$\begin{aligned} \tilde{x}(k+1|k) &= x(k+1)1_N - \hat{x}(k+1|k), \\ \text{and} \quad \tilde{x}(k+1|k+1) &= x(k+1)1_N - \hat{x}(k+1|k+1), \end{aligned} \tag{8.14}$$

respectively. Let

$$\begin{aligned} P(k+1|k) &= \mathbb{E}[\tilde{x}(k+1|k)\tilde{x}(k+1|k)^\top] \\ \text{and} \quad P(k+1|k+1) &= \mathbb{E}[\tilde{x}(k+1|k+1)\tilde{x}(k+1|k+1)^\top] \end{aligned}$$

be predictive and posterior error covariance matrices. We are now ready to state the main result of this section.

Theorem 4 (***Asymptotic Error Covariance***). *For the scalar linear stochastic dynamics* (8.3) *and the distributed linear filtering algorithm with noisy communication defined by* (8.11), (8.12) *and* (8.13), *the following statements hold:*

1. *the asymptotic error covariance is*

$$\begin{aligned}
&\lim_{k\to\infty} P(k|k-1) \\
&\qquad = \ell^2 r \sum_{i=0}^{\infty} (1-\ell)^{2i} Q^{\dagger(i+1)} (Q^{\dagger(i+1)})^\top + \frac{q}{1-(1-\ell)^2} 1_N 1_N^\top \\
&\qquad\qquad + \left(\frac{1-\zeta}{m+1}\right)^2 \sigma_c^2 \sum_{i=0}^{\infty} (1-\ell)^{2i} \sum_{j=0}^{m-1} Q^{\dagger i} Q^j (Q^j)^\top (Q^{\dagger i})^\top, \quad (8.15)
\end{aligned}$$

where $Q^\dagger = \zeta \mathcal{I}_N + \left(\frac{1-\zeta}{m+1}\right) \sum_{i=0}^{m} Q^i$;

2. *the trace of the asymptotic covariance matrix is*

$$\operatorname{tr}\Big(\lim_{k\to\infty} P(k|k-1)\Big) = \frac{\ell^2 r + qN + \dfrac{(1-\zeta)^2\sigma_c^2 m}{(m+1)^2}}{1-(1-\ell)^2} + \ell^2 r \sum_{h=1}^{N-1} \frac{\left|\left(\dfrac{1-\zeta}{m+1}\right)\bar{\lambda}_h + \zeta\right|^2}{1-(1-\ell)^2\left|\left(\dfrac{1-\zeta}{m+1}\right)\bar{\lambda}_h + \zeta\right|^2} + \left(\frac{1-\zeta}{m+1}\right)^2 \sigma_c^2 \sum_{h=1}^{N-1} \frac{\left(\dfrac{1-|\lambda_h|^{2(m-1)}}{1-|\lambda_h|^2}\right)}{1-(1-\ell)^2\left|\left(\dfrac{1-\zeta}{m+1}\right)\bar{\lambda}_h + \zeta\right|^2}, \quad (8.16)$$

where $\bar{\lambda}_h = \sum_{n=0}^{m} \lambda_h^n$.

Proof. From (8.14) and (8.11) it follows that

$$\tilde{x}(k|k) = (1-\ell)\tilde{x}(k|k-1) - \ell n(k). \quad (8.17)$$

Then

$$P(k|k) = (1-\ell)^2 P(k|k-1) + \ell^2 r \mathcal{I}_N. \quad (8.18)$$

Further note that (8.12) can be solved explicitly to obtain

$$z(k+1|k) = \sum_{i=0}^{m} Q^i \hat{x}(k|k) + \sigma_c \sum_{i=1}^{m} Q^{m-i} u\Big(k + \frac{i}{m}\Big). \quad (8.19)$$

Substituting $z(k+1|k)$ in (8.13) and using (8.14), we obtain

$$\tilde{x}(k+1|k) = Q^{\dagger}\tilde{x}(k|k) + w(k+1)1_N - \left(\frac{1-\zeta}{m+1}\right)\sum_{i=1}^{m} Q^{m-i}\sigma_c u\left(k+\frac{i}{m}\right), \quad (8.20)$$

where $Q^{\dagger} = \zeta\mathcal{I}_N + \left(\frac{1-\zeta}{m+1}\right)\sum_{i=0}^{m} Q^i$. It follows that

$$P(k+1|k) = Q^{\dagger}P(k|k)(Q^{\dagger})^{\top} + q1_N1_N^{\top} + \left(\frac{1-\zeta}{m+1}\right)^2 \sigma_c^2 \sum_{i=0}^{m-1} Q^i(Q^i)^{\top}. \quad (8.21)$$

Since Q is row-stochastic with $Q1_N = 1_N$, it can be shown that $Q^{\dagger}1_N = 1_N$, i.e., $Q^{\dagger}$ is also row-stochastic. Using row-stochasticity of $Q^{\dagger}$, (8.18) and (8.21) we obtain

$$\begin{aligned} P(k+1|k) = {} & (1-\ell)^2 Q^{\dagger}P(k|k-1)(Q^{\dagger})^{\top} + \ell^2 r Q^{\dagger}(Q^{\dagger})^{\top} \\ & + q1_N1_N^{\top} + \left(\frac{1-\zeta}{m+1}\right)^2 \sigma_c^2 \sum_{i=0}^{m-1} Q^i(Q^i)^{\top}. \quad (8.22) \end{aligned}$$

We can solve (8.22) with initial condition $P(0|-1)$ to obtain:

$$\begin{aligned} P(k|k-1) = {} & (1-\ell)^{2k} Q^{\dagger k}P(0|-1)(Q^{\dagger k})^{\top} \\ & + \ell^2 r \sum_{i=0}^{k-1}(1-\ell)^{2i} Q^{\dagger(i+1)}(Q^{\dagger(i+1)})^{\top} + q\sum_{i=0}^{k-1}(1-\ell)^{2i} 1_N1_N^{\top} \\ & + \left(\frac{1-\zeta}{m+1}\right)^2 \sigma_c^2 \sum_{i=0}^{k-1}(1-\ell)^{2i} \sum_{j=0}^{m-1} Q^{\dagger i}Q^j(Q^j)^{\top}(Q^{\dagger i})^{\top}. \quad (8.23) \end{aligned}$$

Taking the limit $k \to +\infty$ and using the geometric series summation formula, we establish (i).

The second statement follows using model decomposition of Q and the result that

$$\operatorname{tr}\left(Q^{\dagger im}(Q^{\dagger im})^{\top}\right) = \sum_{h=0}^{N-1} \left| \left(\frac{1-\zeta}{m+1}\right) \bar{\lambda}_h + \zeta \right|^{2im}. \tag{8.24}$$

□

Note that the steady-state error covariance (8.16) is bounded and hence our distributed estimation algorithm is stabilizing in the mean squared sense. Our algorithm is not necessarily optimal since it assumes convex weights ℓ and ζ to be constant. However, given our algorithm, we can choose optimal parameters ℓ and ζ as we discuss in the next section.

8.5.2 Tuning parameters ℓ and ζ

The algorithm defined in §8.4 requires two parameters ℓ and ζ to be tuned. For a given graph structure (fixed Q and N), given process, measurement and communication variance (fixed r, q, and σ_c), and a given number of consensus rounds (fixed m), we choose these parameters to minimize the asymptotic error covariance (8.16). In the following, let $J(\ell, \zeta) = \operatorname{tr}\Big(\lim_{k\to\infty} P(k|k-1)\Big)$.

When determining the optimal (ℓ, ζ) which minimize J, we note a special case for $m = 0$. In this case of no consensus the modified algorithm will only involve (8.11) and (8.13). Inspecting (8.13) we see that the only appropriate formulation would have $\zeta = 1$. With $\zeta = 1$ and $m = 0$ we see that (8.16) simplifies to $J|_{m=0} = \frac{(\ell^2 r+q)N}{1-(1-\ell)^2}$. We then minimize $J|_{m=0}$ using `fmincon` in MATLAB to solve for the optimal ℓ. Likewise, for $m > 0$ we can use `fmincon` to solve for the optimal ℓ and ζ which minimize J.

The trends of optimal ℓ and ζ as a function of number of consensus rounds m and σ_c are

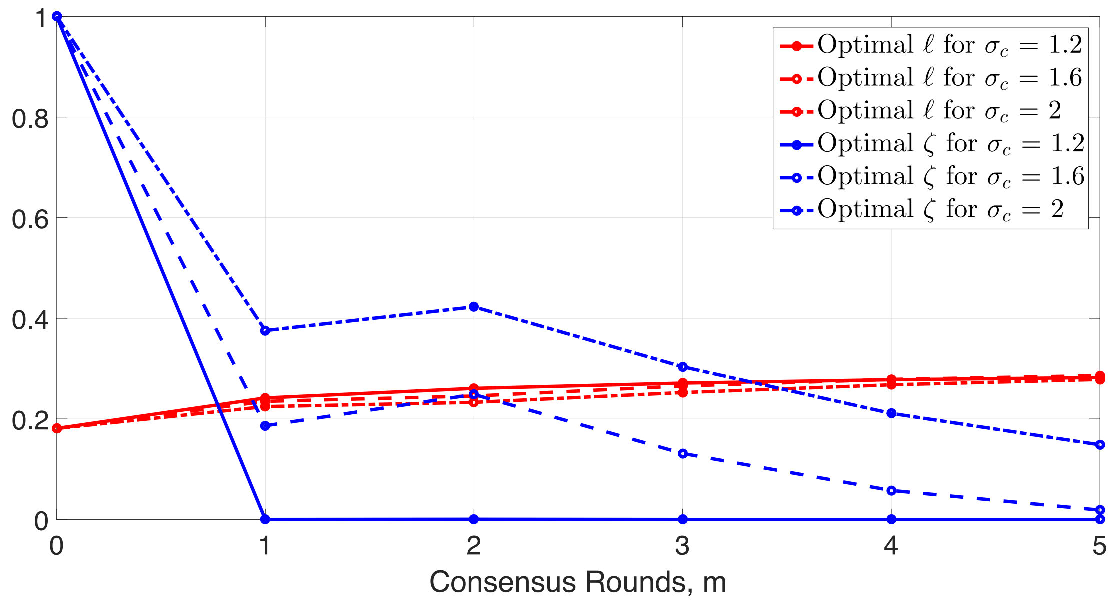


Figure 8.2: Optimal ℓ and ζ as a function of consensus rounds m and σ_c with $N = 3$, $r = 25$, and $q = 1$ for an undirected line graph.

shown in Fig. 8.2. Note that optimal ℓ increases with m, while ζ does not follow a monotonic trend. The initial trend of ζ is attributed to the transient consensus dynamics. As the number of consensus rounds increases, the value ζ goes to zero as discussed in §8.4.

8.5.3 Numerical Simulations

We numerically investigate the performance of the modified estimation algorithm developed in §8.4. We consider an undirected line graph with $N = 3$ nodes. We choose the same parameters as in Fig. 8.1 and again choose as our performance metric $\sum_{i=1}^{3} \mathrm{var}(\tilde{x}_i(4|3))$. For each value of m and σ_c we use the optimal ℓ and ζ in the algorithm as determined in §8.5-B. Fig. 8.3 shows for the modified estimation algorithm the summed error variance metric versus the number of consensus rounds m, with the color of the lines designating the value of σ_c in (8.12). Comparing with Fig. 8.1, we see that the error variance no longer increases

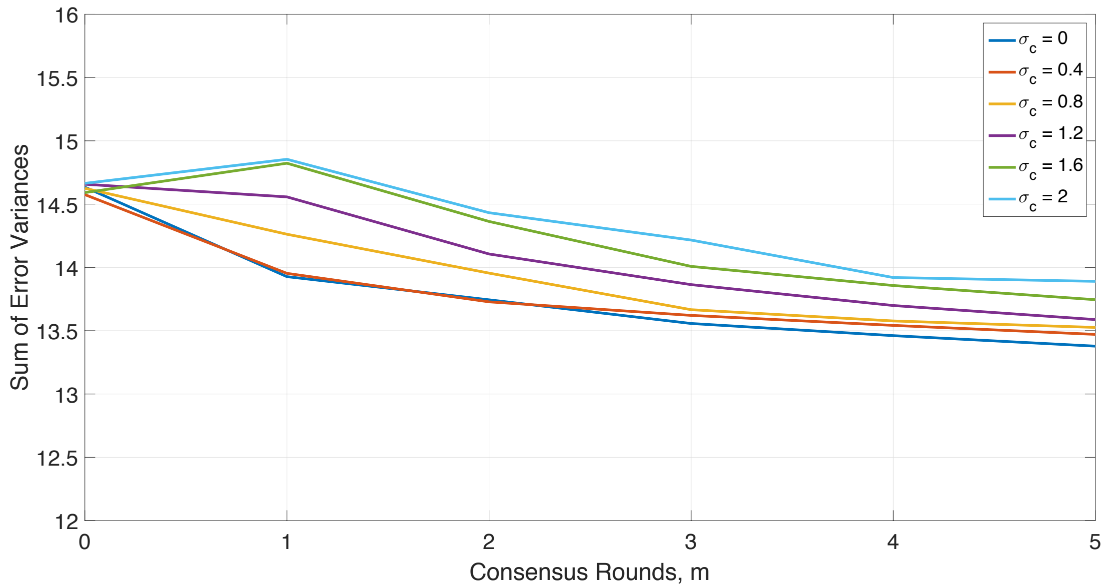


Figure 8.3: Influence of communication noise in consensus dynamics on error variance across 200,000 Monte Carlo runs for the distributed filtering algorithm defined in §8.4 with $N = 3$, $r = 25$, and $q = 1$ for an undirected line graph. Even as σ_c increases the error variance no longer diverges as more consensus rounds are performed.

in an unbounded way as more consensus rounds are performed. Rather, the trend in error variance versus consensus rounds is much closer to the monotonically decreasing trend one expects in a distributed filter without communication noise. Even for larger values of σ_c we see the error variance decreases with additional consensus rounds, which is how an effective estimation algorithm should perform. Note the slight difference in scale between the vertical axes in Figs. 8.1 and 8.3.

8.6 Conclusions

In this paper we studied consensus-based distributed linear filtering under noisy communication. We investigated how noisy communication affects the performance of the distributed filtering algorithm proposed in [56]. We showed that under noisy communication the error covariance of the estimator obtained using this algorithm diverges. We modified the algorithm from [56] to develop a novel distributed filtering algorithm that achieves a bounded asymptotic error covariance under noisy communication. We discussed how the parameters of this new algorithm can be tuned.

Future directions include examining the convexity of the asymptotic error covariance with respect to algorithm parameters, extending to vector-valued dynamical processes, and exploring the influence of the network graph on the performance.

9

A Separation Principle in the Design of Distributed Control for LTI Systems

[*] The separation principle in a (centralized) estimation and control problem gives us the flexibility to design a feedback controller independent of the state estimator. However, the same

[*]This chapter is in preparation for submission and appears as Savas, Park, Poor, and Leonard[24].

principle does not hold when the estimation and control are distributed over a network. In this case, the estimator needs to be redesigned whenever the control strategy is revised, which can be computationally expensive. We investigate a weaker notion of the separation principle in the distributed control of linear time-invariant (LTI) systems. As a main contribution, we characterize the notion using linear matrix inequalities (LMIs) under which agents in the network can change their feedback control without redesigning the estimator. We also analyze how the frequency of information exchange between neighboring agents can extend the regime in which the separation principle holds. We validate our analytical results using a multi-vehicle platooning example through simulations.

9.1 Introduction

Consider the problem of designing a network of agents for estimation and control of discrete-time linear time-invariant (LTI) systems. Each agent in the network assesses a partial output of an LTI system, communicates with its neighbors to estimate the system's state, and applies control input to stabilize the system based on available local information – the partial system output and its neighbors' state estimates. Such design problems arise in a wide range of control engineering domains including formation control of networked robots, efficient power transfer in large-scale power systems, and traffic flow control in transportation networks.

An interesting aspect of the problem is that each agent computes control input based on its own set of information which is different from that of other agents. Hence, such a problem setting imposes a non-classical information structure under which changing the control strategy of an agent to improve the system performance would negatively affect the network's estimation performance and require redesigning the estimation strategy, which is computationally expensive.

In a centralized counterpart, where a single agent estimates and controls the system's state, the *separation principle* gives the agent the freedom to adopt any stabilizing feedback controller independent of the estimator design[74]. However, because of the non-classical information structure, the same separation principle does not hold in our problem unless every agent is broadcasting its locally available information across the network. Hence, the estimation and control strategies across the network would need to be jointly designed.

There is a large literature, including[53,75,76], that presents computational methods and performance analysis for design of networks to estimate the state of LTI systems, where consensus-type algorithms are adopted to exchange and fuse state estimates among neighboring agents. More recent work[77] proposes a control framework based on distributed estimation to design a network of agents to stabilize LTI systems and discusses performance of the proposed framework. LMI-based approaches for joint design of distributed observer and controller are discussed in[78,79].

Also,[80] discusses a decentralized approach for designing a distributed controller for continuous-time LTI systems in which the agents are required to share their local information with neighbors at every time instance over the continuous-time domain. The authors of[81] propose a consensus-based distributed observer over a rate-limited communication network to design linear state feedback systems. The work of[82] investigates a distributed approach to the design of a distributed observer and controller for spatially interconnected systems in the continuous-time domain.

As a complement to the existing literature, we address the following question: *"Under* *what conditions can the agents change their feedback control without revising their estimation strategies?"* Our main contribution is to provide LMI formulations that characterize a weaker notion of the separation principle under which the agents can revise their state feedback strategy without updating parameters of their distributed estimation strategy. In addition, we investigate how the frequency of information exchange through consensus impacts the establishment of the separation principle. Unlike the continuous-time domain problems as in[82,80], the problem of designing distributed control in the discrete-time domain has its own unique technical challenges.

The paper is organized as follows. In §9.2, we present our framework and explain the main problem on the distributed estimation and control design. In §9.3, we propose and prove LMI formulations as a solution to the main problem, and we provide analysis on how the frequency of information exchange affects the feasibility of the LMI. In §9.4, using simulations, we illustrate our main results in a multi-vehicle platooning example. We conclude the paper in §9.5.

9.2 Problem Description

Consider a discrete-time LTI system given by

$$x(k+1) = Ax(k) + \textstyle\sum_{i=1}^{N} B_i u_i(k),\ x(0) \in \mathbb{R}^n \tag{9.1a}$$

$$y_i(k) = C_i x(k),\ i \in \{1, \cdots, N\} \tag{9.1b}$$

where $x(k) \in \mathbb{R}^n$ is the state, $u_i(k) \in \mathbb{R}^{q_i}$ is the i-th input, and $y_i(k) \in \mathbb{R}^{r_i}$ is the i-th output of the system. We design a network of N agents where each agent i assesses its associated system output $y_i(k)$, communicates over a fixed directed graph $\mathcal{G} = (\mathbb{V}, \mathbb{E})$, and assigns control input $u_i(k)$. Each vertex i in $\mathbb{V} = \{1, \cdots, N\}$ represents agent i and each edge $(j, i) \in \mathbb{E}$ indicates that agent j can transmit information to agent i. We define the neighborhood set $\mathbb{N}_i = \{j \in \mathbb{V} \,|\, (j, i) \in \mathbb{E}\}$ to specify a subset of agents that can transmit information to agent i.

We assume that $\mathcal{G}$ is strongly connected and (9.1) is jointly controllable and observable: both pairs (A, B) and (C, A) are controllable and observable, respectively, where $B = (B_1, \cdots, B_N)$ and $C = (C_1^T, \cdots, C_N^T)^T$. However, individual agents do not necessarily have full controllability or observability of the system: For every i in $\mathbb{V}$, pairs (A, B_i) and (C_i, A) may not be controllable and observable, respectively. Hence, without communication with others, each agent can neither estimate the full state of the system nor stabilize it. Below we provide an example of (9.1) and $\mathcal{G}$.

Example 2. *Consider a system of N vehicles moving on the plane where each vehicle can apply a force to control its own motion and can observe its own position. The system is thus jointly controllable and jointly observable. The state and parameters of the system's model* (9.1) *are given as*

$$x = (p_1^T, v_1^T, \cdots, p_N^T, v_N^T)^T \tag{9.2a}$$

$$A = I_N \otimes \begin{pmatrix} I_2 & 0.5 I_2 \\ 0 & I_2 \end{pmatrix} \tag{9.2b}$$

$$B_i = e_i \otimes \begin{pmatrix} 0 \\ I_2 \end{pmatrix}, \; C_i = e_i^T \otimes \begin{pmatrix} I_2 & 0 \end{pmatrix}, \; i \in \{1, \cdots, N\} \tag{9.2c}$$

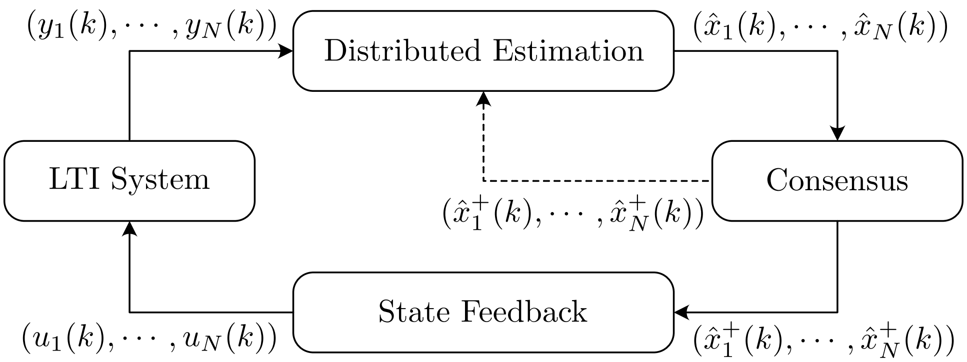


Figure 9.1: A diagram illustrating the closed loop consisting of the LTI system, distributed estimation, (linear) consensus, and state feedback.

where $p_i, v_i \in \mathbb{R}^2$ are, respectively, the position and velocity of the i-th vehicle and e_i is a canonical basis in $\mathbb{R}^N$ whose elements are all zero except the i-th element, which is 1. In §9.4, using this example, we illustrate the main results of our work over line and ring graphs, and apply our framework to a multi-vehicle formation control problem.

Our main goal is to design the network of agents in which each agent i computes a state estimate $\hat{x}_i(k)$, exchanges and fuses the estimate with those of its neighbors, and uses the fused state estimate to compute control input $u_i(k)$ for system stabilization. To design the network, we implement state feedback, distributed estimation, and linear consensus at each agent (see Fig. 9.1 for an illustration of the closed loop consisting of the three components and the LTI system):

State feedback Let $\hat{x}_i(k)$ be the state estimate of agent i. The agent computes $u_i(k)$ according to

$$u_i(k) = K_i \hat{x}_i(k). \tag{9.3}$$

Note that the agent uses only its own state estimate to compute $u_i(k)$. Throughout the paper, we assume that $\{K_i\}_{i\in\mathbb{V}}$ satisfy that $A+\sum_{i\in\mathbb{V}} B_iK_i$ is Schur stable having eigenvalues inside the unit circle in the complex plane.

Distributed estimation Agent i computes $\hat{x}_i(k)$ by recursively updating it according to

$$\hat{x}_i(k+1) = A\hat{x}_i(k) + L_i\left(y_i(k) - C_i\hat{x}_i(k)\right) \\ + \sum_{j\in\mathbb{V}} B_jK_j\hat{x}_i(k) + \sum_{j\in\mathbb{N}_i} W_{ij}\left(\hat{x}_j(k) - \hat{x}_i(k)\right), \quad (9.4)$$

where K_i is the control gain matrix in (9.3) and $W_{ij}\in\mathbb{R}^{n\times n}$, $L_i\in\mathbb{R}^{n\times r_i}$ are the parameters that need to be determined. (9.4), which is motivated by the existing distributed estimation approaches proposed, for instance, in [53,75,76,77], updates $\hat{x}_i(k)$ using the partial output $y_i(k)$, the information $\{\hat{x}_j(k)\}_{j\in\mathbb{N}_i}$ from the agent's neighbors, and the estimate $\sum_{j\in\mathbb{V}} B_jK_j\hat{x}_i(k)$ of the control input applied to (9.1).[†]

m-round linear consensus When additional information exchange is allowed, the agents exchange their state estimates and fuse the exchanged information using m-round linear consensus. Letting $\{\hat{x}_j(k)\}_{j\in\mathbb{V}}$ be the state estimates of the agents at the beginning of the linear consensus, the output $\hat{x}_i^+(k)$ at each agent i is determined as follows:

$$\hat{x}_i^+(k) = \sum_{j\in\mathbb{V}} \bar{P}_{ij}\hat{x}_j(k). \quad (9.5)$$

[†]According to (9.3), the control input applied to (9.1) is given by $\sum_{j\in\mathbb{V}} B_jK_j\hat{x}_j(k)$ which depends on every agent's state estimate.

The parameter $\bar{P}_{ij} \geq 0$ is the i, j-th element of a matrix defined as $\bar{P} = P^m \in \mathbb{R}^{N \times N}$, where m is a non-negative integer and P is a stochastic matrix that conforms with $\mathcal{G} = (\mathbb{V}, \mathbb{E})$, i.e., $(j, i) \notin \mathbb{E} \iff P_{ij} = 0$. Hence, $\hat{x}_i^+(k)$ denotes agent i's updated state estimate after applying the m rounds of linear consensus with matrix P. We refer to m and P as the total number of *consensus rounds* and the *consensus matrix*, respectively. The outcome $\hat{x}_i^+(k)$ is then fed into (9.3) and (9.4) to update each agent's state estimate, $\hat{x}_i(k) = \hat{x}_i^+(k)$, as illustrated in Fig. 9.1.

The work of[83,75,76] presents technical conditions on the system (9.1) and the graph $\mathcal{G}$ under which there are parameters $\{W_{ij}\}_{i,j \in \mathbb{V}}$, $\{L_i\}_{i \in \mathbb{V}}$ that ensure convergence of $\hat{x}_i(k)$ to $x(k)$ for every i in $\mathbb{V}$, when there is no state feedback, i.e., $K_i = 0, \; \forall i \in \mathbb{V}$. More recent work in[77] describes how to jointly compute $\{K_i\}_{i \in \mathbb{V}}$, $\{W_{ij}\}_{i,j \in \mathbb{V}}$, $\{L_i\}_{i \in \mathbb{V}}$ to stabilize (9.1) using (9.3) and (9.4).

In our work, similar to[77], we investigate the problem of designing the state feedback and distributed estimation for system stabilization. However, our work is distinct from[77] in that we consider the case where the gain matrices $\{K_i\}_{i \in \mathbb{V}}$ are designed independently of the parameters $\{W_{ij}\}_{i,j \in \mathbb{V}}$, $\{L_i\}_{i \in \mathbb{V}}$. In particular, our main result establishes technical conditions that ensure the stability of (9.1) when $\{K_i\}_{i \in \mathbb{V}}$ and $\{W_{ij}\}_{i,j \in \mathbb{V}}$, $\{L_i\}_{i \in \mathbb{V}}$ are computed using two decoupled numerical methods. As discussed in §9.3.3 in detail, our result can be applied to scenarios where each agent needs to update its control gain without re-computing the parameters of the distributed estimation over the entire network[‡], or nonlinear state feedback is adopted. In both cases, the analysis of[77] cannot be directly applied.

[‡]As we describe in §9.4.1, finding the parameters for (9.4) involves finding a solution to a large-size linear matrix inequality, which can be computationally expensive. For this reason, whenever possible, it is preferred not to re-compute the parameters of (9.4) when the state feedback is revised.

Also our work allows communication among agents to take place at discrete-time instances, different from the problem setting investigated in[82,80]. This makes our proposed framework and main theorems not only technically distinct but also applicable to a wide range of engineering problems in which agents can only exchange information for a limited number of times over any given finite time interval.

We formalize our main problem as follows.

Problem 2. *For fixed $m \geq 0$, compute the parameters $\{W_{ij}\}_{i,j\in\mathbb{V}}$, $\{L_i\}_{i\in\mathbb{V}}$ and identify the set of state feedback gains $\{K_i\}_{i\in\mathbb{V}}$ for which the control inputs determined by* (9.3) *and* (9.4) *stabilize the system* (9.1).

9.3 Parameter Design for State Feedback and Distributed Estimation

We present a linear matrix inequality (LMI) formulation to compute the parameters of (9.4) that result in the stability of (9.1), and discuss the existence of a solution to the LMI formulation. We start with $m = 0$, i.e., no linear consensus and address the general case, where $m \geq 0$, in §9.3.2.

We begin by defining the estimation error as $\tilde{x}_i(k) = x(k) - \hat{x}_i(k)$. Using (9.1), (9.3)-(9.5), the state equation for $\tilde{x}_i(k)$ can be derived as follows:

$$\tilde{x}_i(k+1) = (A - L_iC_i)\,\tilde{x}_i(k) + \textstyle\sum_{j\in\mathbb{V}} B_jK_j\left(\tilde{x}_i(k) - \tilde{x}_j(k)\right) \\ + \textstyle\sum_{j\in\mathbb{N}_i} W_{ij}\left(\tilde{x}_j(k) - \tilde{x}_i(k)\right). \quad (9.6)$$

In what follows, we cast (9.6) as a feedback interconnection of two components – the *control component* (9.7) and *estimation component* (9.8) defined below – and find sufficient conditions for the feedback interconnection of the two components to attain the convergence $\lim_{k\to\infty} \|\tilde{x}_i(k)\|_2 = 0,\ \forall i \in \mathbb{V}$. We will use this result to address Problem 2.

Let $W_{ij} = W_{ij}^{\mathrm{E}} + W_{ij}^{\mathrm{C}}$ and define

$$\tilde{v}_i(k) = \textstyle\sum_{j\in\mathbb{V}} B_j K_j(\tilde{x}_i(k) - \tilde{x}_j(k)) - \sum_{j\in\mathbb{N}_i} W_{ij}^{\mathrm{C}}(\tilde{x}_i(k) - \tilde{x}_j(k)) \tag{9.7}$$

$$\tilde{x}_i(k+1) = (A - L_i C_i)\,\tilde{x}_i(k) - \textstyle\sum_{j\in\mathbb{N}_i} W_{ij}^{\mathrm{E}}(\tilde{x}_i(k) - \tilde{x}_j(k)) + \tilde{v}_i(k). \tag{9.8}$$

Note that the feedback interconnection of (9.7) and (9.8) is equivalent to (9.6). In (9.7), the first term $\sum_{j\in\mathbb{V}} B_j K_j(\tilde{x}_i(k) - \tilde{x}_j(k))$ denotes the difference between the control input applied to the system and the estimate of it by agent i, and the second term $\sum_{j\in\mathbb{N}_i} W_{ij}^{\mathrm{C}}(\tilde{x}_i(k) - \tilde{x}_j(k))$ is adopted to counteract the error in the control input estimation. A simple choice of W_{ij}^{C} is $W_{ij}^{\mathrm{C}} = B_j K_j$. However, such choice of W_{ij}^{C} would not be optimal in our formulation. In §9.4, we present LMI-based optimization to find the best selection of W_{ij}^{C}. (9.8) is equivalent to (9.6) except that we represent the control input estimation error term by $\tilde{v}_i(k)$ and adopt $\{W_{ij}^{\mathrm{E}}\}_{i,j\in\mathbb{V}}$ in place of $\{W_{ij}\}_{i,j\in\mathbb{V}}$.

We use the small-gain theorem [84, Chapter 5.4] to specify conditions on the parameter selection that ensure the convergence in (9.6). To this end, let us represent (9.8) as an LTI system with state $\tilde{x}(k) = (\tilde{x}_1(k), \cdots, \tilde{x}_N(k)) \in \mathbb{R}^{nN}$ and input $\tilde{v}(k) = (\tilde{v}_1(k), \cdots, \tilde{v}_N(k)) \in \mathbb{R}^{nN}$ as follows:

$$\tilde{x}(k+1) = \tilde{A}\tilde{x}(k) + \tilde{v}(k), \tag{9.9}$$

where $\tilde{A} \in \mathbb{R}^{nN \times nN}$ is defined as

$$\tilde{A} = \mathrm{diag}(A - L_1 C_1, \cdots, A - L_N C_N) + W^{\mathrm{E}} \tag{9.10}$$

and W^{E} is a block matrix whose i, j-th block element is

$$[W^{\mathrm{E}}]_{ij} = \begin{cases} W^{\mathrm{E}}_{ij} & \text{if } j \in \mathbb{N}_i \backslash \{i\} \\ -\sum_{l \in \mathbb{N}_i \backslash \{i\}} W^{\mathrm{E}}_{il} & \text{if } j = i \\ 0_n & \text{otherwise.} \end{cases} \tag{9.11}$$

Also, we rewrite (9.7) as

$$\tilde{v}(k) = \tilde{D}\tilde{x}(k), \tag{9.12}$$

where $\tilde{D}$ is a block matrix whose i, j-th block element is

$$[\tilde{D}]_{ij} = \begin{cases} -B_j K_j + W^{\mathrm{C}}_{ij} & \text{if } j \in \mathbb{N}_i \backslash \{i\} \\ \sum_{l \in \mathbb{V} \backslash \{i\}} B_l K_l - \sum_{l \in \mathbb{N}_i \backslash \{i\}} W^{\mathrm{C}}_{il} & \text{if } j = i \\ -B_j K_j & \text{otherwise.} \end{cases}$$

9.3.1 LMI Formulation for Parameter Design

Let $\tilde{G}$ be the (input-to-state) transfer function matrix of (9.9). As an application of the small-gain theorem [84, Chapter 5.4], the feedback interconnection of the estimation component (9.9) and the control component (9.12) is L_2-stable if it holds that $\|\tilde{G}\|_{H_\infty} \|\tilde{D}\|_2 < 1$.

Adopting the bounded real lemma for discrete-time LTI systems [85,86], we establish the following equivalence:

$$\|\tilde{G}\|_{H_\infty} < \gamma \Leftrightarrow \begin{pmatrix} -X & X\tilde{A} & I_{nN} & 0_{nN} \\ \tilde{A}^T X & -X & 0_{nN} & X \\ I_{nN} & 0_{nN} & -\gamma I_{nN} & 0_{nN} \\ 0_{nN} & X & 0_{nN} & -\gamma I_{nN} \end{pmatrix} \prec 0 \tag{9.13}$$

where γ is a positive real number and $X \in \mathbb{R}^{nN \times nN}$ is a symmetric and positive-definite matrix. In the following lemma, we provide a sufficient condition under which a solution X, γ exists for (9.13).

Lemma 2. *Suppose that there are* $\left\{W_{ij}^E\right\}_{i,j\in\mathbb{V}}$, $\{L_i\}_{i\in\mathbb{V}}$ *for which* $\tilde{A}$ *given in* (9.10) *is Schur stable. Then, a solution* $X = X^T \succ 0, \gamma > 0$ *exists for* (9.13).

Proof. Based on [87, Lemma 5.1 (iii)] and by applying the Schur complement, (9.13) can be equivalently expressed as

$$X \prec \gamma^2 I_{nN} \tag{9.14a}$$

$$\tilde{A}^T X \tilde{A} - X + I_{nN} - \tilde{A}^T X \left(X - \gamma^2 I_{nN}\right)^{-1} X \tilde{A} \prec 0 \tag{9.14b}$$

Note that the last term in (9.14b) goes to zero as γ tends to infinity. Since $\tilde{A}$ is assumed to be Schur stable, there is $X = X^T \succ 0$ satisfying $\tilde{A}^T X \tilde{A} - X \prec -I_{nN}$. Consequently, the same X satisfies (9.14) for sufficiently large γ. This completes the proof. □

Remark 2. *The results of* [76,83,75] *from the distributed estimation literature address the existence of the parameters* $\left\{W_{ij}^E\right\}_{i,j\in\mathbb{V}}$, $\{L_i\}_{i\in\mathbb{V}}$ *for which* $\tilde{A}$ *is Schur stable when the system* (9.1) *is* *jointly observable and the graph $\mathcal{G}$ is strongly connected. The result of*[76,83] *is based on state augmentation ideas and that of*[75] *leverages the structure in the system model* (9.1) *and the graph connectivity. Therefore, in conjunction with the results from those references, Lemma 2 implies that the joint observability of* (9.1) *and the strong connectivity of $\mathcal{G}$ are sufficient for* (9.13) *to have a solution.*

Based on Lemma 2, in the following theorem, we address Problem 2. Given $\gamma_2 > 0$, define

$$\mathbb{K}_{\gamma_2} \quad = \quad \{\{K_i\}_{i\in\mathbb{V}} \,|\, A \;+\; \textstyle\sum_{i\in\mathbb{V}} B_iK_i \text{ is Schur stable}, \min_{\{W_{ij}^C\}_{i,j\in\mathbb{V}}} \|\tilde{D}\|_2 \quad < \quad \gamma_2\}.$$

Theorem 5. *Suppose given parameters $\{W_{ij}^E\}_{i,j\in\mathbb{V}}$, $\{L_i\}_{i\in\mathbb{V}}$ of the estimation component $\tilde{G}$ satisfy $\|\tilde{G}\|_{H_\infty} < \gamma$, e.g.,* (9.13) *has a solution with the same γ. Assuming that $\mathbb{K}_{\gamma^{-1}}$ is nonempty, for any state feedback gains $\{K_i\}_{i\in\mathbb{V}}$ belonging to $\mathbb{K}_{\gamma^{-1}}$ there is $\{W_{ij}^C\}_{i,j\in\mathbb{V}}$ such that the control inputs determined by* (9.3) *and* (9.4) *stabilize the system* (9.1)*.*

Proof. Since, under the assumptions on $\|\tilde{G}\|_{H_\infty}$ and $\|\tilde{D}\|_2$, the small-gain theorem holds, we have that $\lim_{k\to\infty} \|\tilde{x}(k)\|_2 = 0$. By applying (9.3) to (9.1a), we obtain

$$x(k+1) = (A + \textstyle\sum_{i\in\mathbb{V}} B_iK_i)x(k) - \sum_{i\in\mathbb{V}} B_iK_i\tilde{x}_i(k).$$

Since $A + \sum_{i\in\mathbb{V}} B_iK_i$ is Schur stable and the last term converges to zero as k tends to infinity, the state $x(k)$ converges to zero. This completes the proof. $\square$

Note that when γ in the statement of Theorem 5 is too large, $\mathbb{K}_{\gamma^{-1}}$ would be an empty set. In other words, when the H_∞-norm of the estimation component is too large, there is no control gain that stabilizes (9.1) while satisfying the inequality $\min_{\{W_{ij}^C\}_{i,j\in\mathbb{V}}} \|\tilde{D}\|_2 < \gamma^{-1}$ for the small-gain theorem to hold. In §9.3.2, we address the effect of m-round linear consensus on the state feedback design. In particular, we show that with more frequent information exchange (m large), the agents have more options to select state feedback gains that stabilize (9.1) and satisfy the inequality condition for the small-gain theorem to hold.

9.3.2 Effect of Linear Consensus on Stability

Suppose that the agents are allowed to fuse their state estimates using m-round linear consensus (9.5) with a consensus matrix P. Without loss of generality, we assume that m is an even number and $\mathcal{G}$ is undirected (and we select P to be symmetric). Define $\tilde{x}'(k) = Q^{m/2}\tilde{x}(k)$ and $\tilde{v}'(k) = Q^{m/2}\tilde{v}(k)$, where $Q = P \otimes I_n$.[§] By using (9.1), (9.3)-(9.5) and following similar steps to obtain (9.9) and (9.12) in §9.3, we can derive the state equations for the control and estimation components as follows:

$$\tilde{v}'(k) = Q^{m/2}\tilde{D}Q^{m/2}\tilde{x}'(k) \tag{9.15}$$

$$\tilde{x}'(k+1) = Q^{m/2}\tilde{A}Q^{m/2}\tilde{x}'(k) + \tilde{v}'(k) \tag{9.16}$$

We refine the definition of $\mathbb{K}_{\gamma_2}$ as follows: Given $\gamma_2 > 0$,

$$\mathbb{K}_{\gamma_2,m} = \{\{K_i\}_{i\in\mathbb{V}} \mid A+\sum_{i\in\mathbb{V}} B_i K_i \text{ is Schur stable}, \min_{\{W_{ij}^C\}_{i,j\in\mathbb{V}}} \|Q^{m/2}\tilde{D}Q^{m/2}\|_2 < \gamma_2\}.$$

We extend Theorem 5 to the case where the agents are allowed to use m-rounds of linear consensus.

[§]For odd m, we let $\tilde{x}'(k) = Q^{\lfloor m/2 \rfloor}\tilde{x}(k)$ and $\tilde{v}'(k) = Q^{\lfloor m/2 \rfloor}\tilde{v}(k)$.

$$M_1 = \begin{pmatrix} -X^* & X^*\left(A - \sum_{i\in\mathbb{V}} L_i^* C_i\right) & I_n & 0_n \\ \left(A - \sum_{i\in\mathbb{V}} L_i^* C_i\right)^T X^* & -X^* & 0_n & X^* \\ I_n & 0_n & -\gamma^* I_n & 0_n \\ 0_n & X^* & 0_n & -\gamma^* I_n \end{pmatrix},\ M_2 = \begin{pmatrix} -X^* & 0_n & I_n & 0_n \\ 0_n & -X^* & 0_n & X^* \\ I_n & 0_n & -\gamma^* I_n & 0_n \\ 0_n & X^* & 0_n & -\gamma^* I_n \end{pmatrix} \tag{9.17}$$

Theorem 6. *For any given $\gamma_2 > 0$, there is $m^* \geq 0$ for which $\mathbb{K}_{\gamma_2,m}$ is non-empty for $m \geq m^*$. For sufficiently large m, we can design parameters $\{W_{ij}\}_{i,j\in\mathbb{V}}$, $\{L_i\}_{i\in\mathbb{V}}$ for which the LTI system* (9.1) *is stable with any $\{K_i\}_{i\in\mathbb{V}}$ belonging to $\mathbb{K}_{\gamma^{-1},m}$, where γ is the H_∞-norm of* (9.16).

Proof. Notice that since $\tilde{D}(1_N \otimes I_n) = 0$ holds, $\|Q^{m/2}\tilde{D}Q^{m/2}\|_2$ converges to zero as m tends to infinity. Also we note that since Q is a stochastic matrix, $\|Q^{m_1/2}\tilde{D}Q^{m_1/2}\|_2 \leq \|Q^{m_2/2}\tilde{D}Q^{m_2/2}\|_2$ if $m_1 \geq m_2$. Hence, given fixed $\gamma_2 > 0$, we can find m^* for which $\mathbb{K}_{\gamma_2,m}$ is non-empty for all $m \geq m^*$. It remains to show that for sufficiently large m, we can find $\{W_{ij}^{\mathrm{E}}\}_{i,j\in\mathbb{V}}$, $\{L_i\}_{i\in\mathbb{V}}$ for which $\|\tilde{G}'\|_{H_\infty}\|Q^{m/2}\tilde{D}Q^{m/2}\|_2 < 1$ holds, where $\tilde{G}'$ is the (input-to-state) transfer function of (9.16).

Since the system (9.1) is jointly observable, we can find $\{L_i^*\}_{i\in\mathbb{V}}$ such that $A - \sum_{i\in\mathbb{V}} L_i^* C_i$ is Schur stable. Consider the matrix M_1 defined in (9.17). Suppose that we select the matrices $\{L_i^*\}_{i\in\mathbb{V}}$ and a symmetric positive-definite matrix $X^* \in \mathbb{R}^{n\times n}$ that satisfy $M_1 \prec 0$ with smallest $\gamma^* > 0$.[¶] Also we can verify that

$$\frac{1}{N}11^T \otimes M_1 + \left(I_{nN} - \frac{1}{N}11^T\right) \otimes M_2 \prec 0 \tag{9.18}$$

where M_1 and M_2 are defined in (9.17).

[¶]By similar arguments as in Lemma 2 such matrices exist for some γ^*.

Define $W_{ij}^{\mathrm{E}} = P_{ij}A$ and $L_i = NL_i^*$ and express (9.16) as

$$\tilde{x}'(k+1) = (P^{m+1} \otimes A - Q^{m/2}\bar{L}\bar{C}Q^{m/2})\tilde{x}'(k) + \tilde{v}'(k) \tag{9.19}$$

with $\bar{L} = \mathrm{diag}(NL_1^*, \cdots, NL_N^*)$, $\bar{C} = \mathrm{diag}(C_1, \cdots, C_N)$. Using the same LMI formulation as in (9.13), but with $\tilde{A} = P^{m+1} \otimes A - Q^{m/2}\bar{L}\bar{C}Q^{m/2}$ and $X = I_N \otimes X^*$, we have that

$$\|\tilde{G}'\|_{H_\infty} < \gamma^* \Leftrightarrow \begin{pmatrix} -X & X\tilde{A} & I_{nN} & 0_{nN} \\ \tilde{A}^T X & -X & 0_{nN} & X \\ I_{nN} & 0_{nN} & -\gamma^* I_{nN} & 0_{nN} \\ 0_{nN} & X & 0_{nN} & -\gamma^* I_{nN} \end{pmatrix} \prec 0. \tag{9.20}$$

Since $\lim_{m\to\infty} P^{m/2} = \lim_{m\to\infty} P^{m+1} = \frac{1}{N}1_N 1_N^T$, when m tends to infinity, the matrices in (9.18) and (9.20) are identical up to permutation. With the same choice of X^*, γ^* as in (9.18), the LMI (9.20) holds for sufficiently large m. Therefore, the H_∞-norm of (9.19) approaches γ^*, which is the H_∞-norm of the centralized estimator, as a large number of rounds of linear consensus is allowed. In conjunction with the fact that $\lim_{m\to\infty} \|Q^{m/2}\tilde{D}Q^{m/2}\|_2 = 0$, we conclude that when m is sufficiently large, we can compute $\{W_{ij}\}_{i,j\in\mathbb{V}}$, $\{L_i\}_{i\in\mathbb{V}}$ for which $\|\tilde{G}'\|_{H_\infty}\|Q^{m/2}\tilde{D}Q^{m/2}\|_2 < 1$ holds for any $\{K_i\}_{i\in\mathbb{V}}$ belonging to $\mathbb{K}_{(\gamma^*)^{-1},m}$. $\square$

Theorem 6 implies that when sufficiently frequent information exchange is allowed, $\mathbb{K}_{\gamma^{-1},m}$ will be nonempty and we can always find gains $\{K_i\}_{i\in\mathbb{V}}$ to stabilize the system. Furthermore, by increasing m we can design the parameters $\{W_{ij}\}_{i,j\in\mathbb{V}}$, $\{L_i\}_{i\in\mathbb{V}}$ to allow the agents to adopt arbitrarily large state feedback gains $\{K_i\}_{i\in\mathbb{V}}$.

9.3.3 Extension

Suppose that (9.3) consists of linear and nonlinear parts:

$$u_i(k) = K_i \hat{x}_i(k) + \mu_i(\hat{x}_i(k)). \tag{9.21}$$

Such state feedback can be applied, for instance, to maneuver multiple vehicles as in Example 2. The linear part $K_i \hat{x}_i(k)$ can be designed to maintain a desired formation and the nonlinear part $\mu_i(\hat{x}_i(k))$ would be used, whenever necessary, for the vehicles to keep a safe distance from obstacles nearby. In this case, we can perform the stability analysis by representing (9.4) with the following two components:

$$\tilde{v}_i(k) \;=\; \sum_{j \in \mathbb{V}} B_j(\mu_j(\hat{x}_i(k)) - \mu_j(\hat{x}_j(k))) - \sum_{j \in \mathbb{N}_i} B_j(\mu_j(\hat{x}_i(k)) - \mu_j(\hat{x}_j(k))) \tag{9.22}$$

where we assume that there is a constant γ_C for which $\|\tilde{v}(k)\|_2 \leq \gamma_C \|\tilde{x}(k)\|_2$ holds and

$$\tilde{x}_i(k+1) = (A - L_i C_i)\,\tilde{x}_i(k) + \sum_{j \in \mathbb{V}} B_j K_j(\tilde{x}_i(k) - \tilde{x}_j(k)) - \sum_{j \in \mathbb{N}_i} W_{ij}^{\mathrm{E}}(\tilde{x}_i(k) - \tilde{x}_j(k)) + \tilde{v}_i(k). \tag{9.23}$$

Note that, unlike the approach discussed in §9.3, we substitute $W_{ij}(\hat{x}_j(k) - \hat{x}_i(k))$ in (9.4) with the following nonlinear function: $W_{ij}(\hat{x}_j(k), \hat{x}_i(k)) = W_{ij}^{\mathrm{E}}(\hat{x}_j(k) - \hat{x}_i(k)) + B_j(\mu_j(\hat{x}_i(k)) - \mu_j(\hat{x}_j(k)))$. Also, the design of the parameters $\{W_{ij}^{\mathrm{E}}\}_{i,j \in \mathbb{V}}$, $\{L_i\}_{i \in \mathbb{V}}$ for the estimation component will depend on the *linear part* $\{K_i\}_{i \in \mathbb{V}}$ of (9.21). In this case, the small-gain theorem can be used to establish the stability results as in Theorems 5 and 6 if it holds that $\|\tilde{G}\|_{H_\infty} < \gamma_C^{-1}$, where $\tilde{G}$ is the transfer function of (9.23).

9.4 Simulations

9.4.1 Parameter Design

Recall that the control gains $\{K_i\}_{i\in\mathbb{V}}$ are assumed to be given, and the key design parameters in our framework are P, $\left\{W_{ij}^E\right\}_{i,j\in\mathbb{V}}$, $\left\{W_{ij}^C\right\}_{i,j\in\mathbb{V}}$, and $\{L_i\}_{i\in\mathbb{V}}$. We select P to be a stochastic matrix with smallest second eigenvalue and conforming with $\mathcal{G}$ to allow the agents to fuse the estimates as fast as possible. For simplicity, we choose $W_{ij}^E = P_{ij}A$ as motivated by the approach of[76].

We solve the following two optimization formulations to compute $\{L_i\}_{i\in\mathbb{V}}$ and $\{W_{ij}^C\}_{i,j\in\mathbb{V}}$ that minimize the H_∞-norm of (9.16) and the 2-norm of (9.15), respectively.

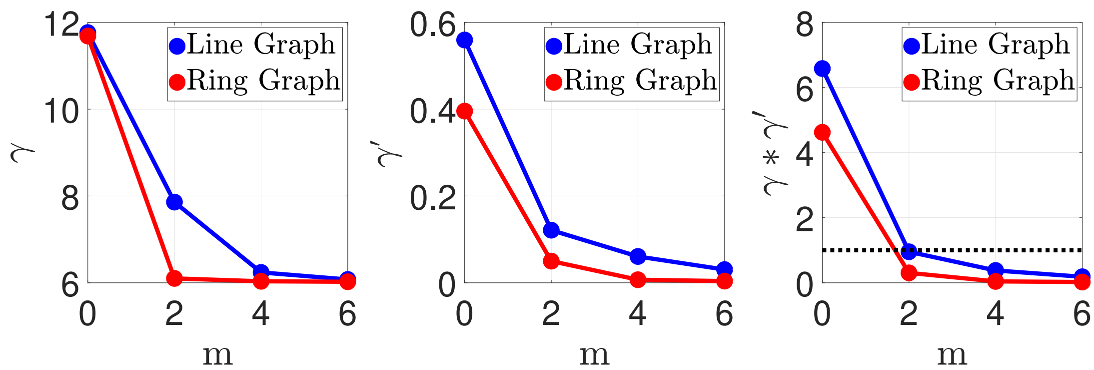
$$\begin{aligned} &\text{minimize}_{\gamma, X, \{L_i\}_{i\in\mathbb{V}}} \gamma \\ &\text{subject to (9.20)} \end{aligned} \tag{9.24}$$

$$\begin{aligned} &\text{minimize}_{\gamma', \left\{W_{ij}^C\right\}_{i,j\in\mathbb{V}}} \gamma' \\ &\text{subject to} \begin{pmatrix} \gamma' I_{nN} & Q^{m/2}\tilde{D}Q^{m/2} \\ \left(Q^{m/2}\tilde{D}Q^{m/2}\right)^T & \gamma' I_{nN} \end{pmatrix} \succ 0 \end{aligned} \tag{9.25}$$

where $\gamma > 0, X = X^T \succ 0$, and $Q = P \otimes I_n$. Note that (9.24) is a non-convex optimization.

9.4.2 Simulation Results

Consider Example 2 with $N = 4$ and two types of communication graphs: undirected line and ring graphs. For each graph, we compute the parameters of (9.4) by solving (9.24) and



(a) (b) (c)

Figure 9.2: Plots of the optimal value of (a) γ for the estimation component design (9.24), (b) γ' for the control component design (9.25), and (c) the product $\gamma * \gamma'$, where the critical value of 1 is drawn as a dotted line.

(9.25) for $m = 0, 2, 4, 6$, where the state feedback gain, motivated by the centralized LQR controller, is given by

$$K_i = e_i^T \otimes \begin{pmatrix} -0.192 & 0 & -0.284 & 0 \\ 0 & -0.192 & 0 & -0.284 \end{pmatrix} \tag{9.26}$$

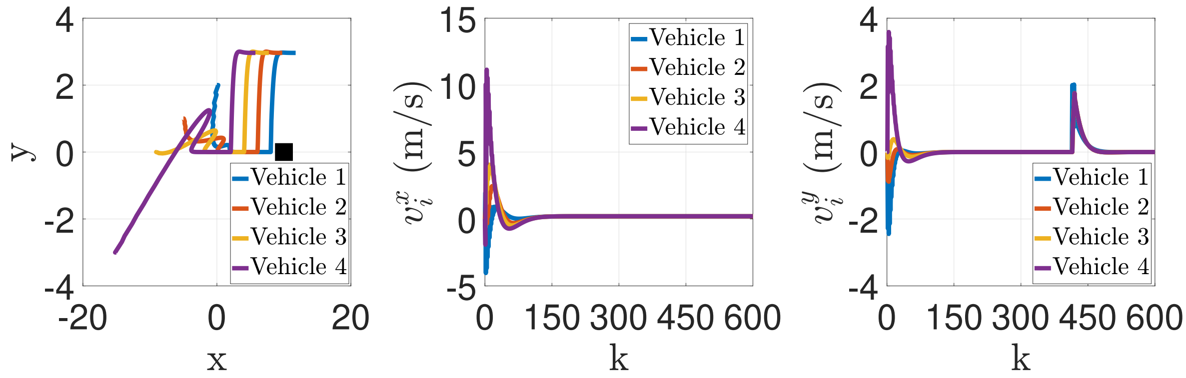
The vector e_i is a canonical basis of dimension 4.

Fig. 9.2 depicts the minimal costs obtained in the optimization (9.24) and (9.25) with the increasing number of consensus rounds m over line and ring graphs. We can observe that as m increases, both norms of the estimation and control components decrease. In conjunction with Theorem 6, this suggests that when the agents are allowed to communicate more frequently (when m is large), they would have more flexibility in selecting the state feedback gains without re-computing the parameters of the estimation strategy. We can also observe that since the ring graph has one additional edge between agents 1 and 4, which gives the agents more paths over which to share state estimates, the consensus on the state estimates takes place faster over the ring graph. Consequently, the ring graph attains smaller norms for both components than does the line graph. Fig. 9.2(c) shows that $m \geq 2$ consensus rounds are needed to satisfy the small-gain theorem for the control gain we select in (9.26).

With the same example, we next consider a formation control scenario: 4 vehicles move along the x-axis at the speed of $0.2\ m/s$ while maintaining a line formation where each vehicle is spaced a distance of $2\ m$ apart. The vehicles communicate over the line graph and vehicle 1 acts as a leader in the formation. When the leader detects an obstacle, it moves to the side to avoid collision while the rest of the vehicles stay in the formation. We adopt (9.21) to design nonlinear control with a different notation. We define $u_i(k) = u_i^{\text{linear}}(k) + \mu_i\,(\hat{x}_i(k))$ with $u_i^{\text{linear}}(k) = (u_i^{\text{linear},x}(k), u_i^{\text{linear},y}(k))$ as

$$
\begin{aligned}
u_1^{\text{linear},x}(k) &= -\textstyle\sum_{j\in\mathbb{V}}(k_p(\hat{p}_1^x(k) - \hat{p}_j^x(k) + d_{1j}) + k_v(\hat{v}_1^x(k) - \hat{v}_j^x(k))) + k_v(\hat{v}_1^x(k) - 0.2))\\
u_1^{\text{linear},y}(k) &= -\textstyle\sum_{j\in\mathbb{V}}(k_p(\hat{p}_1^y(k) - \hat{p}_j^y(k)) + k_v(\hat{v}_1^y(k) - \hat{v}_j^y(k))) + k_v(\hat{v}_1^y(k)))\\
u_i^{\text{linear},x}(k) &= -\textstyle\sum_{j\in\mathbb{V}}(k_p(\hat{p}_i^x(k) - \hat{p}_j^x(k) + d_{ij}) + k_v(\hat{v}_i^x(k) - \hat{v}_j^x(k)))\\
u_i^{\text{linear},y}(k) &= -\textstyle\sum_{j\in\mathbb{V}}(k_p(\hat{p}_i^y(k) - \hat{p}_j^y(k)) + k_v(\hat{v}_i^y(k) - \hat{v}_j^y(k)))
\end{aligned}
$$

for $i \in \{2, 3, 4\}$, where $d_{ij} = 2(i-j)$. We select $k_p = 0.16$ and $k_v = 0.3$. This was motivated by the approach in [88] to achieve the desired formation control.



(a) (b) (c)

Figure 9.3: Formation control with $m = 4$ rounds of linear consensus showing (a) trajectories in the xy-plane, (b) v_i^x, and (c) v_i^y of all 4 vehicles as they maintain a line formation while avoiding a stationary obstacle at $(10, 0)$.

The nonlinear part $\mu_i = (\mu_i^x, \mu_i^y)$ is defined below to achieve obstacle avoidance.

$$\mu_1^x(\hat{x}(k)) = 0, \tag{9.27a}$$

$$\mu_1^y(\hat{x}(k)) = \begin{cases} \frac{\xi}{|\xi|} & \text{if } \|\hat{p}_1(k) - p_o(k)\| \leq 1 \\ -\xi + 2\frac{\xi}{|\xi|} & \text{if } 1 < \|\hat{p}_1(k) - p_o(k)\| \leq 2 \\ 0 & \text{otherwise} \end{cases} \tag{9.27b}$$

where $\xi = \hat{p}_1^y(k) - p_o^y(k)$ and $\mu_i(\hat{x}(k)) = (0, 0)$, $i \in \{2, 3, 4\}$. We assume that the vehicles can measure the location $p_o = (p_o^x, p_o^y)$ of the obstacle. Each vehicle estimates the state of the system based on (9.23), which includes the linear part of the state feedback in computing its parameters.

Fig. 9.3 illustrates the simulation results on the formation control. We observe that under the control $u_i(k)$, the vehicles achieve the desired line formation while maintaining the pre-assigned velocity of $0.2\ m/s$ along the x-axis, as illustrated in Figs. 9.3(a),9.3(b). When vehicle 1 approaches the stationary obstacle, it uses (9.27) to avoid colliding with it; this is illustrated in Figs. 9.3(a),9.3(c).

9.5 Conclusions

We investigated the design of a network of agents for the estimation and control of LTI systems. Since the separation principle does not hold, the estimation and control strategies need to be jointly designed, which involves finding a solution to a large-scale optimization. This could be a disadvantage if the agents need to change their control strategies without re-solving the optimization. We have presented LMI formulations to characterize the conditions under which the design of estimation and control can be decoupled, and shown how the frequency of information exchange between agents affects the establishment of the conditions. Our simulation results illustrate an application of our framework to multi-vehicle platooning.

References

[1] M.E.J. Newman. *Networks: An Introduction*. Oxford University Press, 2010.

[2] J. Baillieul and P.J. Antsaklis. Control and communication challenges in networked real-time systems. *Proceedings of the IEEE*, 95(1):9-28, 2007.

[3] M. Wooldridge. *An introduction to multiagent systems*. John Wiley & Sons, 2009.

[4] S.J. Russell and P. Norvig. *Artificial Intelligence: A Modern Approach*. Prentice Hall, Englewood Cliffs, NJ, 2003.

[5] N.E. Leonard. Multi-agent system dynamics: Bifurcation and behavior of animal groups. *IFAC Annual Reviews in Control*, 38(2):171-183, 2014.

[6] K. Sycara. Multiagent systems. *AI Magazine*, 19(2):79-92, 1998.

[7] N. Vlassis. A concise introduction to multiagent systems and distributed artificial intelligence. *Synthesis Lectures on Artificial Intelligence and Machine Learning*, 1(1):1-71, 2007.

[8] N.E. Leonard and A. Olshevsky. Cooperative learning in multiagent systems from intermittent measurements. *SIAM Journal on Control and Optimization*, 53(1):1-29, 2015.

[9] P. Landgren, V. Srivastava, and N.E. Leonard. On distributed cooperative decision-making in multiarmed bandits. *European Control Conference*, pages 243-248. IEEE, 2016.

[10] W. Ren, R.W. Beard, and E.M. Atkins. A survey of consensus problems in multi-agent coordination. *American Control Conference*, pages 1859-1864. IEEE, 2005.

[11] A.J. Savas, V. Srivastava, and N.E. Leonard. On distributed linear filtering with noisy communication. *American Control Conference*, pages 2699–2704. IEEE, 2017.

[12] M.J. Osborne and A. Rubinstein. *A Course in Game Theory*. MIT Press, Cambridge, MA, 1994.

[13] P. Yang, R. A. Freeman, and K. M. Lynch. Multi-agent coordination by decentralized estimation and control. *IEEE Transactions on Automatic Control*, 53(11):2480–2496, 2008.

[14] N.E. Leonard and E. Fiorelli. Virtual leaders, artificial potentials and coordinated control of groups. *IEEE Conference on Decision and Control*, pages 2968-2973. IEEE, 2001.

[15] J. Lin, A.S. Morse, and B.D.O. Anderson. The multi-agent rendezvous problem. *IEEE Conference on Decision and Control*, pages 1508-1513. IEEE, 2003.

[16] P.B. gen. Dohmann and S. Hirche. Distributed control for cooperative manipulation with event-triggered communication. *IEEE Transactions on Robotics*, 36(4):1038-1052, 2020.

[17] S.K. Mitter. Filtering and stochastic control: a historical perspective. *IEEE Control Systems Magazine*, 16(3):67-76, 1996.

[18] D.P. Joseph and T.J. Tou. On linear control theory. *Transactions of the American Institute of Electrical Engineers, Part II: Applications and Industry*, 80(4):193-196, 1961.

[19] J. Potter. A guidance-navigation separation theorem. *Astrodynamics Guidance and Control Conference*, page 653, 1964.

[20] H. Van de Water and J. Willems. The certainty equivalence property in stochastic control theory. *IEEE Transactions on Automatic Control*, 26(5):1080-1087, 1981.

[21] W.M. Wonham. On the separation theorem of stochastic control. *SIAM Journal on Control*, 6(2):312-326, 1968.

[22] D. Wu, J. Wu, and S. Chen. Separation principle for networked control systems with multiple-packet transmission. *IET Control Theory and Applications*, 5(3):507-513, 2011.

[23] A. Rantzer. A separation principle for distributed control. *IEEE Conference on Decision and Control*, pages 3609-3613. IEEE, 2006.

[24] A.J. Savas, S. Park, H.V. Poor, and N.E. Leonard. A separation principle in the design of distributed control for LTI systems. In Preparation.

[25] S. Joshi and S. Boyd. Sensor selection via convex optimization. *IEEE Transactions on Signal Processing*, 57(2):451-462, 2009.

[26] J.L. Rogers. A parallel approach to optimum actuator selection with a genetic algorithm. *AIAA Guidance, Navigation, and Control Conference*, pages 14-17, 2000.

[27] S. Kondoh, C. Yatomi, and K. Inoue. The positioning of sensors and actuators in the vibration control of flexible systems. *JSME International Journal, Series III*, 33(2):145-152, 1990.

[28] K. Hiramoto, H. Doki, and G. Obinata. Optimal sensor/actuator placement for active vibration control using explicit solution of algebraic Riccati equation. *Journal of Sound and Vibration*, 229(5):1057-1075, 2000.

[29] K.K. Chen and C.W. Rowley. H_2 optimal actuator and sensor placement in the linearised complex Ginzburg-Landau system. *Journal of Fluid Mechanics*, 681:241-260, 2011.

[30] N.K. Dhingra, M.R. Jovanović, and Z.Q. Luo. An ADMM algorithm for optimal sensor and actuator selection. *IEEE Conference on Decision and Control*, pages 4039-4044. IEEE, 2014.

[31] F. Lin, M. Fardad, and M.R. Jovanović. Algorithms for leader selection in stochastically forced consensus networks. *IEEE Transactions on Automatic Control*, 59(7):1789-1802, 2014.

[32] K. Fitch and N.E. Leonard. Information centrality and optimal leader selection in noisy networks. *IEEE Conference on Decision and Control*, pages 7510-7515. IEEE, 2013.

[33] L. Freeman. A set of measures of centrality based upon betweenness. *Sociometry* 40(1): 35-41, 1977.

[34] C. Tallberg. Bayesian and other statistical approaches for analyzing network blockstructures. PhD Thesis, 2003.

[35] D. Ghoshdastidar and U. von Luxburg. Practical methods for graph two-sample testing. NeurIPS 2018, Montréal, Canada.

[36] T.J. van Woudenberg, B. Simoski, E.F. de Mello Araújo, K.E. Bevelander, W.J. Burk, C.R. Smit, L. Buijs, M. Klein, and M. Buijzen. Identifying influence agents that promote physical activity through the simulation of social network interventions: agent-based modeling study. *J. Med. Internet. Res.* 21(8), 2019.

[37] B. Efron and R.J. Tibshirani. *An Introduction to the Bootstrap*. CRC Press, May 15, 1994.

[38] A. Bavelas. Communication patterns in task-oriented groups. *J. Acoust. Soc. Am.* 22(6):725-730, 1950.

[39] M.E.J. Newman. The mathematics of networks.

[40] K. Stephenson and M. Zelen. Rethinking centrality: Methods and examples. *Social Netw.*, vol. 11, 1-37, 1989.

[41] K. Fitch and N.E. Leonard. Joint centrality distinguishes optimal leaders in noisy networks. *IEEE Transactions on Control of Network Systems*, 3(4):366-378, 2016.

[42] P. Erdos and A. Renyi. On random graphs. *Publicationes Mathematicae*, 6:290-297, 1959.

[43] E.N. Gilbert. Random graphs. *Annals of Mathematical Statistics*, 30(4):1141-1144, 1959.

[44] F. Chung and L. Lu. *Complex Graphs and Networks*. Am. Math. Soc., Providence, 2006.

[45] R. Albert and A.L. Barabasi. Statistical mechanics of complex networks. *Reviews of Modern Physics*, 74(47):47-97, 2002.

[46] I. Rausch, P. Simoens, and Y. Khaluf. Adaptive foraging in dynamic environments using scale-free interaction networks. *Front. Robot. AI*, 7(86):1-14, 2020.

[47] D.J. Watts and S.H. Strogatz. Collective dynamics of 'small-world' networks. *Nature*, 393(6684):440-442, 1998.

[48] S.I. Roumeliotis and M.J. Mataric. "Small-world" networks of mobile robots. *AAAI/IAAI*, 1093, 2000.

[49] J.P. Hespanha. *Linear Systems Theory*. Princeton University Press, 2009.

[50] S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004.

[51] I. Matei and J. S. Baras. Consensus-based linear distributed filtering. *Automatica*, 48:1776–1782, 2012.

[52] S. Das and J. M. F. Moura. Consensus+Innovations Distributed Kalman Filter with Optimized Gains. *IEEE Transactions on Signal Processing*, arXiv:1605.06096v2, October 2016.

[53] R. Olfati-Saber. Distributed Kalman filter with embedded consensus filters. In *IEEE Conference on Decision and Control*, pages 8179–8184. IEEE, 2005.

[54] R. Olfati-Saber. Distributed Kalman filtering for sensor networks. In *IEEE Conference on Decision and Control*, pages 5492–5498. IEEE, 2007.

[55] U. A. Khan and J. M. F. Moura. Distributing the Kalman filter for large-scale systems. *IEEE Transactions on Signal Processing*, 56(10):4919–4935, 2008.

[56] R. Carli, A. Chiuso, L. Schenato, and S. Zampieri. Distributed Kalman filtering based on consensus strategies. *IEEE Journal on Selected Areas in Communications*, 26(4):622–633, 2008.

[57] L. Xiao, S. Boyd, and S-J. Kim. Distributed average consensus with least-mean-square deviation. *Journal of Parallel and Distributed Computing*, 67(1):33–46, 2007.

[58] F. Zhang and N. E. Leonard. Cooperative filters and control for cooperative exploration. *IEEE Transactions on Automatic Control*, 55(3):650–663, 2010.

[59] A. Jadbabaie, J. Lin, and A. S. Morse. Coordination of groups of mobile autonomous agents using nearest neighbor rules. *IEEE Transactions on Automatic Control*, 48(6):988–1001, 2003.

[60] R. Olfati-Saber, J. A. Fax, and R. M. Murray. Consensus and cooperation in networked multi-agent systems. *Proceedings of the IEEE*, 95(1):215–233, 2007.

[61] R. Olfati-Saber and J. S. Shamma. Consensus filters for sensor networks and distributed sensor fusion. In *IEEE Conference on Decision and Control*, pages 6698–6703. IEEE, 2005.

[62] D. P. Spanos and R. M. Murray. Distributed sensor fusion using dynamic consensus. In *IFAC World Congress*. Citeseer, 2005.

[63] A. Speranzon, C. Fischione, K. H. Johansson, and A. Sangiovanni-Vincentelli. A distributed minimum variance estimator for sensor networks. *IEEE Journal on Selected Areas in Communications*, 26(4):609–621, 2008.

[64] K. M. Lynch, I. B. Schwartz, P. Yang, and R. A. Freeman. Decentralized environmental modeling by mobile sensor networks. *IEEE Transactions on Robotics*, 24(3):710–724, 2008.

[65] G. F. Young, L. Scardovi, and N. E. Leonard. Robustness of noisy consensus dynamics with directed communication. In *American Control Conference*, pages 6312–6317, 2010.

[66] I. Poulakakis, G. F. Young, L. Scardovi, and N. E. Leonard. Information centrality and ordering of nodes for accuracy in noisy decision-making networks. *IEEE Transactions on Automatic Control*, 61(4):1040-1045, 2016.

[67] V. Srivastava and N. E. Leonard. Collective decision-making in ideal networks: The speed-accuracy trade-off. *IEEE Transactions on Control of Network Systems*, 1(1):121–132, 2014.

[68] M. H. DeGroot. Reaching a consensus. *Journal of the American Statistical Association*, 69(345):118–121, 1974.

[69] N. E. Friedkin. The problem of social control and coordination of complex systems in sociology: A look at the community cleavage problem. *IEEE Control Systems*, 35(3):40–51, 2015.

[70] K. J. Åström. *Introduction to stochastic control theory*, volume 70 of *Mathematics in Science and Engineering*. Academic Press, New York, 1970.

[71] F. Bullo, J. Cortés, and S. Martínez. *Distributed Control of Robotic Networks*. Applied Mathematics Series. Princeton University Press, 2009. Electronically available at http://coordinationbook.info.

[72] J. N. Tsitsiklis. *Problems in Decentralized Decision Making and Computation*. PhD thesis, Massachusetts Institute of Technology, November 1984.

[73] J. Gorski, F. Pfeuffer, and K. Klamroth. Biconvex sets and optimization with biconvex functions: A survey and extensions. *Mathematical Methods of Operations Research*, 66(3):373–407, 2007.

[74] D.P. Bertsekas. *Dynamic Programming and Optimal Control*. 3rd ed., Belmont, MA, USA: Athena Scientific, 2005, vol. 1.

[75] A. Mitra and S. Sundaram. Distributed observers for LTI systems. *IEEE Transactions on Automatic Control*, 63(11):3689–3704, 2018.

[76] S. Park and N. C. Martins. Design of distributed LTI observers for state omniscience. *IEEE Transactions on Automatic Control*, 62(2):561–576, 2017.

[77] L. Wang, D. Fullmer, F. Liu, and A. S. Morse. Distributed control of linear multi-channel systems: summary of results. *American Control Conference*, pages 4576–4581. IEEE, 2020.

[78] M. Razeghi-Jahromi and A. Seyedi. Stabilization of networked control systems with sparse observer-controller networks. *IEEE Transactions on Automatic Control*, 60(6):1686–1691, 2015.

[79] J. Xiong and W. Liu. H_∞ observer-based control for large-scale systems with sparse observer communication network. *Numerical Algebra, Control, and Optimization*, 10(3):331–343, 2020.

[80] T. Kim, D. Lee, and H. Shim. Decentralized design and plug-and-play distributed control for linear multi-channel systems. *arXiv 2011.09735*, 2021.

[81] F.C. Rego, Y. Pu, A. Alessandretti, A.P. Aguiar, A.M. Pascoal, and C.N. Jones. A distributed Luenberger observer for linear state feedback systems with quantized and rate-limited communications. *IEEE Transactions on Automatic Control*, 66(9):3922–3937, 2021.

[82] X. Zhang, K. Hengster-Movrić, M. Šebek, W. Desmet, and C. Faria. Distributed observer and controller design for spatially interconnected systems. *IEEE Transactions on Control Systems Technology*, 27(1):1–13, 2019.

[83] L. Wang and A. S. Morse. A distributed observer for a time-invariant linear system. *IEEE Transactions on Automatic Control*, 63(7):2123–2130, 2018.

[84] H.K. Khalil. *Nonlinear Systems*. 3rd ed., Upper Saddle River, NJ: Prentice-Hall, 2002.

[85] G. Pipeleers, B. Demeulenaere, J. Swevers, and L. Vandenberghe. Extended LMI characterizations for stability and performance of linear systems. *Systems & Control Letters*, 58:510–518, 2009.

[86] R.J. Caverly and J.R. Forbes. LMI properties and applications in systems, stability, and control theory. *arXiv:1903.08599v2*, 2019.

[87] P. Gahinet and P. Apkarian. A linear matrix inequality approach to H_∞ control. *International Journal of Robust and Nonlinear Control*, 4:421–448, 1994.

[88] Y. Zheng, S.E. Li, K. Li, and W. Ren. Platooning of connected vehicles with undirected topologies: Robustness analysis and distributed h-infinity controller synthesis. *IEEE Transactions on Intelligent Transportation Systems*, 19(5):1353–1364, 2018.

[89] A. Mahajan, N. C. Martins, M. C. Rotkowitz, and S. Yüksel. Information structures in optimal decentralized control. *IEEE Conference on Decision and Control*, pages 1291–1306. IEEE, 2012.

[90] H. S. Witsenhausen. A counterexample in stochastic optimum control. *SIAM J. Control*, 6(1):131–147, 1968.

[91] Y. C. Ho. Review of the Witsenhausen problem. *IEEE Conference on Decision and Control*, pages 1611–1613. IEEE, 2008.

[92] V. Ugrinovskii. Distributed robust filtering with H_∞ consensus of estimates. *Automatica*, 47:1–13, 2011.

[93] F. Gama, Q. Li, E. Tolstaya, A. Prorok, and A. Ribeiro. Synthesizing decentralized controllers with graph neural networks and imitation learning. *arXiv:2012.14906v4*, 2022.

[94] F. Gama and S. Sojoudi. Distributed linear-quadratic control with graph neural networks. *Signal Processing*, 196, 2022.