FORMATION CONTROL AND ROBUSTNESS ANALYSIS OF TIME-DELAYED AGENTS

A THESIS

Presented to the Department of Mechanical and Aerospace Engineering

California State University, Long Beach

In Partial Fulfillment

of the Requirements for the Degree

Master of Science in Mechanical Engineering

Committee Members:

Praveen Shankar, Ph.D. (Chair)
Oscar Morales-Ponce, Ph.D.
John F. Quindlen, Ph.D.

College Designee:

Hamid Rahai, Ph.D.

By Andres F. Rivera

B.S., 2015, City University of New York

January 2019

ABSTRACT

FORMATION CONTROL AND ROBUSTNESS ANALYSIS OF TIME-DELAYED AGENTS

By

Andres F. Rivera

January 2019

This thesis considers the formation control problem of a group of homogeneous non-holonomic agents in the presence of two kinds of delays, a time delay in the sensing feedback channel and a time delay in the agent communication network. The agents are assumed to communicate using a fixed and directed communication topology. The formation control problem is tackled using consensus protocols; this work proposes a new consensus protocol that allows for the existence of leader agents (agents that do not receive state information from its neighbors) within the formation, and the algebraic form of the distributed forcing function that solves the formation regulation problem. The time- delayed stability analysis of this formation is analyzed using the CTCR (Cluster Treatment of Characteristic Roots) method under the SDS (Spectral Delay Space) domain. Sufficient conditions for the stability of the time-delayed formation control system are presented. The methodology is implemented and validated with a numerical example evaluating the formation regulation and dynamic formation trajectory tracking capabilities of the scheme, along with a Monte Carlo experiment validating the time- delay robustness assessment. The results of the analysis of this methodology show that the sensing delay can drive the effective stability margin of the multi-agent system. We present an example where the communication delay margin is infinite for a finite range of sensing delays. The example emphasizes the need for the explicit consideration of sensing delays in the design of robust formations of agents and the methodology discussed in this thesis adequately addresses such considerations.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

1.1 Multi-Agent System Control

The area of cooperative control of multi-agent systems has attracted a lot of research in recent decades [1] due to the increased need for engineered systems capable of performing tasks of increasing complexity where a system composed of a single agent is unable to meet the performance characteristics. In this context, an *agent* represents an arbitrary individual dynamic system, and a *multi-agent system* (MAS) represents a collection of such agents, where their cooperation renders increased benefits as defined by their application. Compared to single-agent systems, MAS often achieve greater efficiency and increased capabilities when performing their tasks [1, 2]. In literature, we can find the following applications: A constellation of Earth-orbiting spacecraft flying in formation to perform synthetic aperture radar (SAR) measurements [3], a formation of Unmanned Aerial Vehicles (UAV) for military reconnaissance missions, or civilian applications such as vegetation growth tracking, topographic analysis, fire monitoring among others [4]. While the applications include widely different agents and objectives, the fundamental approaches to the coordination of multiple robots, aircraft or spacecraft are very similar: they all require the coordination of the system's agents to accomplish an objective.

The traditional approach to the design of multiple agent systems involves analyzing the dynamics of the individual agents and designing robust control laws that guarantee stability and meet a particular performance criteria, such as reference tracking or regulation. The process of coordinating the *agents* of the system is achieved by introducing a *centralized* controller, often called commander or supervisor, that collects the state infor-

---

[1] Search results in the scholar citation search engine Google Scholar for the keywords "multi-agent system" show more than 1 million results since the year 2000 and 800 thousand results since the year 2010.

mation of all the agents in the system and computes a set of commands that allow each individual agent to achieve their individual target [5]. This type of control architecture is adequate when the number of agents is small and the control computational effort is sufficiently small to be performed by a monolithic central entity, however, as the scale of the system increases, the centralized approach imposes large communication and computational resource requirements on the centralized computer, rendering the problem unfeasible.

A competing approach that does not suffer from the scalability constraints of the *centralized control* approach involves distributing the communication and decision-making load to each of the individual agents; this approach is called *distributed control* and has been largely motivated by the technological advancements in sensor, communication and processing resource miniaturization. It is important to note those individual agents are typically characterized by having limited communication and processing resources and the analysis of the distributed control problem often involves considering the effects of the resource restrictions.

The research community has produced a wide gamut of approaches toward achieving distributed cooperative control, including: model predictive [6], receding horizon [7], observer-based [8], and finally consensus-based techniques [1, 9, 10].

1.2 The Role of Consensus and Time Delays in Distributed MAS Control

Consensus has been identified as a critical concept to the study of cooperative systems, the primary goal of this section is to introduce the concept of consensus problems and their relevance to the control of MAS'.

In a strict sense, information consensus means reaching "agreement" by the agents in the system to a common value or variable that depends on the state of all agents. A *consensus protocol* is the rule that determines the interaction of the agents and its neighbors[10]. The existence of an information interchange network is implied in the definition of the

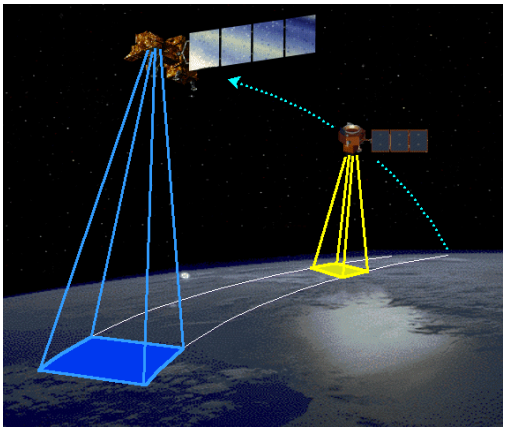


FIGURE 1.1. MAS example - trailing formation of Landsat-7 and EO-1 satellites [11].

consensus protocol. The agents of the system agree on key pieces of information enabling them to work together in a coordinated manner. The agreement variables could be, but are not necessarily, those associated with the physical state of the system, including position, velocity, attitude, and so on. In order to exchange information and take control decisions, the agents must be equipped with sensing and communication capabilities, and processing units to interpret and determine suitable control actions. Based on the information acquired by the sensors, information is exchanged in accordance to a consensus protocol. Intuitively, it can be inferred that the design of the consensus protocol drives the dynamic behavior of the overall system, and its analysis and implementation are fundamental parts of the control problem.

We present an illustrative example of a system of n agents, with moment of inertia I, rotating about a fixed axis trying to synchronize their angular position θ_i $\{i \in \mathbb{N}_n^*, i \leq n\}$ or heading in 2-d space. The first derivative of the heading, also called angular velocity, of the agent is represented by ω. The state of each agent is $\vec{x}_i = [\theta, \omega]$, this system is

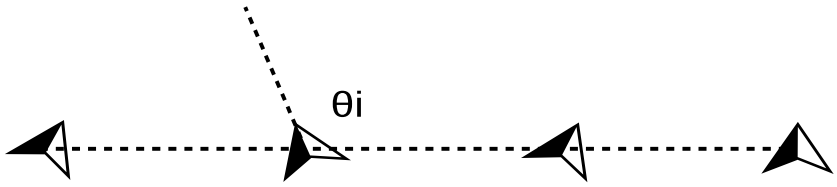


FIGURE 1.2. Consensus MAS control: synchronizing heading angle.

governed by second-order linear system (1.1).

$$\dot{\vec{x}}_i = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \vec{x}_i + \begin{bmatrix} 0 \\ \frac{1}{I} \end{bmatrix} u_i \tag{1.1}$$

Assuming all the agents communicate a single variable θ_i, a linear consensus protocol where each agent i compares its current angular position against the angular position of the other $n-1$ agents in the system can be defined as follows:

$$u_i = -\theta_i + \sum_{j \neq i}^{n} \theta_j \tag{1.2}$$

The consensus protocol equation (1.2) will determine the time evolution of the angular position of the agents. If the consensus problem is solved, the state of the individual agents will be driven to a common value. In this simplified example, the concept of information flow was introduced under the assumption that all agents communicate with one another. In reality, most systems don't behave this way and a mathematical representation of the communication topology must be introduced.

A particularly powerful representation of the information flow can be found in algebraic graph theory. The communication topology is described as a *directed graph* $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ where each vertex v_i in set $\mathcal{V}$ corresponds to an agent in the system, and each edge $e_{ij} = (v_i, v_j) \in \mathcal{E}$ corresponds to a directed communication link between agent$i$ and j. A

graph can be concisely represented by its adjacency matrix, defined in Equation 1.3.

$$\mathcal{A} = [a_{ij}] \in \mathbb{R}_{n\times n} \{a_{ij} = 1 \mid (v_i, v_j) \exists \mathcal{E}, a_{ij} = 0 \mid (v_i, v_j) \nexists \mathcal{E}\} \tag{1.3}$$

The framework introduced in Chapter 2 allows us to describe the MAS by means of linear operators and matrices.

Additionally, a fundamental consideration in the analysis of networked systems is the corrupting effect of time delays. Time delays typically degrade the performance of the control system and have a direct influence on its stability. In networked conditions, time delays are determined by the complexity of the communication topology, the congestion of the network, and the characteristics of the communication protocol for example, Bluetooth RFCOMM, Ethernet, RS-232, and SpaceWire, among others [12]. Another source of delays is associated with the sensing and processing capabilities of the agents. Depending on the agents' processing and communication network capabilities, designing robust consensus protocols with explicit analysis of the influence of time delays is a fundamental task in the control problem.

1.3 Problem Statement

This thesis describes the analysis and design process of consensus protocols for the formation control problem, with particular emphasis in the exact analysis and influence of communication and input/processing time delays, and its implementation on a robotic platform composed of a homogeneous set of differential drive robots. This robotic platform can be modeled by the dynamics of a unicycle, where the system can be described by the following state $z = [x, y, \theta]$, with inputs corresponding to the linear velocity and angular velocity in the out-of-plane direction $u = [v, \omega]$. This is described geometrically in Figure 1.3.

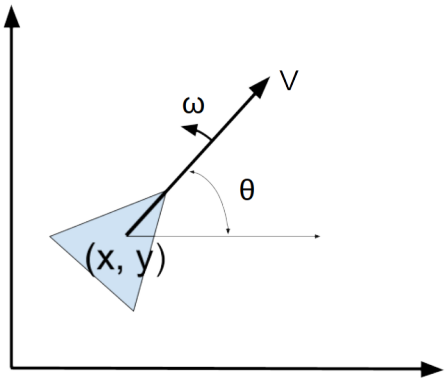


FIGURE 1.3. Unicycle at position x,y heading θ, linear velocity v and angular velocity ω.

The equations of motion of a unicycle are the following:

$$\dot{\vec{z}} = \begin{bmatrix} \dot{x}_i \\ \dot{y}_i \\ \dot{\theta}_i \end{bmatrix} = \begin{bmatrix} cos(\theta_i) & 0 \\ sin(\theta_i) & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v_i \\ \omega_i \end{bmatrix} \tag{1.4}$$

The dynamic system is non-linear and subject to *nonholonomic* constraints. The unicycle has three degrees of freedom, but its motion is restricted by the roll without slipping condition, and can only move in the direction of its heading, or rotate about the axis normal to the plane of motion.

In the formation control problem it is desired for the MAS to achieve a particular formation shape. It is classified in a manner similar to the classic control problem. In the *formation regulation* problem, the formation shape remains stationary or with a constant velocity. In the *formation tracking* problem, it is desired for the formation to track an arbitrary reference signal or path while maintaining formation.

Consensus protocols can be used to couple the states of the agents and satisfy the regulation or formation tracking control problem. Considering the effect of time delays, in its most general form, the consensus function can be described as a function of the delayed

states and a function in time to drive the individual agents towards their position in the formation ϕ.

$$u_i(t) = f(z_i(t-\tau_i), ..., z_n(t-\tau_n), \phi_i(t)) \tag{1.5}$$

The non-linear system can be expressed as a function of the vector of the states of all agents $\vec{x}$ and the vector of individual input consensus functions $\vec{u}$:

$$\dot{\vec{x}} = A(\vec{x}, \vec{u}) \tag{1.6}$$

A review of the theory required to analyze the MAS described by Equations 1.4, 1.5 and 1.6 will be developed in Chapter 2. The design procedure, analysis of the control scheme and simulation examples will be described in Chapters 3 and 4.

1.4 Literature Review

The area of consensus research has been studied for a long time, starting with its introduction in computer science in the area of *distributed computing*. Its use in the context of dynamic systems and control theory started with the work of Borkar and Varaiya [13]. It was first used in a modern context by Jadbabaie, Lin and Morse [14] in terms of consensus protocols where he analyzed Vicseck's swarming model [15] . The theoretical framework for a systematic treatment of the consensus problem in networked dynamic systems was proposed by Olfati-Saber and Murray [10], their work introduced the graph theoretical representation of consensus systems, and introduced their Kronecker multiplication representation. In that seminal paper, they highlight the spectral properties of Graph Laplacians and their connection to the convergence of consensus and alignment algorithms [16], they also introduce the analysis for consensus protocols with a single uniform time delay. This work, however, was originally scoped to systems of first order integrators. Ren and Beard [17] and Feng, Xu and Zhang [18], have expanded on Olfati-Saber and Mur-

ray's work to include results for higher order systems, its applications to multi-vehicle control by Ren and Beard [1], and applications to the unicycle problem [19]. In the area of swarm control, Olfati-Saber [20] used consensus protocols to solve the *flocking* problem in a similar fashion, Xiao et al. [21] developed a methodology for the design of formations. Both methods rely on non-linear consensus protocols, this choice complicates the analysis of the system. The flocking problem is one of two fundamental distributed control problems where all agents are required to move in the same direction with the same speed, the second kind, is the *rendezvous* problem, where all agents are required to reach a common location.

Most authors have identified the role of time delays in the consensus problem, however, few have done an exact and exhaustive study of its effect. In Olfati-Saber and Murray's original paper, they correctly identify the analysis of delayed consensus as a frequency response analysis, and use Nyquist's criterion to define the stability bound for the single-delay, first-order system. The case with second order dynamics and switching topologies was analyzed by Qin, Gao and Zheng [22]; however, this work used the Lyapunov-Krasovskii functional methodology [23], which is conservative in nature, requires the solution of difficult linear matrix inequalities and does not provide an exact assessment of the stability of the system. The work of Cepeda-Gomez and Olgac applies the paradigm of the *Cluster Treatment of Characteristic Roots*, CTCR in short, to analyze different forms of leaderless and virtual leader consensus protocols subject to input delays and time-delayed communications. This paradigm uses the D-Subdivision and a holographic transformation to describe the exact and complete stability map of a *linear time invariant multiple time delayed system* (LTI-MTDS) system. The CTCR method has been the subject of multiple research papers since its introduction by Olgac and Sipahi [25]. Its study has revealed several important characteristics of the general class of multiple, rationally independent, time-delayed systems, including: the discovery of the bounds of imaginary spectra by Gao and

Olgac [26], the introduction of the concept of the Spectral Delay Space by Fazelinia, Sipahi and Olgac [27] and the discovery of the novel delay-scheduling control method [28]. This methodology reveals counterintuitive results, showing that in certain cases, stability can be maintained after increasing the time delays above a certain threshold. It is generally assumed that in order to guarantee the stability of a system, the time delays have to be kept under a static marginal boundary, but the exhaustive analysis of the time-delay stability picture reveals this assumption is invalid. Finally, Cepeda-Gomez and Perico [29] utilize their work in delayed linear consensus protocols to the formation control problem of non-holonomic agents.

1.5 Motivation and Contributions

In Cepeda-Gomez and Perico's work [29], the authors utilize the concepts of *input-output feedback linearization* and a general methodology for the decoupling of the time-delayed linear consensus *proportional-derivative* feedback system to drive the agents towards a static formation shape. The authors analyze the complete stability picture of the system in the case where there are two rationally independent uniform communication time delays, one in the proportional feedback channel and one in the derivative feedback channel. They assume that the agents are not subject to *cognitive or sensing* delays. This assumption is primarily driven by characterizing the agents as capable of acquiring their own state and processing the consensus information in a negligible amount of time. In literature, experimental platforms used for large swarms of robots tend to be low-cost platforms with processing and sensing constraints, in addition to the communication constraints previously considered [30, 31].

To the best of our knowledge, the study of formations of non-holonomic agents with sensing, processing and communication constraints has not been explored. This is the primary motivation for the development of this thesis. In Chapter 3, we describe a methodology for the analysis and design of formations of non-holonomic agents through linear

consensus protocols. We extend the existing decentralized formation control consensus protocol and propose a novel form that allows for the existence of physical leader agents. We prove that this novel protocol solves the consensus problem and perform an exhaustive analysis of the robustness of the protocol to sensing and communication time delays. In the process, we complement the work of Cepeda-Gomez and Olgac [32] and define additional conditions for the factorization property that enables the time-delay stability assessment. We find that depending on the communication topology and the control gains, the sensing delay is a significant factor in the stability picture of the control system, and can drive the effective delay margin. This finding emphasizes the fact that an analysis of time-delay robustness focused only on communication constraints is incomplete. The extension of the consensus protocol to include physical leaders *(agents that don't receive state information from the rest of the formation)* allows us to define tracking goals in a global frame of reference. We analyze the stability conditions of the time-delayed consensus system of non-holonomic agents and find the marginal stability points of sample topologies. The formation control problem is tackled by finding a time-domain factor to decouple the system inputs and drive the agents independently. This factor depends solely on the dynamics of the agents and the target state information of the local communication neighborhood of each agent. This enables us to define the formation shape and dynamic formation in a decentralized manner. We account for the sensing delay in the input-output feedback linearization process by introducing a local non-linear predictor. The predictor uses an internal non-linear model of the plant to predict the delay-free response of each of the agents. In the methodology presented, the predictor is implemented in a decentralized manner by each of the agents. In Chapter 4, we implement the methodology outlined in Chapter 3 and present simulation results that validate the approach, including a sensitivity analysis of the system to various time delays via a Monte Carlo experiment.

CHAPTER 2

PRELIMINARIES

2.1 Time-Delayed Linear Consensus Protocols and Algebraic Graph Properties

As introduced in Section 1.2, a communication topology can be represented by the directed graph $G = (V, E)$, with the set of nodes $V = 1, 2, ..n$ and the set of edges $E \subset V \times V$. An agent i sharing information with agent j is represented by $e_n = (v_i, v_j)$, an edge from $i \longrightarrow j$. The set of neighboring agents of agent i is denoted as:

$$N_i = \{j \in V : (i, j) \in E\} \tag{2.1}$$

This section is a summary of the general framework and main results found by Olfati-Saber and Murray [10], with the addition of some general remarks in literature that followed.

Utilizing the notation, we reformulate Equation (1.2), as an n-order linear system. Each agent is described by the first-order dynamics $\dot{x}_i = u_i$.

$$\dot{x_i} = \sum_{j \in N_i} (x_j(t) - x_i(t)) + c_i(t), x_i(0) = z_i \in \mathbb{R}, c_i(t) = 0. \tag{2.2}$$

The dynamics of the time-delayed MAS in 2.2 can be written as:

$$\dot{x_i} = -Lx(t - \tau) \tag{2.3}$$

Where $L = [l_{ij}]$ is the graph *Laplacian* of the network and its elements are defined as follows:

$$l_{ij} = \begin{cases} |N_i| \,, & i = j \\ -1 \,, & j \in N_i \end{cases} \tag{2.4}$$

The cardinality of set N_i denotes the number of neighbors interacting with agent i, or the

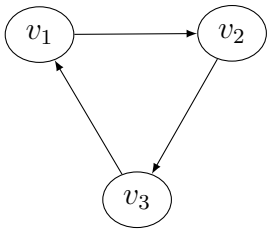


FIGURE 2.1. Strongly connected digraph composed of three nodes.

$in-degree$ of vertex i. According to the definition of the graph Laplacian, this matrix is positive-semidefinite, because the sum of all of its rows $\sum_j l_{ij} = 0$. By Gershgorin's circle theorem [33], each eigenvalue λ_i of the matrix will lay at the union of the circles centered at a_{ii}, with a radius given by the sum of the absolute values of the off-diagonal terms.

$$D_i = \left\{ \lambda \in \mathbb{C} : |z - |N_i|| \leq \sum_{j \in N} |l_{ij}| \right\} \tag{2.5}$$

These regions overlap and the entire spectrum lies in the disk corresponding to the maximum degree of the matrix.

$$D = \{ \lambda \in \mathbb{C} : |z - |N_{max}|| \leq |N_{max}| \} \tag{2.6}$$

If the digraph is *strongly connected*, which means any node i can be reached by following a directed path from any other node j, one of its eigenvalues is guaranteed to be $\lambda_i = 0$. This eigenvalue corresponds to the scaled unity eigenvector $\mathbf{1} = \alpha[1, 1, 1, ..]$, this vector is a member of the null-space of L , $L(L\mathbf{1} = 0)$. This eigenvector is also called the *group decision value*. In the state-space this means the system has an equilibrium point in the form $x^* = \alpha\mathbf{1}$ [10]. An example of a strongly connected graph is show in Figure 2.1.

In a connected graph that does not meet the strongly connected condition, x^* is an equilibrium point of the system in Equation (2.3), unique up to scalar multiples. The number of connected components of a digraph is equal to the multiplicity of the the triv-

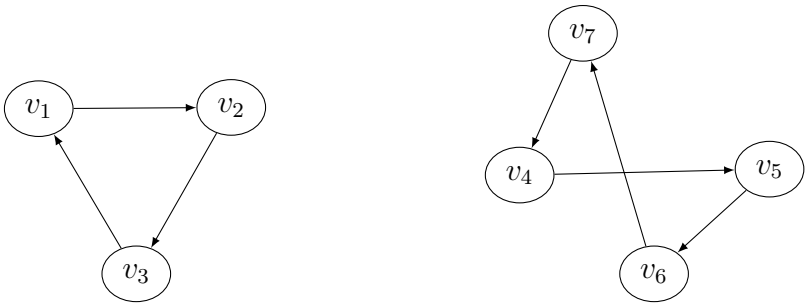


FIGURE 2.2. Digraph with $n = 7$ vertices and $m = 2$ connected components - a system that does not meet the strongly connected condition.

ial eigenvalue, thus defining the scalar multiple of the *group decision value(s)*. A complete proof can be found in [34]. This fact sets the lower bound for the spectrum of the Laplacian matrix at the origin, thus under the strongly-connected graph condition, all non-zero eigenvalues are positive:

$$0 = \lambda_1 < \lambda_2 < .. < \lambda_n < 2\Delta \tag{2.7}$$

The conditions under which the bounds described in Equation (2.7) are valid can be relaxed to include digraphs with at least one spanning tree [1, 35].

The existence of a spanning tree is defined as the case where the digraph includes a set of edges that define a path between a node v_* and every other vertex in the graph, or more formally, a sub-graph of digraph G that includes every vertex of G and is a tree. Vertex v_* is often called the leader vertex. The spectrum characteristics of the Laplacian matrix that meet the spanning tree conditions are summarized in Lemma 1, the proof can be found in [35].

Lemma 1. Let G be a connected digraph of n vertices and L be its Laplacian matrix. Suppose G has $m \geq 1$ strongly connected components. Then $rank(L) = n - m$ and all $n - m$ non-trivial eigenvalues of L have positive real parts.

The spectrum of the matrix defining the consensus protocol has very interesting properties that can be leveraged for the analysis of the stability and performance of the

multi-agent system. In the case of connected topologies, the zero eigenvalue is known as the trivial eigenvalue. The second smallest eigenvalue λ_2 is called the algebraic connectivity, and is a direct measure of the performance of the system and the speed in which the agents reach consensus [10].

The convergence analysis can be done in the frequency domain by analyzing the terms of the system diagonalized by the similarity transformation with its eigenbasis $\boldsymbol{T}$:

$$\begin{aligned} \boldsymbol{x} &= \boldsymbol{T}\boldsymbol{z} \\ \dot{\boldsymbol{z}}_{\boldsymbol{j}} &= -\lambda_j \boldsymbol{z}_j(t-\tau) \end{aligned} \tag{2.8}$$

The characteristic equation of each decoupled subsystem will be in the form:

$$(s + \lambda_i e^{-\tau s}) z_i = 0 \tag{2.9}$$

In the case of connected topologies or topologies with at least one spanning tree, the first n-factors corresponding to the connected components or spanning trees, will have a stationary root at $s = 0$, c In this form it can be readily seen that for systems with higher order dynamics, each connected component in digraph G will generate $n \times m$ stationary roots. These terms are marginally stable at best. For the Laplacian matrix, restating Lemma 1, in the case of strongly connected digraphs or digraphs with a spanning tree, the real part of the Laplacian matrix eigenvalues is positive. The factors generated by the positive non-zero eigenvalues of the Laplacian matrix are called the *disagreement dynamics*. In the case where $\tau = 0$, the disagreement dynamics are stable. The stability picture is completely defined by the analysis of the spectrum of the Laplacian matrix in the case of fixed communication topologies. In the case of switching topologies (i.e., topologies that are described by a time-varying digraph and thus a time-varying Laplacian matrix), the stability analysis has to be completed for every possible topology. Because of this, it may

be desired to restrict the communication graph to one that meets the connected or spanning tree conditions.

Solving the characteristic equation when $\tau \neq 0$, can be untractable due to the infinite dimensionality of the transcendental term. The CTCR method, introduced in Section 2.2 establishes a methodology for determining the exact boundary and conditions under which the disagreement terms are asymptotically stable, the proof can be found in [24].

Lemma 2. The stability of the consensus algorithm is determined by the marginal stability of the group decision dynamics and the stability of the disagreement dynamics

Remark. In the presence of time delays, the stability of the system is determined by the simultaneous stability of the disagreement dynamics, which have an infinite number of roots.

Having established the stability conditions for the factors in Equation (2.9), the convergence properties of the MAS system in Equation (2.3) can be found by considering the spectral properties defined in Lemma 1. The asymptotic stability of the *disagreement dynamics* ensures $\lim_{t\to\infty} z_{i+m}(t) = 0$, and the system will converge to its stationary points. A special case of this condition can be introduced when the communication topology is balanced, the stationary point corresponding to eigenvalue $\lambda_1 = 0$ solves the average consensus problem, i.e. the final state of the system $x(\infty) = Ave(x(0))$. Based on this analysis we introduce the following definitions:

Definition 1 (Global Consensus). Global Consensus is reached if and only all agents in the system reach a single group consensus value:

$$\lim_{x\to\infty} \boldsymbol{x}(t) = \boldsymbol{x}^* \tag{2.10}$$

Theorem 1. If $\lambda_1 = 0$ is an eigenvalue of the Laplacian matrix and has an algebraic multiplicity of 1, the system in Equation (2.3) attains global consensus. Proof follows

by substitution of λ_1. This term is analogous to the system's rigid body mode.

Definition 2 (Local Consensus). Local Consensus is reached if and only if all agents in each group m reach a single group consensus value:

$$\lim_{t\to\infty} \boldsymbol{x_m}(t) = \boldsymbol{x_m}^* \tag{2.11}$$

Theorem 2. If $\lambda_1 = 0$ is an eigenvalue of the Laplacian matrix and has an algebraic multiplicity of m, then the system in Equation (2.3) attains local consensus. Proof follows by substitution of $\lambda_1, ..., \lambda_m$. The system is analogous to a system with multiple rigid body modes.

A complete treatment of the definitions, theorems and conditions can be found in [10].

2.1.1 Alternative Forms of Consensus Protocols

Fax and Murray [36] introduced a geometrically-inspired consensus protocol for a muliple-vehicle formation. We utilize the framework introduced in this section to analyze the properties of the following system. Consider a group of n identical agents with linear dynamics:

$$\dot{\boldsymbol{x}}_{\boldsymbol{i}} = \boldsymbol{A}\boldsymbol{x}_{\boldsymbol{i}} + \boldsymbol{B}\boldsymbol{u}_{\boldsymbol{i}} \tag{2.12}$$

Where $x_i \in \mathbb{R}^m, u_i \in \mathbb{R}^p$ are the agent states and $i \in V = \{1, .., ..., n\}$ is the index for each agent. Each vehicle receives the following measurements:

$$\begin{cases} \boldsymbol{y_i} = C_1\boldsymbol{x_i} \\ \boldsymbol{z_{ij}} = C_2(\boldsymbol{x_i} - \boldsymbol{x_j}) \end{cases} \tag{2.13}$$

Where $z_{ij} \in R_m$ represents the relative external states of the agents and C_1 and C_2 are general output matrices representing the self-sensing and relative-sensing measurements. Assuming $N_i \neq \emptyset$, all the relative state signals are fused into a single error signal, the aver-

age of the relative state measurements:

$$\boldsymbol{z_i} = \frac{1}{|N_i|}\boldsymbol{z_{ij}} \tag{2.14}$$

We can also define a distributed controller K, which maps the relative state measurements to the agents input u_i. Matrices D_1 and D_2 are arbitrary gain matrices defining controller K, corresponding to the global state feedback channel and relative state channel, respectively.

$$u_i = D_1 y_i + D_2 z_i \tag{2.15}$$

Fax and Murray found that on relative-position based formations, a local controller K stabilizes the formation dynamics if an only if it stabilizes al m n-th order systems. This condition is equivalent to Lemma 2 and the frequency response methods to assess the robustness and stability of the decoupled system can be utilized to assess the stability of the MAS. The individual agents in the system are coupled by the relative state consensus term, and decoupled otherwise. One can guarantee stability by utilizing inner-loop techniques.

$$\begin{aligned} \dot{\boldsymbol{x}}_{\boldsymbol{i}} &= \boldsymbol{A}\boldsymbol{x}_{\boldsymbol{i}} + \boldsymbol{B}\boldsymbol{u}_{\boldsymbol{i}} \\ \dot{\boldsymbol{y}}_{\boldsymbol{i}} &= C_1\boldsymbol{x}_{\boldsymbol{i}} \\ \dot{\boldsymbol{z}}_{\boldsymbol{i}} &= \lambda_i C_2 \boldsymbol{x}_{\boldsymbol{i}} \end{aligned} \tag{2.16}$$

Where $\{\lambda_1, .., \lambda_n\} \in \mathbb{Z}^n$ is the set of eigenvalues of the row-normalized graph Laplacian.

Remark. The normalized Laplacian of a digraph can have complex eigenvalues, opening the door for Robust Control theory applications.

2.1.2 Notation for Consensus Protocols of Higher-Order Systems

So far, the notation used in Equation (2.12) to represent a higher-order formation of systems with uniform dynamics has been incomplete as it does include the dynamics of the integrated system. Fax and Murray in [36] have introduced a general form to represent the system using the *Kronecker Product* operator. We introduce the definition of the Kronecker Product of matrices and some properties relevant to the discussion in this thesis. [37]

Definition 3 (Kronecker Product). The Kronecker product $\boldsymbol{R}$ between two matrices $\boldsymbol{P} = [p_{ij}]$ and $\boldsymbol{Q} = [n_{ij}]$ is defined as:

$$\boldsymbol{R} = \boldsymbol{P} \otimes \boldsymbol{Q} = [p_{ij}\boldsymbol{Q}] \tag{2.17}$$

Remark. R is a block matrix of size $mn \times mn$

$$\boldsymbol{P} \in \mathbb{Z}_{m\times m},\ \boldsymbol{Q} \in \mathbb{Z}_{n\times n} \rightarrow \boldsymbol{R} \in \mathbb{Z}_{mn\times mn}$$

The following conclusions follow:

If $\boldsymbol{P}$ is a diagonal matrix, then R is a block-diagonal matrix, where each block is of size $n \times n$.

If $\boldsymbol{P}$ is a block-diagonal matrix, where each block is of size $i \times i$, and $\boldsymbol{Q}$ is a matrix of size $n \times n$, then $\boldsymbol{R}$ is a block-diagonal matrix, where each block is of size $(n+i) \times (n+i)$.

Property 1. Mixed Product of the Kronecker Products:

If $\boldsymbol{A}$, $\boldsymbol{B}$, $\boldsymbol{C}$, $\boldsymbol{D}$ are compatible matrices under matrix multiplication $\boldsymbol{AB}$ and $\boldsymbol{CD}$:

$$(\boldsymbol{A} \otimes \boldsymbol{B})(\boldsymbol{C} \otimes \boldsymbol{D}) = (\boldsymbol{AC}) \otimes (\boldsymbol{BD}) \tag{2.18}$$

Property 2. Invertibility of the Kronecker Product:

$\boldsymbol{A} \otimes \boldsymbol{B}$ is invertible if and only if both $\boldsymbol{A}$ and $\boldsymbol{B}$ are invertible, and the inverse is given by:

$$(\boldsymbol{A} \otimes \boldsymbol{B})^{-1} = \boldsymbol{A}^{-1} \otimes \boldsymbol{B}^{-1} \tag{2.19}$$

Property 3. Left distributivity of the Kronecker Product:

$$(A + B) \otimes C = A \otimes C + B \otimes C \tag{2.20}$$

Property 4. Right distributivity of the Kronecker Product:

$$(A) \otimes (B + C) = A \otimes B + A \otimes C \tag{2.21}$$

Property 5. Spectrum of the Kronecker Product: The spectrum of product $\boldsymbol{A} \otimes \boldsymbol{B}$ is equal to the product of the spectrums of $\boldsymbol{A}$ and $\boldsymbol{B}$

$$\begin{aligned} \sigma(\boldsymbol{A}) &= \lambda_0, \lambda_1, ..., \lambda_n \\ \sigma(\boldsymbol{B}) &= \mu_0, \mu_1, ..., \mu_n \\ \sigma(\boldsymbol{A} \otimes \boldsymbol{B}) &= \mu_0\lambda_0, \mu_1\lambda_1, ..., \mu_n\lambda_n \end{aligned} \tag{2.22}$$

Using the Kronecker Product, we describe the general linear system of homogeneous agents below:

$$\dot{\boldsymbol{x}} = (I_n \otimes A)\boldsymbol{x} + (I_n \otimes B)\boldsymbol{u} \tag{2.23}$$

Introducing local controller K in Equation (2.15), using global state measurements y and relative state measurements z, the general representation for the linear consensus system

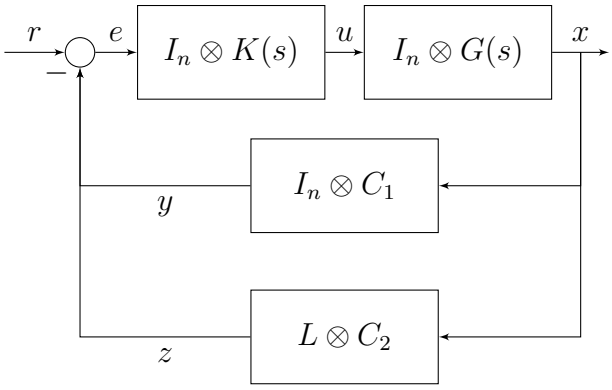


FIGURE 2.3. Block diagram representation of distributed controller - high-order system w. global and relative feedback channels.

of n agents of m order is:

$$\begin{aligned} \boldsymbol{y} &= (I_n \otimes D_1 C_1)\boldsymbol{x} \\ \boldsymbol{z} &= (L \otimes D_2 C_2)\boldsymbol{x} \\ \boldsymbol{u} &= (I_n \otimes D_1 C_1)\boldsymbol{x} + (L \otimes D_2 C_2)\boldsymbol{x} \end{aligned} \tag{2.24}$$

Introducing two types of delays which may be present in the global state measurement channel and the relative measurement channel, the fully assembled system representation can be expressed as:

$$\dot{\boldsymbol{x}} = (I_n \otimes A)\, \boldsymbol{x} + (I_n \otimes BD_1C_1)\, \boldsymbol{x}(t-\tau_1) + (L \otimes BD_2C_2)\, \boldsymbol{x}(t-\tau_2) \tag{2.25}$$

It is important to note that L is not necessarily the traditional digraph Laplacian but a matrix representing the information exchange topology. In Equation (2.16), the matrix is the *in-degree* normalized graph Laplacian. Additionally, τ_1 and τ_2 represent two general kinds of time delays. The block diagram representation of the system is presented in Figure 2.3.

2.2 Exhaustive Analysis of the Stability of Time-Delayed Linear Consensus Systems

In order to analyze the time-delay robustness of linear consensus protocols introduced in the prior sections, we introduce the main findings of the CTCR methodology [25, 26, 27, 38]. Consider the linear time invariant multiple time delayed system (LTI-MTDS) with l delayed feedback terms, and delay vector $\boldsymbol{\tau} = \{\tau_1, \tau_2, ..., \tau_n\}$:

$$\dot{\boldsymbol{x}} = A\boldsymbol{x} + \sum_{i=1}^{l} B_i x(t - \tau_i) \tag{2.26}$$

The characteristic equation for this system:

$$CE(s, \tau) = det\left(s\boldsymbol{I} - \boldsymbol{A} - \sum_{i=1}^{l} B_i e^{-\tau_i s}\right) \tag{2.27}$$

The CTCR method provides an exact and exhaustive stability assessment of the LTI-MTDS system for an arbitrary delay vector τ. This methodology can be applied to cases where the non-linear system can be linearized at its operating points or through dynamic input-output feedback linearization. In its general form, the characteristic equation can be expressed as the pseudo-polynomial:

$$CE(s, \tau) = A_0 s + A_{p+1} + \sum_{j=1}^{p} e^{-n_j \tau_j s} A_j(s, \boldsymbol{\tau}) \tag{2.28}$$

A_0 is an n^{th} degree polynomial in s, A_j are quasi-polynomials in s and τ. n_j is the highest order of commensuracy of delay τ_j. Since $A_0(s)$ is free of delays and it is the highest order term on s, this type of system is classified as "retarded" LTI-MTDS. This form is compatible with the delayed systems that arise in the development of linear consensus protocols.

The primary objective of the CTCR method is determining the number of *unstable characteristic roots* (NU) in the delay-space $\tau \in R^{l+}$. Although the number of char-

acteristic roots of the trascendental system is infinite, the CTCR method utilizes the D-Subdivision theorem to assert that the roots only change from the LHS of the complex plane to the RHS, rendering the system unstable, at a particular continuous loci in the time-delay domain [25]. The points in the loci correspond to the marginal stability points of the trascendental equation. This transition can only happen at the locations where the characteristic root crosses the imaginary axis, at a frequency ω_c. The CTCR method provides the methods to find the loci where the transitions happen. The two primary methods to find this are equivalent and are known in literature as the *Rekausius Substitution* method and the *Spectral Delay Space* by *Rekausius Substitution* or *Half-Angle Tangent* substitution method. We introduce the main concepts in the CTCR paradigm without proof:

Definition 4 (Imaginary Spectra). The complete set of imaginary spectra of the LTI-MTDS system for delay vector $\tau \in \mathbb{R}^{l+}$ is defined as:

$$\begin{aligned} \boldsymbol{\Omega} &= \{\omega_c | CE(s = \omega_i, \boldsymbol{\tau}) = 0, \tau \in \mathbb{R}^{l+}, \omega_c \in \mathbb{R}\} \\ \boldsymbol{\Omega} &= \{\omega_c | \langle \tau, \omega_c \rangle, \tau \in \mathbb{R}^{l+}, \omega_c \in \mathbb{R}\} \end{aligned} \tag{2.29}$$

$\langle \tau, \omega_c \rangle$ is the locus in the delay spaces corresponding to imaginary root ω_c

Definition 5 (Kernel Hypersurfaces $\wp_0$). The curves spanning all the points in the locus $\langle \tau, \omega_c \rangle$ satisfying the constraint $\{0 < \tau_k < 2\pi \quad | \quad k = 1, 2, ..., l\}$ are called the kernel hypersurfaces. The points on this curve contain the smallest delay composition corresponding to all the possible imaginary roots. By this definition, the kernel hypersurfaces are unique for a given system, and the boundaries of the imaginary spectra are bounded. The characteristic equation of the system contains an infinite number of roots, which can be reconstructed from the kernel hypersurfaces.

Definition 6 (Offspring Hypersurfaces $\wp$). The curves obtained from the set

of Kernel Hypersurfaces $\wp$ by applying the pointwise non-linear transformation:

$$\{\langle \tau_1 \pm \frac{2\pi}{\omega_c} j_1, \tau_1 \pm \frac{2\pi}{\omega_c} j_2, ..., \tau_l \pm \frac{2\pi}{\omega_c} j_l \rangle \quad | \quad j_1, j_2, ..., j_l \in \mathbb{N}\} \tag{2.30}$$

This definition agrees with the infinite dimensionality of the characteristic roots of a transcendental equation

Definition 7 (Root Tendency - RT). The root tendency for each imaginary crossing frequency $\omega_c i$ is defined as:

$$RT|^{\tau_j}_{s=\omega_c i} = sgn[Re(S^s_{\tau_j}|_{s=\omega_c i})] \tag{2.31}$$

Where $S^s_{\tau_j} = \frac{\partial s}{\partial \tau_j}$ is the root sensitivity for delay τ_j, which can be computed by partial implicit differentiation of the characteristic equation, evaluated at each kernel loci.

Remark. The root tendency is equivalent under the DS $\leftrightarrow$ SDS transformation, proof can be found in [38]. A Root Tendency of $+1$ is destabilizing and a Root Tendency of -1 is stabilizing.

2.2.1 Spectral Delay Space and The Building Block Concept

The spectral delay space concept consists on transforming the loci $\langle \tau_1, \tau_2, .., \tau_j \rangle$ and its corresponding crossing frequency ω_c to the coordinates $\langle \tau_1 \omega_c, \tau_2 \omega_c, ..., \tau_j \omega_c \rangle$. The primary advantage of representing the kernel hypersurfaces $\wp$ in this domain is that the kernel hypersurface is bound to the *l dimensional* hypercube of edge length 2π. This property will be readily apparent after transforming the transcendental equation from the exponential form to the trigonometric form. Some properties follow from this fact, they can be found in [27].

Having established that the root transitions from stable to unstable can only occur at the purely imaginary marginal stability points $s = \omega_c i$, we use Euler's identity to trans-

form the transcendental terms:

$$e^{\tau_j \omega_c i} = cos(v_j) - isin(v_j) \quad , \quad v_j = \tau_j \omega_c \tag{2.32}$$

Parametrizing the trigonometric terms by a single parameter z_j, the half-angle tangent:

$$z_j = tan(\frac{v_j}{2}) \quad , \quad cos(v_k) = \frac{1 - z_k^2}{1 + z_k^2} \quad , \quad sin(v_k) = \frac{2z_k}{1 + z_k^2} \tag{2.33}$$

The relationship between z_j and t_j is many-to-many. Performing this substitution allows us to use the methods of algebraic geometry to find the loci of marginal stability points. For the two-delay case, polynomial reduction theory and *Sylvester's Resultant* has been used to find the contours in 2D space corresponding to the stability transitions [27]. For the multiple-delay case *Caley's Resultant* and the frequency sweeping method has been used to find the loci of the intersection of two arbitrary delays, while the others are fixed [26].

2.2.2 CTCR Method

The algorithm to find the number of unstable roots in a LTI-MTDS system is as follows:

Algorithm 1 (CTCR Method).

1. Find the number of unstable roots NU_0 of the undelayed system $CE(s, \tau = 0)$.
2. Find the marginal stability loci in the Spectral Delay Space or Delay Space.
3. Trace a ray from the origin of the DS to the point $\tau_t = [\tau_1, \tau_2, ..]$ that needs be assessed.
4. Find the points of intersection between the ray traced and the root transition hypercurves and order them by their euclidian norm .
5. Compute the Root Tendency at every point $\mathcal{P}_j$ of the intersection.

6. Starting from the origin, and the number of stable roots NU_0, iterate through $\mathcal{P}$ in order:

   - Add 2 unstable roots to NU when $RT|_{\mathcal{P}} = +1$
   - Subtract 2 unstable roots $RT|_{\mathcal{P}} = -1$

We proceed to explain the CTCR process with an illustrative example.

A sample system of the form equation in 2.26 with two delays has a characteristic equation:

$$CE(s, \tau_1, \tau_2) = s^2 - \frac{1}{2}se^{-s\tau_1 - s\tau_2} - \frac{1}{2}e^{-s\tau_1 - s\tau_2} + se^{-s\tau_2} + e^{-s\tau_2} \tag{2.34}$$

The number of unstable roots of the system can be computed:

$$\left\{s \to \frac{1}{4}\left(-1 - i\sqrt{7}\right), s \to \frac{1}{4}\left(-1 + i\sqrt{7}\right)\right\}$$

The undelayed system has no unstable roots, $NU_0 = 0$. Having established by the continuity argument that the stability of the system only changes at its marginal stability points, we perform the half-tangent substitution:

$$CE(\omega, z_1, z_2) = \omega^2 + \omega\left(\frac{2iz1z2}{(z1^2+1)(z2^2+1)} - \frac{(1-z1^2)z2}{(z1^2+1)(z2^2+1)} - \frac{z1(1-z2^2)}{(z1^2+1)(z2^2+1)} - \frac{i(1-z1^2)(1-z2^2)}{2(z1^2+1)(z2^2+1)} + \frac{2z2}{z2^2+1} + \frac{i(1-z2^2)}{z2^2+1}\right)$$
$$\frac{2z1z2}{(z1^2+1)(z2^2+1)} + \frac{i(1-z1^2)z2}{(z1^2+1)(z2^2+1)} + \frac{iz1(1-z2^2)}{(z1^2+1)(z2^2+1)} - \frac{(1-z1^2)(1-z2^2)}{2(z1^2+1)(z2^2+1)} - \frac{2iz2}{z2^2+1} + \frac{1-z2^2}{z2^2+1} \tag{2.35}$$

This is quasi-polynomial on ω with complex coefficients $f_0(z_1, z_2)$, $f_1(z_1, z_2)$, $f_2(1)$. For

$CE(\omega, z_1, z_2) = 0$, both real and imaginary parts have to be simultaneously equal to zero.

$$\begin{aligned} Re[CE(\omega i, z_1, z_2)] = &- \omega^2 \left(z_1^2 + 1\right)\left(z_2^2 + 1\right) + \omega\left(z_2 + z_1\left(z_2^2 + 3z_1z_2 - 1\right)\right) \\ &+ \frac{1}{2}\left(-3z_2^2z_1^2 + 3z_1^2 + 4z_2z_1 - z_2^2 + 1\right) = 0 \end{aligned} \tag{2.36}$$

$$\begin{aligned} Im[CE(\omega i, z_1, z_2)] = &\frac{1}{2}\omega\left(-3z_2^2z_1^2 + 3z_1^2 + 4z_2z_1 - z_2^2 + 1\right) \\ &+ \frac{1}{2}\left(-6z_2z_1^2 - 2z_2^2z_1 - 2z_2 + 2\right) = 0 \end{aligned}$$

This process is equivalent to finding the set of common roots for the real and imaginary polynomials, which can be found by using Sylvester's, Bezout's or Macaulay's Resultant [39]. These methods are available in modern computer algebra systems, like *Wolfram Mathematica* [40], *Maple* or *Mathwork's Matlab/MuPad*. Without elaborating on the specifics of the root finding process, we find the following common root:

$$\begin{aligned} &(-\tfrac{1}{8}27z_2^6 - \tfrac{99z_2^4}{8} - \tfrac{45z_2^2}{8} + \tfrac{27}{8})z_1^6 + (-\tfrac{1}{2}3z_2^5 + 9z_2^3 + \tfrac{21z_2}{2})z_1^5 + (-\tfrac{1}{8}47z_2^6 - \tfrac{151z_2^4}{8} - \tfrac{73z_2^2}{8} + \tfrac{31}{8})z_1^4 \\ &+(-3z_2^5 + 10z_2^3 + 13z_2)z_1^3 + (-\tfrac{1}{8}21z_2^6 - \tfrac{61z_2^4}{8} - \tfrac{35z_2^2}{8} + \tfrac{5}{8})z_1^2 + (-\tfrac{1}{2}3z_2^5 + z_2^3 + \tfrac{5z_2}{2})z_1 - \tfrac{z_2^6}{8} - \tfrac{9z_2^4}{8} - \tfrac{7z_2^2}{8} + \tfrac{1}{8} \\ &= 0 \end{aligned}$$

Replacing back $z_1 = tan(\frac{v_1}{2})$ and $z_2 = tan(\frac{v_2}{2})$, the loci in the SDS domain can be found by numerically finding the 0-level contour, or solving for z_1 in terms of z_2 and sweeping the parameter across the domain of the building block $0 < z_2 < 2\pi$. This will generate the Kernel Curves in the SDS domain. In order to transform back to the τ_1, τ_2 (DS) Space, the crossing frequency must be found by solving the fully determined system with one equation and one unknown in Equation (2.35). We find two reflected curves and a single continuous curve on the DS space, this can be seen on Figure 2.4. We can also calculate the following 6 offspring curves to complete the stability map in the $0 < \tau_1 < 10, 0 < \tau_2 < 10 <$

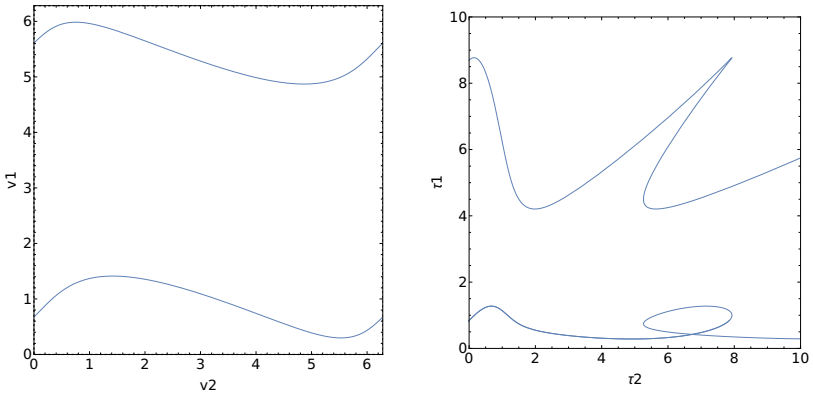


FIGURE 2.4. Kernel curves for second order system with 2 delays.

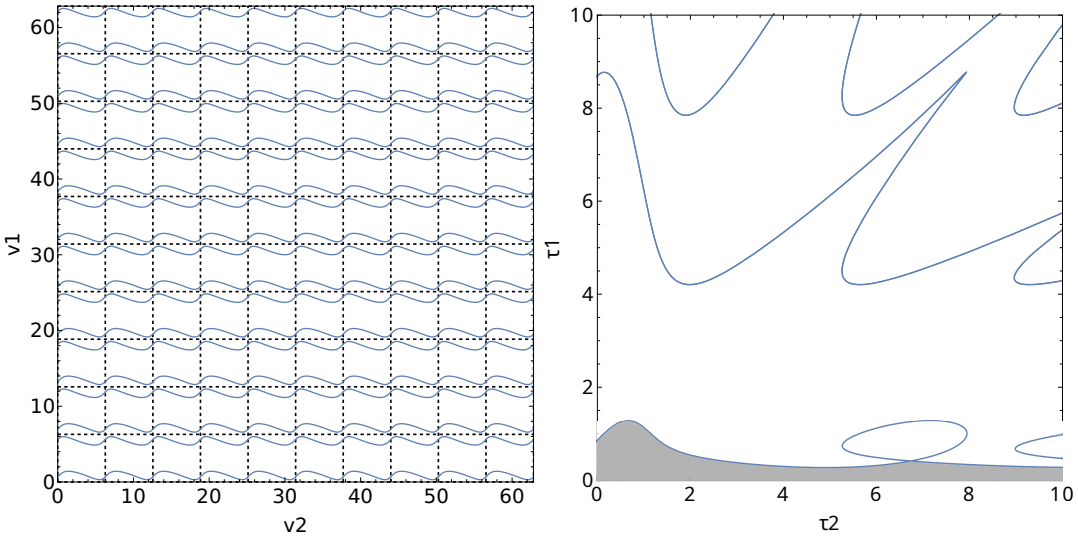


FIGURE 2.5. Offspring curves for second-order system with 2 delays - stable region shaded in grey.

domain. This can be seen in Figure 2.5. The *root tendencies* at the boundaries can be calculated to graphically visualize the stable regions. We see that the system is stable for all delays τ_1 as long as τ_2 is below the vertical boundary. This boundaries are exact and exhaustive in the swept region.

The symbolic deployment of the CTCR method is computationally intensive and systems of higher order may not be suited for this, this makes it difficult to deal with a high-order MAS; however, if the system can be block diagonalized, we an analyze each individual block, similar to the process in equation 2.8. This methodology was introduced by

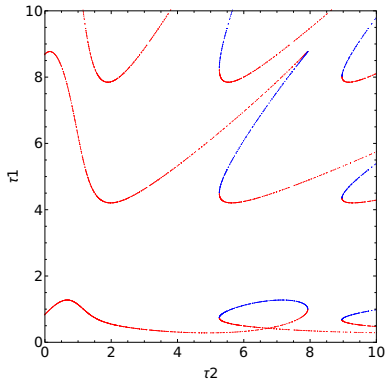


FIGURE 2.6. Root tendency wrt. τ_2 for second order system with 2 time delays. red corresponds to destabilizing transitions, blue corresponds to stabilizing transitions.

Cepeda-Gomez and Olgac [24] and simplifies the process for cases where the communication topology matrix is non-defective. We utilize the methodology in Chapter 3.

2.3 Control of Non-Holonomic Systems by Input-Output Feedback Stabilization

As outlined in the previous section, the CTCR paradigm allows us to study the stability and robustness of LTI-MTDS, however, the model for DWMR's 1.4 is non-holonomic and non-linear.

We restate the general control problem, for a general system:

$$\dot{\boldsymbol{x}} = A(\boldsymbol{x}, \boldsymbol{u}) \tag{2.37}$$

One would like to design a feedback control law of the form:

$$\boldsymbol{u} = \alpha(\boldsymbol{x}) + \beta(\boldsymbol{x})\boldsymbol{v} \tag{2.38}$$

Such that:

- A desired closed-loop equilibrium point is asymptotically stable (regulation)

- A desired closed-loop trajectory $x_d(t)$ is asymptotically stable (tracking)

In linear systems, the controllability condition implies asymptotic and exponential stability at x_e by a linear state feedback controller [41]:

$$\alpha(x) = K(\boldsymbol{x} - \boldsymbol{x_e}) \tag{2.39}$$

Linearizing the system at operating point x_e:

$$\dot{\delta}x = A\delta x + B\delta u \quad , \quad \delta x = x - x_e \quad , \quad \delta u = K\delta x \tag{2.40}$$

If the system linearized system is controllable, then the original non-linear system can be locally stabilized at x_e [42].

Performing operating point linearization for the unicycle in Equation 1.4 by writing it in affine form and computing the Jacobian:

$$\begin{gathered} \dot{\boldsymbol{x}} = f(\boldsymbol{x}) + g(\boldsymbol{x})\boldsymbol{u} \\ f(\boldsymbol{x}) = 0 \quad \boldsymbol{g(x,u)} = g(x)u \\ \frac{\delta \boldsymbol{g(x,u)}}{\delta x} = \begin{bmatrix} 0 & 0 & -vsin(\theta) \\ 0 & 0 & vcos(\theta) \\ 0 & 0 & 0 \end{bmatrix} \\ A \equiv 0 \quad rank(B(x,u)) = 2 < 3 \end{gathered} \tag{2.41}$$

By inspection we can see that the eigenvalues of the jacobian matrix $\{\lambda_J\} = \varnothing$ for all states, and the system is rank deficient, hence it is uncontrollable and cannot be controlled by means of smooth linear feedback. This is equivalent to not meeting Brockett's stability condition [43] and applies for all non-holonomic systems.

Oriolo, Luca and Vendittelli [44] introduced the method of dynamic input-output

feedback linerization by chained forms to stabilize the unicycle using time-varying feedback:

$$u = \alpha(x, t) \tag{2.42}$$

Deriving the chained forms, we define the output $\eta = \{x, y\}$ and take the first derivative of the output:

$$\dot{\eta} = \begin{pmatrix} \dot{x} \\ \dot{y} \end{pmatrix} = \begin{pmatrix} cos(\theta) & 0 \\ sin(\theta) & 0 \end{pmatrix} \begin{pmatrix} u_1 \\ u_2 \end{pmatrix} \tag{2.43}$$

The input u_2 does not affect the output, we recover the dependence on $u_2 = \omega$ by performing the input-output linearization.

Integrating the linear velocity input:

$$u_1 = \varepsilon, \quad \dot{\varepsilon} = a \quad \Rightarrow \quad \dot{\eta} = \varepsilon \begin{pmatrix} cos(\theta) \\ sin(\theta) \end{pmatrix} \tag{2.44}$$

Differentiating $\dot{\eta}$ using the chain and product rules, and substituting $\dot{\varepsilon} = a$ and $\dot{\theta} = \omega$ results in a system with inputs a and ω:

$$\ddot{\eta} = \begin{pmatrix} cos\theta & -\varepsilon sin\theta \\ sin\theta & \varepsilon cos\theta \end{pmatrix} \begin{pmatrix} a \\ \omega \end{pmatrix} \tag{2.45}$$

Using the second derivative of the output $\ddot{\eta}$ as the input vector $\vec{v} = [v_1, v_2]$, and assuming $\varepsilon \neq 0$, we can take the inverse and find:

$$\begin{pmatrix} a \\ \omega \end{pmatrix} = \begin{pmatrix} cos\theta & -\varepsilon sin\theta \\ sin\theta & \varepsilon cos\theta \end{pmatrix}^{-1} \begin{pmatrix} v_1 \\ v_2 \end{pmatrix} \tag{2.46}$$

The original system input in terms of the dynamic compensator:

$$\begin{aligned} \dot{\varepsilon} &= v_1 cos\theta + v_2 sin\theta \\ u_1 &= \varepsilon \\ u_2 &= \frac{v_2 cos(\theta) - v_1 sin(\theta)}{\varepsilon} \end{aligned} \tag{2.47}$$

By construction the system is exactly linearized as two second order chained forms, or the fourth order system with state $\boldsymbol{\phi} = [x, \dot{x}, y, \dot{y}]$ and input $\boldsymbol{u} = [u_1, u_2]$:

$$\ddot{\boldsymbol{\phi}} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix} \boldsymbol{\phi} + \begin{bmatrix} 0 & 0 \\ 1 & 0 \\ 0 & 0 \\ 0 & 1 \end{bmatrix} \boldsymbol{u} \tag{2.48}$$

The system is block diagonal with 2 second order blocks. Oriolo, Luca and Vendittelli [44] also introduce the tracking controller that guarantees asymptotic tracking [42] for trajectory $x_d(t), y_d(t)$ in the form:

$$\begin{aligned} v_1 &= \ddot{x_d}(t) + k_{p1}(x_d(t) - x) + k_{d1}(\dot{x_d}(t) - \dot{x}) \\ v_2 &= \ddot{y_d}(t) + k_{p2}(y_d(t) - y) + k_{d2}(\dot{y_d}(t) - \dot{y}) \end{aligned} \tag{2.49}$$

Remark (Trajectory Restrictions). The desired trajectory $x_d(t), y_d(t)$ must be smooth and persistent (i.e $u_{d1}^2 = x_d(t)^2 + y_d(t)^2$ must never go to zero), this is due to the ε term in the denominator in Equation (2.47). A deadband may be implemented in the regions where $\epsilon = 0$

CHAPTER 3

FORMATION CONTROL AND ROBUSTNESS ANALYSIS

3.1 Linear Consensus Protocol Design

In Section 2.1.1 we introduced an alternative form of consensus protocols for leaderless formations. In this section, we extend the analysis to the case of formations with one or multiple leaders and prove the asymptotic stability of the resulting consensus protocol. Beginning with the general $n \times m$ order representation of the linear consensus system in (2.25) and the results of the dynamic linearization of the unicycle, we design two identical decoupled consensus controllers, one for each output variable $\boldsymbol{\eta} = \{x, y\}$. The process is performed for output variable x but the analysis is identical for output variable y. We define the $2n$-dimensional vector of state variables $\boldsymbol{x} = [x_1, \dot{x_1}, .., x_n, \dot{x_n}]$ and build the full-state feedback system in Equation (3.1) under the assumption that the system is fully observable in the presence of two arbitrary time-delays τ_1 and τ_2, where A_n is the system matrix of each of the agents in the system, I_n is the $n \times n$ identity matrix and D_1 and D_2 are arbitrary feed-forward matrices of size 2×2.

$$\dot{\boldsymbol{x}} = (I_n \otimes A_n)\boldsymbol{x} - (I_n \otimes D_1)\, \boldsymbol{x}(t - \tau_1) + (L \otimes D_2)\, \boldsymbol{x}(t - \tau_2) \tag{3.1}$$

The process of designing a consensus protocol for the MAS of n agents for this system encompasses finding a set of suitable D_1, D_2 control matrices, corresponding to the global and distributed state measurements, respectively, and finding the general form of L such that the system is globally stable. Such system may solve the *formation regulation* and *formation tracking* problems. We start this process by defining L.

Fax and Murray [36] and Cepeda-Gomez and Perico [29] utilize the *in-degree* normalized graph Laplacian Matrix L 3.2 by scaling the digraph laplacian by the inverse of the degree matrix $\mathcal{D}$.

$$L = -\mathcal{D}^{-1}(\mathcal{D} - A_g) = -I_n + \mathcal{D}^{-1}A_g \tag{3.2}$$

We recall that the degree matrix is defined as the diagonal matrix formed by the cardinality of the local neighborhoods of each agent i, $\mathcal{D} = diag(|\mathcal{N}_1|, |\mathcal{N}_2|, .., |\mathcal{N}_n|)$. This form is only valid in the cases where the cardinality of all the agents is not zero, i.e. the universal quantifier holds $\{\mathcal{N}_i \neq \emptyset : \forall \quad i \in \mathbb{N}, 1 < i < n\}$, or, every agent has a non-empty informer neighborhood. In leader-follower architectures, by definition, an agent with the role of leader is independent from the state of its followers, and thus it does not meet the cardinality condition. In this case, the inverse of the degree matrix, $\mathcal{D}^{-1}$, is undefined. Motivated by the abundant references of leader-follower architectures for distributed control systems [45] [5, 46, 47], we introduce a consensus matrix L_{comm} for multi-agent systems that permits the existence of leaders as part of the formation. This matrix is obtained by scaling the adjacency matrix A_g by the *Moore - Penrose* pseudoinverse of the degree matrix, $\mathcal{D}^{+}$.

$$L_{rel} = -I_n + D^{+}A_g \tag{3.3}$$

This form for matrix L corresponds to a communication matrix of scaled relative state measurements. In the cases where the agents are not capable of communicating relative state measurements, we define matrix L_g as the global state communication matrix.

$$L_g = D^{+}A_g \tag{3.4}$$

In order to prove the convergence of the system defined in 3.1, we need to analyze the spectral characteristics of matrix L and L_g.

Lemma 3. The spectrum of $A + B$, where A and B are a commuting pair $AB = BA$, is contained in the set $\{\lambda_1 + \lambda_2 : \lambda_1 \in \sigma(A), \lambda_2 \in \sigma(B)\}$, where $\sigma(A)$ is the spectrum of A.

Proof. Found in [48].

Lemma 4. The spectrum of product $\mathcal{D}^{+}A_g$ for a system with m leaders has m

zero eigenvalues.

Proof. By definition, the adjacency matrix of a graph corresponding to the communication topology of n agents, $A_g \in R_{n\times m}$ with m leaders is rank deficient, the same applies to the degree matrix D. By direct consequence of the singular decomposition property of the *Moore - Penrose* pseudoinverse, the operation does not alter the rank of the matrix, and the product $\mathcal{D}^+ A_g$ shares the same properties:

$$\begin{gathered} rank(A_g) \equiv rank(\mathcal{D}) \equiv rank(\mathcal{D}^+) \equiv rank(\mathcal{D}^+ A_g) = n - m \\ \{\lambda_1 = 0, \lambda_2 = 0, ..., \lambda_m = 0, \lambda_{m+1}, .., \lambda_n\} \in \sigma(\mathcal{D}^+ A_g) \end{gathered} \tag{3.5}$$

Lemma 5. The *Moore-Penrose* pseudoinverse of a diagonal matrix with m zero eigenvalues:

$$\mathcal{D}^+ = diag(0_1, 0_2.., 0_m, \frac{1}{d_{m+l}}, .., \frac{1}{d_n}) \tag{3.6}$$

Theorem 3. The spectrum of a communication matrix L_g containing scaled global state measurements is contained in Gershgorin's disk of radius 1, centered at the origin.

Proof. By Gershgorin's circle theorem [49], the spectrum σ of $\mathcal{D}^+ A_g$ is contained in the disks centered around $diag(\mathcal{D}^+ A_g) = [dag_{ii}]$ with radii $R_i = \sum_{i\neq j}^{n} |a_{ij}| = 0$, for rows corresponding to the m leaders, and $R_j = \sum_{i\neq j}^{\delta_i} |\frac{1}{\delta_i}| = 1$. Recalling $diag(A_g) = \{0\}_n$,$dag_{ii} = 0$, all the circles are centered at the origin of the complex plane with radii of 0 or 1. The eigenvalues corresponding to leader agents are $\lambda_l = 0$.

$$\sigma(\mathcal{D}^+ A_g) = D(0, 0) \cup D(0, 1) \tag{3.7}$$

Theorem 4. The real parts of the eigenvalues of a communication matrix L containing scaled relative state measurements are zero or negative.

Proof. By Lemma 3 and 3, the sum $L = -I_n + D^+ A_g$ shifts the eigenvalues of

$L_g = D^+ A_g$ one unit to the left, thus all eigenvalues are contained in the Gershgorin's disk $D(-1, 1)$ and the eigenvalues corresponding to the leader are $\lambda_l = -1$.

$$\begin{aligned} \{\sigma(\mathcal{D}^+ A_g) : \sigma(\mathcal{D}^+ A_g) = \{\lambda_1, ..., \lambda_n\} \quad | -1 \leq Re(\lambda_i) \leq 1\} \\ \{\sigma(L) : \sigma(L) = \{\lambda_1, ..., \lambda_n\} \quad | -2 \leq Re(\lambda_i) \leq 0\} \quad . \end{aligned} \tag{3.8}$$

So far we have proved that the spectrum of a matrix L is negative and bounded by the origin, which makes matrix L negative semidefinite, this form corresponds to the distributed sensing matrix of relative state measurements. In order to simplify the analysis, we consider the case where $D_1 = D_2 = K$, this condition implies that the relative and self-sensing state feedback signals are uniformly scaled. Under this condition, the general system in Equation (3.1) can be rewritten as:

$$\dot{\boldsymbol{x}} = (I_n \otimes A_n)\boldsymbol{x} - (I_n \otimes K)\, \boldsymbol{x}(t - \tau_1) + (-I_n + \mathcal{D}^+ A_g \otimes K)\, \boldsymbol{x}(t - \tau_2) \tag{3.9}$$

$$K = \begin{bmatrix} 0 & 0 \\ K_p & K_d \end{bmatrix} \tag{3.10}$$

Having analyzed the characteristics of the possible communication matrices L_{comm}, we can further specify the system by defining the nature of the two rationally independent time-delays τ_1 and τ_2. We consider τ_1 as the time-delay incurred by each agent while obtaining it's own state information, we call this delay the *sensing delay*. We also define τ_2 as the linear combination of the *sensing delay* and the *communication delay* induced by the interaction of agents in a communication network, $\tau_2 = \tau_s + \tau_{comm}$. As proven in Theorem 3, the real part of the spectrum of matrix L_g may contain positive eigenvalues and constitutes the limiting case in the stability analysis.

$$\dot{\boldsymbol{x}} = (I_n \otimes A_n)\boldsymbol{x} - (I_n \otimes K)\, \boldsymbol{x}(t - \tau_s) + (L_{comm} \otimes K)\, \boldsymbol{x}(t - (\tau_s + \tau_{comm})) \tag{3.11}$$

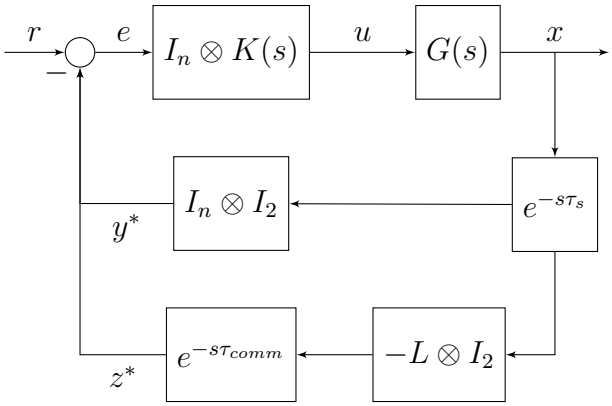


FIGURE 3.1. Block diagram of consensus system with sensing and communication delays.

3.2 Stability Analysis of the Time-Delayed Consensus System

The stability analysis of the consensus system in 3.11 continues by utilizing the decoupling procedure first identified by Cepeda-Gomez and Olgac [24] for time-delayed consensus systems. By virtue of the design of the system and Property 3, the matrices in the left-hand side of the operator "$\otimes$", namely I_n, I_n, L_{comm}, form a commuting set and can be simultaneously diagonalized.

Theorem 5 (Block Diagonalization Property of Linear Consensus Systems). The system in 3.11 can be represented by the block-diagonal system of n blocks, whose block sizes are equal to the order of agents, m. The characteristic equation of the transformed system can be expressed as a product of the n subsystem characteristic equation factors. This theorem holds true if and only if communication matrix $L_{comm} = L$ *or* L_g is non-defective,

$$\begin{aligned} CE(s, K_p, K_d, \tau_s, \tau_{comm}) &= det(sI_{mn} - I_m \otimes A_n - I_m \otimes Ke^{-\tau_s s} - L_{comm} \otimes Ke^{-(\tau_s+\tau_{comm})s}) \\ &= \prod_{j=1}^{n} q_j(s, \lambda_j, K_p, K_d,) \end{aligned} \tag{3.12}$$

Proof. Assuming the communication matrix L_{comm} is non-defective, there exists a non-singular matrix $\boldsymbol{T}$ such that the similarity transformation is the diagonal matrix of eigenvalues $\boldsymbol{T^{-1}}(L_{comm})\boldsymbol{T} \rightarrow \boldsymbol{\Lambda}$, this statement follows the definition of *defective matrices* and the proof can be found in any standard linear algebra book [50]. Finding this matrix is equivalent to transforming the system to modal coordinates $\boldsymbol{z} = T\boldsymbol{x}$. Constructing the Kronecker products $(T \otimes I_2)$ and $(T^{-1} \otimes I_2)$, we perform the similarity transformation:

$$\begin{aligned} A^* &= (T^{-1} \otimes I_2)A(T \otimes I_2) = (T^{-1}I_nT) \otimes (I_2A_nI_2) = I_n \otimes A_n \\ B_1^* &= -(T^{-1} \otimes I_2)I_n(T \otimes I_2) = -(T^{-1}I_nT) \otimes (I_2KI_2) = -I_n \otimes K \\ B_2^* &= (T^{-1} \otimes I_2)L(T \otimes I_2) = (T^{-1}L_{comm}T) \otimes (I_2KI_2) = \Lambda \otimes K \end{aligned} \tag{3.13}$$

$$\dot{\boldsymbol{z}} = A^*\boldsymbol{z} + B_1^*\boldsymbol{z}(t - \tau_s) + B_2^*\boldsymbol{z}(t - (\tau_s + \tau_{comm})) \tag{3.14}$$

Matrices in Equation (3.14) are block diagonal with n blocks $A_i^* \in \mathbb{R}_{m\times m},\ B_{1i}^* \in \mathbb{R}_{m\times m},\ B_{2i}^* \in \mathbb{Z}_{m\times m}$:

$$\begin{aligned} A_i^* &= A_n \\ B_{1i}^* &= -K \\ B_{2i}^* &= \lambda_i K \end{aligned} \tag{3.15}$$

The characteristic equation of each block:

$$CE(s, K_p, K_d, \tau_s, \tau_{comm})^* = det(sI_m - A_n + Ke^{-\tau_s s} - \lambda_i Ke^{-(\tau_s+\tau_{comm})s}) \tag{3.16}$$

Remark (Defective Matrices L_{comm}). In the case of defective matrices, the characteristic equation of the system in Equation (3.11) can be factorized by factors of order equals to $2(m + 1)$, where $m = \mathcal{M}_a(\lambda_i) - \mathcal{M}_g(\lambda_i)$, the algebraic and geometric multiplicities of eigenvalue λ_i, respectively. This can be proven by the previous process and the

Jordan decomposition theorem [50].

3.2.1 CTCR Deployment for Time-Delayed Factors

The time-delayed factors in Equation (3.16) correspond to a LTI-MTDS system of the neutral type. The time-delayed MAS system will be stable if an only if the *disagreement dynamics* factors are stable and the *group decision dynamics* are stable as well. Applying the CTCR paradigm introduced in Section 2.2, we can find the stability maps of the system by following Algorithm 1. The first step is establishing the number of unstable roots NU_0 of the undelayed system, i.e Equation (3.16) where $\tau_{comm} = 0$ and $\tau_s = 0$. In order to carryout the numerical portion of the algorithm we substitute the system dynamics of the linearized unicycle:

$$A_n = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \tag{3.17}$$

$$CE(s, K_p, K_d)^* = s^2 + K_d(1 - \lambda_j)s + K_p(1 - \lambda_j) = 0 \tag{3.18}$$

Recalling λ_j is an arbitrary eigenvalue of the communication matrix, L_{comm}, and the results in Theorem 4, the limiting case of the analysis corresponds to $L_{comm} = L_g$, we continue with this assumption, however, the analysis can be completed considering the spectrum of $L_{comm} = L_{rel}$. Accordingly, the eigenvalues λ_j are in the disk $D(0, 1)$ and the largest value is $\lambda_j = 1$. This corresponds to the marginal stability factor, equivalent to the *group consensus dynamics* and occurs in systems where the *strongly connected* condition, introduced in Section 2.1, is met. This factor produces two roots at the origin. All other factors are asymptotically stable for $K_p > 0, K_d > 0$, these factors correspond to the *disagreement dynamics*.

The second step involves finding the exact marginal stability curves of the system by performing the half-angle tangent substitution. This substitution transforms each factor 3.16 to pseudo-polynomial form in ω, parametrized by coefficients K_p, K_d, λ and the

half-angle tangent spectral delay factors $\tau_s \to z_1, \tau_{comm} \to z_2$ defined in Equation (2.33):

$$
\begin{aligned}
&CE(\omega, K_p, K_d, z1, z2)^* \\
&\quad = -\omega^2 \\
&\quad + \omega K_d \left(\lambda_j \left(\frac{4iz_1z_2}{(z_1^2+1)(z_2^2+1)} - \frac{2(1-z_1^2)z_2}{(z_1^2+1)(z_2^2+1)} - \frac{2z_1(1-z_2^2)}{(z_1^2+1)(z_2^2+1)} - \frac{i(1-z_1^2)(1-z_2^2)}{(z_1^2+1)(z_2^2+1)} \right) \right. \\
&\qquad \left. + \frac{2z_1}{z_1^2+1} + \frac{i(1-z_1^2)}{z_1^2+1} \right) \\
&\quad + K_p \left(\lambda \left(\frac{4z_1z_2}{(z_1^2+1)(z_2^2+1)} + \frac{2i(1-z_1^2)z_2}{(z_1^2+1)(z_2^2+1)} + \frac{2iz_1(1-z_2^2)}{(z_1^2+1)(z_2^2+1)} + \frac{(z_1^2-1)(1-z_2^2)}{(z_1^2+1)(z_2^2+1)} \right) \right. \\
&\qquad \left. - \frac{2iz_1}{z_1^2+1} + \frac{1-z_1^2}{z_1^2+1} \right) = 0
\end{aligned}
\tag{3.19}
$$

The common root of the imaginary and real parts of Equation (3.19) is a non-linear ex-

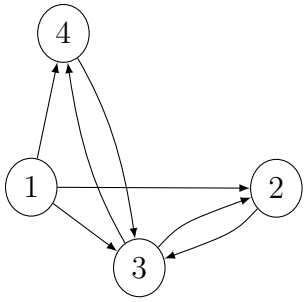


FIGURE 3.2. Topology example - one connected leader (1) , three fully connected followers (2,3,4).

pression with 126 irreducible terms. In order to illustrate the technique we analyze the stability of the system defined by the graph in Figure 3.2. This system has one connected component and one leader. The scaled adjacency matrix for this system is shown in Equation 3.20, with its diagonal form [2].

[2]This matrix has two trivial eigenvalues, $\lambda_1 = \lambda_2 = 0$, one corresponding to the spanning tree where the root is the leader, and one corresponding to the connected com-

$$\mathcal{D}^{+}A_g = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \end{bmatrix}$$
$$\Lambda = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & -0.57735 & 0 \\ 0 & 0 & 0 & 0.57735 \end{bmatrix} \tag{3.20}$$

The time delayed factors are in the form 3.16. By Equation (3.18), the number of unstable roots $NU_0 = 0$ when $\boldsymbol{\tau} \to 0$. By selecting gain values $K_p = K_d = 1$ we find the loci of root transitions or the points of marginal stability by finding the zero level contour of the common roots of the imaginary and real parts of Equation 2.33. The points in this contour, corresponding to the marginal stability loci in the SDS space, are used to find the corresponding values of the crossing frequencies ω_c, these factors generate the continuous 3D $[z_1, z_2, \omega_c]$ kernel curves shown in Figure 3.4. The *Kernel Curves* in Figure 3.3 correspond to the $0 < v_1 < 2\pi$, $0 < v_2 < 2\pi$ domain projection of the 3D curves in the v_1, v_2 plane. This projection can be seen in Figure 3.3, along with its representation in the $\{\tau_s, \tau_{comm}\}$ domain.

We can readily observe that the MAS stability is independent of τ_{comm} for small τ_s. We zoom into the square region $0 < \tau_s < 5$, $0 < \tau_{comm} < 5$ region to emphasize this fact, and mark the stability region in gray, along with the number of unstable roots in each of the areas bound by the transition curves. This can be seen in Figure 3.5. From this analysis we can conclude that the disagreement factors generated by eigenvalues $\lambda_3 = -0.57735$

---

ponent of followers, this matrix is non-defective and the repeated trivial eigenvalue has a geometric multiplicity of 2

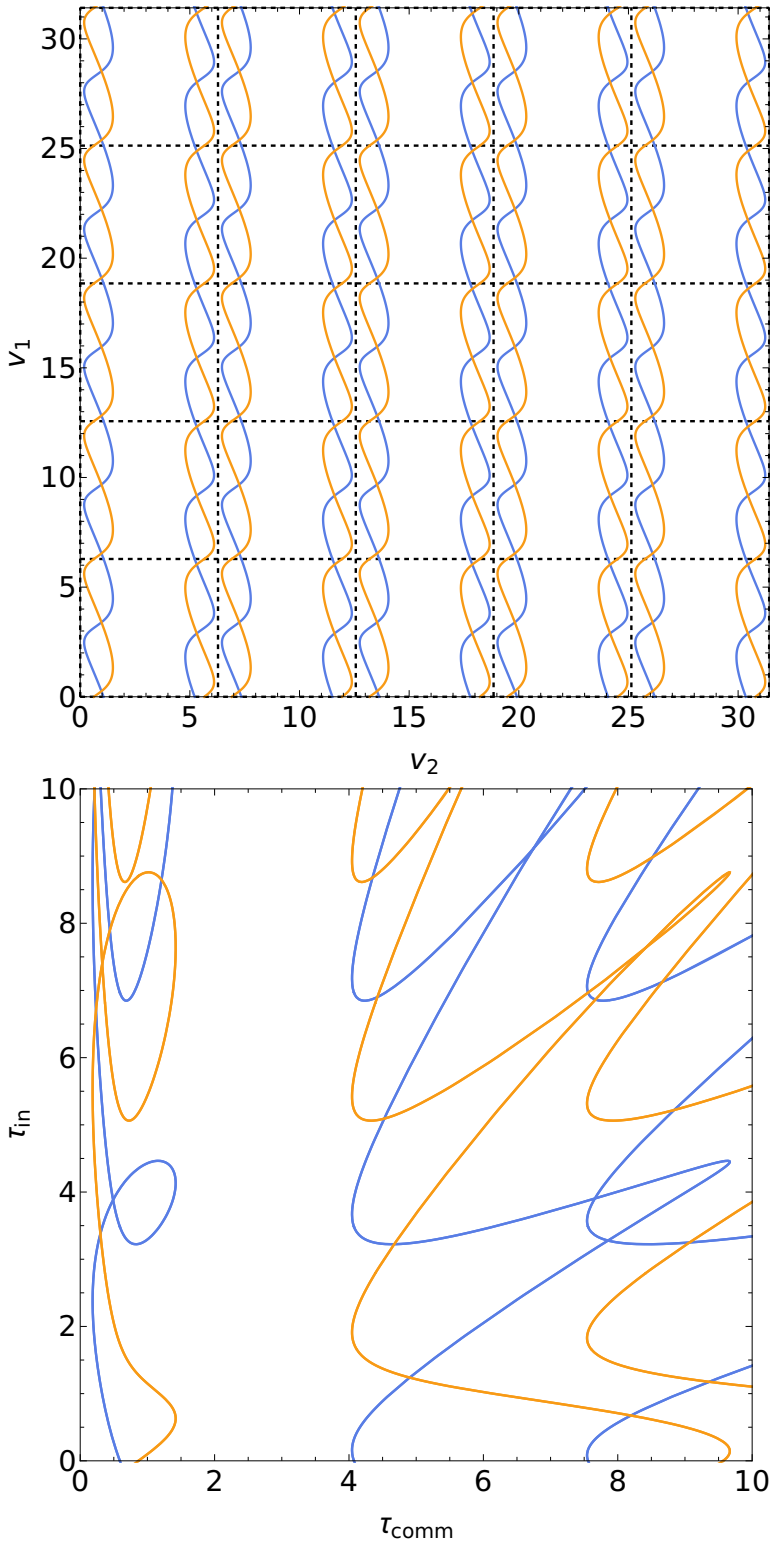


FIGURE 3.3. Marginal stability curves for example topology in Figure 3.2 - $0 < \boldsymbol{\tau} < 10$**.**

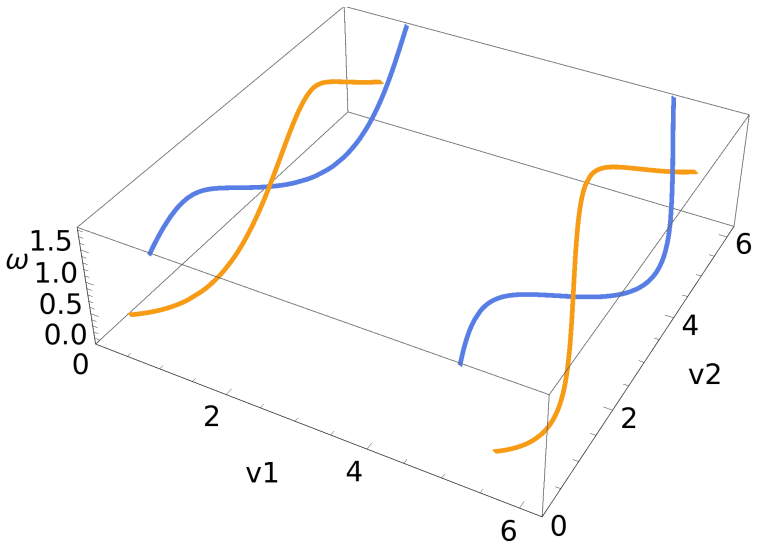


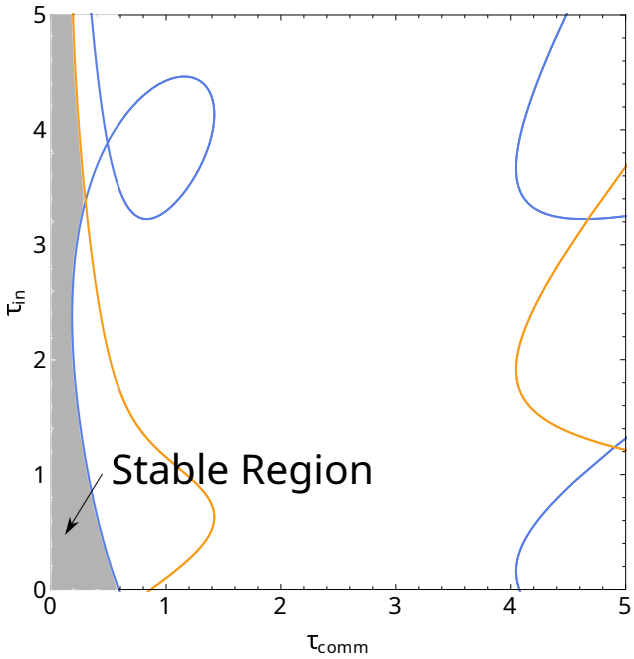
FIGURE 3.4. Marginal stability 3D kernel curves in SDS domain for topology in Figure 3.2.

and $\lambda_4 = 0.57735$ represent the absolute stability bounds of the system, these curves correspond to those represented by red and green lines, respectively, in Figures 3.5a, 3.4 and 3.3. A system characterized by delays below this 2D boundary will be asymptotically stable.

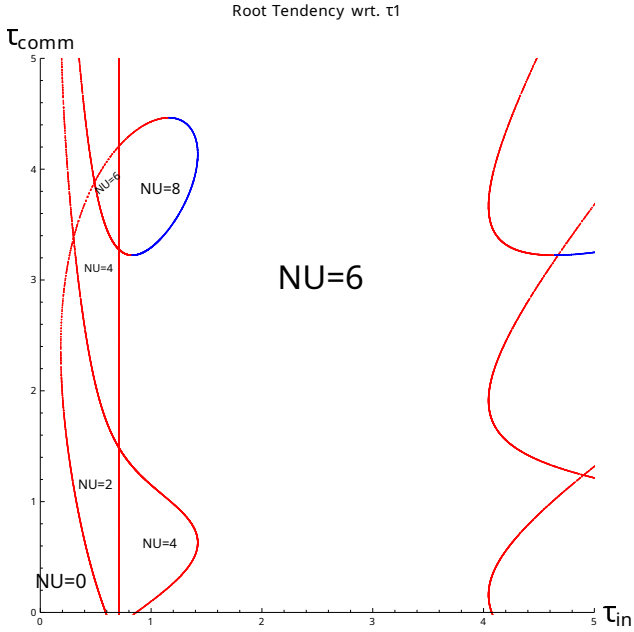
Remark (Asymptotic Stability of Time-Delayed Factors). Consensus protocol factors in Equation (3.12) are asymptotically stable and converge to zero if and only if the time delay vector $\boldsymbol{\tau} = \{\tau_s, \tau_{comm}\}$ is within the region of stability defined by the number of unstable roots found by utilizing the CTCR method.

3.3 Formation Control of Time-Delayed Multi-Agent Systems

The stability of the *disagreement factors* in the closed system 3.11 guarantee that the agents will reach a common state, this may imply that the agents in the formation will collide with one another. In formation problems it is desired for the agents to form a specific shape and/or track a trajectory.



(a) **Stable region.**



(b) **Root tendencies as τ_{comm} increases - red denotes unstable transitions, blue denotes stable transitions.**

FIGURE 3.5. Stability picture for topology in Figure 3.2.

3.3.1 Formation Regulation Control in Time-Delayed Consensus Systems

We introduce a time-varying forcing term $\boldsymbol{\phi}(t)$, containing the relative states of the agents defining the formation structure in the un-transformed domain of x. Function $\boldsymbol{\phi}(t)$ is expressed in terms of decoupling matrix $F(s)$, such that it solves the formation regulation problem. Following the convention in this thesis, $\boldsymbol{\phi(t)}$ refers to the vertically concatenated forcing functions for the multi-agent system, as defined in Equation (3.21).

$$\boldsymbol{\Phi}(t) = (F_r \otimes K)\boldsymbol{\phi}(t) \tag{3.21}$$

$$\dot{\boldsymbol{x}} = (I_n \otimes A_n)\boldsymbol{x} - (I_n \otimes K)\, \boldsymbol{x}(t-\tau_s) + (L \otimes K)\, \boldsymbol{x}(t-(\tau_s+\tau_{comm})) + \boldsymbol{\Phi}(t) \tag{3.22}$$

Theorem 6 (Formation Regulation). For the general block-diagonalizable delayed consensus system of second-order agents defined by system matrix A, feedback matrices I_n and L, control matrix K and control input $\boldsymbol{\Phi(t)}$.

$$(I_n \otimes A)\boldsymbol{x} - (I_n \otimes K)\, \boldsymbol{x}(t-\tau_1) + (L \otimes K)\, \boldsymbol{x}(t-\tau_2) + \boldsymbol{\Phi}(t) \tag{3.23}$$

$\boldsymbol{\Phi}(t)$ sufficiently solves the formation regulation problem if $F_r = (I_n - L)$ and the n decoupled factors z_j are stable.

Proof. The frequency response analysis of the decoupled system with the system matrices defined in Equation (3.15) can be utilized to find the form of matrix F_r. Decoupling the system by Theorem 5, we refer to Equation (3.13) for the definition of matrices A^*, B_1^* and B_2^*. Matrices with superscript $*$ are block-diagonal matrices.

$$\begin{aligned} \boldsymbol{\chi}(t) &= (T \otimes I_2)\boldsymbol{\phi}(t) \\ \boldsymbol{\Phi^*}(t) = (T^{-1} \otimes I_2)\boldsymbol{\chi}(t) &= (T^{-1}F_rT \otimes K)\boldsymbol{\phi}(t) \end{aligned} \tag{3.24}$$

$$\dot{\boldsymbol{z}} = A^*\boldsymbol{z} + B_1^*\boldsymbol{z}(t-\tau_1) + B_2^*\boldsymbol{z}(t-\tau_2) + \boldsymbol{\Phi^*}(t) \tag{3.25}$$

The frequency output response of the transformed MIMO system 3.25:

$$\boldsymbol{Z(s)} = [sI_{2n} - A^* - I_n^* e^{-\tau_1 s} - L^* e^{-\tau_2 s}]^{-1} \boldsymbol{\Phi^*(s)} \tag{3.26}$$

In its general form $\boldsymbol{\Phi^*(s)}$ is not block diagonal, or if it is, it is not guaranteed to have block sizes compatible with the blocks of matrices A^*, B_1^*, B_2^* appearing in the consensus system. However, we can restrict the form of matrix F_r such that the system can be decoupled by ensuring the similarity transformation $T^{-1}F_rT$ diagonalizes matrix F_r. The choice of F_r evident considering modal matrix T contains the spectrum of the feedback matrices, in the left-hand side of the Kronecker product, and simultaneously diagonalizes their linear combination $-\beta I_n + \gamma L$. Hence, without loss of generality, matrix F_r can have the following form:

$$\begin{aligned} F_r &= (\beta B_1 + \gamma B_2) \\ T^{-1}F_rT &= (\beta B_1^* + \gamma B_2^*) \end{aligned} \tag{3.27}$$

Substituting into Equations 3.24 and 3.25, by virtue of the block diagonal properties and compatibility of the matrices in the system, we can express the transfer function matrix [41] of the system in terms of the n second order subsystems:

$$\boldsymbol{H_j(s)} = [sI_2 - A_j^* - B_{1j}^* e^{-\tau_s s} - B_{2j}^* e^{-(\tau_s + \tau_{comm})s}]^{-1} \begin{bmatrix} 0 & 0 \\ K_p(\gamma\lambda_j - \beta) & K_d(\gamma\lambda_j - \beta) \end{bmatrix} \tag{3.28}$$

This expression can be rewritten in terms of the determinant/adjoint form of the matrix inverse, this results in an expression in terms of the n characteristic equations in Equation (3.12) and the product of the matrix adjoint and the decoupling factor:

$$\boldsymbol{H_j(s)} = \frac{1}{CE_j^*(s, K_p, K_d, \tau_1, \tau_2)} \begin{bmatrix} K_p\left(\gamma\lambda_j - \beta\right) & K_d\left(\beta + \gamma\lambda_j\right) \\ sK_p\left(\gamma\lambda_j - \beta\right) & sK_d\left(\gamma\lambda_j - \beta\right) \end{bmatrix} \tag{3.29}$$

Assuming all decoupled factors are desired to reach a stationary relative position $z_j(\infty) \rightarrow z_{jf}$, we find the steady-state response to a scaled step input.

$$Z_j^*(t \rightarrow \infty) = \lim_{s \rightarrow 0} sH_j(s) \begin{bmatrix} \frac{z_{jf}}{s} \\ 0 \end{bmatrix} = \begin{bmatrix} \frac{z_{jf}(\beta - \gamma\lambda_j)}{\lambda_j - 1} \\ 0 \end{bmatrix} = \begin{bmatrix} z_{if} \\ 0 \end{bmatrix} \tag{3.30}$$

We find that for matrix F_s to decouple the system, constants $\gamma = \beta = -1$, for any eigenvalue λ_j if and only if the factors are stable and the closed loop system matrix $sI_2 - A_j^* - B_{1j}^* e^{-\tau_1 s} - B_{2j}^* e^{-\tau_2 s}$ is non-singular. These are the sufficient conditions to solve the formation regulation problem in a system governed by the control scheme in Equation (3.22), thus proving Theorem 6. Using this form, the steady-state error to sudden step-like formation shape changes is zero. The final form of the forcing function for the system with global feedback is defined as:

$$\mathbf{\Phi}(t) = \left(\left(I_n - \mathcal{D}^+ A_g\right) \otimes K\right) \boldsymbol{\phi}(t) \tag{3.31}$$

The form for Equation (3.31) is convenient for implementation in distributed systems since it does not require any ahead-of-time calculations, does not depend on the formation shape or the dynamics of the individual agents and depends solely on the general structure of the feedback scheme (i.e. the communication and sensing capabilitites of the system). An agent, cognizant of it's position in the formation, may interchange this knowledge with its informers and each member of the system can calculate the forcing factor required to participate in the formation in a distributed manner.

3.3.2 Formation Tracking Control in Time-Delayed Consensus Systems by Connected Leaders

In a system with multiple leaders as part of the formation, each leader will be independent of the state of the rest of the group, the analysis of the leaders is equivalent to the analysis of the disagreement factor corresponding to the trivial eigenvalue of $\mathcal{D}^+ A_g$. In

the state-space the dynamics of the leader are presented below, where $\boldsymbol{x}$ is the state of the leader $\boldsymbol{x} = [x_1, \dot{x_1}]$:

$$\dot{\boldsymbol{x}} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \boldsymbol{x} - \begin{bmatrix} 0 & 0 \\ K_p & K_d \end{bmatrix} \boldsymbol{x}(t - \tau_1) \tag{3.32}$$

This is a LTI system with a single time delay in the feedback channel. The stability of the leader factor can be analyzed using the CTCR method, and was part of the analysis performed to produce the complete stability maps of the sample system in Figure 3.3. The stability margin for this factor is the curve in yellow in Figure 3.5. For the leader to track a desired trajectory $\boldsymbol{x_d} = [x_d(t), \dot{x_d}(t)]$, we introduce the exogenous input T $= \begin{bmatrix} 0 \\ K_p x_d + K_d \dot{x}_d \end{bmatrix}$ and rewrite the equation in terms of error $e(t) = x_d(t) - x(t)$ and exogenous input $x_d(t)$. For conciseness non-delayed terms $x_d(t)$, $e(t)$ and $x(t)$ are written without explicit mention of their time dependence. We find the $H(s) = \frac{E(s)}{X_d(s)}$ transfer function and analyze the steady-state response of the system to a step input $1/s$, and to a ramp input $1/s^2$:

$$\begin{gathered} \ddot{x}_d - \dot{x}_d - x_d + K_d \dot{x_d}(t - \tau_s) + K_p x_d(t - \tau_s) = \ddot{e} + K_d \dot{e}(t - \tau_s) + K_p e(t - \tau_s) \\ H(s) = \frac{s^2 + (K_d e^{-s\tau_s} - 1)s + (K_p e^{-s\tau_s} - 1)}{s^2 + K_d e^{-s\tau_s} s + K_p e^{-s\tau_s}} \end{gathered} \tag{3.33}$$

$$e(t \to \infty) = \lim_{s \to 0} sH(s) \frac{1}{s} = 0 \tag{3.34}$$

$$e(t \to \infty) = \lim_{s \to 0} sH(s) \frac{1}{s^2} = \tau_s \tag{3.35}$$

We can see that the tracking error for step inputs is equal to 0, and the tracking error for ramp input is bounded by the value of the sensing delay τ_s. External access to the dynamics of the leader allows us to drive a single or multiple leaders to set the formation in a global frame. As mentioned in the formation regulation problem, the time-varying formation shape defines the relative position of the agents in space. For formations with one or more leaders, the decoupled nature of the leader agent guarantees the shape is fixed at

the positions of the leaders. We assert, without proof, that accurate formation tracking is guaranteed when the *disagreement dynamics* are stable and the tracking performance of the leader is acceptable. A proof may follow from results in cascading systems theory, focusing on the relative strength of the inner loop, corresponding to individual global measurement feedback, and the decoupled nature of the leader terms in the consensus system. We can represent the tracking input as part of the control scheme by noting that the matrix P , $P \in \mathbb{Z}_{n\times 1}$ of row sums of matrix $Q = I_n - \mathcal{D}^+ A_g$ has entries equal to one, in rows corresponding to the leaders, and equal to zero, in rows corresponding to the followers. Recalling $\boldsymbol{x_d}(t) = [x_d(t), \dot{x}_d(t)]$ and multiplying:

$$P = \left[\sum_{i=1}^{n} q_{ij}\right] \tag{3.36}$$

$$\mathbf{T}(t) = (P \otimes K)\boldsymbol{x_d}(t) \tag{3.37}$$

The final form of the consensus system with formation tracking and regulation capabilities is defined in terms of exogenous inputs $\mathbf{\Phi}(t)$ in Equation (3.31) and $\boldsymbol{T}(t)$ in Equation (3.37) as follows:

$$\dot{\boldsymbol{x}} = (I_n \otimes A_n)\boldsymbol{x} - (I_n \otimes K)\ \boldsymbol{x}(t-\tau_s) + (\mathcal{D}^+ A_g \otimes K)\ \boldsymbol{x}(t-(\tau_s+\tau_{comm})) + \mathbf{\Phi}(t) + \mathbf{T}(t) \tag{3.38}$$

3.4 Distributed Non-Linear Predictor for Input-Output Feedback Linearization of Unicycles with Sensing Delays

Recalling Equation (2.47), the input-output feedback compensator for each DWMR depends on non-delayed state variable θ. The compensation has to be exact in order to establish that the marginal stability points found in the analysis of the linear consensus system are equivalent to those of the non-linear non-holonomic system. In the literature of networked control systems we find the use of predictors[51, 52] to mitigate the corrupt-

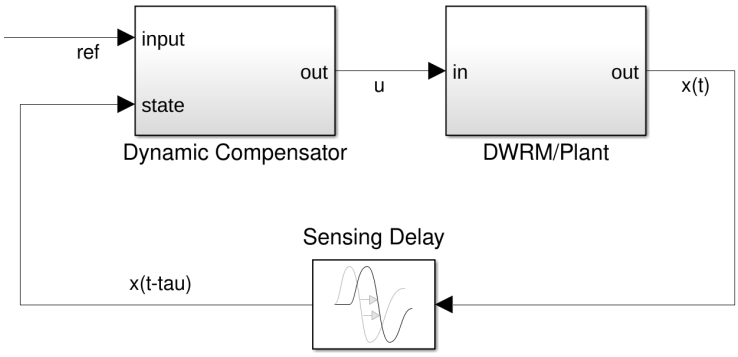


FIGURE 3.6. Distributed agent - without predictor.

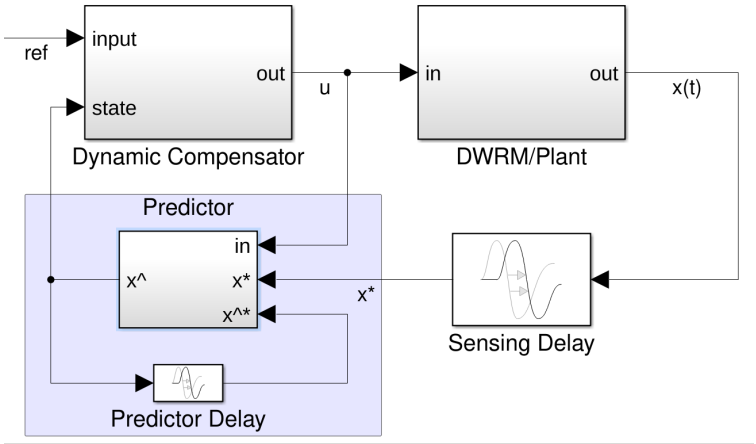


FIGURE 3.7. Distributed agent - with delayed state predictor.

ing effect of time-delays in feedback systems. We adapt the work of Kojima et al. [53] to design a *local* non-linear predictor to use in conjunction with the dynamic compensator in Equation (2.47) to linearize the system. We focus on a distributed prediction scheme, where each agent is in charge of predicting their state based on time-delayed measurements. The predictor is a full-order predictor due to the coupling of the state and the inputs of the non-holonomic system. The agent without a predictor is shown in Figure 3.6. The agent with the predictor is shown in Figure 3.7. Defining the following quantities, $\hat{\boldsymbol{x}}$ is the predictor state, $\boldsymbol{x}^*$ is the time-delayed state measurement, $\hat{\boldsymbol{x}}^*$ is the time-delayed

predictor state. The delayed unicycle has the following dynamics:

$$\dot{\boldsymbol{x}}^* = \begin{cases} \dot{x^*} = v(t)cos\left(\theta(t-\tau_s)\right) \\ \dot{y^*} = v(t)sin\left(\theta(t-\tau_s)\right) \\ \omega^* = \theta(t-\tau_s) \end{cases} \tag{3.39}$$

The state predictor dynamics are defined below, where $K_{pred} = diag(k_1, k_2, k_3)$:

$$\begin{bmatrix} \dot{\hat{x}} \\ \dot{\hat{y}} \\ \dot{\hat{\theta}} \end{bmatrix} = \begin{bmatrix} v(t)cos\ \hat{\theta}(t) \\ v(t)sin\ \hat{\theta}(t) \\ \omega(t) \end{bmatrix} - K_{pred} \begin{bmatrix} \hat{x}^* - x^* \\ \hat{y}^* - y^* \\ \hat{\theta}^* - \theta^* \end{bmatrix} \tag{3.40}$$

Kojima et al. [53] present sufficient conditions for the uniform asymptotic convergence to zero of the prediction error $\boldsymbol{e} = \hat{\boldsymbol{x}} - \boldsymbol{x}$, the conditions are presented below, and depend on the matrix K_{pred}, the constant sensing delay τ_s, and the maximum velocity input $\bar{v}$.

Theorem 7 (Asymptotic Convergence of Predictor). The non-linear predictor in Equation 3.40, for a constant time-delay τ_s, asymptotically converges to zero if:

$$\begin{cases} (k_1\bar{v} + 2k_1^2)\tau_s < -\bar{v} + 2k_1 \\ (k_2\bar{v} + 2k_2^2)\tau_s < -\bar{v} + 2k_2 \\ (k_1 + k_2)\bar{v} + k_3^2)\tau_s < -2\bar{v} + k_3 \end{cases} \tag{3.41}$$

Under the simplifying assumptions $k_1 = k_2$, the sufficient conditions for asymptotic convergence are presented below:

$$\begin{cases} k_1 > \frac{\bar{v}}{2} \\ k_3 > 2\bar{v} \\ \tau_s < min(\frac{2k_1 - \bar{v}}{k_3 - 2\bar{v}}, \frac{k_1(2k_1 + \bar{v})}{2k_1\bar{v} + k_3^2}) \end{cases} \tag{3.42}$$

Proof. Follows Lyapunov-Razumikhin's theorem and can be found in Kojima et al. [53].

CHAPTER 4

NUMERICAL EXAMPLES, VALIDATION AND DISCUSSION

4.1 Example MAS of Non-Holonomic Agents

We continue with a numerical example corresponding to a directed butterfly graph of 12 agents. This graph is shown in Figure 4.1, in this example the layout of the graph also defines the positions of the agents in the formation, but it can be completely arbitrary as long as the communication topology does not change. We highlight the 2 identical connected components in red and orange. The two connected components are coupled by intermediate agents 6 and 9. The leaders of this formation are agents 3 and 12, we expect

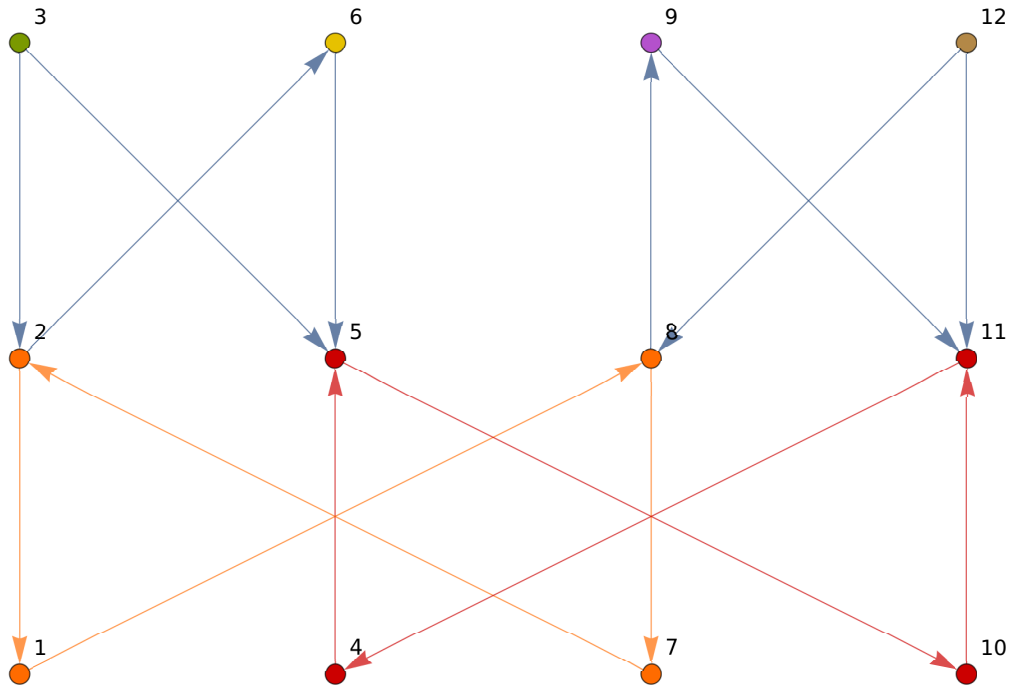


FIGURE 4.1. Directed butterfly topology - 12 agents, 2 leaders, 2 connected components.

to see 2 trivial eigenvalues corresponding to the leaders, and 2 corresponding to the connected components. For 12 linearized unicycle agents, the identity matrix is of size 12×12 $I_n = I_{12}$, A_n is defined in Equation (3.17) and gain matrix K is defined in Equation (3.10), with gains $K_p = 0.2, K_d = 1$. These gains can be set using optimal control techniques

on the dynamics of the leader agents. The scaled adjacency $\mathcal{D}^+A_g$ for this topology is expressed below, along with a list of eigenvalues.

$$\mathcal{D}^+A_g = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{1}{2} & 0 & 0 & 0 & \frac{1}{2} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & \frac{1}{3} & \frac{1}{3} & 0 & \frac{1}{3} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ \frac{1}{2} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{2} \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{3} & \frac{1}{3} & 0 & \frac{1}{3} \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{4.1}$$

This matrix is non-defective and has two complex conjugate eigenvalue pairs; the complex conjugate eigenvalue pairs don't pose a problem to the CTCR method.

$$\sigma\left(\mathcal{D}^+A_g\right) = \left\{-\frac{1}{\sqrt{2}}, \frac{i}{\sqrt{2}}, -\frac{i}{\sqrt{2}}, \frac{1}{\sqrt{2}}, -\frac{1}{\sqrt{3}}, \frac{i}{\sqrt{3}}, -\frac{i}{\sqrt{3}}, \frac{1}{\sqrt{3}}, 0, 0, 0, 0\right\} \tag{4.2}$$

The marginal stability curves for this system are presented in Figure 4.2. For presentation purposes we only show the kernel curves and the curves in the $0 < \tau < 3$ region.

We see that the stability of the system is independent of the communication delay for small τ_{in}. The maximum allowable sensing delay for the formation is $\tau_s = 0.63$ for any τ_{comm}. The formation regulation matrix, defined in Equation 3.31, is numerically expressed in Equation (4.3).The leader mapping matrix, defined in Equation (3.37) is computed for

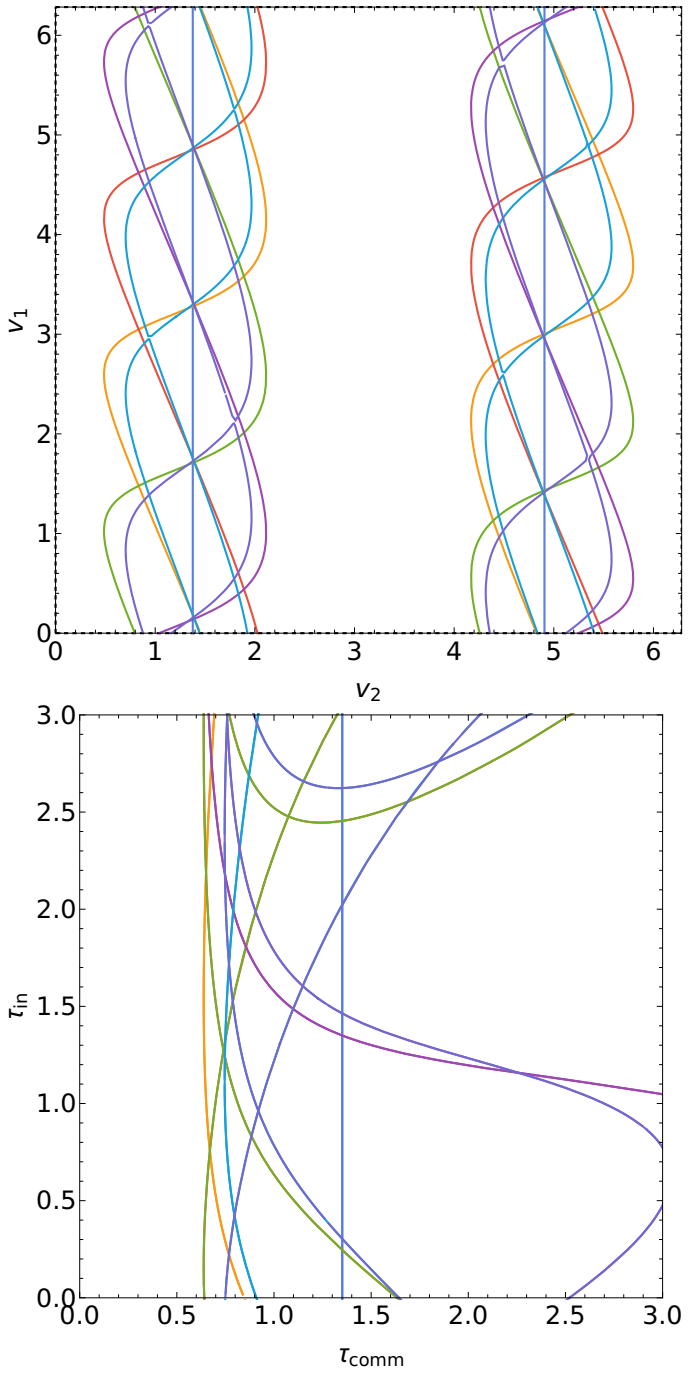


FIGURE 4.2. Marginal stability loci for directed butterfly topology - kernel curves for directed butterfly topology in Figure 4.1 and loci in $0 < \tau < 3$.

the formation with two leaders.

$$\boldsymbol{F_r} = \begin{bmatrix} 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & -\frac{1}{2} & 0 & 0 & 0 & -\frac{1}{2} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & -\frac{1}{3} & -\frac{1}{3} & 1 & -\frac{1}{3} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 \\ -\frac{1}{2} & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & -\frac{1}{2} \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -\frac{1}{3} & -\frac{1}{3} & 1 & -\frac{1}{3} \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \quad \boldsymbol{P} = \begin{Bmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{Bmatrix} \tag{4.3}$$

The relative position vectors for the x and y direction consensus controllers, corresponding to the grid position generates constant exogenous inputs $\phi(t)$:

$$\phi(t)_x = \begin{Bmatrix} 1 \\ 0 \\ 1 \\ 0 \\ 1 \\ 0 \\ 2 \\ 0 \\ 2 \\ 0 \\ 2 \\ 0 \\ 3 \\ 0 \\ 3 \\ 0 \\ 3 \\ 0 \\ 4 \\ 0 \\ 4 \\ 0 \\ 4 \\ 0 \end{Bmatrix} \quad \phi(t)_y = \begin{Bmatrix} 1 \\ 0 \\ 2 \\ 0 \\ 3 \\ 0 \\ 1 \\ 0 \\ 2 \\ 0 \\ 3 \\ 0 \\ 1 \\ 0 \\ 2 \\ 0 \\ 3 \\ 0 \\ 1 \\ 0 \\ 2 \\ 0 \\ 3 \\ 0 \end{Bmatrix} \tag{4.4}$$

In the control scheme the only unknown is the gain matrix K_{pred} for the distributed delayed state predictor. For the experimental plant [31], the maximum linear velocity input was found to be $\bar{v} = 0.1m/s$, we analyze the maximum allowable sensing delay in the predictor by using inequalities in 3.42, this limit is conservative in nature

due to the *Lyapunov-Krasovskii* functional methodology to use the boundary. Sweeping $0 < k_1 < 5, 0 < k_3 < 0$ we obtain the surface plot defining the conservative maximum allowable sensing delay τ_s for a combination k_1, k_3. Restricting the domain to the maximum allowable delay of the plant, we set $k_1 = 0.9$, and any gain $0.2 > k_3 > 6$ will ensure the predictor error has asymptotic stability and converges to zero. This is shown in Figure 4.3.

4.1.1 Model Implementation

A model of the consensus system was implement in *Mathwork's Matlab-Simulink*, the general structure of the model is composed of an input subystem that generates the time-dependent spacing and trajectory factors. Two identical consensus controllers, one for each coordinate x and y, and a subsystem including all of the agents, each containing the input-output linearization compensator, the observer and the unicycle model. The agent subsystem outputs state pairs $x, \dot{x}$, $y, \dot{y}$ and θ, ω, for the entire formation independently. The state is delayed by a transport delay block and fed back into the consensus controllers. The outer layer of the model can be seen in Figure 4.4. The consensus controllers graphically implements the linear equation system, it takes three inputs, the delayed state, the inter-agent formation signal $\boldsymbol{\phi}(t)$, and the trajectory signal $\boldsymbol{x_d}(t)$, this can be seen in Figure 4.5. Each agent block, shown in Figure 4.6 includes the distributed control logic, including the non-linear predictor and the input-output compensator. It has two input ports, one for each direction, and a single output signal, containing the full predicted state, as well as its derivatives. Blocks with labels referring to the state variables x, y, θ represent routing blocks to extract the specified variables. The unicycle, predictor and input-output compensator are implemented as compiled *C Level-2 s-functions*. Each agent exports its prediction error for further analysis. One addition to the non-linear unicycle model was the inclusion of actuator saturation limits, for both linear and angular velocity units. The *e-puck* robot [31] has a maximum linear velocity $\bar{v} = 0.1$ m/s, an a wheelbase of 4.1 cm, resulting in a maximum angular velocity of 4.87 rad/s.

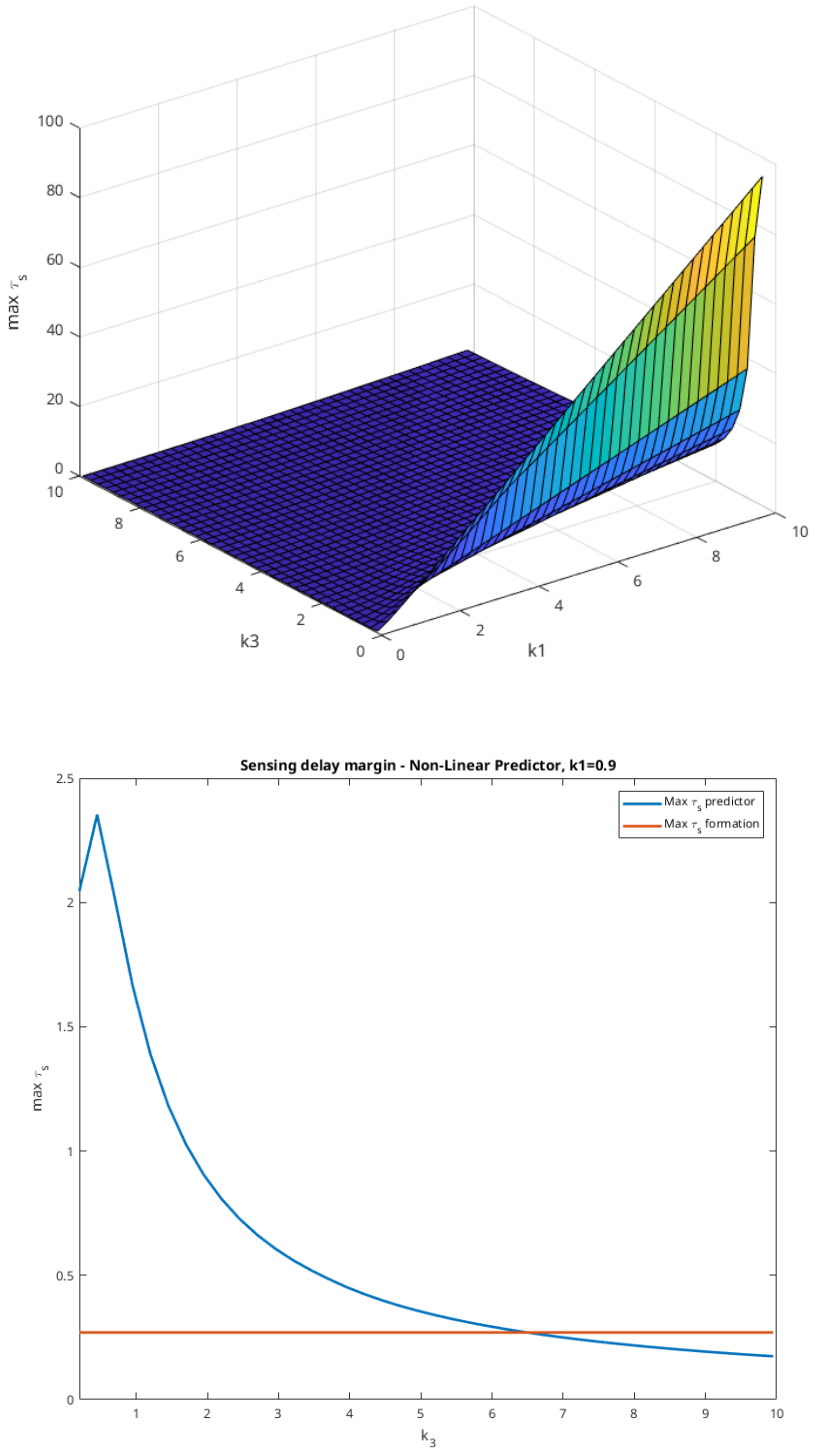


FIGURE 4.3. Time-delay margins vs gain selection - non-linear predictor.

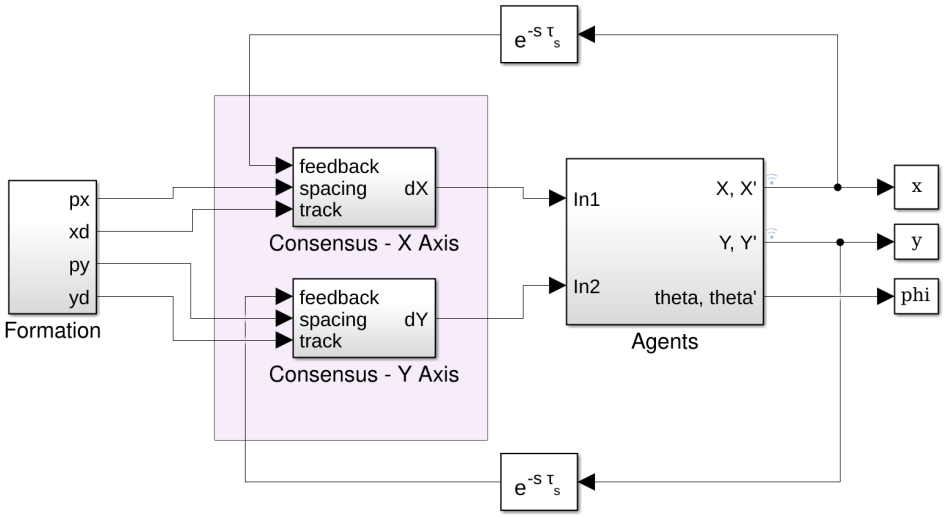


FIGURE 4.4. Consensus system model - outer layer.

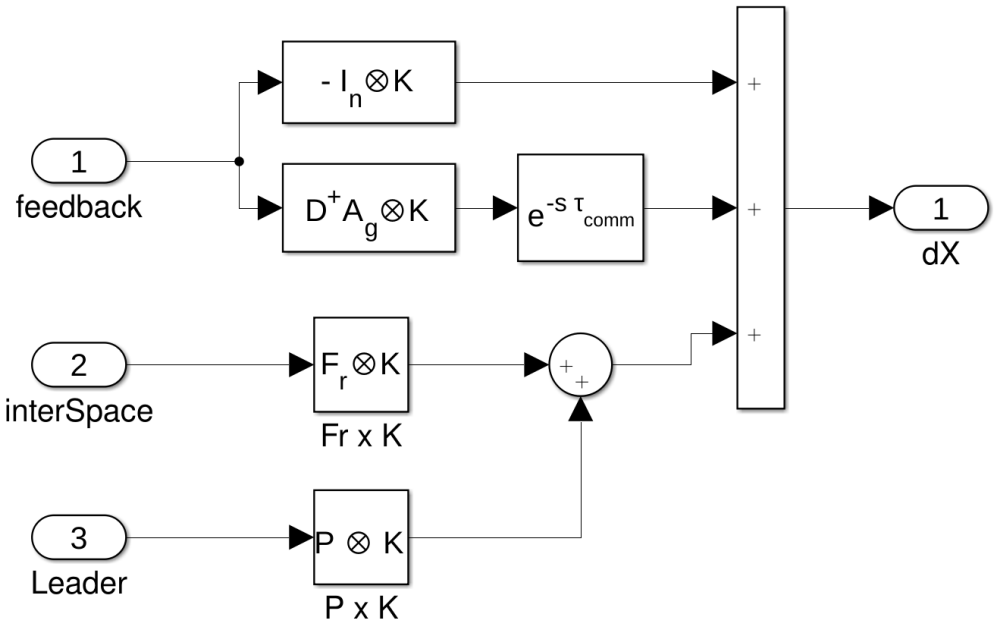


FIGURE 4.5. Consensus system model - consensus controller.

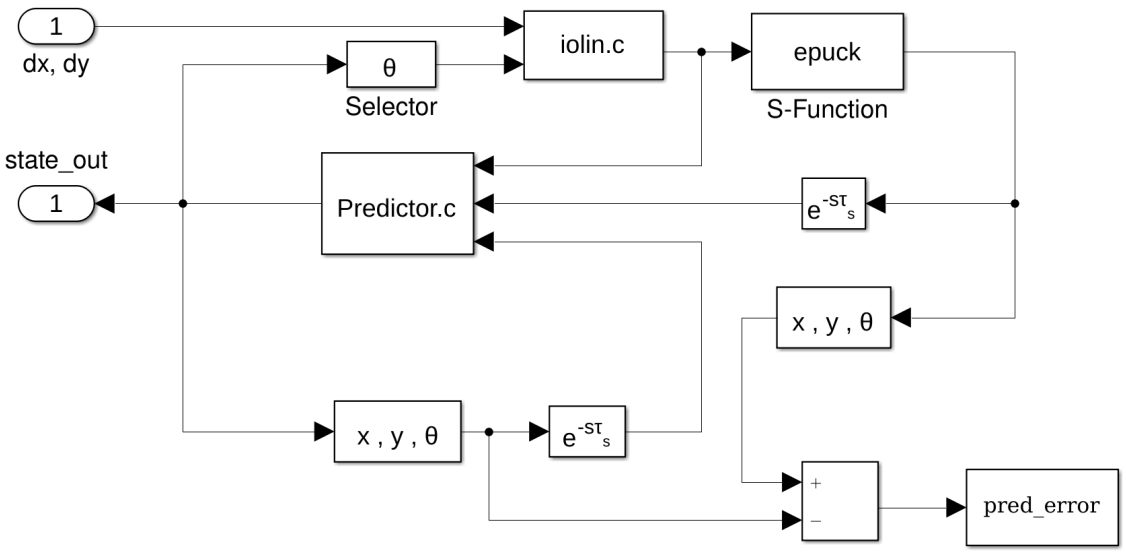


FIGURE 4.6. Agent model - predictor, IO compensator and non-linear plant.

Signal routing between the consensus controllers and the agents was handled by defining two s-functions, a *breakCat* function that takes in two channels, each with n input signals, and separates them into n channels each with two signals, one per control input in x and y, and a *joinCat* function where the states of each individual agent is concatenated and separated into three signals, one for each state variable and its derivative. The number of inputs and outputs of each signal routing block is dynamically calculated by providing the number of agents as a parameter. The trajectory generation block contains the sources for the formation keeping and tracking signals and is not shown.

4.2 Simulation Results

In this section we execute the model defined previously and analyze each of the aspects discussed in Chapter 3. We summarize the simulation parameters in Table 4.1.

4.2.1 Prediction Error

Before analyzing the overall performance of the formation, we show the convergence of the distributed non-linear predictor. As expected by the design conditions, the error asymptotically converges down to zero. The initial conditions of the predictor and the agent

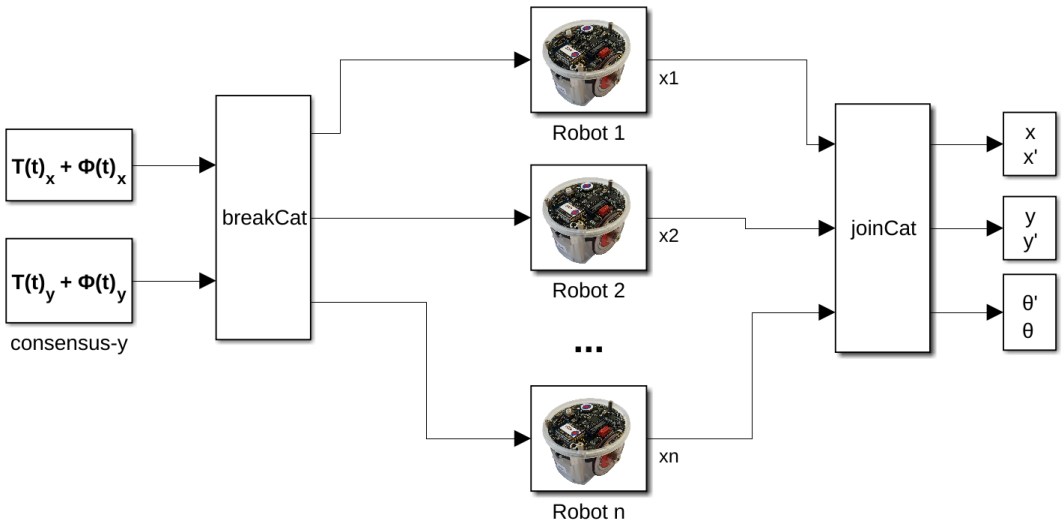


FIGURE 4.7. Signal routing model.

TABLE 4.1. Simulation Parameters

Parameter	Symbol	Value
Solver	-	ode45 - *Runge-Kutta (4,5)*
Step-Size	δt	4e−3 seconds
Proportional Gain	K_p	0.2
Derivative Gain	K_d	1
Predictor Gains	K_{pred}	$\{0.9, 0.9, 1.8\}$
Sensing Delay	τ_s	8 ms
Communication Delay	τ_{comm}	150 ms

were initialized at random. The error dynamics are shown in Figure 4.8.

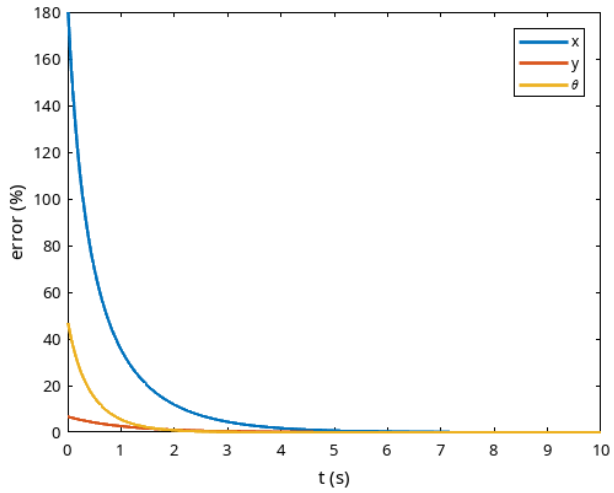


FIGURE 4.8. Prediction error with random initial conditions - states $0 < x, y < 10$, $0 < \theta < 2\pi$.

4.2.2 Consensus Variable Convergence

Simulating the unforced system shows that the agents reach a common state at the origin of the formation. Figure 4.9 shows the individual consensus variables x and y and the trajectories taken by the agents to reach this position in the cartesian plane. The settling time of the system is about 30 seconds, with the leaders 3 and 12 settling first. The response of the system evidences that the coupling of the remaining agents causes them to behave and experience similar transient response characteristics. Comparing the simulation results with the expected results of the linear plant can be achieved by deployment of the QPmR method [54], which allows us to compute all the poles within a specified region of a linear system at a particular time-delay combination. The QPmR method indentifies the dominant root of the system with a time-constant of $\frac{1}{\omega_n \zeta} = 6.9$ s , corresponding to $\sim\frac{1}{4.3}$ the settling time of the simulated non-linear system. This value agrees within 10% of the classical settling-time expression for linear second order systems, reinforcing the presumption that the agents behave linearly. The transient response characteristics of the

system found using the QPmR method can be found in Table 4.2.

4.2.3 Formation Regulation

The system forced by the relative spacing factor $\phi(t)$ reaches the error band of the desired positions in the grid after 30 seconds and remains stationary. This is consistent with the expected step response characteristics of the linear system, however, the response is slightly slower, likely due to the actuator saturation constraints. As an example of the response to sudden inputs, we excited the system with a periodic switching signal between two different formations, the one specified by the positions in Figure 4.1 and a new one that resembles a inscribed circle, shown in Figure 4.11a. The period of the switching signal was 50 s, we show the graph of x and y trajectories in Figure 4.11b. The system reaches zero error after settling in each formation.

TABLE 4.2. Frequency Response Characteristics - Butterfly Topology of 12 Agents

Pole	Damping	Frequency	Time Constant
		(rad/seconds)	(seconds)
-1.43e-01 + 1.83e-01i	6.16e-01	2.32e-01	6.9
-2.06e-01 + 1.94e-01i	7.27e-01	2.83e-01	4.86
-2.25e-01 + 3.23e-02i	9.90e-01	2.27e-01	4.44
-2.30e-01 + 2.90e-34i	1.00e+00	2.30e-01	4.35
-2.31e-01 + 3.37e-02i	9.90e-01	2.34e-01	4.33
-2.33e-01 - 3.37e-33i	1.00e+00	2.33e-01	4.28
-2.73e-01 + 5.72e-10i	1.00e+00	2.73e-01	3.67
-2.73e-01 - 5.72e-10i	1.00e+00	2.73e-01	3.67
-2.73e-01 + 6.33e-10i	1.00e+00	2.73e-01	3.67
-2.73e-01 - 1.86e-10i	1.00e+00	2.73e-01	3.67
-6.62e-01 + 9.17e-01i	5.85e-01	1.13e+00	1.51
-7.13e-01 + 7.71e-01i	6.79e-01	1.05e+00	1.40
-7.99e-01 + 7.84e-11i	1.00e+00	7.99e-01	1.25
-7.99e-01 - 5.96e-10i	1.00e+00	7.99e-01	1.25
-7.99e-01 + 6.07e-10i	1.00e+00	7.99e-01	1.25
-7.99e-01 - 1.72e-10i	1.00e+00	7.99e-01	1.25

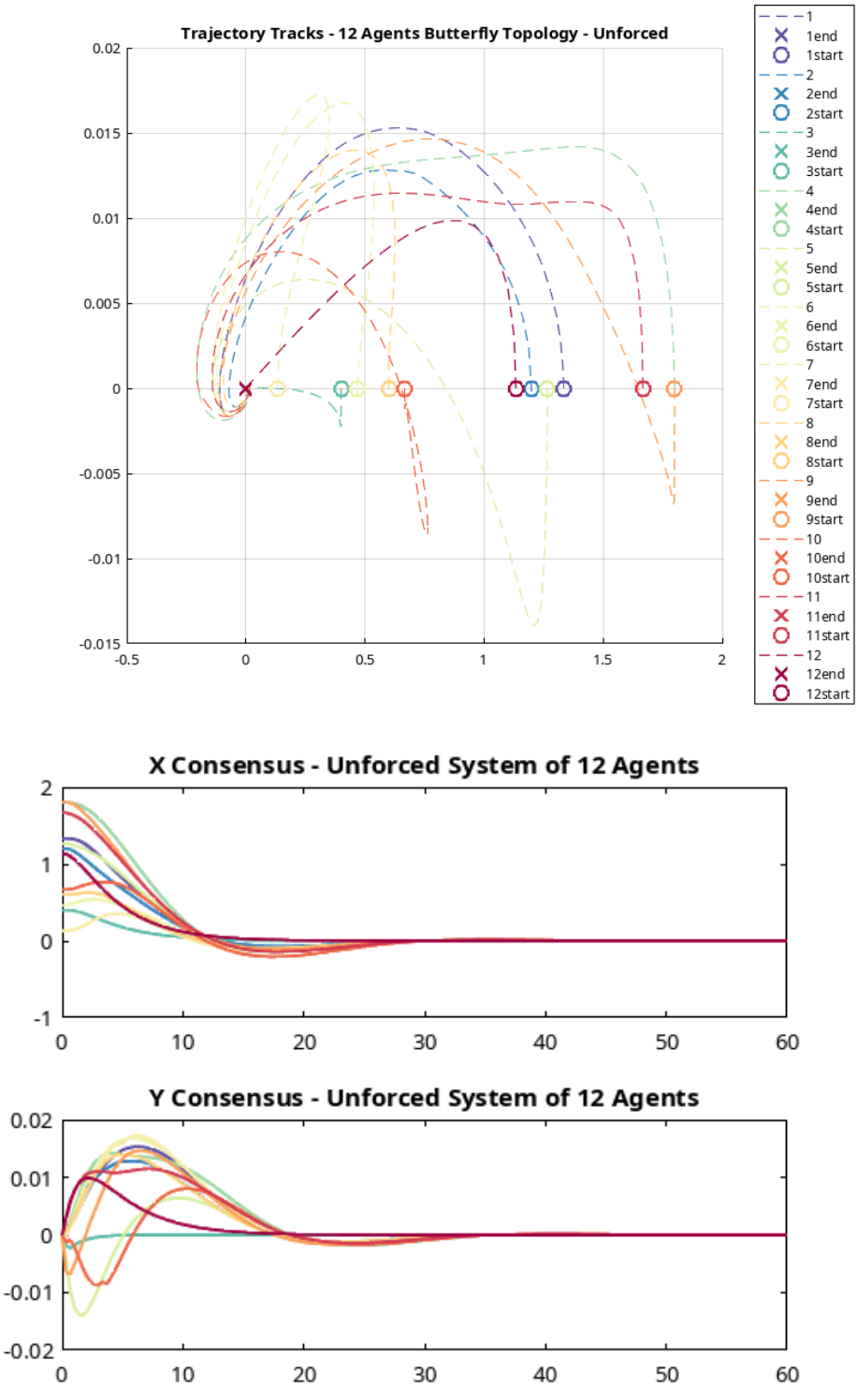


FIGURE 4.9. Global consensus - convergence of agent states x and y.

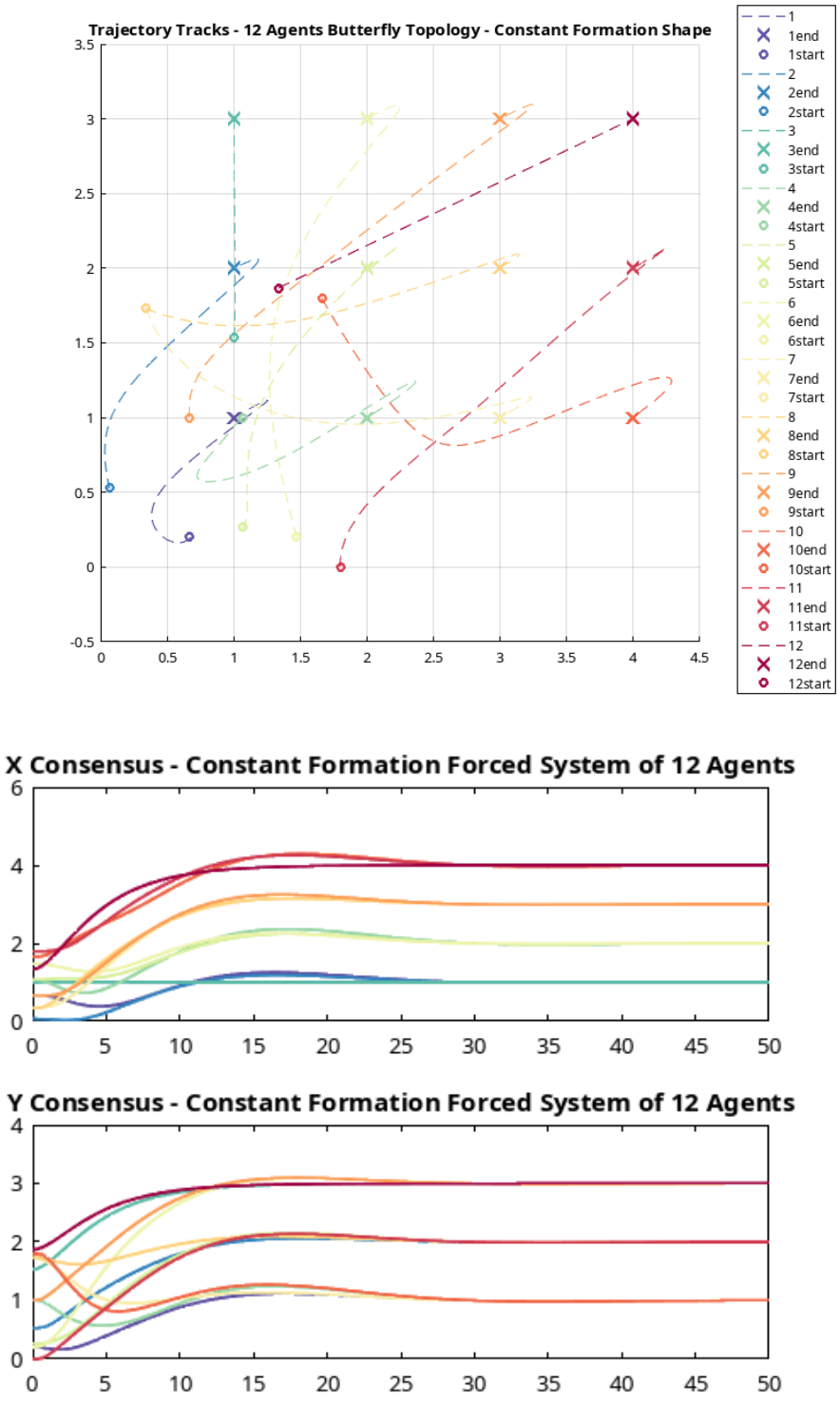
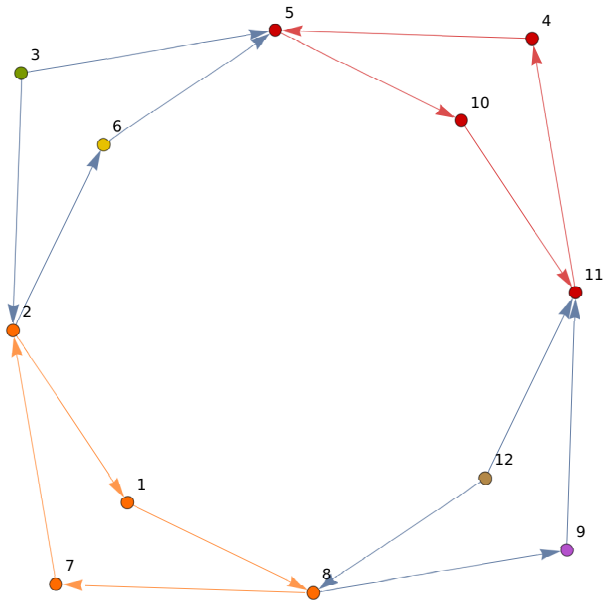
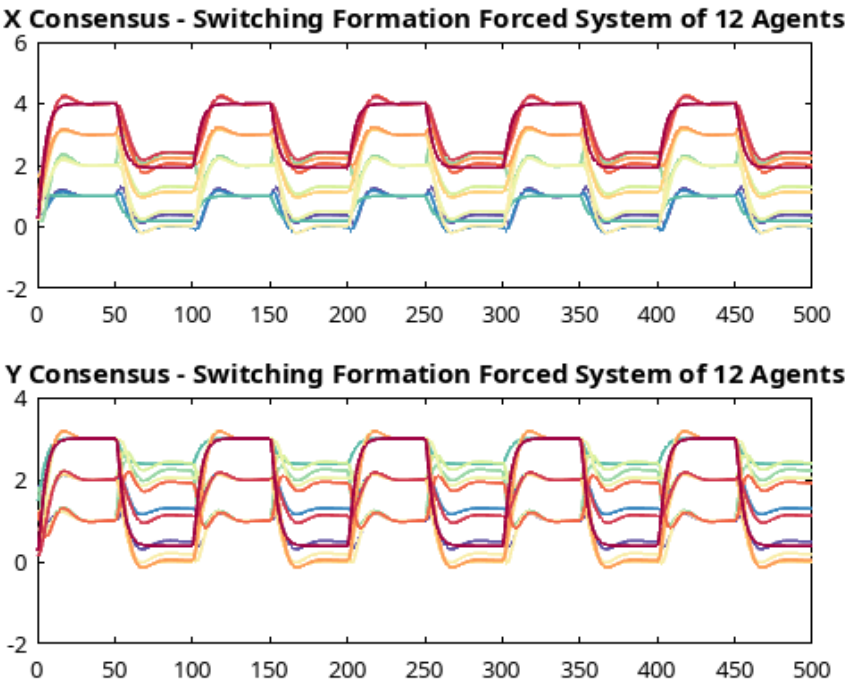


FIGURE 4.10. Square formation - 12 agents trajectory tracks.



(a) Directed butterfly topology - circle-like formation of 12 agents.



(b) Switching formation shape - X and Y trajectories, $\frac{1}{50}$ Hz.

FIGURE 4.11. Circle-like formation shape and state response.

4.2.4 Dynamic Formation Keeping

To display the bounded velocity error of the formation to constant velocity inputs, we enhance the constant formation forcing function with a time-dependent periodic term to dynamically scale the formation shape. Using a triangle wave of amplitude α and frequency ω_{scale}, the scaling of the formation terms is done uniformly across all the agents. The scaling function is show in Equation (4.5).

$$\begin{aligned} \boldsymbol{\phi_{xx}(t)} &= \boldsymbol{\phi_x} + \boldsymbol{\phi_x} \alpha \, triangle(t, \omega_{scale}) \\ \boldsymbol{\phi_{yy}(t)} &= \boldsymbol{\phi_y} + \boldsymbol{\phi_y} \alpha \, triangle(t, \omega_{scale}) \end{aligned} \tag{4.5}$$

The response of the system and the error with respect to the inputs are shown in Figure 4.12, we see a steady state relative error of less than 3% of the desired dynamically scaled formation, with zero error at the zero velocity points. Although there is a time-varying non-zero relative error, the formation shape is retained before, during and after scaling. This behavior is also seen when the formation shape is rotated about its centroid, $\bar{\phi}_x, \bar{\phi}_y$ with initial heading θ_0 with the forcing function in Equation (4.6).

$$\begin{aligned} \phi_{xxj}(t) &= \alpha \left((\phi_{xj} - \bar{\phi}_x) cos(\omega_r t - \theta_0) - (\phi_{yj} - \bar{\phi}_y) sin(\omega_r t - \theta_0) \right) \\ \phi_{xxj}(t) &= \alpha \left((\phi_{xj} - \bar{\phi}_x) sin(\omega_r t - \theta_0) + (\phi_{yj} - \bar{\phi}_y) cos(\omega_r t - \theta_0) \right) \end{aligned} \tag{4.6}$$

The steady-state relative error to the sinusoid input with frequency 0.01Hz is time-varying and periodic, with a maximum value of 6% error. The error and a top view of the trajectory tracks can be seen in Figure 4.13. From the response, we see that the agents in the formation settle to a circular trajectory with constant radius, the radius of each circle is within 1% of the distance from the centroid of the static formation to each agents position, as seen in Figure 4.14.

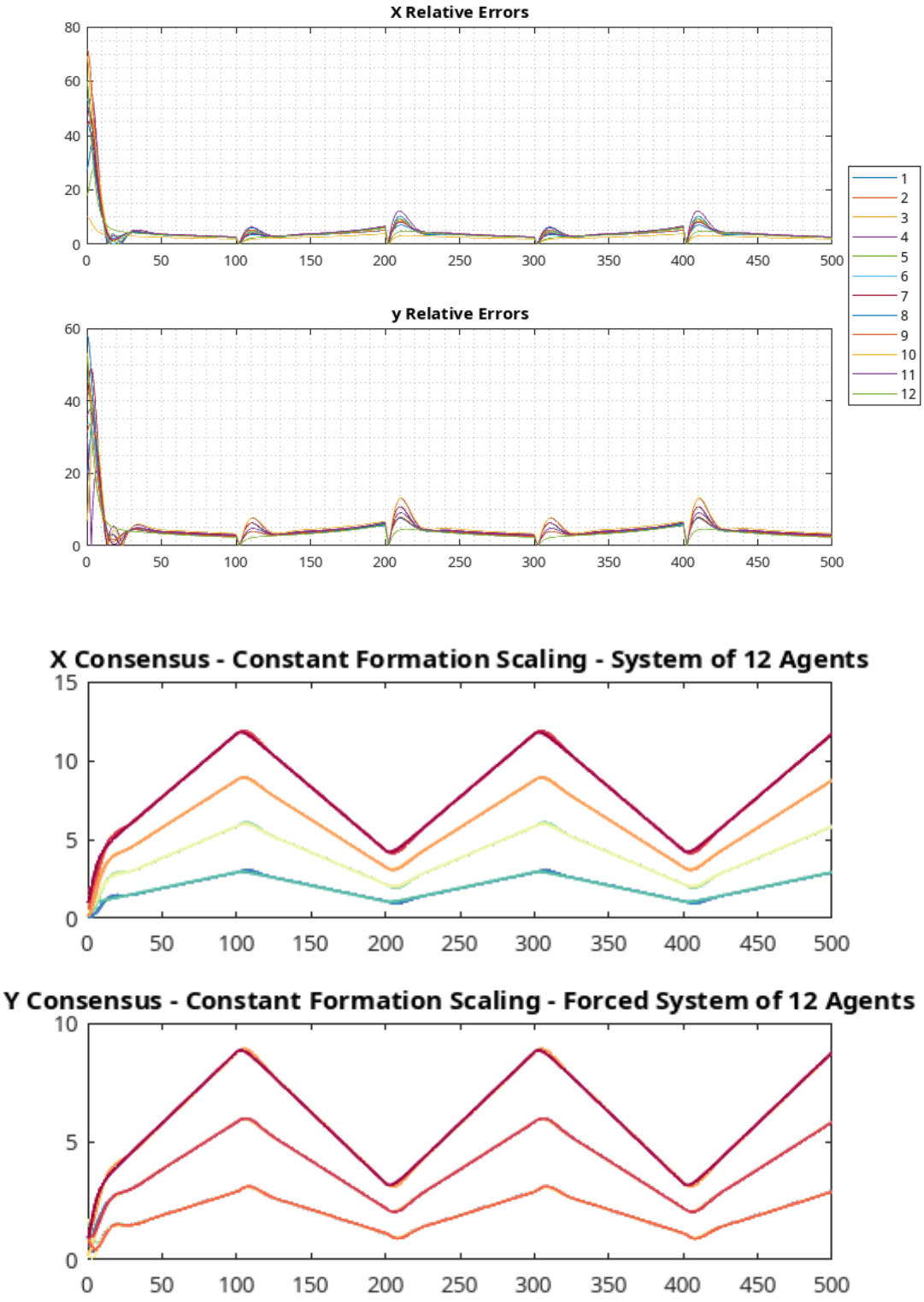


FIGURE 4.12. Dynamic formation - linear scaling.

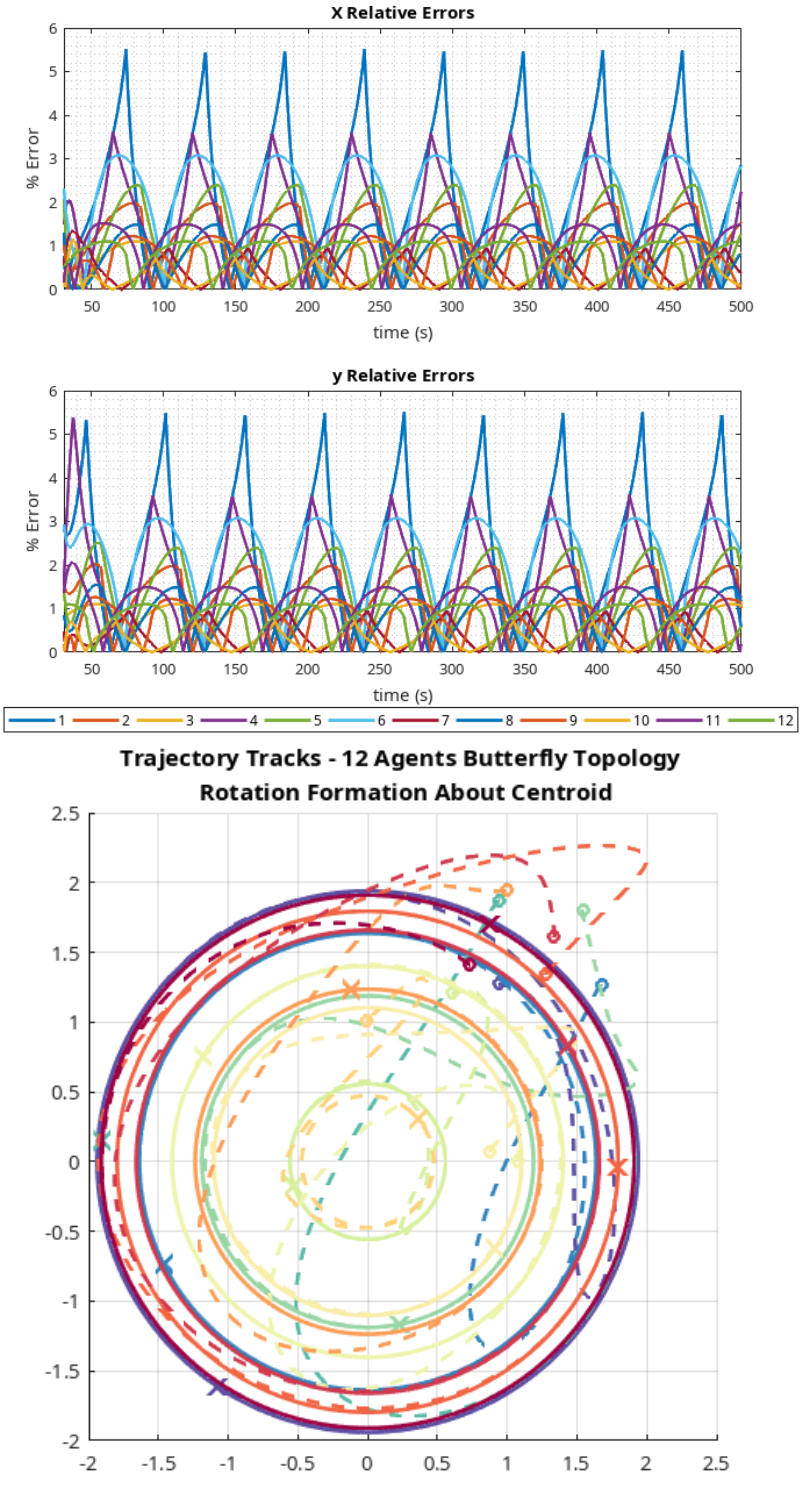


FIGURE 4.13. Dynamic formation - virtual structure rotation about centroid.

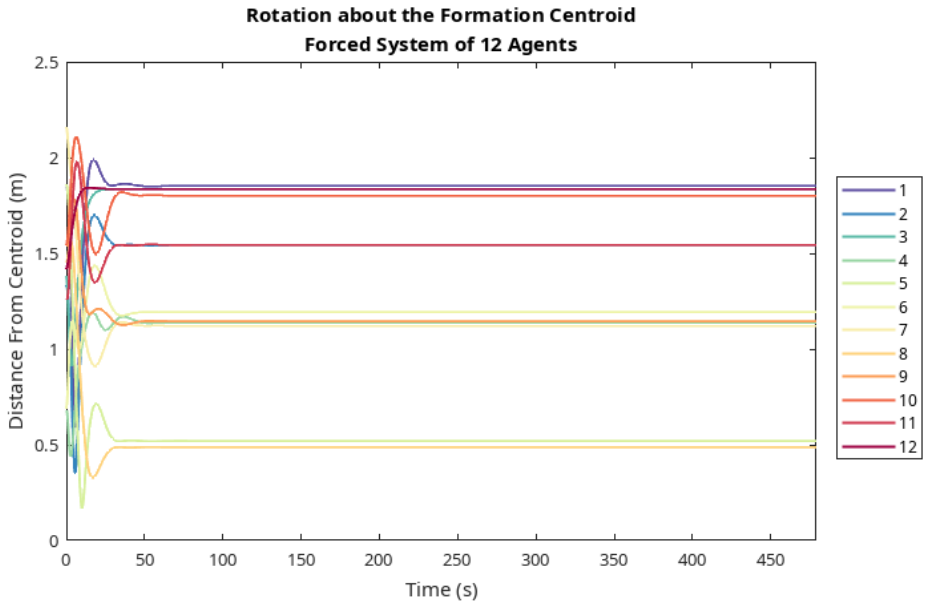


FIGURE 4.14. Dynamic formation - distance to formation centroid.

4.2.5 Formation Trajectory Tracking

When defining the inter-agent distances with respect to the formation centroid, a desired trajectory factor in the form of Equation (3.37) can be used to guide the entire formation centroid to a trajectory. We use Bernoulli's Lemniscate to generate a persistent cartesian parametric trajectory for the consensus control system in terms of time, similar to the formation dynamic formation keeping factors. The desired trajectory is expressed as:

$$\begin{aligned} x_d(t) &= \alpha\sqrt{2}\frac{cos\left(\omega_{dyn}t\right)}{sin\left(\omega_{dyn}t\right)^2+1} \\ y_d(t) &= \alpha\sqrt{2}\frac{cos\left(\omega_{dyn}t\right)sin\left(\omega_{dyn}t\right)}{sin\left(\omega_{dyn}t\right)^2+1} \end{aligned} \tag{4.7}$$

The response of the system under the lemniscate excitation may exceed the saturation velocity limit of the actuators, this drove selection of the scaling factor to be $\alpha = 2$ m and $\omega_{dyn} = 0.005$Hz. The trajectory tracks of the formation centroid and the desired trajectory can be seen in Figure 4.16. The response of the centroid of the formation with respect to

the desired trajectory has a non constant relative tracking error of less than 7.5% and an RMS error of 3.33%. The response of the system and the relative percent errors in the x, y coordinates can be seen in Figure 4.16.

4.3 Model Validation Using Monte Carlo Methods

Due to the introduction of the non-linear predictor and the saturation effects into the plant, we cannot truly assert that the system behaves exactly as a linear system, and thus, the analysis of the marginal stability points with respect to time delays may not be an accurate representation of the stability of the actual system. In order to establish whether the assessment is correct, we randomly sample the delay space in the region close to the stability bounds found by the CTCR method and simulate the unforced consensus system at these points. The initial conditions of the agents are also selected at random, for each run, in the unit square centered at the origin. We selected 500 points in the delay space, the sampled points and the histogram of the distributions can be found in Figure 4.17. Introducing the marginal stability points found by using the CTCR algorithm, the points were classified based on whether they corresponded to points within a stable or unstable region. Using this classification, the probabilities of finding a unstable point in the sampled set is $P_s = 0.374$, conversely, the probability of finding a stable point $P_u = 0.626$. After performing the simulation, the proportion of points that were found to be stable was $P_{ss} = 0.3580$, representing a 4% difference in the expected result. The simulated classification of the points and aggregated responses of the simulated system are shown in Figures 4.18a and 4.18b. Red crosses correspond to unstable points, the exact marginal stability bounds found by the CTCR method are shown as red triangle markers and the stable points are shown as blue circles.

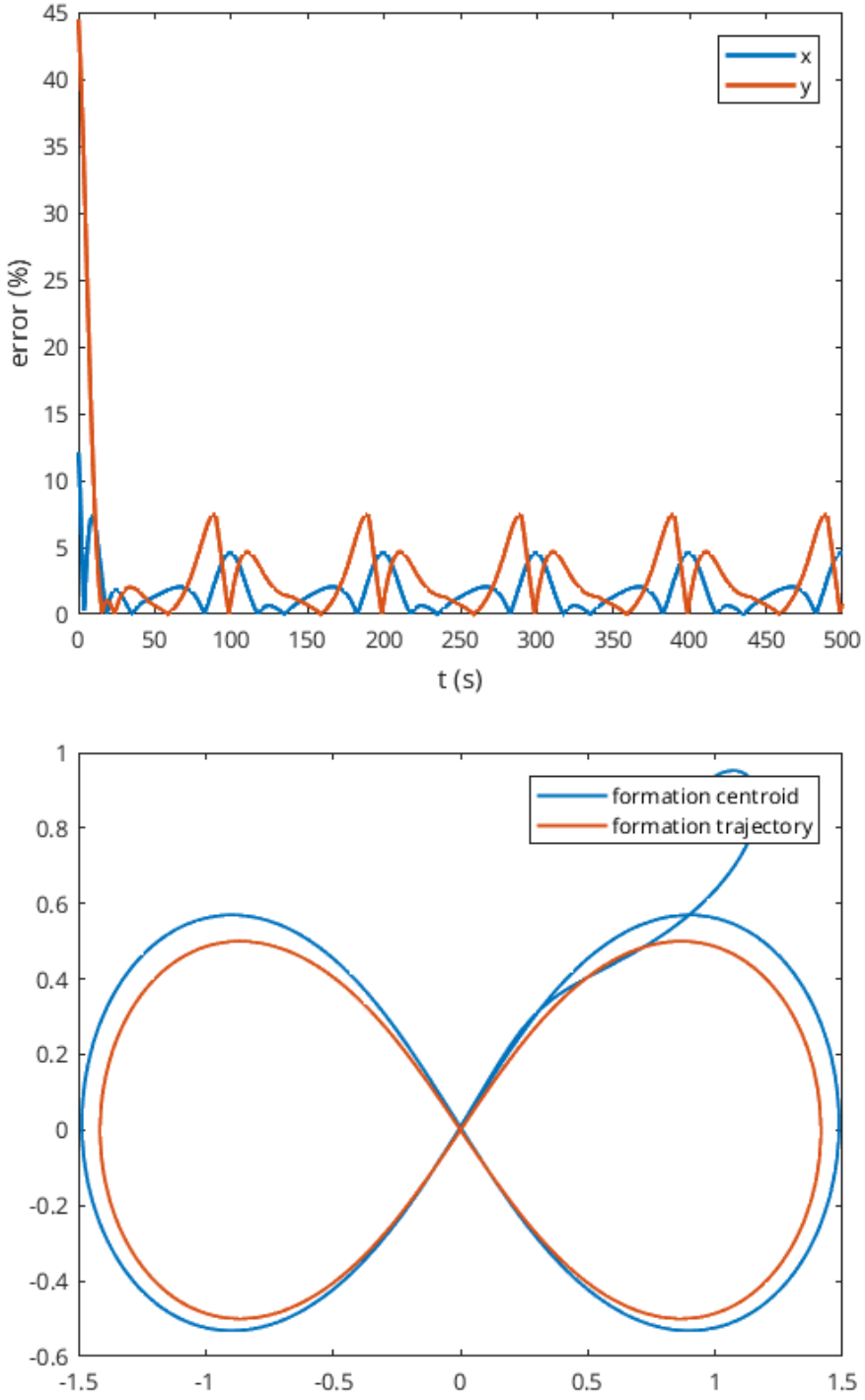


FIGURE 4.15. Formation tracking - centroid error and trajectory tracks.

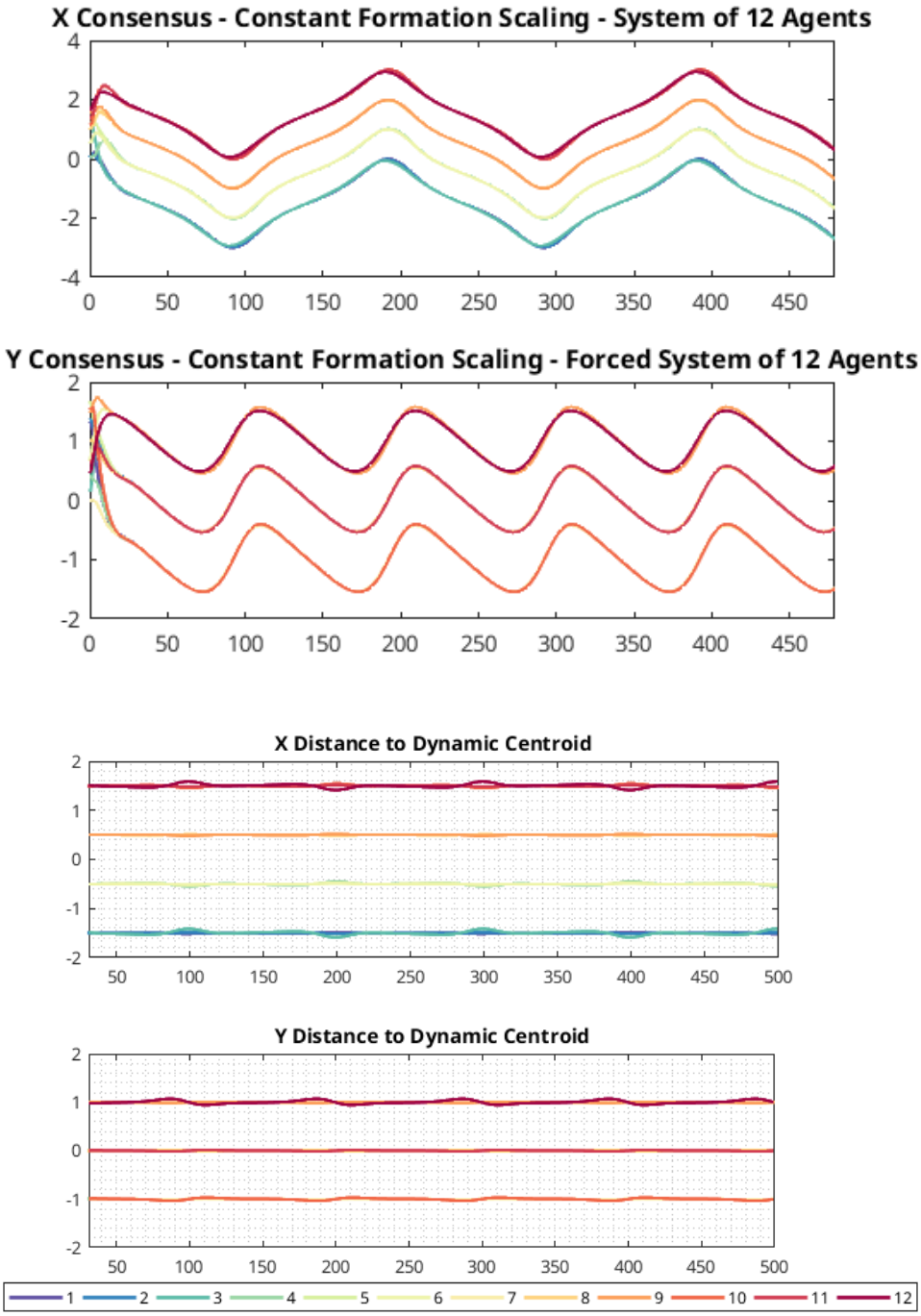


FIGURE 4.16. Formation tracking - system response and distance relative to centroid.

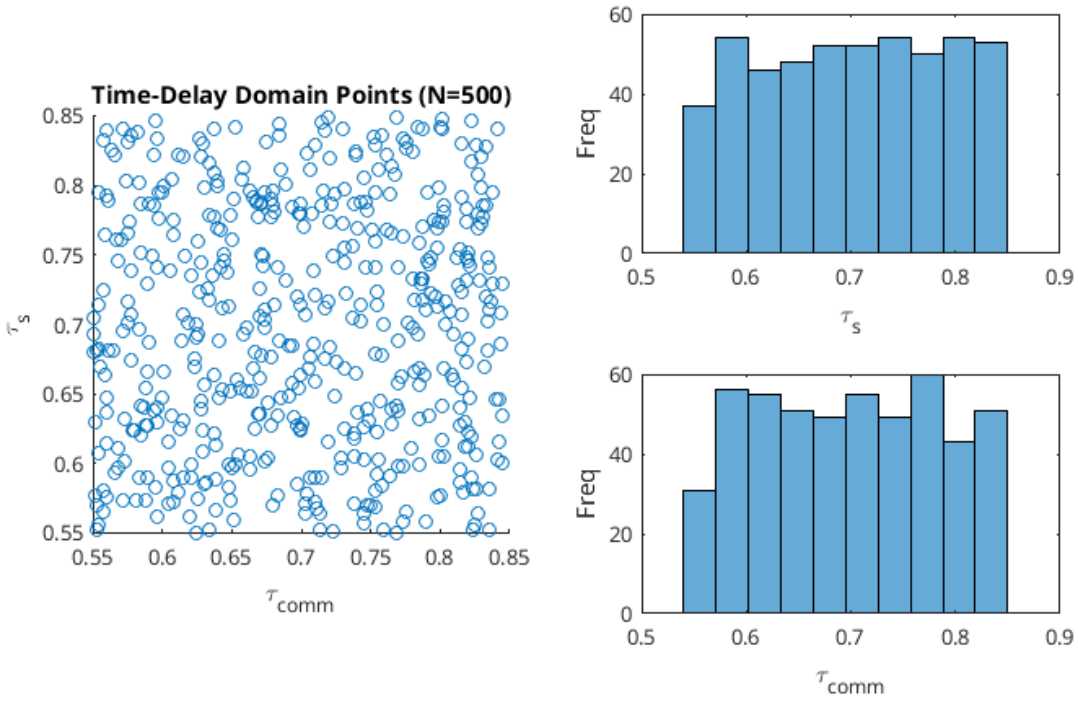
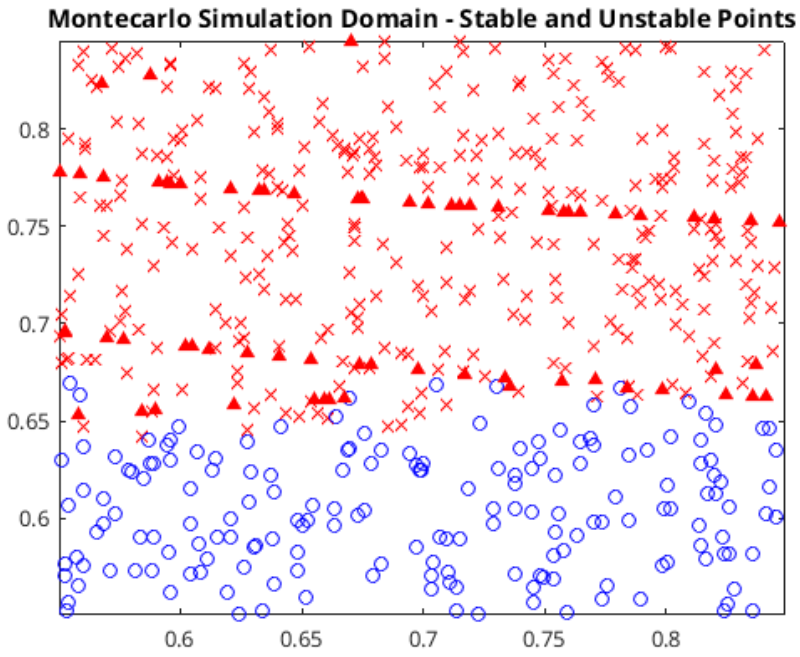
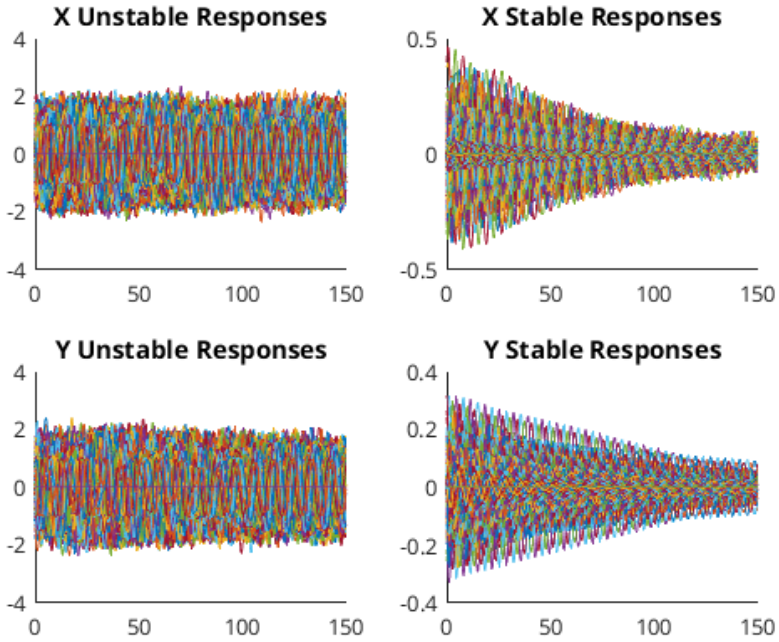


FIGURE 4.17. Monte Carlo simulation - sampling of delay space.



(a) Monte Carlo simulation - sampled point classification.



(b) Monte Carlo simulation - system responses.

FIGURE 4.18. Stability comparison of linear system analysis with sampled system - response classification.

4.4 Conclusion

The approach developed in Chapter 3 to analyze and design a robust consensus formation proportional-derivative control scheme for input-output linearizable non-holonomic systems in the presence of multiple time delays has been implemented and verified for sample communication topologies. The consensus control architecture was augmented to support the analysis of systems with "physical" leader agents, and the architecture was shown to solve the consensus problem by analyzing the spectrum of the communication and sensing matrices. The special case of self-sensing delays for formations of non-holonomic agents was considered for the first time in literature and with the introduction of a non-linear predictor, the performance of the non-linear system was found to be acceptable and comparable with that of the linearized system, as shown in Section 4.3. The results presented in Figure 3.5 showed that in the presence of self-sensing and communication time delays, careful selection of the uniform controller gains can yield a system with infinite communication time delay margin. Results from linear control theory apply to the formation scheme and considering the plant's order, the steady-response characteristics are that of a linear system under proportional-derivative control, with a zero steady-state error for step inputs, and bounded errors for ramp response. In addition to this, the formation shape specification methodology introduced in Equation (3.31) was shown to be convenient in the definition of static and dynamic formation shapes, specially when defined in terms of the centroid of the formation. For formations with the presence of a physical leader, exciting the leader with a persistent trajectory results in the trajectory tracking of the entire formation with performance characteristics typical of a type 1 system. The transient response of the system was partially analyzed by using the QPmR method on the linear system, and the results were confirmed by simulating the multiagent system with different exogenous inputs corresponding to the dynamic formation keeping and dynamic formation tracking terms.

The simulation results introduced in this section have validated the design methodology introduced in this thesis and showed that the tools used to analyze the performance of the system adequately and exhaustively describe the communication and sensing constraints of a formation of agents. Future work can include a complete frequency response analysis of the multi-agent system and an assessment of the trajectory tracking performance and the robustness characteristics of a formation under PID linear consensus control.

APPENDIX:
COMPUTER CODE

CTCR Method Mathematica Library

The CTCR algorithm was implemented in Wolfram Mathematica [40] in order to efficiently analyze systems expressed symbolically. It uses a graphical approach to obtain the zero-level contour so it is restricted to 2 delay systems. Example usage of the library is provided in the example below.

```
nAgents = Range[1, 3];
(* Define graph of 3 agents *)
commEdges = {1 -> 2, 2 -> 1, 3 -> 1, 3 -> 2};
Gcomm = Graph[Join[nAgents], Join[commEdges], VertexLabels -> "Name"]
n = VertexCount[Gcomm];
DMcomm = PseudoInverse@DiagonalMatrix[VertexInDegree[Gcomm]];
selfEdges = # -> # & /@ nAgents;
Gself = Graph[selfEdges, VertexLabels -> "Name"];
DMself = IdentityMatrix[n];

Aplant = {{0, 1}, {0, 0}};
Gains = {{0, 0}, {Kp, Kd}};

(*Define Characteristic Equation 2 delay system *)
GetTF[A_, B_, C_] := (s*IdentityMatrix[Length[A]]) -
   Chop@N[A] - ((Chop@N[B]*Exp[-t2*s]) + (Chop@N[C]*Exp[-t1*s]*
       Exp[-t2*s]));
GetCharEQ[A_, B_, C_] := Det[GetTF[A, B, C]];

Id = IdentityMatrix[n];
Lpl = Transpose@AdjacencyMatrix@Gcomm;
SelfM = AdjacencyMatrix@Gself;
InformerM = DMcomm.Lpl;
{T, DInf} = JordanDecomposition[InformerM];

A = KroneckerProduct[Id, Aplant];
Bm = {{0, 0}, {0, 1}};
B1 = -KroneckerProduct[SelfM, Bm.Gains]
B2 = KroneckerProduct[InformerM, Bm.Gains]

Ay = SimilarityTransformation[A, Tf];
B1y = SimilarityTransformation[B1, Tf];
B2y = SimilarityTransformation[B2, Tf];
```

```
(* Obtain list of second order factors *)
twoOrderIndices={1,3,5}
charEqns2nd =
  GetCharEQ[GetBlock[Ay, #, 2], GetBlock[B1y, #, 2],
     GetBlock[B2y, #, 2]] & /@ twoOrderIndices;
desiredGains={Kp -> 0.2, Kd -> 0.5};

(* Obtain Analysis results for first factor, 10 offspring curves *)
nn = 10;
analysisResults = CTCRAnalysis[(charEqns2nd[[1]]/.desiredGains), s, {t1,
 ↪  t2}, c, nn];

(* SDS Plot *)
sdsFunctions = analysisResults[["functions"]][["SDS"]];
sdsFunPlot =
  ParametricPlot[sdsFunctions[[All, 1 ;; 2]], {c, 0, 1},
   PlotRange -> {{0, 2*nn*Pi}, {0, 2*nn*Pi}},
   GridLines -> {2*Pi*Range[0, nn, 1], 2*Pi*Range[0, nn, 1]},
   GridLinesStyle -> Directive[Black, Dashed, Thick],
   PerformanceGoal -> "Speed", Frame -> True, AspectRatio -> 1];

(* DS Plot *)
dsFunctions = analysisResults[["functions"]][["DS"]];
dsFunPlot =
  ParametricPlot[dsFunctions[[All, 1 ;; 2]], {c, 0, 1},
   PlotRange -> {{0, 10}, {0, 10}}, PerformanceGoal -> "Speed",
   Frame -> True, FrameLabel -> {{"\[Tau]1", ""}, {"\[Tau]2", ""}}];

(* Root Tendecy Plot *)
rtgroups = GroupBy[analysisResults[["RT"]][["t1"]], #[[3]] &];
rtmap = Map[
   ListPlot[rtgroups[#][[All, 1 ;; 2]],
     PlotRange -> {{0, 10}, {0, 10}},
     PlotStyle -> Switch[#, -1, Blue, 1, Red], AspectRatio -> 1,
     ImageSize -> Large, Frame -> True] &, Keys[rtgroups]];
```

TimeDelayLib - Mathematica [40] Library for TimeDelay System Analysis

```
BeginPackage["TimeDelayLib`"]

SimilarityTransformation::usage = "SimilarityTransformation[A,Tf] \
performs the similarity transformation of matrix A with matrix Tf"

GetBlock::usage = "GetBlock[A, n, size] returns blocks at diagonal \
position \"n\" of size \"size\" from block diagonalized matrix \"A\""

CTCRAnalysis::usage = "
CTCRAnalysis[charEqn, s, taus, c, numberOffspring, resolution:60,
 ↪  splitCoefficient:1]
Returns an association in the form of:
Association[
    \"points\" -> Association[
        \"SDS\" -> {{v1,v2,omega}},
        \"DS\"->{{t1,t2,omega}}}
        ],
    \"functions\" -> Association[
        \"SDS\" -> InterpolationFunction[],
        \"DS\" -> InterpolationFunction[]
        ]
    \"RT\" -> Association[
        \"t1\"->{{t1,t2,RT}},
        \"t2\" -> {{t1,t2, RT}}
        ]
    ]
""

FindKernelSDS::usage =
"FindKernelSDS[chareq,s,taus,resolution_:60,recursion_:2] returns a
nested list of lists of length 3 corresponding to the Kernel Curves
of characteristic equation \"chareq\", with laplace variable \"s\" a
list of a maximum size of 2 time delays \"taus\={t1,t2}\"."

propagateSDS::usage = "propagateSDS[kernList, n] generates the
offspring curves from a list of points defining the kernel \"kernList\" in
 ↪  SDS space."
transformSDSToDS::usage = "transformSDSToDS[kernList,n] generates the
offspring curves in DS space from a list of points defining the
kernel \"kernList\" in SDS space by performing a point-wise
```

```
transformation."

RootTendency::usage = "RootTendency[a,s,taudir,taus,ptz] computes the
root tendency of characteristic equation \"a\" with Laplace variable \"s\",
 ↪  along direction \"taudir\= 1,..,n\" corresponding to the index
of time-delay list \"taus\"={t1,t2,..,tn}"

RealJordanDecomposition::usage = ""
SortedJordanDecomposition::usage = ""

Begin["`Private`"]
SimilarityTransformation[A_, Tf_] := Inverse[Tf].A.Tf;
GetBlock[A_, n_, size_] :=Take[A, {n, n + (size - 1)}, {n, n + (size -
 ↪  1)}];
GetReal[x_] := ComplexExpand[Re[x]];
GetIm[x_] := ComplexExpand[Im[x]];

CTCRAnalysis[eqn_, s_, taus_, c_, nn_: 5, res_: 60, coff_: 1] :=
  Module[{kernelSDS, kernelDS, toCurves, sortedCurvesDS,
    sortedCurvesSDS, kernelDSInterpolatingFunctions,
    kernelSDSInterpolatingFunctions, kernelDSFunctions,
    kernelSDSFunctions, propagationVector, dsFunctions, sdsFunctions,
    offSDS, offDS, RT},
   toCurves[ptz_, cutoff_] :=
    Split[ptz, EuclideanDistance[#, #2] < cutoff &];
   kernelSDS = FindKernelSDS[eqn, s, taus, res];
   If[SameQ[kernelSDS, {}], Return[None], None];
   kernelDS = SDS2Tau[kernelSDS];
   sortedCurvesSDS = toCurves[kernelSDS, coff];
   sortedCurvesDS = toCurves[kernelDS, coff];
   kernelSDSInterpolatingFunctions =
    Map[Quiet@
       Interpolation[
        Transpose[{N@Range[0, 1, 1/(Length[#] - 1)], #}]] &,
     sortedCurvesSDS];
   kernelDSInterpolatingFunctions =
    Map[Quiet@
       Interpolation[
        Transpose[{N@Range[0, 1, 1/(Length[#] - 1)], #}]] &,
     sortedCurvesDS];
   kernelSDSFunctions =
    Map[kernelComponents[#, c] &, kernelSDSInterpolatingFunctions];
```

```
   kernelDSFunctions =
    Map[kernelComponents[#, c] &, kernelDSInterpolatingFunctions];
   propagationVector =
    Flatten[Table[{i, j}, {i, 0, nn}, {j, 0, nn}], 1];
   dsFunctions =
    Flatten[Map[propagateDS[#, propagationVector] &,
      kernelDSFunctions], 1];
   sdsFunctions =
    Flatten[Map[propagateSDS[#, propagationVector] &,
      kernelSDSFunctions], 1];
   offSDS =
    Flatten[Map[propagateSDS[#, propagationVector] &, kernelSDS], 1];
   offDS =
    Flatten[Map[propagateDS[#, propagationVector] &, kernelDS], 1];
   RT = Association@
     Map[String@# -> RootTendency[eqn, s, #, taus, offDS] &, taus];
   Return[
    Association[
     "points" -> Association["SDS" -> offSDS, "DS" -> offDS],
     "functions" ->
      Association["SDS" -> sdsFunctions, "DS" -> dsFunctions],
     "RT" -> RT]];
   ];

char2KernelFn[chareq_, s_, taus_] :=
    Module[{toHT, htPolyCoeffs, realPoly, imagPoly, ZkToVk,
    HalfAngleTanRl, tau2Z, halfSub, zi, CoeffsToPol, zs, rts,
    fn, \[Omega]}, zi[i_] := Symbol["z" <> ToString@i];
    \[Omega] = Symbol["\[Omega]"];
    halfSub = {Cos[x_] :> (1 - Tan[x/2]^2)/(1 + Tan[x/2]^2),
    Sin[x_] :> (2 Tan[x/2])/(1 + Tan[x/2]^2)};
    tau2Z[taus2_] :=
    Table[Tan[taus2[[i]] \[Omega]/2] -> zi[i], {i, Length@taus2}];
    HalfAngleTanRl[eqn_,
    taus2_] := ((TrigExpand[
        ExpToTrig[PowerExpand[eqn, taus] /. {s :> \[Omega] I}]]) /.
        halfSub) /. tau2Z[taus2];
    ZkToVk[zk_, vk_] :=
    Table[zk[[i]] -> Tan[vk[[i]]/2], {i, Length[vk]}];
    CoeffsToPol[l_, sym_] := sym^Range[0, Length[l] - 1].l;
    toHT = Numerator@Together[HalfAngleTanRl[chareq, taus]];
    htPolyCoeffs = CoefficientList[toHT, \[Omega]];
```

```
    realPoly = CoeffsToPol[GetReal[htPolyCoeffs], \[Omega]];
    imagPoly = CoeffsToPol[GetIm[htPolyCoeffs], \[Omega]];
    zs = zi /@ Range[Length@taus];
    rts = Resultant[realPoly, imagPoly, \[Omega],
    Method -> "Subresultants"];
    fn[vs_] := rts /. (ZkToVk[zs, vs]);
    Return[fn]
]

PolySurf[chareq_, nparm_, resolution_: 60, recursion_: 2] :=
    Module[{toKern, kernRefl, rt, myPlot1, myPlot2, fn, res, v1, v2},
    toKern[plt_] :=
    Flatten[Cases[Normal@plt, Line[data_] :> data, -1], 1];
    fn = chareq[{v1, v2}];
    rt = Power[#[[1]], #[[2]]] & /@ Rest@FactorList[fn];
    myPlot1 =
    RegionPlot[
        chareq[{v1, v2}] <= 0., {v1, 0., 2. Pi}, {v2, 0., 2. Pi},
        PlotPoints -> resolution, MaxRecursion -> 3];
    myPlot2 :=
    Show[ContourPlot[# == 0, {v1, 0, 2 Pi}, {v2, 0, 2 Pi},
        Contours -> {0}, Exclusions -> None, PlotPoints -> resolution,
            MaxRecursion -> 3, PerformanceGoal -> "Quality"] & /@ rt];
    myPlot1 =
    If[Length@Flatten@toKern@myPlot1 <= 3, myPlot2, myPlot1];
    res = toKern@myPlot1;
    Return[res];
    ];

FindKernelSDS[chareq_, s_, taus_, resolution_: 60, recursion_: 2] :=
    FindKernelSDS[chareq, s, taus, resolution, recursion] =
    Module[{dsRoots, sdsBounds, EvalCharSDS, FindSDSOmegas,
        commonRoots, \[Omega], repl},
    FindSDSOmegas[eqn_, vs_] :=
        Module[{subl, res},
        subl = GetReal[
        Select[Flatten[Solve[EvalCharSDS[eqn, vs] == 0, \[Omega]],
            1][[All, 2]], (Abs@Re@# > 0 && Abs@Im@# < 0.0001) &]];
        res =
        If[Length[subl] > 0, Map[{vs[[1]], vs[[2]], #} &, subl], {}];
        Return[res];];
    EvalCharSDS[eqn_, vs_] :=
```

```
        Module[{rule},
        rule = Map[(\[Omega] taus[[#]] -> vs[[#]]) &,
        Range[Length@vs]];
        Return[eqn /. {s -> \[Omega] I} /. rule]];
    commonRoots = char2KernelFn[chareq, s, taus];
    sdsBounds =
        PolySurf[commonRoots, Length@taus, resolution, recursion];
    Return[
        Select[Flatten[Join[FindSDSOmegas[chareq, #] & /@ sdsBounds], 1],
        UnsameQ[#, {}] &]];

propagateSDS[fn_,
    list_] := {(fn[[1]] + 2 Pi #[[1]]), (fn[[2]] + 2 Pi #[[2]]),
        Abs@fn[[3]]} & /@ list;

propagateDS[fn_,
    list_] := {(fn[[1]] + 2 Pi #[[1]]/Abs@fn[[3]]), (fn[[2]] +
        2 Pi #[[2]]/Abs@fn[[3]]), Abs@fn[[3]]} & /@ list;

SDS2Tau[points_] :=
    Map[{#[[1]]/#[[3]], #[[2]]/#[[3]], #[[3]]} &, points];
kernelComponents[fun_, c_] := Indexed[fun[c], #] & /@ Range[3];

transformSDSToDS[fn_,
    list_] := {(fn[[1]] + 2 Pi #[[1]])/
        Abs@fn[[3]], (fn[[2]] + 2 Pi #[[2]])/Abs@fn[[3]],
        Abs@fn[[3]]} & /@ list;

RootTendency[equation_, s_, taudir_, taus_, ptz_] :=
    Module[{totalD, solExpr, sss},
    SetAttributes[Evaluate@Select[taus, UnsameQ[#1, taudir] &],
    Constant];
    totalD = Dt[equation == 0, taudir];
    solExpr = Dt[s, taudir] /. Solve[totalD, Dt[s, taudir]];
    solExpr =
    solExpr /. {s -> I*ptz[[All, 3]],
        Sequence @@ MapIndexed[#1 -> ptz[[All, Sequence @@ #2]] &, taus]};
    Transpose[{ptz[[All, 1]], ptz[[All, 2]], Sign[Re[solExpr[[1]]]]}]
End[]
EndPackage[]
```

REFERENCES

REFERENCES

[1] Ren, W., and Beard, R. W., 2008, Distributed Consensus in Multi-vehicle Cooperative Control. Springer, London, United Kingdom.

[2] Shamma, J. S., ed., 2007, Cooperative Control of Distributed Multi-agent Systems. John Wiley & Sons, Hoboken, NJ.

[3] Krieger, G., Hajnsek, I., Papathanassiou, K. P., Younis, M., and Moreira, A., 2010, "Interferometric Synthetic Aperture Radar (SAR) Missions Employing Formation Flying," Proc. of the IEEE, 98(5), pp. 816–843. Doi: 10.1109/jproc.2009.2038948.

[4] Stipanović, D. M., İnalhan, G., Teo, R., and Tomlin, C. J., 2004, "Decentralized Overlapping Control of a Formation of Unmanned Aerial Vehicles," Automatica, 40(8), pp. 1285–1296. Doi: 10.1016/j.automatica.2004.02.017.

[5] Beard, R. W., Lawton, J., and Hadaegh, F. Y., 2001, "A Coordination Architecture for Spacecraft Formation Control," IEEE Transactions on Control Systems Technology, 9(6), pp. 777–790. Doi: 10.1109/87.960341.

[6] Camponogara, E., Jia, D., Krogh, B. H., and Talukdar, S., 2002, "Distributed Model Predictive Control," IEEE Control Systems, 22(1), pp. 44–52.

[7] Dunbar, W. B., and Murray, R. M., 2006, "Distributed Receding Horizon Control for Multi-vehicle Formation Stabilization," Automatica, 42(4), pp. 549–558.

[8] Hong, Y., Chen, G., and Bushnell, L., 2008, "Distributed Observers Design for Leader-following Control of Multi-agent Networks," Automatica, 44(3), pp. 846–850.

[9] Ren, W., Beard, R., and Atkins, E., 2005, "A Survey of Consensus Problems in Multi-agent Coordination". Proc. of the 2005 American Control Conference, Portland, OR, June 8-10, 2005, pp.1859-1864. Doi: 10.1109/acc.2005.1470239.

[10] Olfati-Saber, R., and Murray, R., 2004, "Consensus Problems in Networks of Agents with Switching Topology and Time-delays," IEEE Transactions on Automatic Control, 49(9), pp. 1520–1533. Doi: 10.1109/tac.2004.834113.

[11] Goddard Space Flight Center, NASA, "Landsat-7 Trailed by EO-1," June 18, 2018, Electronic Image,https://eo1.gsfc.nasa.gov/new/Technology/FormFly.html.

[12] Zhang, W., Branicky, M., and Phillips, S., 2001, "Stability of Networked Control Systems," IEEE Control Systems Magazine, 21(1), pp. 84–99. Doi: 10.1109/37.898794.

[13] Borkar, V., and Varaiya, P., 1982, "Asymptotic Agreement in Distributed Estimation," IEEE Transactions on Automatic Control, 27(3), pp. 650–655.

[14] Jadbabaie, A., Lin, J., and Morse, A., 2003, "Coordination of Groups of Mobile Autonomous Agents Using Nearest Neighbor Rules," IEEE Transactions on Automatic Control, 48(6), pp. 988–1001. Doi: 10.1109/tac.2003.812781.

[15] Vicsek, T., Czirók, A., Ben-Jacob, E., Cohen, I., and Shochet, O., 1995, "Novel Type of Phase Transition in a System of Self-driven Particles," Physical Review Letters, 75(6), pp. 1226–1229. Doi: 10.1103/physrevlett.75.1226.

[16] Mohar, B., Alavi, Y., Chartrand, G., and Oellermann, O., 1991, "The Laplacian Spectrum of Graphs," Graph Theory, Combinatorics, and Applications, 2(871-898), p. 12.

[17] Ren, W., and Beard, R. W., 2008, "Consensus Algorithms for Double-integrator Dynamics," Distributed Consensus in Multi-vehicle Cooperative Control: Theory and Applications, pp. 77–104.

[18] Feng, Y., Xu, S., and Zhang, B., 2012, "Group Consensus Control for Double-integrator Dynamic Multiagent Systems with Fixed Communication Topology," International Journal of Robust and Nonlinear Control, 24(3), pp. 532–547. Doi: 10.1002/rnc.2904.

[19] Lin, Z., Francis, B., and Maggiore, M., 2005, "Necessary and Sufficient Graphical Conditions for Formation Control of Unicycles," IEEE Transactions on Automatic Control, 50(1), pp. 121–127. Doi: 10.1109/TAC.2004.841121.

[20] Olfati-Saber, R., 2006, "Flocking for Multi-agent Dynamic Systems: Algorithms and Theory," IEEE Transactions on Automatic Control, 51(3), pp. 401–420. Doi: 10.1109/tac.2005.864190.

[21] Xiao, F., Wang, L., Chen, J., and Gao, Y., 2009, "Finite-time Formation Control for Multi-agent Systems," Automatica, 45(11), pp. 2605–2611. Doi: 10.1016/j.automatica.2009.07.012.

[22] Qin, J., Gao, H., and Zheng, W. X., 2011, "Second-order Consensus for Multi-agent Systems with Switching Topology and Communication Delay," Systems & Control Letters, 60(6), pp. 390–397. Doi: 10.1016/j.sysconle.2011.03.004.

[23] Fridman, E., 2001, "New Lyapunov–krasovskii Functionals for Stability of Linear Retarded and Neutral Type Systems," Systems & Control Letters, 43(4), pp. 309–319.

[24] Cepeda-Gomez, R., and Olgac, N., 2011, "An Exact Method for the Stability Analysis of Linear Consensus Protocols with Time Delay," IEEE Transactions on Automatic Control, 56(7), pp. 1734–1740. Doi: 10.1109/tac.2011.2152510.

[25] Olgac, N., and Sipahi, R., 2002, "An Exact Method for the Stability Analysis of Time-delayed Linear Time-invariant (LTI) Systems," IEEE Transactions on Automatic Control, 47(5), pp. 793–797. Doi: 10.1109/tac.2002.1000275.

[26] Gao, Q., and Olgac, N., 2016, "Bounds of Imaginary Spectra of LTI Systems in the Domain of Two of the Multiple Time Delays," Automatica, 72, pp. 235–241. Doi: 10.1016/j.automatica.2016.05.011.

[27] Fazelinia, H., Sipahi, R., and Olgac, N., 2007, "Stability Robustness Analysis of Multiple Time- Delayed Systems Using "building Block" Concept," IEEE Transactions on Automatic Control, 52(5), pp. 799–810. Doi: 10.1109/tac.2007.898076.

[28] Olgac, N., Ergenc, A. F., and Sipahi, R., 2005, ""delay Scheduling": A New Concept for Stabilization in Multiple Delay Systems," Modal Analysis, 11(9), pp. 1159–1172. Doi: 10.1177/1077546305055777.

[29] Cepeda-Gomez, R., and Perico, L. F., 2015, "Formation Control of Nonholonomic Vehicles under Time Delayed Communications," IEEE Transactions on Automation Science and Engineering, 12(3), pp. 819–826. Doi: 10.1109/tase.2015.2424252.

[30] Herbrechtsmeier, S., Witkowski, U., and Rückert, U., 2009, "BeBot: A Modular Mobile Miniature Robot Platform Supporting Hardware Reconfiguration and Multi-standard Communication," Progress in Robotics. Springer, Berlin, Germany, pp. 346–356.

[31] Mondada, F., Bonani, M., Raemy, X., Pugh, J., Cianci, C., Klaptocz, A., Magnenat, S., Zufferey, J.-C., Floreano, D., and Martinoli, A., 2009. "The E-puck, a Robot Designed for Education in Engineering," Proc. of the 9th Conference on Autonomous Robot Systems and Competitions, Vol. 1, Instituto Politécnico de Castelo Branco, Portugal, pp. 59–65.

[32] Cepeda-Gomez, R., and Olgac, N., 2013, "Exact Stability Analysis of Second-order Leaderless and Leader–follower Consensus Protocols with Rationally-independent Multiple Time Delays," Systems & Control Letters, 62(6), pp. 482–495. Doi: 10.1016/j.sysconle.2013.02.011.

[33] Godsil, C., and Royle, G., 2001, "The Laplacian of a Graph," Graduate Texts in Mathematics. Springer, New York, NY, pp. 279–306.

[34] Marsden, A., 2013, "Eigenvalues of the Laplacian and Their Relationship to the Connectedness of a Graph," University of Chicago, IL.

[35] Brualdi, R. A., 2010, "Spectra of Digraphs," Linear Algebra and its Applications, 432(9), pp. 2181–2213. Doi: 10.1016/j.laa.2009.02.033.

[36] Fax, J., and Murray, R., 2004, "Information Flow and Cooperative Control of Vehicle Formations," IEEE Transactions on Automatic Control, 49(9), pp. 1465–1476. Doi: 10.1109/tac.2004.834433.

[37] Roger A. Horn, C. R. J., 2012, Matrix Analysis. Cambridge University Press.

[38] Gao, Q., Zalluhoglu, U., and Olgac, N., 2014, "Investigation of Local Stability Transitions in the Spectral Delay Space and Delay Space," Journal of Dynamic Systems, Measurement, and Control, 136(5), p. 051011.

[39] Sturmfels, B., 1991, Sparse Elimination Theory, Vol. 91. Mathematical Sciences Institute, Cornell University, Ithaca, NY.

[40] Wolfram Mathematica, version 11.3, 2018, Champaign, IL.

[41] Ogata, K., 2009, Modern Control Engineering (5^{th} Edition), Pearson, London, United Kingdom.

[42] De Luca, A., Oriolo, G., and Samson, C., 1998, Feedback Control of a Nonholonomic Car-like Robot. Springer, Berlin, Germany, pp. 171–253.

[43] Brockett, R. W., 1983. "Asymptotic Stability and Feedback Stabilization". Differential Geometric Control Theory, Birkhauser, Basel, Switzerland, pp. 181–191.

[44] Oriolo, G., Luca, A. D., and Vendittelli, M., 2002, "WMR Control Via Dynamic Feedback Linearization: Design, Implementation, and Experimental Validation," IEEE Transactions on Control Systems Technology, 10(6), pp. 835–852.

[45] Consolini, L., Morbidi, F., Prattichizzo, D., and Tosques, M., 2008, "Leader–follower Formation Control of Nonholonomic Mobile Robots with Input Constraints," Automatica, 44(5), pp. 1343–1349.

[46] Hu, J., and Feng, G., 2010, "Distributed Tracking Control of Leader–follower Multi-agent Systems under Noisy Measurement," Automatica, 46(8), pp. 1382–1387.

[47] Cui, R., Ge, S. S., How, B. V. E., and Choo, Y. S., 2010, "Leader–follower Formation Control of Underactuated Autonomous Underwater Vehicles," Ocean Engineering, 37(17-18), pp. 1491–1502.

[48] Horn, R. A., Horn, R. A., and Johnson, C. R., 1990, Matrix Analysis, Cambridge University Press, Cambridge, MA.

[49] Varga, R. S., 2004, Geršgorin and His Circles. Springer, Berlin, Germany.

[50] Strang, G., Strang, G., Strang, G., and Strang, G., 1993, Introduction to Linear Alge-

bra, Vol. 3. Wellesley-Cambridge Press, Wellesley, MA.

[51] Huang, J.-Q., and Lewis, F. L., 2003, "Neural-network Predictive Control for Nonlinear Dynamic Systems with Time-delay," IEEE Transactions on Neural Networks, 14(2), pp. 377–389.

[52] Lai, C.-L., and Hsu, P.-L., 2010, "Design the Remote Control System with the Time-delay Estimator and the Adaptive Smith Predictor," IEEE Transactions on Industrial Informatics, 6(1), pp. 73–80.

[53] Kojima, K., Oguchi, T., Alvarez-Aguirre, A., and Nijmeijer, H., 2010, "Predictor-based Tracking Control of a Mobile Robot with Time-delays," IFAC Proceedings Volumes, 43(14), pp. 167–172.

[54] Vyhlídal, T., and Zítek, P., 2014, QPmR - Quasi-Polynomial Root-Finder: Algorithm Update and Examples. Springer, New York, NY, pp. 299–312.