

Experiments in modeling and control of multi-agent systems

Submitted in partial fulfillment of the requirements

of the degree of

Doctor of Philosophy

of the

Indian Institute of Technology Bombay, India

and

Monash University, Australia

by

Apurva Joshi

Supervisors:

Dr. Debraj Chakraborty (IIT Bombay)

and

Dr. Hoam Chung (Monash University)



The course of study for this award was developed jointly by Monash University, Australia and the Indian Institute of Technology, Bombay and was given academic recognition by both of them. The program was administered by The IITB-Monash Research Academy

January, 2019

To the curious,
Madhusudan Limaye

Thesis Approval

The thesis entitled

Experiments in modeling and control of multi-agent systems

by

Apurva Joshi

is approved for the degree of

Doctor of Philosophy

Examiner

Examiner

IITB Supervisor

Monash Supervisor

Chairman

Date: _____

Place: _____

Declaration

I declare that this written submission represents my ideas in my own words and where others ideas or words have been included, I have adequately cited and referenced the original sources. I also declare that I have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in my submission. I understand that any violation of the above will be cause for disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

Notice 1

Under the Copyright Act 1968, this thesis must be used only under the normal conditions of scholarly fair dealing. In particular no results or conclusions should be extracted from it, nor should it be copied or closely paraphrased in whole or in part without the written consent of the author. Proper written acknowledgement should be made for any assistance obtained from this thesis.

Notice 2

I certify that I have made all reasonable efforts to secure copyright permissions for third-party content included in this thesis and have not knowingly added copyright content to my work without the owners permission.

Student Name: Apurva Joshi

IITB ID: 144074002

Monash ID: 26414538

Copyright © 2019 by Apurva Joshi
All Rights Reserved

Abstract

This thesis presents experimental work in modeling and control of physical multi-vehicle systems. The first part focuses on *control* of multi-agent systems. We present the implementation of decentralized control algorithms for autonomous cooperative flight of a quadrotor team. In the second part, the focus is on *modeling* of multi-agent systems. We propose a model for vehicular traffic on broad roads, where lane discipline is not followed by human drivers.

Cooperative control of quadrotor teams: In this part we present experiments in decentralized cooperative control of multiple quadrotors. In the first set of experiments, a *min-max* time *consensus tracking* law is implemented on a fleet of quadrotors in an indoor environment. The *follower* quadrotors converge onto the reference trajectory generated by a *leader* quadrotor in min-max time using a local feedback control strategy which is known to be globally optimal. Further, experiments are performed to study the effect of finite communication/measurement rate on consensus tracking.

In the second set of experiments, a decentralized *consensus* law is implemented on a fleet of quadrotors in an outdoor environment. The proposed consensus law requires the exchange of only position information among agents and guarantees that the agents will asymptotically reach consensus, assuming that the communication graph remains connected at each instant of time. We demonstrate that for arbitrary initial positions of the quadrotors, the consensus law drives them to an autonomously decided common point. A novel communication protocol which guarantees reliable, real-time information exchange for such coordinated motion, is proposed. This protocol avoids data collisions and operates on-board a fully airborne system, without dependence on a ground station. It is also capable of handling changing communication graph topologies, temporary link-breaks and link-additions. Using this communication protocol, the quadrotors attain consensus for static and dynamic communication graphs. Experiments to observe the effect of communication rate on consensus performance are also conducted.

Model for lane-less vehicular traffic: In this part, a model is introduced for traffic on broad roads, where lane-discipline is not followed by human drivers. It is assumed that the driver reactions are influenced by possibly a number of vehicles, obstacles and other unmodeled entities in *visibility cones* to the front and to the sides of each vehicle, for motion along the longitudinal and lateral directions. *Influence graphs* are used to model the network of influences and resultant interactions. We review a model developed to predict motion of vehicles in congested traffic situations, where all drivers are forced to behave homogeneously. A model is proposed to extend these predictions for the motion of vehicles in sparse traffic conditions. It is shown

that in sparse traffic conditions, the velocity and inter-vehicle separations in the set of modeled vehicles are uniformly bounded. Experiments are performed to verify the traffic model. Videos of typical traffic on a sample road in Mumbai city, India, are recorded. Detailed information of vehicular motion is extracted through image processing techniques. This data is utilized to compare the actual trajectories of the vehicles with the model-predicted trajectories. It is verified that the proposed model can accurately predict complex maneuvers like overtaking, lateral movements and collision avoidance with slower moving vehicles, in addition to macroscopic patterns.

Contents

Preface	ix
Abstract	ix
List of Figures	xv
List of Tables	xvii
1 Introduction	1
1.1 Overview of work done	2
1.2 Organization of the thesis	6
2 Literature Review	7
2.1 Cooperative control of multi-agent systems	7
2.2 Modeling of lane-less traffic	12
2.3 Contributions	13
I Cooperative control of multi-agent systems	17
3 Basics of quadrotor motion	19
3.1 Frames of reference	19
3.2 Quadrotor dynamics	20
3.3 Basic quadrotor control	22
3.3.1 Inner control loops	22
3.3.2 Approximation as double integrators	24
3.3.3 Outer control loop: motion control	25
4 Testbed details	29
4.1 Indoor testbed	29
4.1.1 Vehicle details	29
4.1.2 Localization platform details	29
4.1.3 The host server	30
4.2 Outdoor testbed	30

4.2.1	The physical unit	31
4.2.2	The virtual unit	32
4.2.3	The communication module	32
4.2.4	Combining the physical and virtual units	32
5	Consensus tracking	35
5.1	Graphs	35
5.2	System description	36
5.3	Communication graph	36
5.4	Problem formulation	36
5.5	Solution to the min-max time consensus tracking problem	38
5.6	Experiments	39
5.6.1	Motion planning in \mathbb{R}^3	39
5.6.2	Implementation details	40
5.6.3	Experiment 1: One leader and one follower	41
5.6.4	Experiment 2: Multi-agent consensus tracking	41
5.6.5	Experiment 3: Effect of finite sampling rate	46
6	Consensus	49
6.1	Proposed consensus law	49
6.2	Communication Protocol	51
6.2.1	Steps involved	52
6.2.2	Addressing contingencies	54
6.3	Experiments	55
6.3.1	Communication protocol experiments	56
6.3.2	Cooperative control experiments	58
II	Modeling of multi-agent systems	67
7	Review of a model for lane-less traffic	69
7.1	Notation from graph theory	69
7.2	Modeling assumptions	70
7.2.1	For longitudinal (Y) direction	70
7.2.2	For lateral (X) direction	73
7.3	Homogeneous Traffic Model: Fixed Graph	74
7.3.1	Driver model for longitudinal direction	74
7.3.2	Analysis of longitudinal dynamics	75
7.3.3	Driver model for lateral direction	76
7.3.4	Analysis of lateral dynamics	76
7.3.5	Coupling between longitudinal and lateral motion	77

8	Proposed traffic model and verification	79
8.1	Modeling assumptions for heterogeneous traffic	79
8.1.1	Assumptions for longitudinal (Y) direction	79
8.1.2	Assumptions for lateral (X) direction	80
8.2	Heterogeneous Traffic Model: Time Varying Graph	80
8.3	Calibration of the model	84
8.3.1	Transformation from camera frame to local level frame	84
8.3.2	Conversion to X - Y axes	86
8.4	Model Validation	86
8.4.1	Congested regime: Time invariant graph	88
8.4.2	Sparse regime: Time varying graph, overtaking and lane changing	89
9	Conclusions and Future Work	93
9.1	Future Work	95
9.1.1	Cooperative control of multi-agent systems	95
9.1.2	Modeling of lane-less traffic	95
	Publications from this thesis	98
	Bibliography	99

List of Figures

1.1	Consensus tracking on an indoor testbed of quadrotors	4
1.2	Consensus on an outdoor testbed of quadrotors	5
3.1	Frames of reference: $\{E\}$ and $\{L\}$	19
3.2	Frames of reference: $\{L\}, \{B\}$ and $\{V\}$	21
3.3	Basic control architecture for a quadrotor	23
3.4	Setup to tune constants of switching surface for implementation of time optimal control	27
4.1	A photo of the indoor flying arena	30
4.2	Overall system architecture for the consensus tracking experiments	31
4.3	A block diagram of the physical quadrotor unit	31
4.4	A team of quadrotors constructed at IIT Bombay	32
4.5	A block diagram of the virtual quadrotor unit	32
5.1	Switching curve and optimal feedback control of agent a_i	39
5.2	A photo of the indoor flying arena	40
5.3	Consensus tracking: Two agent case, comparison between the difference trajectories of the quadrotor and ideal double integrator leader-follower pair	41
5.4	Consensus tracking: Two agent case, plot of the position, velocity and control input to the follower quadrotor	42
5.5	Communication graphs used in the multi-agent consensus tracking experiments	42
5.6	Consensus tracking: Three follower agents tracking a stationary leader	43
5.7	Consensus tracking: Three follower agents tracking a stationary leader	44
5.8	Consensus tracking: Three follower agents tracking a stationary leader	45
5.9	Consensus tracking: Three follower agents tracking a moving leader	46
5.10	Comparison between the difference trajectory of the leader-follower pair for the sampling time of 5 ms and 20 ms	47
5.11	Comparison between the difference trajectory of the leader-follower pair for the sampling time of 5 ms and 10 ms	48
6.1	A block diagram of the quadrotor control loops	52
6.2	Workflow of the communication protocol	53

6.3	Steps involved in the initial time synchronization	53
6.4	Data packet structure of Xbee in API mode	54
6.5	An instance of case of link break between Nodes 1 and 3 due to the presence of an obstacle	56
6.6	Data transfer during the link break mode of operation	58
6.7	A snapshot of nine time-frames of the communication protocol	59
6.8	Consensus: communication graph	60
6.9	Consensus: Comparison between simulated and experimental trajectories of three agents	61
6.10	Consensus: trajectories of three physical quadrotors	61
6.11	Consensus: trajectories of three physical quadrotors and three virtual quadrotors	62
6.12	Consensus: communication graph switching	63
6.13	Centroid tracking in the case of switching communication topology	63
6.14	GPS plot of consensus between three physical quadrotors with a dynamic com- munication graph	64
6.15	Plot of quadrotor trajectories for different data exchange rates	65
6.16	Plot of quadrotor trajectories for different data exchange rates	65
7.1	Different influence graphs for Y direction depending on the angle of view (aov) of the vehicles	72
7.2	Influence graph for X direction depending on the angle of view (aov) of the vehicles	74
8.1	Influence graph for Y direction with unmodeled vehicles	80
8.2	The convoy of vehicles with the influence graphs	81
8.3	Switching influence graphs	81
8.4	The camera frame $\{C\}$	85
8.5	The local level frame $\{L\}$:	85
8.6	Boundary polynomials	85
8.7	Traffic model: tracking errors (RMS) for the 7 cases along X -axis	87
8.8	Traffic model: tracking errors (RMS) for the 7 cases along Y -axis	87
8.9	Traffic model: comparison between actual and simulated trajectories	90
8.10	Traffic model: tracking errors (absolute) of each vehicle along Y -axis	91

List of Tables

2.1	A literature review of vertical take-off and landing (VTOL) multi-vehicle testbeds	11
5.1	Consensus tracking details for Experiment 2, case 1	43
5.2	Consensus tracking details for Experiment 2, case 1	44
5.3	Consensus tracking details for Experiment 2, case 2	44
5.4	Time taken by follower to track stationary leader for different sampling times	47
6.1	Sample packet transmission times of Node 0 during initialization phase	56
6.2	A comparison of number of nodes in the network and communication efficiency	57
6.3	A comparison of communication efficiency in indoor and outdoor environment	58
6.4	Position data of the quadrotors for Experiment 2(a)	60
6.5	Position data of the quadrotors for Experiment 2(b)	62
6.6	Position data of the quadrotors for Experiment 3(b)	63
8.1	Consolidated cumulative errors for all cases.	88
8.2	Cumulative tracking errors for each vehicle: time invariant graph, case S_1 .	89
8.3	Cumulative tracking errors for each vehicle: time varying graph, case S_3 .	89

Chapter 1

Introduction

A multi-agent system (MAS) is a collection of agents¹, wherein each agent possesses limited processing, local sensing and communication capabilities. Cooperative control deals with the design and analysis of actions that the agents take in order to achieve a common objective. These *distributed* actions are based on exchange of local information between agents. This information exchange gives rise to a networked system. Networked systems have been studied in a variety of subjects. For example, they are used in biology to model the working and wiring of transcriptional regulatory circuits [RC10] and molecular programming [HSM18]. Sociologists have used networks to analyze and predict behavior of techno-social systems [ME10], [ÖDF⁺18], [SL14]. In physics, networks are used for modeling and prediction of the emergence of behavior norms [RC10], [VCBJ⁺95], [SL18]. In engineering, technological advancements have facilitated the synthesis of networked dynamical systems, such as multi-vehicle systems and sensor networks [COSS11].

Our focus in this thesis is on multi-vehicle systems. It is known that teams of autonomous vehicles operating in coordination can lead to a greater operational capability and efficiency compared to autonomous vehicles that perform solo missions [RB08]. There are ample applications of autonomous multi-vehicle systems in the civilian and military domain: distributed sensor networks [EGPS01], hazardous material handling [OMS01], humanitarian aid and disaster relief [GFS⁺16], space-based interferometers [SHP04], combat, surveillance and reconnaissance systems [FTB03] to name a few. To enable these applications, it is essential to develop and implement various cooperative control capabilities like rendezvous, formation control, flocking, foraging, attitude alignment, etc.

Over the past couple of decades there has been a lot of research on the theoretical development of cooperative control laws for multi-agent systems [CYRC13]. However, implementation of cooperative control laws on physical systems poses significant challenges:

- A stable system of systems needs to be developed rather than just a single system.
- Due to the wireless medium and decentralized mode of communication, information ex-

¹Agent: an entity which can perceive its environment through sensors and act on it through actuators [RN16]

change between the agents can be unreliable.

- Limited computational resources are available for each vehicle.

While the engineering challenges in construction and control of such a multi-vehicle fleet are addressed in the first part of the thesis, it is equally important to note that significantly more complex multi-agent interactions exist in nature [Rey87], [SMW⁺14]. Even more interestingly, complex human-engineered multi-agent systems exist almost everywhere in the world. Consider the case of vehicular traffic. Multiple vehicles ply on shared roads everyday. It is important to observe and infer laws of interaction from this highly complex but very successful interactive behavior. There have been several attempts to model behavior of human drivers maneuvering vehicles to understand traffic flow at macroscopic and microscopic level [May90]. However, most work in this domain has been focused on modeling of vehicles which strictly follow lane discipline [GHR61]. Such models fail to predict motion of vehicles in developing countries, where strict lane discipline is not observed, resulting in a more complex interactive behavior. This drawback is addressed in this thesis and a novel model for lane-less traffic is proposed. Apart from immediate uses in transportation planning and analysis, it will help us engineer better networked systems in future.

1.1 Overview of work done

The work done in this thesis is divided in two parts. In the first part, we shall look into the design and analysis of cooperative control laws and their implementation on physical systems. In the second part, we introduce a model developed for vehicular traffic applicable to cases where lane-following is not adhered to. An overview of these two parts now follows.

Part I. Cooperative control of multi-quadrotor systems

Quadrotor helicopters (generally referred to as quadrotors) are flying vehicles capable of vertical take-off and landing (VTOL). This makes them suitable for cluttered and urban environments. Their hovering capabilities can be exploited in applications involving surveillance [W⁺06], videography [JRT⁺15] and distributed sensor networks [VSB⁺11]. Availability of cheap miniature embedded systems for computation, communication and sensing have made it possible to easily construct teams of quadrotors. Hence, a team of quadrotors is chosen as the platform to test cooperative control laws.

Cooperation requires multiple vehicles to agree on the value of a variable of interest. The two problems in cooperative control considered in this work are that of *consensus* and *consensus tracking* of multiple vehicles. To achieve consensus, the vehicles are required to agree on a value of the variable of interest. In consensus tracking, all vehicles are required to track a constant or time-varying consensus state. The solutions of these problems can be applied to common multi-vehicle mission maneuvers like rendezvous and formation flight. This is illustrated in

this thesis through the implementation of the consensus-based cooperative control laws on the multi-quadrotor platform. During practical implementation, there are several issues that need to be tackled, such as imperfect sensors with limited accuracy and update rates, and limited on-board computation. Wireless communication between agents gives rise to issues like limited range, packet drop, interference, etc. We conduct experiments to study the effect of such issues on the multi-agent system performance. Further, a novel fully airborne communication protocol is designed and experimentally verified in an outdoor environment. Let us now look at these problems in detail.

Consensus tracking

In the consensus tracking problem, a special agent, referred to as a *leader* generates a reference trajectory on to which the remaining agents, referred to as *followers*, need to converge using information available locally to each agent [RB08]. Follower agents generally only communicate with *neighboring* agents, where proximity might be based on a network architecture or geographical distance [BCM09] [ME10]. It is essential in some applications that consensus tracking be achieved in minimum time [ST94] [KSBY00]. For example, in the case of multiple unmanned aerial vehicles operating in hostile environments, it is essential that rogue invading agents be captured in minimum time [RWB⁺05] [OS06].

A local information based feedback control strategy is developed in [MC18], to achieve *min-max* time consensus tracking of double-integrator agents with bounded inputs. In [MC18], communication between any two agents is assumed to be possible only if the distance between them is less than a fixed threshold. A decentralized algorithm to extract a rooted spanning tree from a connected communication graph is given. A spanning tree rooted at the leader agent is obtained from the initial connected graph using this algorithm. Each pair of connected agents in this tree is modeled as a pursuer-evader pair and it is proposed that the local time optimal feedback pursuit policy be used by each agent. For the selected spanning tree, this proposed local control is shown to be the communication preserving global min-max time strategy.

The control strategy proposed in [MC18] is implemented in this work on a testbed of aerial robots under various practical situations. In order to implement the feedback control strategies proposed in [MC18], each agent needs to measure its own state and that of its neighbors, instantaneously. In practice, agents measure their states (position and velocity) using various sensors [LPLK12]. Such measurements can only be done at a finite rate [ANT10]. Further, each agent needs to communicate with its neighbors in order to obtain their states. The communication over any communication channel also happens at a finite rate. Due to these limitations, agents cannot obtain the instantaneous values of their states and that of their neighbors. In this thesis, experiments to observe the effect of these limitations are performed. Figure 1.1 shows the implementation of consensus tracking carried out on a quadrotor fleet in an indoor environment.

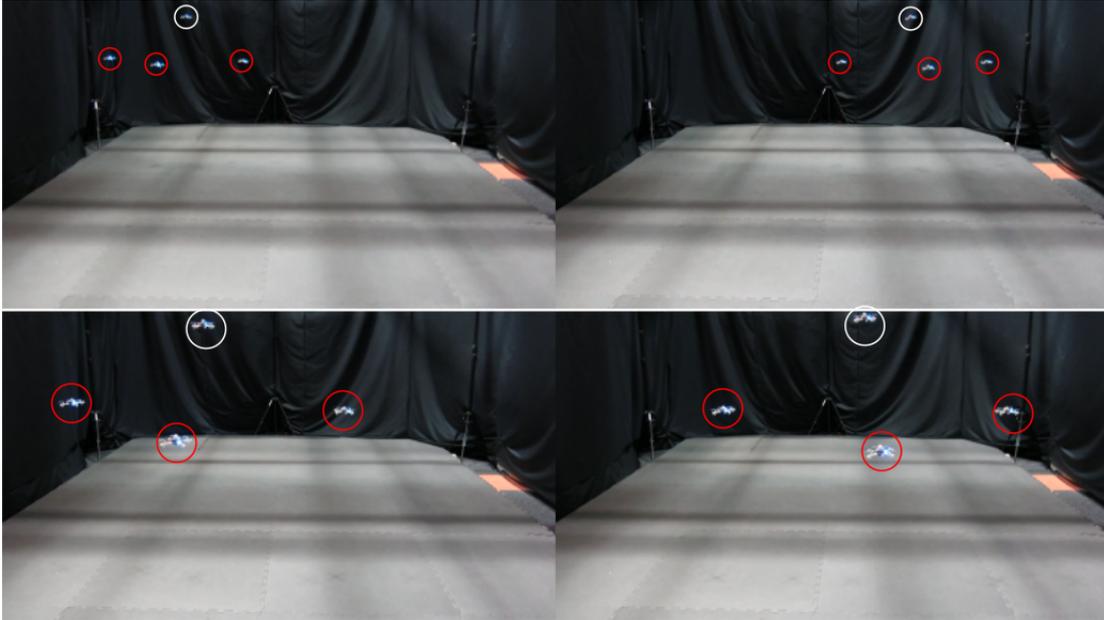


Figure 1.1: Consensus tracking being performed on an indoor testbed of four quadrotors. The *follower* quadrotors (red) track an arbitrary trajectory generated by a *leader* quadrotor (white)

Consensus

For a team of vehicles to reach consensus they need converge onto a common value of state (position and velocity). In this work, a decentralized consensus law, which is a variation of [RA07], is proposed to guarantee that the agents will asymptotically reach an autonomously decided common location, assuming that the communication graph between the agents remains connected at each instant of time. This consensus law requires only the exchange of position information among neighboring agents in order to drive them to a common point. Due to the decentralized nature of the consensus law, the trajectory computations are done completely on-board the quadrotors. This consensus law is implemented on-board a testbed of six (three physical and three virtual) quadrotors and a rendezvous maneuver is demonstrated through several experiments. Figure 1.2 shows the implementation of the consensus law on a quadrotor fleet in an outdoor environment.

Further, a novel communication protocol is proposed for facilitating the real-time exchange of data between multiple agents. This protocol avoids data packet collisions using time-slotting. During initialization, the clocks of all the agents are synchronized and each agent transmits data in an allotted time slot. The protocol has the ability to maintain synchronization and handle changes in the communication topology due to: obstacles, or the motion of the agents entering and leaving each others' communication range. This protocol is implemented on-board the quadrotors and is fully airborne, thus eliminating the requirement for a ground station.

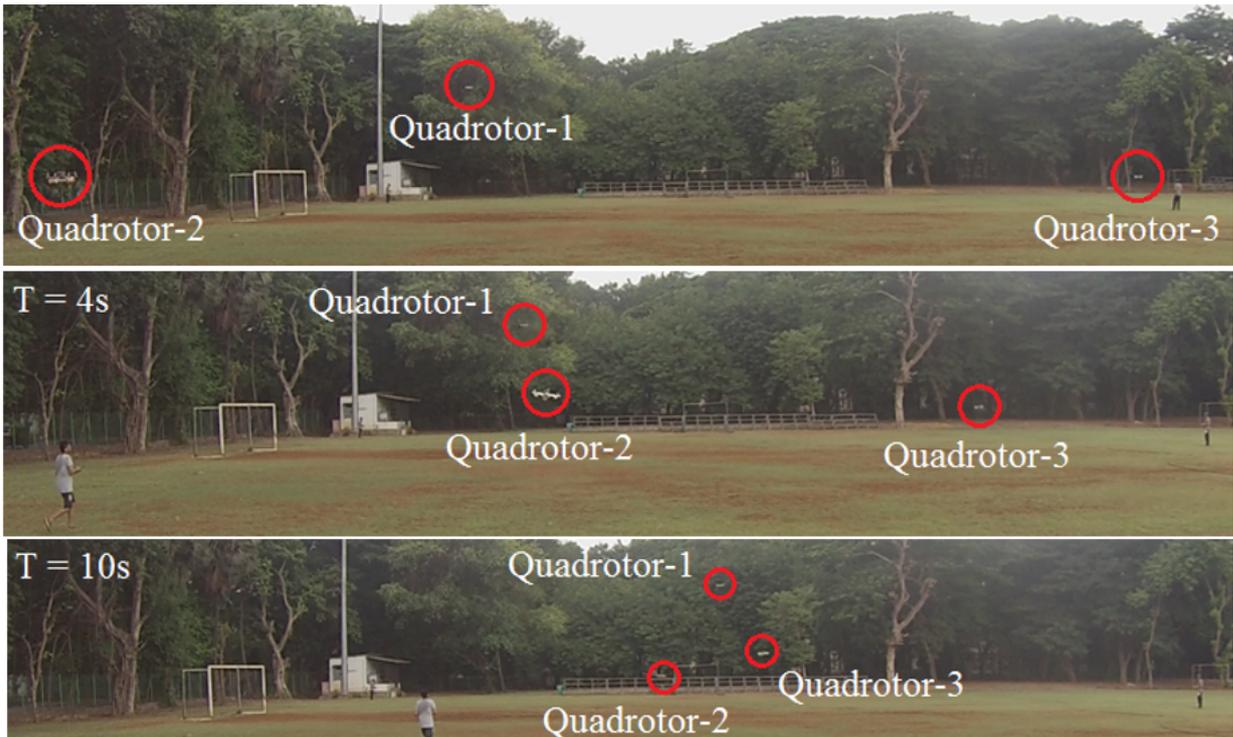


Figure 1.2: Consensus being achieved on an outdoor testbed of three quadrotors. All quadrotors (red) are seen to autonomously navigate to a common point (rendezvous)

Part II. Modeling of lane-less traffic

In this part of the thesis, we focus on understanding the manner in which human drivers successfully coordinate in order to navigate vehicles in traffic conditions where lane discipline is not followed. In Indian conditions where the roads are often wide, multiple vehicles drive abreast, but lane discipline is never observed. Such “lawless” driving is often characterized as chaotic [OH08]. In an effort to model such behavior, a dynamic model is introduced in [CMCM16]. The primary hypothesis in the lane-less model introduced in [CMCM16] is that it is not enough to consider only the vehicle directly in front, since vehicles to the side can suddenly move in. Consequently, each driver needs to watch all vehicles in a ‘cone’ rather than only the vehicle right ahead. Similarly, for lateral motion (similar to lane changing in laned traffic), each driver needs to watch vehicles in two separate cones to his/her right and left. Each driver is affected by vehicles in these cones and in turn affects other drivers who can see his/her vehicle, creating connected *influence graphs*. Using these graphs, the laws governing the microscopic behavior of vehicles in dense traffic conditions is proposed in [MJC⁺18]. In this thesis we present an extension of this model to predict behavior of vehicles in sparse traffic conditions, where complex maneuvers like overtaking are observed. The validity of these models and the underlying assumptions is established by comparing the model predicted trajectories with actual motion of real vehicles extracted from video recorded traffic data on a sample Indian road.

1.2 Organization of the thesis

The thesis is organized as follows: we begin by reviewing relevant literature and highlighting the contributions of our work in Chapter 2. In the first part of the thesis, we will look at different algorithms for control of quadrotor teams. Chapter 3 introduces the basics of quadrotor motion. Details regarding the testbeds constructed for implementation of the cooperative control algorithms are provided in Chapter 4. A solution to the consensus tracking problem is reviewed and its implementation is discussed in Chapter 5. Chapter 6 contains details regarding the implementation of a decentralized consensus algorithm along with a novel communication protocol. The second part of the thesis presents a mathematical model for lane-less traffic. A model for dense traffic is reviewed in Chapter 7. This model is extended for sparse traffic in Chapter 8 and then a validation of both these models is presented.

Chapter 2

Literature Review

In this section, a brief review of the literature related to control of multi-agent systems and modeling of vehicular traffic is presented. The literature related to cooperative control of quadrotor fleets is summarized in Section 2.1. In Section 2.2, work related to modeling of vehicular traffic is presented. The contributions of this thesis are highlighted in Section 2.3

2.1 Cooperative control of multi-agent systems

The study of collective behavior where multiple agents influence each others' dynamics through local interaction laws has been studied since [VCBJ⁺95]. Control laws have been designed for such systems in order to impose a desired collective behavior to serve potential applications like localization and rescue during calamities [CKBM06], estimation of traffic density [Pur05], surveillance and intelligence gathering [AAMO13], mapping [MSM⁺12], construction [LMK11], load transportation [LSK13] etc.

A survey of theoretical tools for cooperative control of multi-agent systems with general linear and nonlinear dynamics can be found in [RBA05], [OPA15], [QMSW16] and [CPD⁺18]. An extensive review of literature for consensus based distributed cooperative control of multi-vehicle systems can be found in [RB08], [BCM09] (and references therein).

We shall now look at literature addressing two problems considered in the thesis: (a) consensus problem and (b) consensus tracking problem. In the consensus problem, the consensus state is decided collectively by the agents based on their initial condition. A variant of this problem is the consensus tracking problem, wherein a trajectory is generated autonomously by a special agent(s) which acts as the reference trajectory onto which the remaining agents need to converge using local control laws. Let us now look at some of the solutions to these problems found in literature.

Consensus problem

The observations made in [VCBJ⁺95] are first explained mathematically in [JLM03], wherein consensus laws are proposed for discrete time multi-agent systems with single integrator dy-

namics. A continuous time multi-agent system with single integrator dynamics is considered in [OSM04a]. In this paper, various scenarios like directed and undirected communication, fixed and varying communication graph are studied. The effect of time delays in communication for multi-agent systems with a fixed undirected communication graph are also studied.

In [RA07], [GWJ09], [AT10], consensus laws for double integrators with position and velocity as state variables, are presented. Results for multi-agent systems with single integrator agents are extended for double integrator agents in [RA07]. Necessary and sufficient conditions for consensus given a directed communication topology are also derived. In [GWJ09], consensus law without velocity measurement is proposed. The case of bounded inputs is considered in [AT10]. Results for consensus of general linear systems are presented in [FM04].

Control laws have also been designed for consensus of nonlinear systems. In [YCCCK09], second order consensus laws for directed topologies have been proposed and notions of generalized algebraic connectivity have been proposed. An adaptive consensus law for all undirected connected graphs is proposed in [LRLF12]. In [MRM13], a control law is proposed to drive nonlinear MAS to consensus, under undirected communication graphs, only using relative position measurements.

Consensus tracking problem

The consensus tracking problem can be viewed as an extension of the trajectory tracking problem for the multi-agent case. This problem is discussed in [Ren08], [CR12]. In [JLM03], the consensus tracking problem with fixed leader state and undirected communication between agents, is addressed. The case where leaders are dynamic, the directed communication graph is fixed and inputs to the agents are bounded, is considered in [Ren08]. In [ML07], the case of multi-agent systems with changing graphs and dynamic leaders is considered.

Consensus tracking laws have also been developed for higher order nonlinear systems. In [ZL12], a robust, adaptive neural network based controller is designed to synchronize all follower nodes to the leader node with bounded residual errors, wherein all agents communicate over a weighted directed graph with, fixed topology. An adaptive backstepping consensus controller is presented in [CWLL15] for consensus tracking over time-invariant undirected graphs. A neural network-based state observer for each follower is designed to solve the unmeasurable state problem of high-order nonlinear multi-agent systems. Robust and adaptive consensus tracking problems with parameter uncertainties are studied in [ZLD13], and a class of discontinuous protocols for agents communicating over undirected time-invariant graphs are proposed. The case of time-varying communication graphs is considered in [WDCY13].

Time to consensus

The speed at which agents of the multi-agent system reach consensus is an important parameter in designing the control laws. There are several control laws (see [RB08], [ME10] and references therein) which achieve asymptotic consensus. In such control laws, the speed of

consensus is characterized by the *algebraic connectivity* i.e. the second largest eigenvalue of the Laplacian of the communication graph [OSM04b], [RB08].

The problem of designing control laws to achieve finite time consensus among multiple single integrator agents is studied in [Cor06]. Control laws which achieve consensus in finite time for double integrator dynamics are proposed in [WH08]. In [ZW12], finite time consensus algorithm for a heterogeneous multi-agent system with single and double integrator agents is proposed. The finite time consensus tracking problem is considered in several papers like [CRM10], [LDL11], etc. In [CRM10], finite time consensus tracking is achieved using distributed sliding mode control. A finite time consensus tracking algorithm for double integrator agents without velocity information from neighboring agents, is proposed in [LDL11]. In [ZDWC15], [ZDWZ13], [ZWC16], algorithms have been developed for finite time consensus tracking of double integrator agents under different scenarios like: single and multiple leaders, with/without velocity information from neighboring agents, etc.

The problem of time optimal consensus is studied in [YSS⁺13], in which a first order discrete time multi-agent system is considered. The agents use the consensus law proposed in [JLM03]. A decentralized algorithm to compute the final consensus value of the network in finite time and minimum number of steps, is proposed. However, the actual consensus is achieved asymptotically. In [MC18], the authors propose a local information based feedback control strategy to achieve min-max time consensus tracking of double integrator agents with bounded inputs. Each agent is assumed to have limited communication range. A decentralized algorithm is proposed to extract a spanning tree rooted at the leader node, from a given connected communication graph. Each pair of agents in this tree is modeled as a pursuit-evasion pair and a local time optimal feedback pursuit policy is proposed for each agent. It is shown that for the selected spanning tree, the proposed control strategy is the communication preserving global min-max time strategy. In this thesis, we implement the control strategy proposed in [MC18] on a testbed of quadrotors. However in practice, due to limitations in sensing equipments, there are constraints on the rate at which state information is available to each agent. An analysis of the effect of these limitations on the performance of the min-max time consensus proposed in [MC18], is presented in [JSCC19].

Practical implementation

On the theoretical front, substantial literature is available on distributed cooperative control of multi-vehicle systems. However, there is limited work done on implementation and testing of cooperative control algorithms on practical systems. A brief survey of the vertical take-off and landing (VTOL) multi-vehicle testbeds that implement cooperative control for formation flight and trajectory tracking is as follows:

Indoor testbeds: In [KMPK13], the authors describe a centralized control algorithm to demonstrate tight flight formations, while a decentralized control algorithm is implemented in [TMK12] to achieve linear and rectangular-shaped formation flight. An open-loop control strategy is de-

veloped and implemented to achieve consensus tracking of a group of quadrotors while minimizing an integral cost function. In both cases, actual execution of the algorithm takes place on a central computer. Formations are achieved in [PHSA17] by generating piecewise polynomials which satisfy given waypoint and continuity constraints, but trajectories are computed and uploaded from the base station. Double-integrator based decentralized consensus algorithms have been used to achieve formation control among quadrotors in [SFZR15]. However, these experiments have been performed in a controlled indoor environment. The problem of formation control with the goal of maintaining bearing rigidity was addressed in [SG17] and a decentralized gradient-based control law was developed and implemented.

Outdoor testbeds: Formation flight in outdoor conditions with two physical industrial grade unmanned helicopters, and seven simulated helicopters is reported in [SCHS07]. However, the control laws used are not decentralized in nature. In [BSK11], formation control was achieved using centralized controller and point-to-point communication between agents and ground station. Collision-free formation flight using Nash bargaining was demonstrated in [HHWT11]. Successful execution of decentralized multi-quadrotor flock which performs stable autonomous outdoor flight is given in [VVS⁺14], [KPA17]. While [VVS⁺14] uses an empirical control law, [KPA17] proves persistence of formation with a single integrator model. In the absence of low level access to the autopilot control loops in either of these papers, it is not clear how the proposed theoretical models are applicable to quadrotors. In [DZRZ16] and [DYSZ15], a Riccati equation based consensus tracking strategy is proposed and implemented on an outdoor testbed of quadrotors. Formation control using visual inertial odometry is achieved in [WCLK18]. A summary of vertical take-off and landing (VTOL) multi-vehicle testbeds is given in Table 2.1.

Table 2.1: A literature review of vertical take-off and landing (VTOL) multi-vehicle testbeds

Author	Year	Arena	Decentralized				No. of agents	Dependencies	Remarks
			Control	Communication	Control	Communication			
Turpin et. al. [TMK12]	2012	Indoor	Software yes, hardware - no	No	No	4	Motion capture system, central computation	Open loop control strategy	
Kushleyev et. al. [KMPK13]	2013	Indoor	No	No	No	20	Motion capture system, central computation	Centralized control	
Saif et. al. [SFZR15]	2015	Indoor	Software yes, hardware - no	No	No	4	Motion capture system, central computation	Double integrator consensus algorithms, time invariant graphs	
Preiss et. al. [PHSA17]	2017	Indoor	Software yes, hardware - no	No	No	49	Motion capture system, central computation	Formation flight by generating piecewise polynomials a-proxi	
Schiano et. al. [SG17]	2017	Indoor	Yes	Yes	Yes	5	On-board cameras (failed); Mo- tion capture system used	Formation control using bearing rigidity	
Shaw et. al. [SCHS07]	2007	Outdoor	No, leader info with everyone	Token ring	Token ring	2 + 6*	GPS	Scalability issues	
Bürkle et al. [BSK11]	2011	Outdoor	No	No	No		GPS, point-to-point communi- cation	Intensive control processing at ground sta- tion	
Hoffmann et. al. [HHWT11]	2011	Outdoor, Indoor	Yes	Not available	Not available	2-3	GPS, on-board cameras	Collision-free flight using algorithms based on Nash Bargaining	
Vasarhelyi et. al [VVS ⁺ 14]	2014	Outdoor	Yes	Xbee, CSMA/CA	Xbee, CSMA/CA	10	GPS	Empirical control laws	
Dong et. al. [DYSZ15]	2015	Outdoor	Yes	Xbee, CSMA/CA	Xbee, CSMA/CA	5	GPS	Double integrator consensus algorithms, time invariant graphs	
Dong et. al. [DZRZ16]	2016	Outdoor	Yes	Xbee, CSMA/CA	Xbee, CSMA/CA	4	GPS	ARE based formation control protocol for time varying graphs	
Kang et. al. [KPA17]	2017	Outdoor	Yes	Xbee, CSMA/CA	Xbee, CSMA/CA	5	GPS	Single integrator consensus laws applied to physical system	
Weinstein et. al [WCLK18]	2018	Outdoor, Indoor	Yes	WiFi	WiFi	12	On-board cameras	Centralized task assignment, heavy on- board computation	

* simulated agents

Communication for cooperative motion

The thrust of research in recent times has been towards development of control algorithms, wherein inter-agent communication is assumed to be perfect. The state of the art in practical multi-UAV communication is available in [BST13], [NCKS17] (and references therein). For communication in indoor environments, the agents keep sending data packets repeatedly to make up for unreliability in data reception [LHM⁺14]. In [PHSA17], to achieve low latency, the authors aim for high probability of communication by repeating request-response commands until acknowledged or a timeout occurs. There is no mechanism to avoid data collisions. In outdoor experiments performed in [SCHS07], a token ring communication protocol is used for collision-free inter-agent communication. Packet loss during wireless communication was one of the contributing factors to the experimental results differing from the ideal theoretical results [SCHS07]. In [VVS⁺14], to achieve consensus among quadrotors, XBee modules working in broadcast mode, which use standard MAC protocols with randomized collision resolution, are used for information exchange. The unpredictable behavior of such protocols lead to delays in data transmission and deterioration in convergence accuracy/rates which are unacceptable in real-time applications [GGC13]. A study of the effect of control gain, communication update rate and graph topologies performed in [MSKR14] shows that timely dissemination of information and its reception are vital for consensus to occur among agents. A survey of such issues in UAV communication networks is presented in [GJV16].

The review of literature in distributed cooperative control of quadrotor teams shows that there is scope for implementation of decentralized cooperative control laws with mathematically provable convergence guarantees on practical systems. There is also a need to develop the underlying communication protocols which guarantee time-bound exchange of information in lightweight airborne systems. These issues are addressed in the first part of the thesis.

2.2 Modeling of lane-less traffic

Mathematical models of traffic are important for predictions [Cla03] and real-time control of traffic [HSL⁺15, CC17], infrastructural planning [WSD⁺05], study of traffic conditions on economy and environments and various other applications (e.g., see [Tre75]-[WSCW17] and references therein). Traffic models are broadly classified into two types [MM16]: the *macroscopic models* attempt to describe the speed-flow-density relations of a traffic stream. Examples include the Lighthill-Whitham-Richards (LWR) model [LW55], the force models [Hel95], fluid flow [KW98] and gaseous flow analogies [AR00], stochastic models [PS15] and wave models [JGL15]. On the other hand, *microscopic* models describe the behavior of individual vehicles in various traffic scenarios [GHR61, TMH15]. The most common microscopic model is the car-following model [GHR61], whereas variations and extensions are studied in [CLZA12]-[Bex68]. Later developments include models of lateral movements like lane-changing [TMH15, Zhe14]-[HÖR⁺03] and obstacle avoidance [SM03]. Various forms of sta-

bility such as input-to-state [TKP02], string stability [SH96] and mesh stability [SPH99]-[SZ12] have also been considered. For a detailed survey, we refer the reader to [MM16, Zhe14, AKT16] and references therein.

Most of the literature mentioned above, is focused on modeling laned traffic, i.e., the situation where the drivers abide by lane-discipline. Similar models, in the absence of more realistic theory, have been used to model Indian traffic [AKSS16, BGC03] which, however, does not obey lane discipline. To address this gap, several attempts have been made recently to simulate traffic in typical Indian conditions (e.g., see [MM16] and references therein): highly heterogeneous vehicles with maximum speed capabilities and driver behavior differing widely across modern high-powered passenger cars to old overloaded goods carriers [BM12]-[FW15]. However, these models are based on ad-hoc rules, with no tractable mathematical properties. Hence, no a priori guarantees of convergence or boundedness of simulated trajectories can be given. Given the importance of such models for understanding road traffic behavior, traffic simulation and transportation planning, we believe a more mathematically rigorous model, with provable stability properties, is necessary.

A theoretical model and consequent predictions need empirical validation. Various attempts have been made to record traffic data using simulations [PS15, CLZA12], video recording through helicopters [OH08, Tol74] or high-mounted cameras [MBK13, Adm17], using vehicles equipped with sensors [CC17, Roc68]-[MR13], and loops embedded in roads [Cla03]. A well-known example of traffic data set is Next Generation SIMulation (NGSIM) [Adm17], which has been used widely for traffic model verification [WSCW17, TMH15, CLZA12, TLR10, Lav14, KM11] - [LVH09]. However, the NGSIM data is recorded for laned traffic and hence is not suitable for the validation of the models presented in [MJC⁺18]. Hence, for verifying that the proposed model is able to predict trajectories of real cars on Indian roads, the actual trajectories of vehicles, during normal traffic conditions, are recorded using a camera mounted atop a building adjoining a sample road in Mumbai, India.

2.3 Contributions

Cooperative control of multi-quadrotor teams

In this work, we show that using the low level access to quadrotor control loops, quadrotors can be approximated as two independent double integrator systems. Thus, available theory for consensus and consensus tracking of double integrator agents can be modified and applied to quadrotors.

Consensus tracking

The contributions of this work are as follows:

- (a) *Implementation of consensus tracking*: Min-max time consensus tracking on a testbed

of four quadrotors has been performed in an indoor environment. To the best of the author's knowledge, minimum-time consensus tracking has not been implemented on a testbed of multiple quadrotors. The trajectories of the leader agent are generated manually. The three follower agents converge onto this reference trajectory in min-max time using the local feedback control strategy proposed in [MC18]. The source-code of the implementation is made available at [Jos18]. The implemented algorithm is decentralized and is computationally cheap due to the availability of closed form expressions of the feedback law. Thus, unlike open-loop cooperative control implementations (see [TMK12]), computationally intensive numerical methods are not required.

(b) *Effect of non-zero sampling time on consensus tracking*: Experiments are performed to observe the effect of non-zero sampling time (i.e., the time interval between two consecutive time instants at which agents receive the states of their neighbors) on the min-max time consensus tracking control strategy proposed in [MC18].

Consensus

The contributions in this work are as follows:

(1) *Theoretically provable, experimentally verified consensus law*: A consensus law for double integrator dynamics, which is a variation of [RA07] and requires the exchange of only the position information among agents, is proposed. It guarantees that the agents will asymptotically reach an autonomously decided consensus point, assuming the communication graph remains connected at each instant of time. The trajectory generation is done on-board and there is no requirement of a ground station except for telemetry purposes. In order to implement the consensus law, the quadrotor motion is decoupled along two orthogonal axes in the horizontal plane. The quadrotors are kept at different altitudes in order to avoid collisions. Experiments are performed to provide a comparison between the trajectories generated by double integrator agents and physical units, and to demonstrate successful consensus with six quadrotors.

(2) *Implementation of a scalable airborne communication protocol*: A novel communication protocol is proposed for facilitating the real-time exchange of data between multiple agents. This protocol avoids data packet collisions using time-slotting. During initialization, the clocks of all the agents are synchronized and each agent transmits data in an allotted time slot. The protocol has the ability to maintain synchronization and handle changes in the communication topology due to: obstacles, or the motion of the agents entering and leaving each others' communication range. This protocol is implemented on-board the quadrotors and is fully airborne, thus eliminating the requirement for a ground station. An average transmission and reception efficiency of 98.04% and 98.52% respectively is achieved for complete information exchange between three to six agents for 18 runs of 300 s each in indoor and outdoor environments.

(3) *Effect of communication on control and convergence*: Experiments are performed wherein the rate at which data is exchanged between the agents using the proposed law to reach consensus, is altered. It is observed that slower communication rates progressively lead

to degraded performance in reaching consensus.

Modeling of lane-less traffic

In this thesis, the model proposed in [MJC⁺18] for predicting the motion of vehicles in dense traffic conditions is extended. A model to predict motion of vehicles in sparse traffic conditions is proposed. For verifying that these models are able to predict trajectories of real cars on Indian roads, the actual trajectories of vehicles during normal traffic conditions are recorded using a camera mounted atop a building adjoining a sample road in Mumbai, India. For our analysis and comparison with the proposed model we extract, using image processing techniques, information about vehicle trajectories in \mathbb{R}^2 . These trajectories are then compared with the trajectories generated by the proposed model initialized with measured data. Due to infrastructural and feasibility constraints, our recordings are limited to approximately 230 m of the road. As the average speed of vehicles in this stretch of road is 40-50 km/h, the duration of the analysis is around 15 s. To compensate for this short duration, the analysis is repeated for multiple sets of vehicles and the statistical performance of the proposed model is characterized. We demonstrate that in dense traffic, the vehicles do converge to a layered platoon like structure with fixed inter-vehicle and inter-layer distances. In sparse traffic, the proposed model correctly predicts dynamic lane-changing based on availability of free space, overtaking slower vehicles and maneuvering to avoid slower vehicles. Comparisons of the model predicted and actual vehicle trajectories, evolution of RMS error and cumulative values of prediction errors over time are also presented.

Part I

Cooperative control of multi-agent systems

Chapter 3

Basics of quadrotor motion

In this chapter we review the basics of a quadrotor system. We start by introducing the frames of reference that are essential in analyzing the motion of agents in Section 3.1. The quadrotor dynamics are presented in Section 3.2. The control architecture is presented in Section 3.3. In Section 3.3.2, we justify the approximation of quadrotor dynamics as two independent double integrator systems.

3.1 Frames of reference

To analyse the motion of the multi-agent system comprising of quadrotors as agents, standard definitions [NKG12] for the frames of reference are used in the analysis (see Fig. 3.1 and Fig. 3.2).

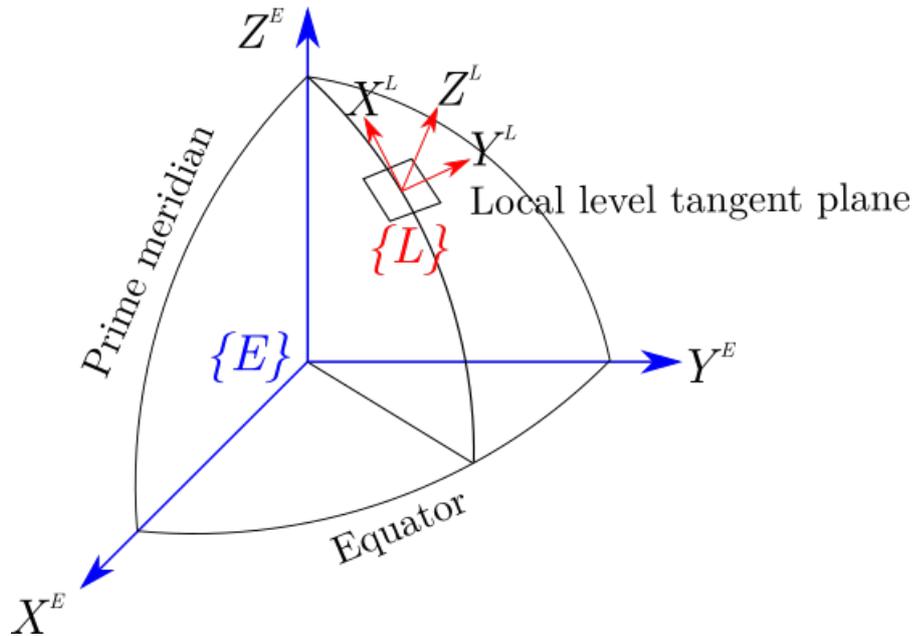


Figure 3.1: A depiction of the $\{E\}$ and $\{L\}$ frames of reference

Earth-centered Earth-fixed frame, $\{E\}$

This is considered to be an inertial frame of reference and is defined as follows: (a) The origin is at the centre of mass of the Earth (b) The Z^E – axis passes through the geographical north pole (c) The X^E – axis passes through the intersection of the equatorial plane and the Greenwich meridian (d) The Y^E – axis completes the right-hand coordinate system.

Local-level frame, $\{L\}$

The local-level frame or the navigation frame is set up as follows: (a) The origin is at a fixed point on Earth (b) The X^L – axis points towards the north (c) The Y^L – axis points towards the east (d) The Z^L – axis points upwards.

Body frame, $\{B\}$

The body frame $\{B\}$ is oriented as follows: (a) The origin is at the centre of mass of the quadrotor (b) The X^B – axis points towards the front end of the quadrotor (c) The Y^B – axis points towards the right end of the quadrotor (d) The Z^B – axis points in the downward direction as shown in Fig. 3.2.

Body-fixed frame, $\{V\}$

A body-fixed coordinate frame $\{V\}$ is attached to the vehicle [Cor11] such that (a) It has the same origin as $\{B\}$ (b) The Z^V – axis is parallel to Z^L – axis (c) X^V, Y^V – axes are projections of X^B, Y^B onto a plane parallel to the $X^L Y^L$ plane in $\{E\}$ and passing through the origin of $\{V\}$ as shown in Fig. 3.2.

The angles θ_p, θ_r and θ_y correspond to pitch, roll and yaw of the quadrotor respectively as shown in Fig. 3.2. The $X^V Y^V$ plane is constrained to always remain parallel to the $X^L Y^L$ plane. However, it can change its orientation by some θ_y about Z^V . This change in orientation between $\{V\}$ and $\{L\}$ is given by a rotation matrix

$$\mathbf{R}_V^L = \begin{bmatrix} c\theta_y & s\theta_y & 0 \\ -s\theta_y & c\theta_y & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad (3.1)$$

where $c(\cdot) := \cos(\cdot)$ and $s(\cdot) := \sin(\cdot)$.

3.2 Quadrotor dynamics

The six degrees of freedom of motion in a quadrotor are achieved by varying the rotor speeds, $\bar{\omega}_i, i = 1, 2, 3, 4$ of the four motors. The thrust F_i generated by each motor along the Z^B – axis is given by

$$F_i = b\bar{\omega}_i^2, \quad (3.2)$$

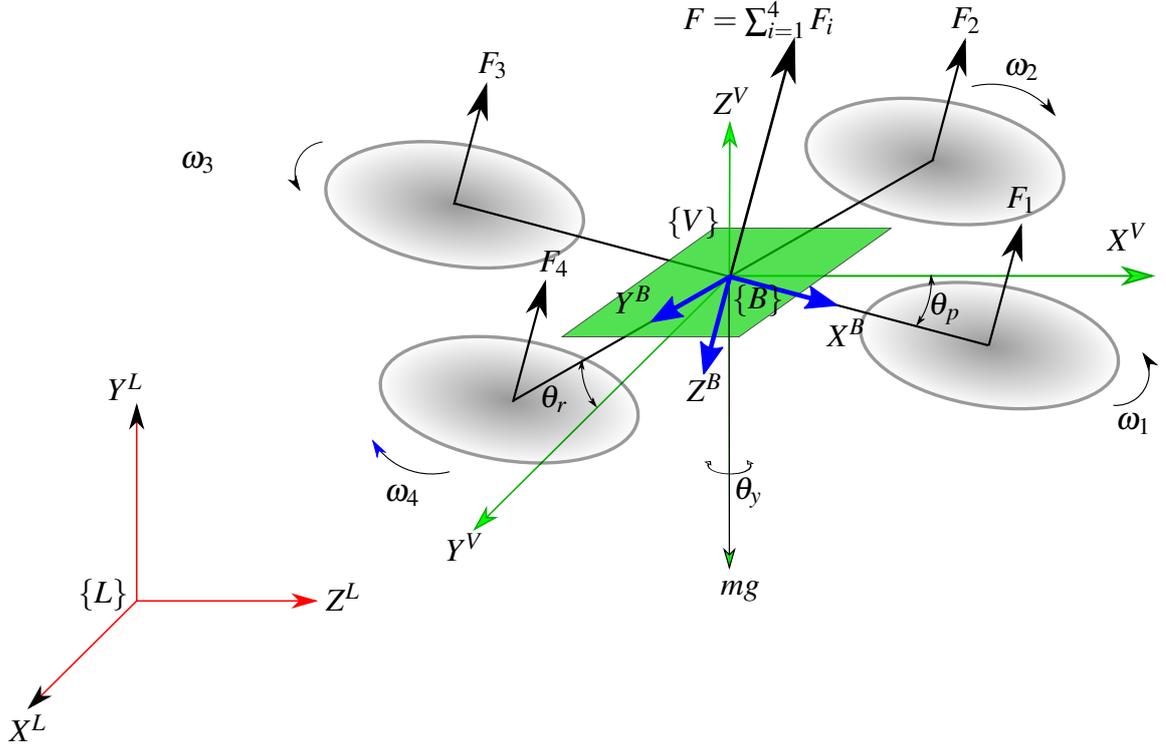


Figure 3.2: The quadrotor system setup along with the $\{L\}$, $\{B\}$ and $\{V\}$ frames of reference

where $b > 0$ is the lift constant. It is known [Cor11] that b depends on the air density, the cube of the rotor blade radius and the chord length of the blade.

In the Earth-fixed frame $\{E\}$, let $\bar{\mathbf{p}}^E = \begin{bmatrix} x^E & y^E & h^E \end{bmatrix}^T \in \mathbb{R}^3$ denote the position of the quadrotor. Then, the translational dynamics of the quadrotor are given by Newton's second law [Cor11] to be

$$m\dot{\bar{\mathbf{v}}}^E = \begin{bmatrix} 0 \\ 0 \\ mg \end{bmatrix} - \mathbf{R}_B^E \begin{bmatrix} 0 \\ 0 \\ F \end{bmatrix} \quad (3.3)$$

where $\bar{\mathbf{v}}^E = \begin{bmatrix} \dot{x}^E & \dot{y}^E & \dot{h}^E \end{bmatrix}^T \in \mathbb{R}^3$ is the velocity of the vehicle in the Earth frame, g is the acceleration due to gravity, m is the total mass of the quadrotor and $F = \sum_{i=1}^4 F_i$ is the combined upward thrust generated by the four motors. The first term in (3.3) refers to the force of gravity which always acts downwards in $\{E\}$ and the second term corresponds to the combined thrust in the body frame $\{B\}$ rotated into $\{E\}$. This rotation is given by the matrix

$$\mathbf{R}_B^E = \begin{bmatrix} c\theta_p c\theta_y & -s\theta_r s\theta_p c\theta_y + c\theta_r s\theta_y & c\theta_r s\theta_p c\theta_y + s\theta_r s\theta_y \\ -c\theta_p s\theta_y & s\theta_r s\theta_p s\theta_y + c\theta_r c\theta_y & -c\theta_r s\theta_p s\theta_y + s\theta_r c\theta_y \\ -s\theta_p & -s\theta_r c\theta_p & c\theta_r c\theta_p \end{bmatrix}.$$

The rotational motion of the quadrotor is achieved by generating a pairwise difference in the rotor thrusts. The rotational acceleration of the quadrotor is given [Cor11] by Euler's equation

of motion

$$\mathbf{J}\dot{\boldsymbol{\omega}} = -\boldsymbol{\omega} \times \mathbf{J}\boldsymbol{\omega} + \boldsymbol{\Gamma}, \quad (3.4)$$

where \mathbf{J} is the 3×3 inertia matrix, $\boldsymbol{\omega}$ is the body angular velocity vector and $\boldsymbol{\Gamma} = [\tau_x \ \tau_y \ \tau_z]^T$ is the torque applied to the quadrotor. The following equation relating T , $\boldsymbol{\Gamma}$ and $\bar{\omega}_i, i = 1, \dots, 4$ is derived in [Cor11]:

$$\begin{bmatrix} T \\ \boldsymbol{\Gamma} \end{bmatrix} = \begin{bmatrix} -b & -b & -b & -b \\ 0 & -db & 0 & db \\ db & 0 & -db & 0 \\ k & -k & k & -k \end{bmatrix} \begin{bmatrix} \bar{\omega}_1^2 \\ \bar{\omega}_2^2 \\ \bar{\omega}_3^2 \\ \bar{\omega}_4^2 \end{bmatrix} =: \mathbf{Q} \begin{bmatrix} \bar{\omega}_1^2 \\ \bar{\omega}_2^2 \\ \bar{\omega}_3^2 \\ \bar{\omega}_4^2 \end{bmatrix}, \quad (3.5)$$

where k is the drag constant which depends on the same factors as b , and d is the distance of the motor from the center of gravity of the quadrotor. The matrix \mathbf{Q} is of full rank if $b, k, d > 0$ and thus can be inverted to get the rotor speeds required to generate specific forces and torques as follows

$$\begin{bmatrix} \bar{\omega}_1^2 \\ \bar{\omega}_2^2 \\ \bar{\omega}_3^2 \\ \bar{\omega}_4^2 \end{bmatrix} = \mathbf{Q}^{-1} \begin{bmatrix} T \\ \boldsymbol{\Gamma} \end{bmatrix} \quad (3.6)$$

The overall motion of the quadrotor can be obtained by integrating the dynamics (3.3) and (3.4) where the forces and torques acting on the quadrotor are functions of rotor speeds.

3.3 Basic quadrotor control

In our experiments, a nested control structure is employed to control the quadrotor. A block diagram of the control architecture is shown in Figure 3.3. The inner control loops are used to control the attitude (roll, pitch and yaw angles) and altitude of the quadrotor. The outer control loops are used to control the position, and subsequently the motion of the quadrotor in the $X^L Y^L$ plane. Various methods can be used for motion control of the quadrotor [Qual17]. In particular, we shall look at two methods:

1. Proportional-derivative (PD) controller
2. Time optimal controller

Let us now see the control loops in detail.

3.3.1 Inner control loops

Attitude control

Roll and pitch control: Let $\boldsymbol{\Theta}^* = [\theta_p^* \ \theta_r^*]^T \in \mathbb{R}^2$ be the desired value of roll and pitch angles and $\boldsymbol{\Theta} = [\theta_p \ \theta_r]^T \in \mathbb{R}^2$ be measured roll and pitch angles. In order to attain $\boldsymbol{\Theta}^*$, torques

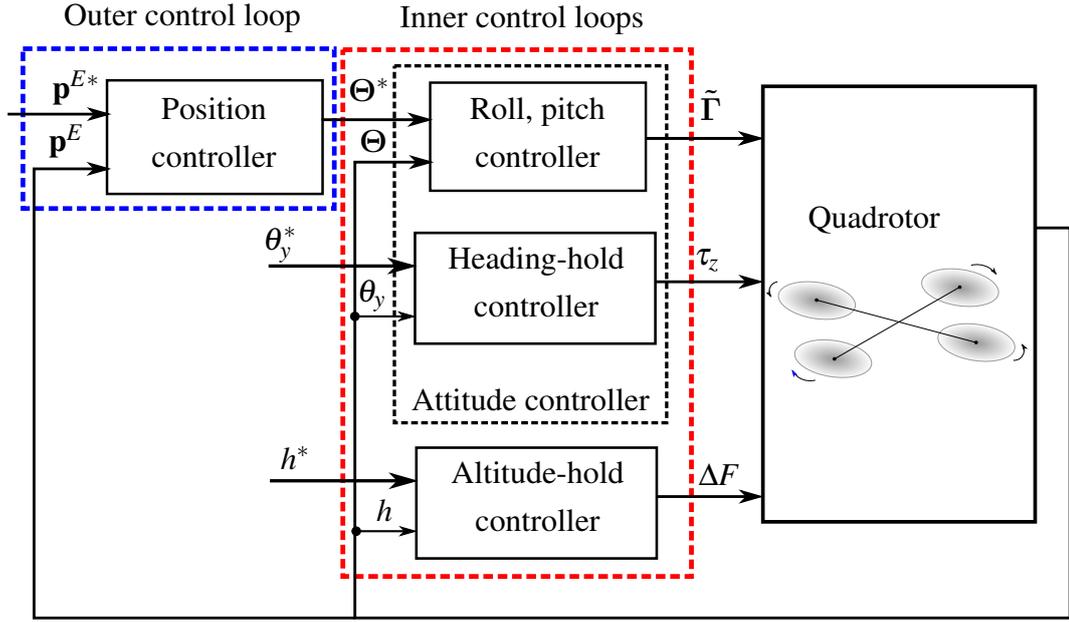


Figure 3.3: Basic control architecture for a quadrotor

$\tilde{\Gamma} = \begin{bmatrix} \tau_x & \tau_y \end{bmatrix}^T \in \mathbb{R}^2$ are generated using a proportional-derivative controller given by

$$\tilde{\Gamma} = K_{p,r,p}(\Theta^* - \Theta) + K_{d,r,p}(\dot{\Theta}^* - \dot{\Theta}), \quad (3.7)$$

where $K_{p,r,p} = \begin{bmatrix} K_{p_r} & 0 \\ 0 & K_{p_p} \end{bmatrix} \in \mathbb{R}^{2 \times 2}$ and $K_{d,r,p} = \begin{bmatrix} K_{d_r} & 0 \\ 0 & K_{d_p} \end{bmatrix} \in \mathbb{R}^{2 \times 2}$ are the control gains.

Yaw control (Heading hold): The heading of the quadrotor is the angle between X^B and X^L as seen in Fig. 3.2. This angle is represented by the yaw angle, θ_y . The heading of the quadrotor is kept constant at $\theta_y = 0^\circ$, in order to decouple the motion of the quadrotor along the roll and pitch axis. As we shall see later, this decoupling is useful in design of the cooperative control algorithms. Hence, each quadrotor maneuvers in $X^L Y^L$ plane by only changing its roll and pitch angles.

The required yaw torque is generated using the following proportional-derivative (PD) control law

$$\tau_z = K_{p_y} e_H + K_{d_y} \dot{e}_H, \quad (3.8)$$

where $e_H = (\theta_y^* - \theta_y)$ is the error, θ_y is the measured yaw angle, θ_y^* is the desired yaw angle and K_{p_y}, K_{d_y} are the control gains.

Altitude Hold

In the case of the quadrotors used in the outdoor testbed, the altitude is held constant using PD gain scheduling. Let h be the actual altitude, h^* the desired altitude and \bar{F} the thrust value corresponding to h^* . Error bands $B_k := [b_k, b_{k+1})$, where $b_k \in \mathbb{R}$ and $b_0 < \dots < b_k < \dots < b_m$ are defined. The error $e_A = (h^* - h) \in B_k$ for some k . Different set of gains $K_p^{B_k}, K_d^{B_k}$ corresponding to each error band B_k are assigned. Then the required cumulative thrust correction is generated using

$$\Delta F = K_p^{B_k} e_A + K_d^{B_k} \dot{e}_A + \omega_0, \quad (3.9)$$

where ω_0 is the rotor speed bias required to produce thrust to counter the weight of the vehicle.

3.3.2 Approximation as double integrators

Motion of the quadrotor in the $X^V Y^V$ plane is achieved by generating required rolling and pitching torques. The total force $\mathbf{f}^V = \begin{bmatrix} f_x^V & f_y^V & f_z^V \end{bmatrix}^T \in \mathbb{R}^3$ acting on the quadrotor is given by

$$\mathbf{f}^V = \mathbf{R}_x(\theta_r) \mathbf{R}_y(\theta_p) \begin{bmatrix} 0 \\ 0 \\ F \end{bmatrix}, \text{ where} \quad (3.10)$$

$$\mathbf{R}_x(\theta_r) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & c\theta_r & s\theta_r \\ 0 & -s\theta_r & c\theta_r \end{bmatrix}, \quad \mathbf{R}_y(\theta_p) = \begin{bmatrix} c\theta_p & 0 & s\theta_p \\ 0 & 1 & 0 \\ -s\theta_p & 0 & c\theta_p \end{bmatrix}.$$

For small θ_r and θ_p , (3.10) can be simplified to get

$$f_x^V = F \sin \theta_p \approx F \theta_p \quad (3.11)$$

$$f_y^V = F \sin \theta_r \cos \theta_p \approx F \theta_r. \quad (3.12)$$

In $\{E\}$, the total force acting on the quadrotor is given by

$$\mathbf{f}^E = \mathbf{R}_V^E \mathbf{f}^V. \quad (3.13)$$

Now, the dynamics of the quadrotor in the $X^E Y^E$ plane is given by

$$\dot{\mathbf{p}}^E = \mathbf{v}^E, \quad \dot{\mathbf{v}}^E = \mathbf{f}^E, \quad (3.14)$$

where \mathbf{f}^E can be controlled through (3.13). To change \mathbf{f}^E arbitrarily, θ_p and θ_r are adjusted according to (3.13). If θ_p and θ_r can be varied independently and instantaneously, then due to (3.13) and (3.14), the effective motion of the quadrotor in the $X^E Y^E$ plane can be approximated as a double integrator. The attitude control loop is designed such that $\theta_p \rightarrow \theta_p^*$ and $\theta_r \rightarrow \theta_r^*$ in milliseconds. Due to low moment of inertia, it is observed on the indoor and outdoor testbeds (see Chapter 4) that a change in roll and pitch angles is attained much faster (within 50 ms) than translational motion, which is in the range of 3 to 5 m/s.

Remark 1. The heading control loop given by (3.8) and is designed such that $\theta_y \rightarrow \theta_y^*$ within around 500 ms and is held constant on both, the indoor and outdoor testbeds. Hence, the matrix given by (3.1) can be assumed to remain constant for translational motion considerations.

Remark 2. As θ_p and θ_r of the quadrotor change, the vertical component of the total thrust vector, F reduces by a factor of the cosine of θ_p and θ_r . The angles being small, the reduction is not severe. The altitude control loop given by (3.9) is designed to be fast and corrects the experienced drop at a rate of 25 cm/s on the outdoor testbed and 50 cm/s on the indoor testbed.

Thus the quadrotor can be approximated as two independent double integrator systems. This is verified (refer Sections 5.6 and 6.3) by comparing the theoretically predicted trajectories of double integrator agents with the actual trajectories of physical quadrotors in both, indoor and outdoor environments.

3.3.3 Outer control loop: motion control

Let us now review the techniques used for quadrotor motion control using the outer control loop (see Figure 3.3).

Proportional-derivative (PD) controller

Let the measured position of the quadrotor be $\mathbf{p}^E = \begin{bmatrix} x^E & y^E \end{bmatrix}^T \in \mathbb{R}^2$ in $\{E\}$. The desired position is denoted by $\mathbf{p}^{E*} = \begin{bmatrix} x^{E*} & y^{E*} \end{bmatrix}^T$. This is achieved by varying θ_p and θ_r while θ_y is held constant by the heading control loop.

For the sake of analysis, $\{E\}$ is selected as reference to describe quadrotor motion using \mathbf{p}^E and \mathbf{v}^E . For successful motion control, \mathbf{f}^E can be varied according to the following law [Cor11]:

$$\mathbf{f}^E = mK_f[K_p(\mathbf{p}^{E*} - \mathbf{p}^E) - \mathbf{v}^E], \quad (3.15)$$

where K_f and K_p are constants. The corresponding desired values of angles, $\Theta^* = \begin{bmatrix} \theta_p^* & \theta_r^* \end{bmatrix}^T \in \mathbb{R}^2$ from the control law (3.15) navigate the quadrotor to the desired location \mathbf{p}^{E*} . Manipulating (3.11), (3.12), (3.13) and (3.15), the expression

$$\Theta^* = \mathbf{R}_E^V \frac{mK_f}{F} [K_p(\mathbf{p}^{E*} - \mathbf{p}^E) - \mathbf{v}^E]. \quad (3.16)$$

is obtained. In order to attain Θ^* , torques $\tilde{\Gamma} = \begin{bmatrix} \tau_x & \tau_y \end{bmatrix}^T \in \mathbb{R}^2$ are generated using a proportional-derivative controller given by (3.7).

Time optimal controller

A time optimal controller is used when the objective is to transfer a system from an arbitrary initial state to a target set in *minimum* time [Kir12]. Let the measured position of the quadrotor be $\mathbf{p}^L = \begin{bmatrix} x^L & y^L \end{bmatrix}^T \in \mathbb{R}^2$ in $\{L\}$. The desired position is denoted by $\mathbf{p}^{L*} = \begin{bmatrix} x^{L*} & y^{L*} \end{bmatrix}^T$ (In

this case, we choose $\{L\}$ as reference to describe quadrotor motion). Then, the time optimal controller is designed to drive the quadrotor from \mathbf{p}^L to \mathbf{p}^{L*} in minimum time. We now look at the design and implementation of a time optimal controller for a quadrotor constrained to small angle rotations.

It is shown in Section 3.3.2 that the quadrotor dynamics can be approximated as two independent double integrator systems. Hence, for simplicity, we present the implementation of the time optimal controller only for motion along one direction, say X^L . Similar calculations are valid for implementation along Y^L .

For small values of θ_p , using (3.11), (3.13) and the fact that θ_y is held constant by the heading hold loop, we have

$$\ddot{x}^L = k_{\theta,x}\theta_p \quad (3.17)$$

where $k_{\theta,x}$ is assumed to be constant. Let the double integrator system along the X^L direction be denoted by

$$\dot{\mathbf{x}}^L(t) = \mathbf{A}\mathbf{x}^L(t) + \mathbf{b}u(t) \quad (3.18)$$

where

$$\mathbf{x}^L = \begin{bmatrix} x^L \\ \dot{x}^L \end{bmatrix}, \mathbf{A} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}, \mathbf{b} = \begin{bmatrix} 0 \\ 1 \end{bmatrix}.$$

Then using 3.17 and 3.18, the control input can be written as $u = \dot{x}^L = k_{\theta,x}\theta_p$. Let us assume that $|\theta_p| \leq \beta$, where β is the maximum rotation angle for which the approximation holds. Then the switching curve governing the time optimal control law that drives system (3.18) from $\mathbf{x}^L(0)$ to origin in minimum time can be found out [Kir12] to be

$$s(\mathbf{x}^L) := x^L + \frac{\dot{x}^L|\dot{x}^L|}{2k_{\theta,x}\beta} \quad (3.19)$$

Now, the time optimal control input u^* can be calculated using

$$u^*(\mathbf{x}^L) := \begin{cases} -\beta \text{sign}(\dot{x}^L) & \text{if } s(\mathbf{x}^L) = 0 \\ -\beta \text{sign}(s(\mathbf{x}^L)) & \text{if } s(\mathbf{x}^L) \neq 0 \end{cases} \quad (3.20)$$

Practical implementation: To implement the time optimal control law, it is essential to ascertain the value of $k_{\theta,x}$ in order to correctly determine the switching curve. The value of β is known a-priori. An estimate of the value of $k_{\theta,x}$ can be obtained by applying maximum input $u = \pm\beta$ and observing/computing the acceleration along the X^L axis.

However, due to differentiation errors or imperfect sensors, there may be situations wherein velocity and acceleration measurements are noisy. Practically, the value of F does not remain exactly constant. Thus, the following empirical method is proposed to calculate the value of this constant. This method is applied to tune the value of $k_{\theta,x}$ and $k_{\theta,y}$ along the X^L and Y^L directions respectively.

Tuning method¹:

1. A grid is set up along the X^L and Y^L axis as shown in Figure 3.4 with each block length equal to one unit distance.
2. Initialize the quadrotor at point A_1 such that X^V and Y^V is aligned with X^L and Y^L respectively. Run the time optimal control law (3.20) with the estimated value of $k_{\theta,y}$ plugged in to (3.19). Ideally, the quadrotor should reach the origin using one switch from $u = -\beta$ to $u = +\beta$ [Kir12]. If the quadrotor reaches the origin in more than one switch, adjust the value of $k_{\theta,y}$ till the quadrotor requires one switch to reach origin.
3. Initialize the quadrotor at point A_5 such that X^V and Y^V is aligned with X^L and Y^L respectively. Run the time optimal control law (3.20) with the updated estimate of $k_{\theta,y}$. Ideally, the quadrotor should reach the origin using one switch from $u = +\beta$ to $u = -\beta$. If the quadrotor reaches the origin in more than one switch, adjust the value of $k_{\theta,y}$ till the quadrotor requires one switch to reach origin. Again with this updated value of $k_{\theta,y}$, verify whether the time optimal control law 3.20 drives the quadrotor from point A_1 to origin in one switch.
4. Repeat steps 2 and 3 at points A_3 and A_7 to tune the value of $k_{\theta,x}$.
5. Verify whether the time optimal control law (3.20) drives the quadrotor from points A_2, A_4, A_6 and A_8 using the updated values of $k_{\theta,x}$ and $k_{\theta,y}$.
6. To check repeatability, repeat step 5 at points B_1, \dots, B_8 .

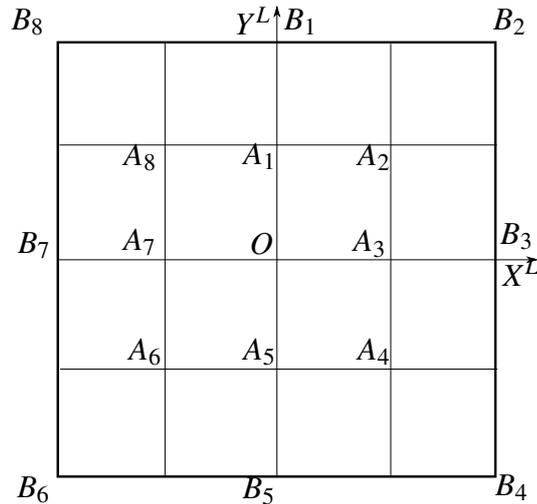


Figure 3.4: Setup to tune constants of switching surface for implementation of time optimal control

¹The code for implementation of the time optimal controller for a quadrotor with approximated dynamics is available at https://github.com/zapurva/crazygames_ws/blob/master/src/crazyflie_ros/crazyflie_controller/src/time_optimal_controller.cpp.

Chapter 4

Testbed details

One of the aims of this thesis is to tackle the engineering challenges in implementation of cooperative control on physical systems. Hence, indoor and outdoor testbeds are created in order to implement the cooperative control laws. This chapter provides details regarding the system architecture and integration required to setup these testbeds. These specifications provide a performance benchmark for other experimental work done in this domain. Details of the indoor testbed are presented in Section 4.1 and the outdoor testbed in Section 4.2.

4.1 Indoor testbed

An indoor testbed is created for conducting the consensus tracking experiments. The testbed on which the experiments are performed comprises of three main systems: (a) The quadrotor vehicles (b) The localization system (c) The host server. A picture of the setup is shown in Figure 4.1.

4.1.1 Vehicle details

The Crazyflie 2.0 quadrotor [cra18], an open-source hardware and software platform, is used for conducting the experiments. The vehicle weighs 27 g with battery and has dimensions of 92x92x29 mm (LxBxH). It has a 32-bit 168 MHz ARM micro-controller onboard. It communicates with the host server using a 2.4 GHz USB radio (called Crazyradio PA) using 32 byte packets with transmission speeds of up to 2 Mbps. With a payload of motion-capture markers and 3.7 V, 300 mAh batteries, the quadrotor has an approximate flying time of 7 minutes. Four such quadrotors are used in the experimental testbed.

4.1.2 Localization platform details

A motion-capture system from Optitrack is used for localization of the quadrotors. The system comprises of 16 cameras [fle18] with infrared sensors which detect the markers fixed atop the quadrotors. A proprietary software (Motive, by Optitrack) uses data in the form of images

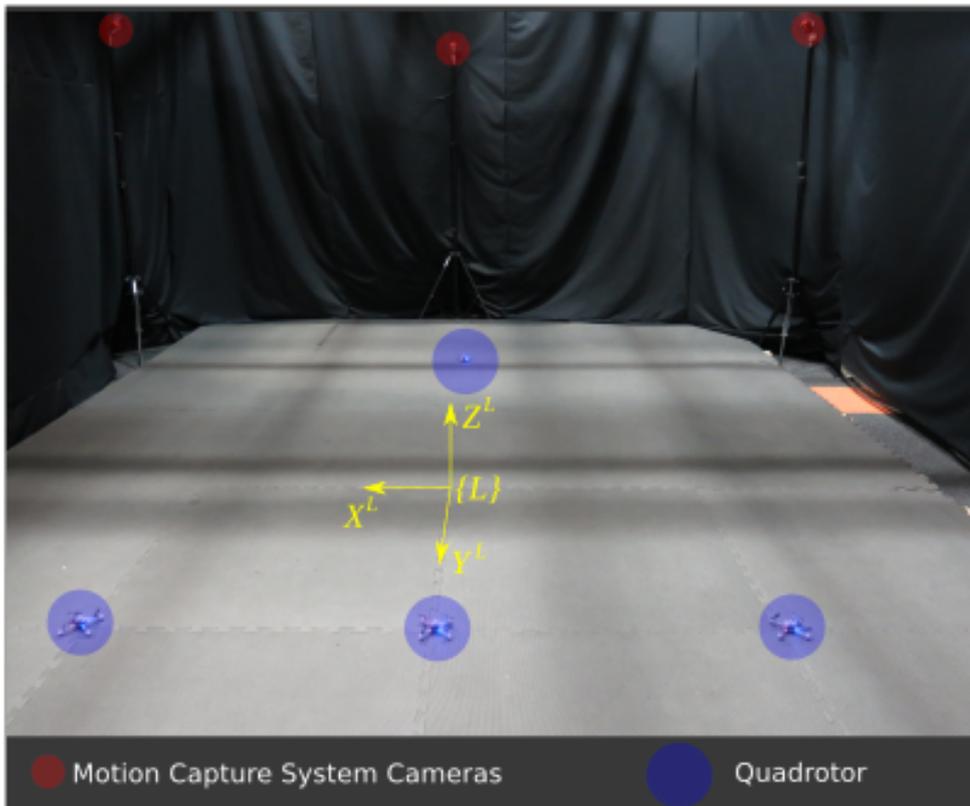


Figure 4.1: A photo of the indoor flying arena

captured by the cameras, performs localization computations and provides position data for all the quadrotors within the workspace. This data is streamed over the local network using the Virtual-Reality Peripheral Network (VRPN) protocol [vrp18]. The localization system provides millimeter-level precision at high frequencies of up to 200 Hz.

4.1.3 The host server

A host server running Robot Operating System (ROS) is used as a common interface. It subscribes to the position data of the quadrotors made available over the local network using the VRPN protocol. The high level control algorithms are executed on the server and the desired setpoint values of roll, pitch and yaw angles are sent to each quadrotor using ROS [HA17]. The on-board flight controller then takes control action based on these received setpoint values.

The overall system architecture is as shown in Figure 4.2.

4.2 Outdoor testbed

An outdoor testbed is created for conducting the consensus experiments. The testbed comprises of three main systems: (a) The physical quadrotor unit (b) A virtual unit (c) A communication module. Let us now look at these systems in detail.

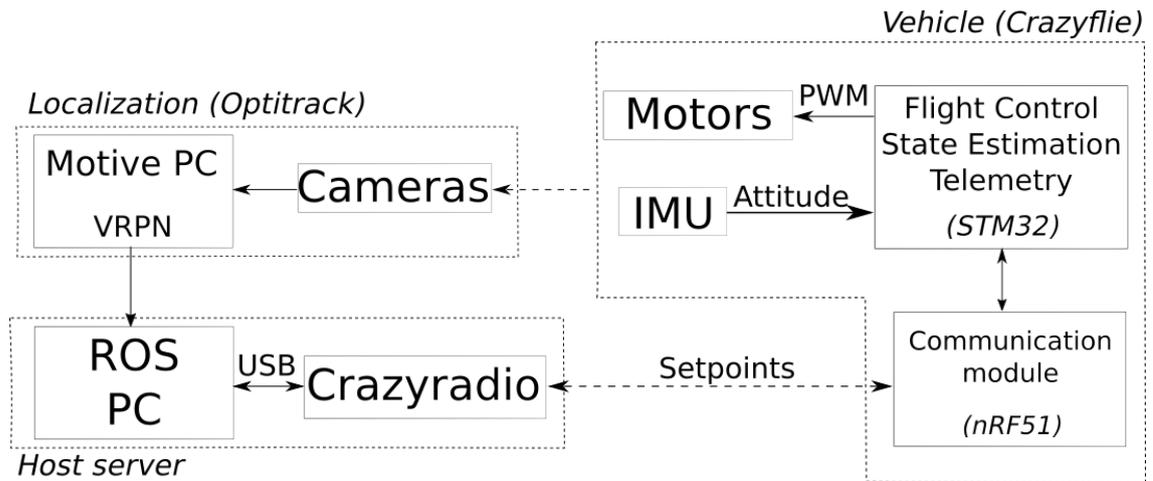


Figure 4.2: Overall system architecture for the consensus tracking experiments

4.2.1 The physical unit

The physical quadrotors are constructed in-house using commercial off the shelf components as seen in Fig. 4.4. The system architecture is as shown in Fig. 4.3. Microcontroller-1 is the ATmega 2560 based APM 2.6 board [apm15] which is used for flight control. Microcontroller-2 is an ATmega 2560 microcontroller which is used to orchestrate the communication between the multiple quadrotors. A u-blox LEA-6H GPS receiver provides localization information of the quadrotors in terms of latitudes and longitudes.

Each quadrotor is capable of manual and autonomous flight. The auto-pilot running on-board Microcontroller-1 is a modified version of the AeroQuad [aer15] open-source autopilot.

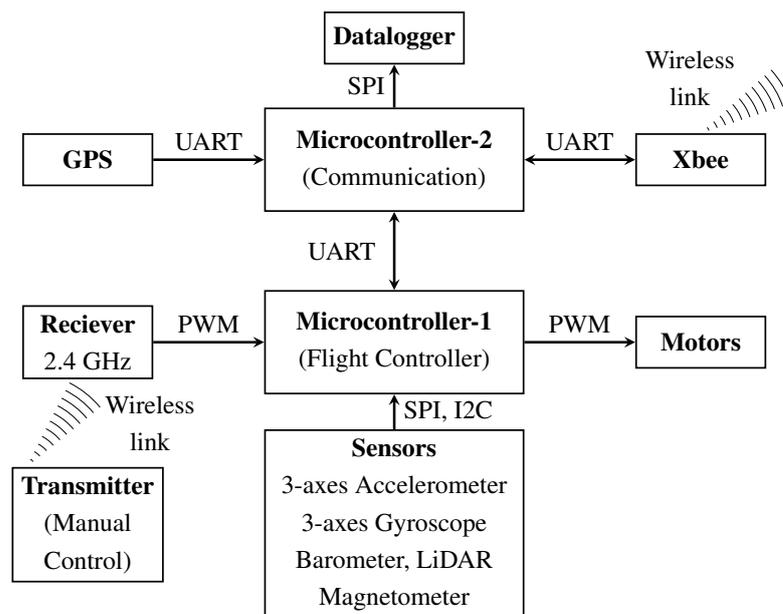


Figure 4.3: A block diagram of the physical quadrotor unit

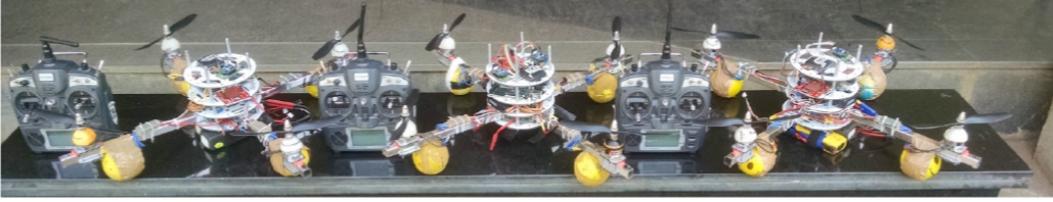


Figure 4.4: A team of quadrotors constructed at IIT Bombay

4.2.2 The virtual unit

Due to unavailability of resources to physically construct more quadrotors, three real-time virtual quadrotor emulators are created in addition to the physical units. Each emulator consists of a laptop computer running ROS [ROS17] and Gazebo [Gaz17] platforms, used to simulate the dynamics of a single quadrotor described in Section 3.2. A pre-existing stack [MSK⁺12] which provides ROS packages related to modeling, control and simulation of a quadrotor, is modified to generate real-time quadrotor trajectories. A communication module with the same hardware as on-board the physical unit, is interfaced with the laptop computer as shown in Fig 4.5.

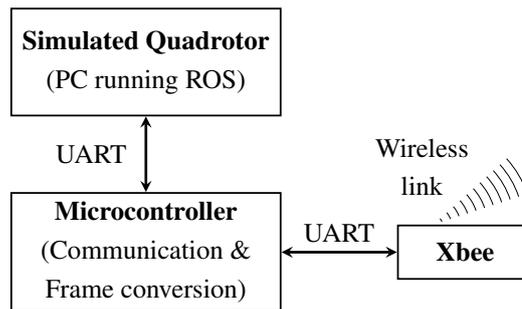


Figure 4.5: A block diagram of the virtual quadrotor unit

4.2.3 The communication module

The communication module facilitates the information exchange between all the agents, physical and virtual. The hardware comprises of a Xbee modem which is interfaced with an ATmega 2560 microcontroller as shown in Fig. 4.3 and 4.5. The microcontroller runs the communication protocol which is described in Section 6.2 and thus a wireless link is set up between the agents.

4.2.4 Combining the physical and virtual units

The motions of the physical and virtual units are inherently described in different coordinate systems. The physical units use geodetic coordinates (latitude, φ and longitude, λ) obtained from the GPS for localization and are usually converted to the navigation frame $\{L\}$. The motion of the virtual units in the simulator is described in a virtual reference frame, say $\{S\}$.

To facilitate interaction between the physical and virtual units, equirectangular projections and the WGS84 system [Cly02] are used for transformation between these two coordinate systems, $\{L\}$ and $\{S\}$. A derivation of this transformation is given in below.

Transformation between Local-level frame ($\{L\}$) and virtual reference frame ($\{S\}$)

The fact that over very small distances, the latitude and longitude can be assumed to vary linearly in the $\{L\}$ frame [Cly02] is used.

Let the coordinates of a quadrotor be $\begin{bmatrix} x^L(t) & y^L(t) \end{bmatrix}^T$ in the $\{L\}$ frame and $\begin{bmatrix} x^S(t) & y^S(t) \end{bmatrix}^T$ in the $\{S\}$ frame. The reference latitude and longitude, $\begin{bmatrix} \varphi^* & \lambda^* \end{bmatrix}^T$ are set to be the actual GPS coordinates of a point in the outdoor arena where the experiments are being carried out. Then, $\begin{bmatrix} x^S(t) & y^S(t) \end{bmatrix}^T = (0,0)$ at $\begin{bmatrix} x^L(t) & y^L(t) \end{bmatrix}^T = \begin{bmatrix} \varphi^* & \lambda^* \end{bmatrix}^T$.

The following constants as per the WGS84 system [Cly02] are used for the transformation

- Equatorial Radius, $R_{eq} = 6378137.00$ m
- Polar Radius, $R_p = 6356752.31$ m
- Flattening, $f = \frac{R_{eq} - R_p}{R_{eq}} = \frac{1}{298.26}$
- Eccentricity, e is defined as $e^2 = 1 - \frac{R_p^2}{R_{eq}^2} = 2f - f^2$
- Radius of Curvature in Prime Vertical, $R_{PV} = R_N = R_{eq} \frac{1}{\sqrt{1 - e^2 \sin^2(\varphi^*)}}$
- Radius of Curvature in Meridian $R_M = \frac{R_{PV}(1 - e^2)}{1 - e^2 \sin^2(\varphi^*)}$

Conversion from $\{L\}$ to $\{S\}$ frame

The $\begin{bmatrix} x^S(t) & y^S(t) \end{bmatrix}^T$ coordinates in $\{S\}$ can be calculated using the following two equations:

$$x^S = R_M \times (x^L - \varphi^*) \times \frac{\pi}{180} \quad (4.1)$$

$$y^S = R_{PV} \cos(\varphi^*) \times (y^L - \lambda^*) \times \frac{\pi}{180} \quad (4.2)$$

Conversion from $\{S\}$ to $\{E\}$ frame

The above two equations can be rearranged to give the GPS coordinates in $\{E\}$ frame as follows:

$$x^L = \varphi^* + \frac{x^S}{R_M} \times \frac{180}{\pi} \quad (4.3)$$

$$y^L = \lambda^* - \frac{y^S}{R_{PV} \cos(\varphi^*)} \times \frac{180}{\pi} \quad (4.4)$$

Chapter 5

Consensus tracking

In this chapter we present the consensus tracking experiments performed on a multi-quadrotor testbed in an indoor environment. We start by reviewing the local information based feedback control strategy proposed in [MC18] to achieve min-max time consensus tracking of double integrator agents with bounded inputs. This is required for completeness, since we will be basing our experiments on these theoretical results. We begin by defining the notation used to model the interaction between the agents in Section 5.1. Then, the multi-agent system under consideration is introduced in Section 5.2. The problem of min-max time consensus tracking is formulated in Section 5.4. In Section 5.5, the solution of the consensus tracking problem is presented. Results of various experiments on implementation of the consensus tracking law on a testbed of four quadrotors are presented in Section 5.6.

5.1 Graphs

An undirected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ is a finite set of nodes \mathcal{V} connected by a set of edges $\mathcal{E} \subseteq (\mathcal{V} \times \mathcal{V})$. If two nodes $a_i, a_j \in \mathcal{V}$ are connected to each other, then the graph \mathcal{G} has an edge $(a_i, a_j) \in \mathcal{E}$ between nodes a_i and a_j . A graph $\mathcal{G}_s = (\mathcal{V}_s, \mathcal{E}_s)$ is a subgraph of \mathcal{G} if $\mathcal{V}_s \subseteq \mathcal{V}$ and $\mathcal{E}_s \subseteq \mathcal{E}$. A sequence of edges $(a_{i_k}, a_{j_k}) \in \mathcal{E}$ where $k = 1, \dots, r$ such that $a_{i_{k+1}} = a_{j_k}$, with $a_{i_k} \neq a_{i_l}$ for $k \neq l$ is called a path of length r . If for a path, $a_{j_r} = a_{i_1}$, then it is called a cycle. If there exists a path between any two nodes, then the graph is said to be connected. A connected graph without cycles is called a spanning tree, denoted by $\mathcal{G}_{tree} = (\mathcal{V}_{tree}, \mathcal{E}_{tree})$. In an undirected graph, if $(a_i, a_j) \in \mathcal{E}$, then $(a_j, a_i) \in \mathcal{E}$. In a directed graph (or digraph) $\vec{\mathcal{G}} = (\mathcal{V}, \vec{\mathcal{E}})$, the edges are assigned a direction, i.e. $(a_i, a_j) \in \vec{\mathcal{E}} \not\Rightarrow (a_j, a_i) \in \vec{\mathcal{E}}$. A directed path and a directed cycle are defined analogously to the undirected case. If there exists a root node from which there is a directed path to every other node in the digraph, then the digraph is called a rooted directed tree. If there exists a rooted directed tree $\vec{\mathcal{G}}_{tree} = (\mathcal{V}_{tree}, \vec{\mathcal{E}}_{tree})$ such that $\mathcal{V}_{tree} = \mathcal{V}$ and $\vec{\mathcal{E}}_{tree} \subseteq \vec{\mathcal{E}}$, then the digraph $\vec{\mathcal{G}} = (\mathcal{V}, \vec{\mathcal{E}})$ is said to contain a directed spanning tree.

5.2 System description

Consider a multi-agent system (MAS) of $N + 1$ agents which are labeled as a_0, a_1, \dots, a_N and have identical double-integrator dynamics

$$\dot{\mathbf{x}}_i(t) = \mathbf{A}\mathbf{x}_i(t) + \mathbf{b}u_i(t) \quad i = 0, 1, \dots, N \quad (5.1)$$

where

$$\mathbf{x}_i = \begin{bmatrix} x_i \\ \dot{x}_i \end{bmatrix}, \mathbf{A} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}, \mathbf{b} = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

The control input u_i of agent a_i belongs to the set

$$U(\beta_i) := \{u \in \mathcal{U} \mid |u(t)| \leq \beta_i, \forall t \geq 0\}$$

where \mathcal{U} denotes the set of measurable functions from \mathbb{R} to \mathbb{R} . The agent a_0 , also referred to as the *leader*, generates the reference trajectory by using an admissible input $u_0 \in U(\beta_0)$. The agents a_1, \dots, a_N are referred to as the *followers*. Notice that the state $\mathbf{x}_i(t)$ is a function of time t , initial condition $\mathbf{x}_i(0)$ and the control input u_i . Thus, the notations $\mathbf{x}_i(t)$ and $\mathbf{x}_i(t, \mathbf{x}_i(0), u_i)$ are used interchangeably depending on the context.

5.3 Communication graph

Let $\mathcal{G}(t) = (\mathcal{V}(t), \mathcal{E}(t))$ denote the communication graph of MAS (5.1) at instant t . It is assumed that agents a_i and a_j can communicate with each other at time t , only if the distance between them is less than some threshold r , called the *communication radius*, i.e., $(a_i, a_j) \in \mathcal{E}(t)$ only if $|x_i(t) - x_j(t)| \leq r$. Further, we assume that the initial graph $\mathcal{G}(0)$ contains at least one directed spanning tree rooted at a_0 .

In [MC18], a decentralized, run-time algorithm is proposed to extract a rooted directed spanning tree from the connected graph. By applying this algorithm on $\mathcal{G}(0)$, a directed spanning tree \mathcal{G}_{tree} rooted at a_0 is obtained. Then, the control inputs of the non-leader agents to ensure that the graph \mathcal{G}_{tree} is preserved, i.e., $\mathcal{G}_{tree} \subseteq \mathcal{G}(t), \forall t \geq 0$, are designed.

Define the set $P = \{0, 1, \dots, N\}$ and a map $p : P \setminus \{0\} \rightarrow P$ such that $a_{p(i)}$ denotes the parent agent of a_i under the communication tree \mathcal{G}_{tree} . Notice that there is a communication link between agent a_i and $a_{p(i)}$ over which $a_{p(i)}$ can transmit its state information to a_i . Using this information, agent a_i can track the state trajectory of $a_{p(i)}$.

5.4 Problem formulation

Let $\mathcal{X}(0) := [\mathbf{x}_0(0), \mathbf{x}_1(0), \dots, \mathbf{x}_N(0)]$ be an initial condition of MAS (5.1). Clearly, $\mathcal{X}(0)$ leads to a communication graph $\mathcal{G}(0)$ from which \mathcal{G}_{tree} can be extracted. However, due to the communication constraints, the follower agents do not know $\mathcal{X}(0)$ completely. This lack

of information is inculcated into the problem formulation by considering all possible initial conditions allowed by a given spanning tree \mathcal{G}_{tree} . For this purpose, the set

$$\Xi(\mathcal{G}_{tree}) := \{ \mathcal{X}(0) \in \mathbb{R}^{2(N+1)} \mid \mathcal{E}_{tree} \subseteq \mathcal{E}(0) \} \quad (5.2)$$

is defined.

Let $u_0 \in U(\beta_0)$ be the control input of agent a_0 . Define $\mathbf{z}_i := \mathbf{x}_i - \mathbf{x}_{p(i)}$. Let $\mathbf{z}_i(0) = \mathbf{x}_i(0) - \mathbf{x}_{p(i)}(0)$. Then, define $\mathbf{z}_i(t, \mathbf{z}_i(0), u_i, u_{p(i)}) := \mathbf{x}_i(t, \mathbf{x}_i(0), u_i) - \mathbf{x}_{p(i)}(t, \mathbf{x}_{p(i)}(0), u_{p(i)})$ for every $i \in P \setminus \{0\}$. Let $\mathbf{u}(t) = (u_1(t), \dots, u_N(t))$.

Definition 1. MAS (5.1) is said to have achieved consensus on the reference trajectory \mathbf{x}_0 at time \bar{t} if $\mathbf{x}_i(t) = \mathbf{x}_0(t)$, $\forall t \geq \bar{t}$, $\forall i \in P \setminus \{0\}$.

For the fixed communication tree \mathcal{G}_{tree} , the time required to achieve consensus on the reference trajectory depends on \mathcal{G}_{tree} , $\mathcal{X}(0)$, u_0 and \mathbf{u} . Define *time to consensus* $T(\mathcal{G}_{tree}, \mathcal{X}(0), u_0, \mathbf{u})$ as

$$T(\mathcal{G}_{tree}, \mathcal{X}(0), u_0, \mathbf{u}) := \inf \bar{t}, \text{ s.t. } \mathbf{x}_i(t) = \mathbf{x}_0(t), \forall t \geq \bar{t}, \forall i \in P \setminus \{0\}$$

The objective is to obtain control $\mathbf{u}^* = (u_1^*, \dots, u_N^*)$ which minimizes the time to consensus. Furthermore, it is required that each control u_i^* adheres to the communication graph \mathcal{G}_{tree} and preserves the initial graph \mathcal{G}_{tree} , i.e., $\mathcal{E}_{tree} \subseteq \mathcal{E}(t)$, $\forall t \geq 0$. Notice that agent a_i does not know the complete initial condition $\mathcal{X}(0)$. In addition, the leader agent a_0 can select any admissible input $u_0 \in U(\beta_0)$. Therefore, while minimizing $T(\mathcal{G}_{tree}, \mathcal{X}(0), u_0, \mathbf{u})$, one needs to consider the worst-case realization of $u_0 \in U(\beta_0)$ and the worst-case initial condition allowed by \mathcal{G}_{tree} , i.e. $\mathcal{X}_0 \in \Xi(\mathcal{G}_{tree})$. Such a min-max formulation of $T(\mathcal{G}_{tree}, \mathcal{X}(0), u_0, \mathbf{u})$ is given in [MC18] as follows:

Problem 2. Consider MAS (5.1) with an underlying graph \mathcal{G}_{tree} , initial condition $\mathcal{X}(0) \in \Xi(\mathcal{G}_{tree})$, and leader agent a_0 . Compute u_i^* based on states $\mathbf{x}_{p(i)}$ for all $i = 1, \dots, N$, which is the solution of the following min-max optimization:

$$\begin{aligned} \mathbf{u}^* &= \arg \min_{\substack{\mathbf{u} \in \prod U(\beta_i) \\ i \in P \setminus \{0\}}} \max_{\substack{u_0 \in U(\beta_0), \\ \mathcal{X}(0) \in \Xi(\mathcal{G}_{tree})}} T(\mathcal{G}_{tree}, \mathcal{X}(0), u_0, \mathbf{u}) \text{ s.t. } \mathcal{E}_{tree} \subseteq \mathcal{E}(t), \forall t \geq 0 \\ T^* &= \min_{\substack{\mathbf{u} \in \prod U(\beta_i) \\ i \in P \setminus \{0\}}} \max_{\substack{u_0 \in U(\beta_0), \\ \mathcal{X}(0) \in \Xi(\mathcal{G}_{tree})}} T(\mathcal{G}_{tree}, \mathcal{X}(0), u_0, \mathbf{u}) \text{ s.t. } \mathcal{E}_{tree} \subseteq \mathcal{E}(t), \forall t \geq 0 \end{aligned}$$

The optimal solution $\mathbf{u}^* = (u_1^*, \dots, u_N^*)$ of Problem 2 is obtained in [MC18]. It is also shown that the optimal control u_i^* of each non-leader agent a_i is a local feedback control and is a function of \mathbf{z}_i only. Thus, for the implementation of u_i^* , agent a_i needs the instantaneous value of $\mathbf{z}_i = \mathbf{x}_i - \mathbf{x}_{p(i)}$. However, each agent a_i cannot have the instantaneous value of \mathbf{z}_i due to finite communication and measurement rates. Instead, it is forced to use the last obtained value of \mathbf{z}_i . As a result, the actual inputs of non-leader agents are different from u_i^* 's.

Definition 3. The *sampling time* T_s of MAS (5.1) is defined as the duration between the two successive time instants at which each non-leader agent a_i receives the value of its difference state \mathbf{z}_i .

Let \mathcal{N} be the set of natural numbers. Define the set $\mathcal{T} := \{t_k \in \mathbb{R} \mid t_k = kT_s, k \in \mathcal{N}\}$. Then, each non-leader agent a_i receives the value of $\mathbf{z}_i(t_k)$ at the time instants $t_k \in \mathcal{T}$ which are referred to as the *sampling instants*. Let $u_i^{T_s}$ denote the sampled version of u_i^* corresponding to the sampling time T_s .

$$u_i^{T_s}(t) = u_i^*(\mathbf{z}_i(kT_s)), \quad \forall t \in [kT_s, (k+1)T_s), \quad \forall k \in \mathcal{N} \quad (5.3)$$

As the actual inputs of the non-leader agent a_i 's are different from the optimal inputs u_i^* 's, their actual trajectories are the deviated versions of those corresponding to u_i^* 's. Experiments have been performed to analyze the effect of the sampling time T_s on these deviations.

5.5 Solution to the min-max time consensus tracking problem

In this section, the solution of Problem 2 obtained in [MC18] is given briefly. This is required to make the presentation self-contained with respect to the experiments, based on the theory developed in [MC18], presented later. Let a_i be any non-leader agent and $a_{p(i)}$ be its parent agent. Let β_i and $\beta_{p(i)}$ be the bounds on the control inputs of a_i and $a_{p(i)}$, respectively. Recall that $\mathbf{z}_i = [z_i, \dot{z}_i]^T = \mathbf{x}_i - \mathbf{x}_{p(i)}$. Then, it follows from (5.1) that

$$\dot{\mathbf{z}}_i(t) = \mathbf{A}\mathbf{z}_i(t) + \mathbf{b}(u_i(t) - u_{p(i)}(t)) \quad (5.4)$$

where $u_i \in U(\beta_i)$ and $u_{p(i)} \in U(\beta_{p(i)})$. Define a function $s_i : \mathbb{R}^2 \rightarrow \mathbb{R}$ as

$$s_i(\mathbf{z}_i) := z_i + \frac{\dot{z}_i |z_i|}{2(\beta_i - \beta_{p(i)})} \quad (5.5)$$

Then, define a function $u_i^* : \mathbb{R}^2 \rightarrow [-\beta_i, \beta_i]$ as

$$u_i^*(\mathbf{z}_i) := \begin{cases} +\beta_i \text{sign}(z_i) & \text{if } s(\mathbf{z}_i) = 0 \\ -\beta_i \text{sign}(s(\mathbf{z}_i)) & \text{if } s(\mathbf{z}_i) \neq 0 \end{cases} \quad (5.6)$$

and then define

$$\mathbf{u}^* := (u_1^*, \dots, u_N^*) \quad (5.7)$$

Recall that r is the communication radius. It has been shown in [MC18] that the solution of Problem 2 exists if and only if

$$\beta_i > \beta_{p(i)} + \frac{\dot{z}_i^2(0)}{2(r - |z_i(0)|)}, \quad \forall i \in P \setminus \{0\} \quad (5.8)$$

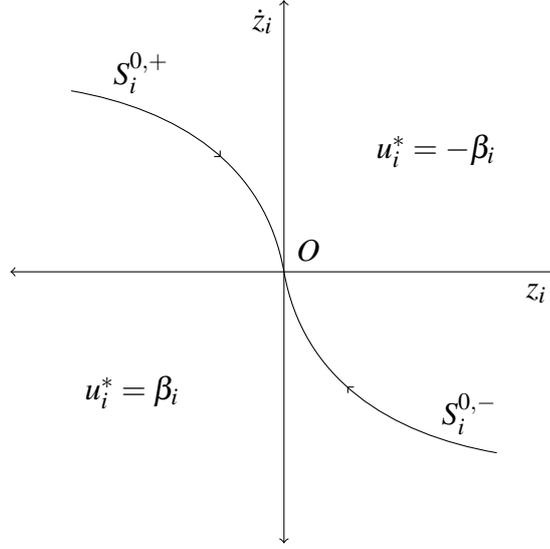


Figure 5.1: Switching curve and optimal feedback control of agent a_i

It is also shown in [MC18] that when the condition (5.8) is satisfied, then \mathbf{u}^* defined in (5.7) is the optimal solution of Problem 2, i.e., \mathbf{u}^* achieves the consensus tracking of MAS (5.1) in time T^* (defined in Problem 2). Thus, the following assumption about MAS (5.1) is made.

Assumptions 4. The control bounds β_i 's of all agents in MAS (5.1) satisfy (5.8).

Define the curve $S_i^0 := \{\mathbf{z} = [z, \dot{z}]^T \in \mathbb{R}^2 \mid s_i(\mathbf{z}) = 0\}$. Then, it follows from (5.6) that the control u_i^* switches from β_i to $-\beta_i$ (or from $-\beta_i$ to β_i) when the difference state \mathbf{z}_i crosses the curve S_i^0 . Hence, the curve S_i^0 is known as the *switching curve* of agent a_i . Define $S_i^{0,+} := \{\mathbf{z} = [z, \dot{z}]^T \in S_i^0 \mid \dot{z} \geq 0\}$ and $S_i^{0,-} := \{\mathbf{z} = [z, \dot{z}]^T \in S_i^0 \mid \dot{z} \leq 0\}$. Notice that $S_i^0 = S_i^{0,+} \cup S_i^{0,-}$. The switching curve S_i^0 and the control u_i^* of agent a_i are illustrated in Figure 5.1.

5.6 Experiments

In this section we present the implementation of the feedback control law (5.6) on the indoor quadrotor testbed in Sections 5.6.1 and 5.6.2. Details of the indoor testbed can be found in Section 4.1. The results of the consensus tracking experiments are presented in Section 5.6.

5.6.1 Motion planning in \mathbb{R}^3

The indoor setup for the motion of quadrotors in the $\{L\}$ frame is depicted in Figure 5.2. It is shown in Section 3.3.2 that for small roll and pitch angles, the motion of the quadrotor along the lateral (X^V) and longitudinal (Y^V) directions can be decoupled and approximated as independent double integrators [JLA⁺16]. Thus, the min-max time consensus tracking feedback control law in (5.6) is implemented for motion planning in both X^L and Y^L directions. An altitude-hold control loop running onboard the vehicle maintains the altitude of the quadrotor at the desired

value, independent of the motion in the X^L and Y^L directions (refer Section 3.3.1 for details). The yaw angle (θ_y) is held constant using a heading-hold control loop (refer Section 3.3.1 for details).

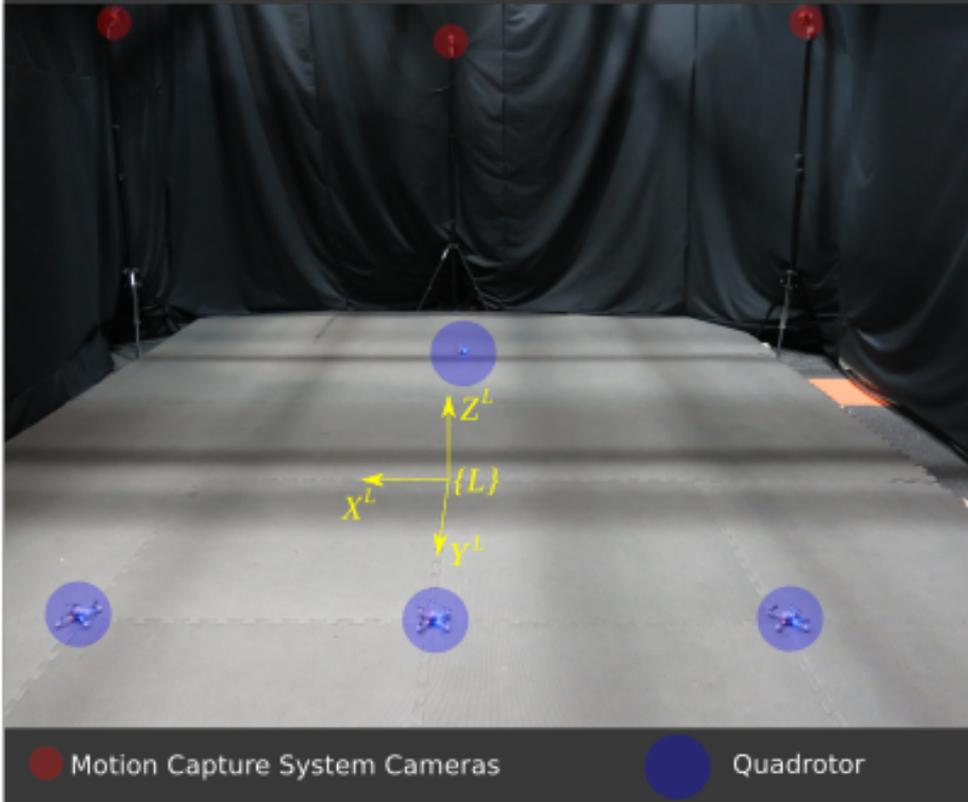


Figure 5.2: A photo of the indoor flying arena

5.6.2 Implementation details

The feedback control law presented in (5.6) achieves consensus tracking of MAS (5.1) in min-max time T^* (defined in Problem 2), i.e., $\mathbf{x}_i(t) = \mathbf{x}_0(t)$, $\forall t \geq T^*$, $\forall i \in P \setminus \{0\}$. Under this control law, each non-leader agent a_i tracks the trajectory of its parent agent $a_{p(i)}$. This eventually leads to consensus tracking on the reference trajectory generated by the leader agent a_0 .

If agent a_i attempts to maintain $\mathbf{x}_i = \mathbf{x}_{p(i)}$ using the control u_i^* (defined in (5.6)) or u_i^{Ts} (defined in (5.3)), then it follows from (5.6) and (5.3) that there will be frequent switching in the control input u_i between $+\beta_i$ and $-\beta_i$. It is well-known that such frequent switching is harmful to the system [Lib03]. In addition, achieving $\mathbf{x}_i = \mathbf{x}_{p(i)}$ would cause collisions between agents. Hence, in the experiments, each non-leader agent a_i implements the control u_i^{Ts} only if $|x_i - x_{p(i)}| > d_p$, where $d_p \in \mathbb{R}$ is a pre-specified safe distance. After that, agent a_i employs a PID controller to ensure $|x_i - x_{p(i)}| \leq d_p$ and that the difference in velocity between the parent and child agent remains bounded. Figure 5.3 shows the region (green) in which the PID controller is active.

5.6.3 Experiment 1: One leader and one follower

This experiment is performed with two quadrotors: one leader a_0 and one follower a_i . The difference trajectory \mathbf{z}_i starting at point A is shown in Figure 5.3. The red curve represents the segment $S_i^{0,-}$ of the switching curve S_i^0 of agent a_i (defined in Section 5.5). If agents a_0 and a_i had ideal double integrator dynamics, then under control $u_i = -\beta_i$ (according to (5.6)), the difference trajectory would have followed the curve AB . Then, at $\mathbf{z}_i(t) = B$, the control u_i would have switched to $u_i = +\beta_i$ (according to (5.6)) and the difference trajectory would have followed the curve $S_i^{0,-}$, till it enters inside the PID controller region shown in green. However, the quadrotor dynamics is an approximation of a double integrator. Thus, under the control $u_i = -\beta_i$, the difference trajectory follows the curve AB' . Then at point $\mathbf{z}_i(t) = B'$, the control u_i switches to $u_i = +\beta_i$. This causes the pitch angle of the quadrotor agent a_i to change from $\theta_p = -\beta_i$ to $\theta_p = +\beta_i$ quickly. Thus, there is a sudden change in the thrust vector F which results in the observed overshoot at point B' . After that the difference trajectory moves parallel to the curve $S_i^{0,-}$, and eventually enters inside the PID controller region. This shows that the control strategy (5.6), proposed in [MC18] for double integrator agents, achieves consensus tracking on a pair of quadrotors constrained to small angle rotations. Figure 5.4 is a plot of difference of position, velocity and control input of the follower quadrotor a_i with $a_{p(i)}$.

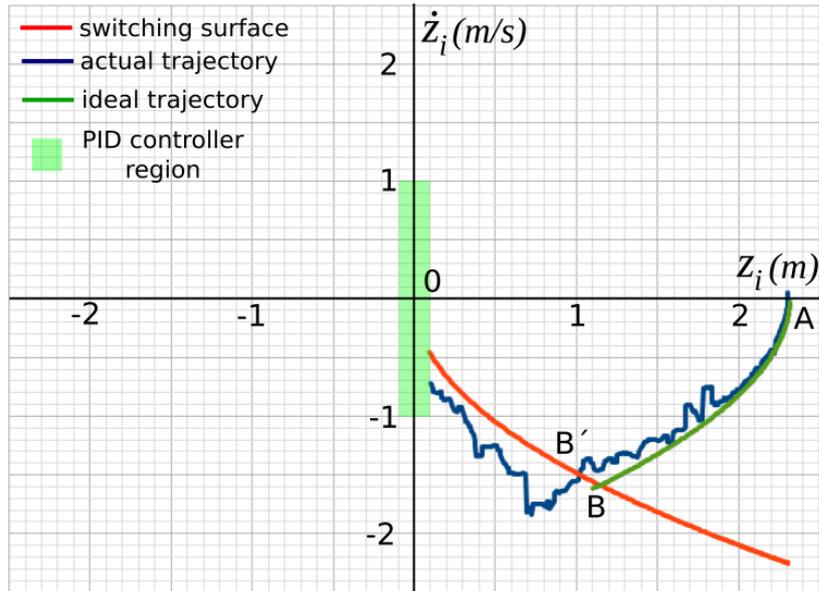


Figure 5.3: Comparison between the difference trajectories of the quadrotor and ideal double integrator leader-follower pair

5.6.4 Experiment 2: Multi-agent consensus tracking

This experiment is performed with four quadrotors: one leader a_0 and three followers a_1, a_2, a_3 which communicate over the graphs shown in Figure 5.5.

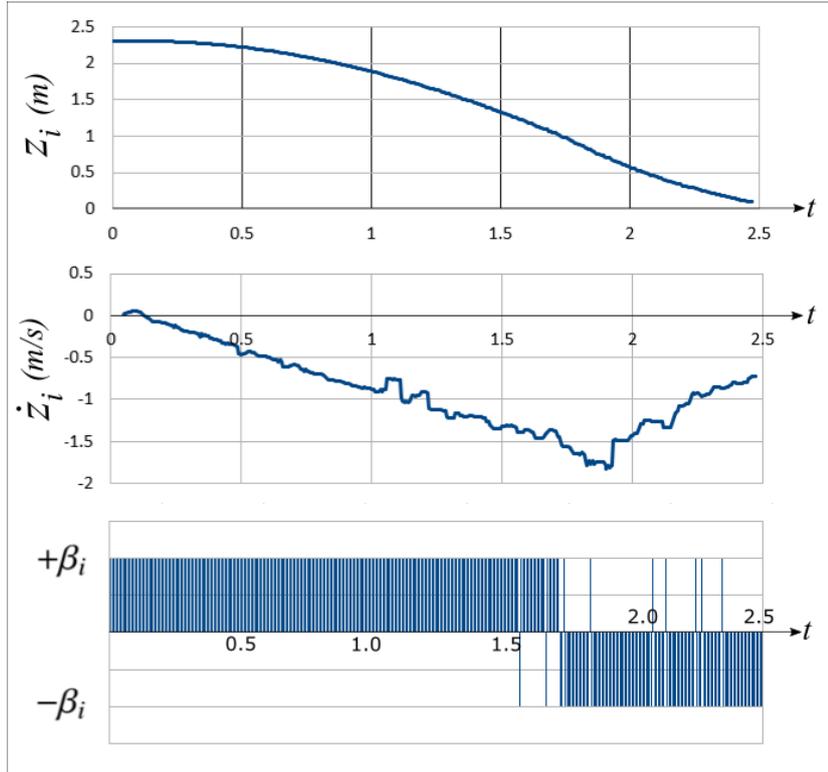


Figure 5.4: Plot of the position, velocity and control input to the follower quadrotor in the two agent case

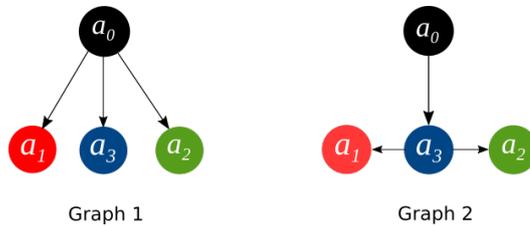


Figure 5.5: Communication graphs used in the multi-agent consensus tracking experiments

Case 1: Stationary leader, communication graph 1

In this case, the agents communicate over graph 1 shown in Figure 5.5. The three follower agents track the stationary leader as seen in Figure 5.6. The initial and final positions, safe distance d_p (See Section 5.6.2), total distance covered and time to consensus for each agent are given in Table 5.1.

The experiment is repeated with different initial conditions as shown in Figure 5.7. The follower agents are able to successfully track the leader as seen in Figure 5.7. The initial and final positions, safe distance d_p (see Section 5.6.2), total distance covered and time to consensus for each agent are given in Table 5.2.

The simulated trajectories of the agents obtained by approximating each of them as two independent double integrators along the X^V and Y^V direction (see Section 3.3.2) are also shown in Figures 5.6 and 5.7. The difference between the actual and simulated trajectories is due to

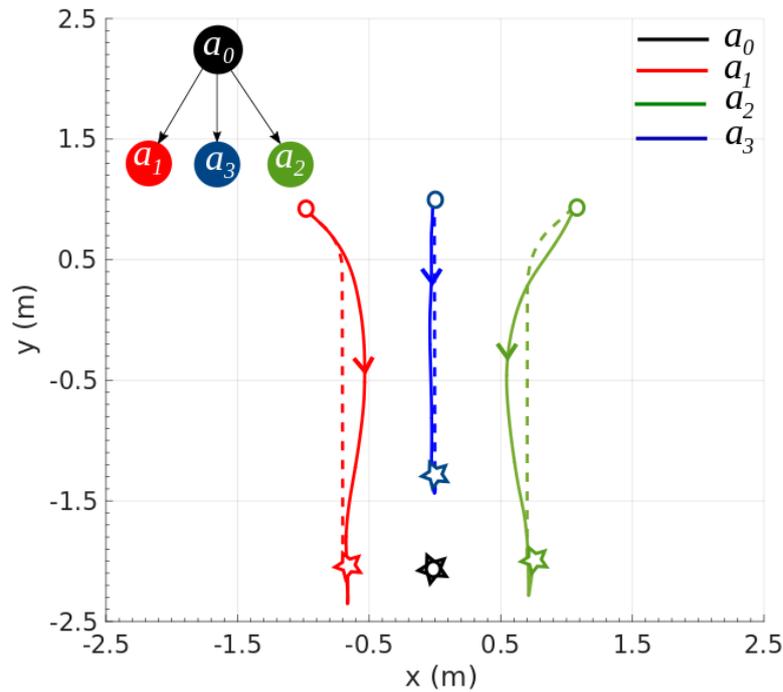


Figure 5.6: Three follower agents tracking a stationary leader. The initial conditions of the agents are marked with a circle and the final conditions with a star. The solid lines denote the actual trajectories and the dashed lines denote the simulated trajectories

Table 5.1: Consensus tracking details for Experiment 2, case 1

Node	Initial coordinates	Final coordinates	Safe distance		Consensus time		Distance covered	
			x (m)	y (m)	x (s)	y (s)	x (m)	y (m)
a_0	(0.01,-2.01)	(0.01,-2.01)	-	-	-	-	0	0
a_1	(-0.98,0.91)	(-0.66,-2.03)	-0.70	0.00	0.59	2.63	0.28	2.93
a_2	(1.06,0.94)	(0.74,-2.00)	0.70	0.00	0.95	2.61	0.35	2.96
a_3	(-0.01,1.00)	(0.00,-1.27)	0.00	0.70	-	2.48	0.00	2.32

the double integrator approximation.

Case 2: Stationary leader, communication graph 2

In this case, the agents communicate using graph 2 shown in Figure 5.5. The three follower agents track the stationary leader as seen in Figure 5.8. The initial and final positions, safe distance, total distance covered and time to consensus for each agent are given in Table 5.3. A video of this experiment is provided here ¹.

¹<https://youtu.be/vc1RCuZWsvI>

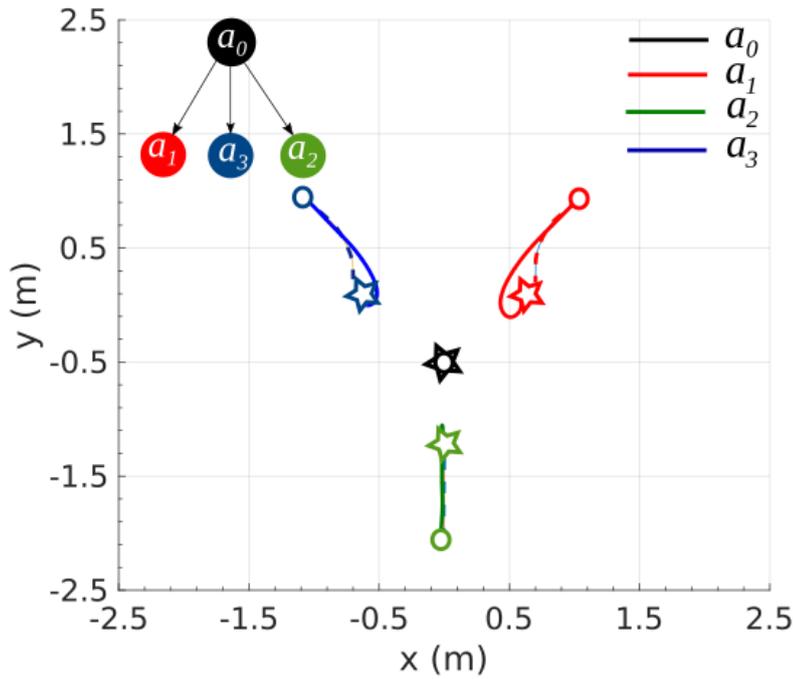


Figure 5.7: Three follower agents tracking a stationary leader. The initial conditions of the agents are marked with a circle and the final conditions with a star. The solid lines denote the actual trajectories and the dashed lines denote the simulated trajectories

Table 5.2: Consensus tracking details for Experiment 2, case 1

Node	Initial coordinates	Final coordinates	Safe distance		Consensus time		Distance covered	
			x (m)	y (m)	x (s)	y (s)	x (m)	y (m)
a_0	(-0.01,-0.50)	(0.00,-0.50)	-	-	-	-	0	0
a_1	(1.03,0.93)	(0.63,0.10)	0.70	0.70	0.95	1.31	0.44	0.83
a_2	(-1.09,0.96)	(-0.64,0.12)	0.00	-0.70	-	1.44	0.00	0.84
a_3	(-0.03,-2.05)	(0.00,-1.24)	-0.70	0.70	1.03	1.4	0.48	0.85

Table 5.3: Consensus tracking details for Experiment 2, case 2

Node	Initial coordinates	Final coordinates	Safe distance		Consensus time		Distance covered	
			x (m)	y (m)	x (s)	y (s)	x (m)	y (m)
a_0	(-0.01, 2.00)	(-0.01, 2.00)	-	-	-	-	0	0
a_1	(-1.00, 0.97)	(-0.74, -1.32)	-0.70	0.00	0.67	-	0.28	-
a_2	(1.02, 0.96)	(0.67, -1.31)	0.70	0.00	0.93	-	0.34	-
a_3	(-0.02, 1.00)	(-0.04, -1.29)	0.00	0.70	-	2.47	-	2.30

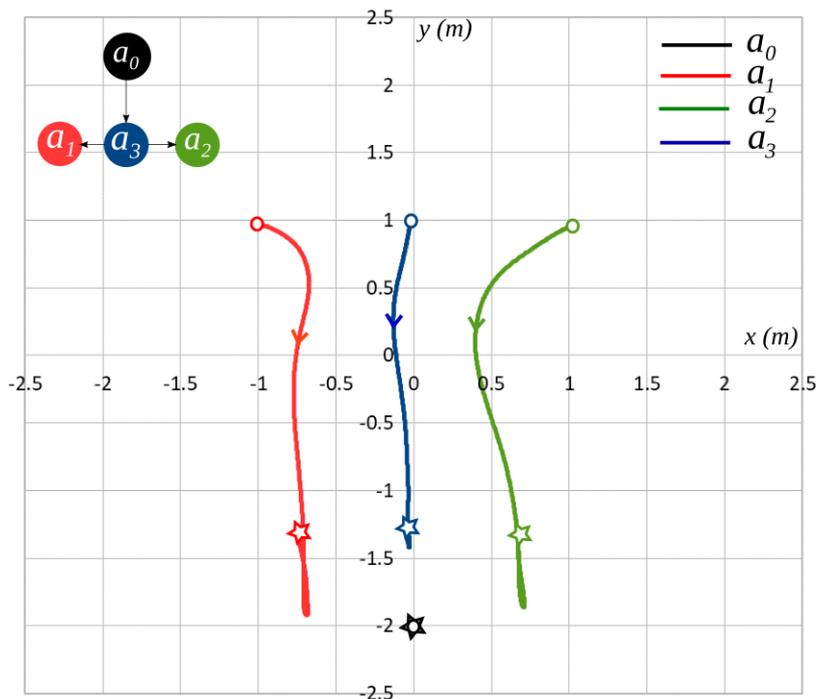


Figure 5.8: Three follower agents tracking a stationary leader. The initial conditions of the agents are marked with a circle and the final conditions with a star

Case 3: Moving evader, communication graph 1

In this case, the agents communicate using graph 1 shown in Figure 5.5. Consensus tracking is enabled whilst the leader agent is in motion as depicted in the video here ². The three follower agents successfully track the leader, which is moving away from the follower agents, as seen in Figure 5.9.

Experiment 2 shows that the control strategy in (5.6) achieves consensus tracking on a group of four quadrotors, for different initial conditions and communication graphs. It is shown that the control strategy also works when the leader agent is in motion. In Figures 5.6 - 5.9, an overshoot prior to reaching the endpoints is observed. Consider the motion depicted in Figures 5.6 and 5.8 under two different communication topologies. Note that as the length of the path originating leader node increases, the overshoot before reaching the endpoints increases. This behavior is similar to the one observed for vehicular traffic on roadways, where a slight overreaction to a small disturbance by a leader vehicle (e.g. sudden slight deceleration), leads to the propagation of a backward chain reaction in follower vehicles, causing them to slow down, far away from the origin of the disturbance. This phenomenon is well studied and known as *string stability* [SH96].

²<https://youtu.be/IAdKYFgpeZg>

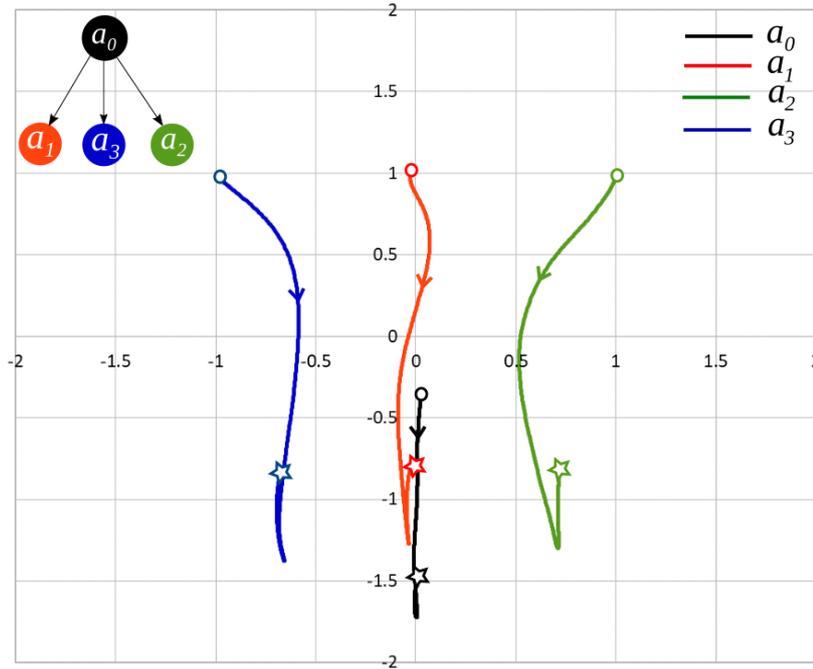


Figure 5.9: Three follower agents tracking a moving leader

5.6.5 Experiment 3: Effect of finite sampling rate

This experiment is performed with two quadrotors: one leader a_0 and one follower a_i (as in Experiment 1) for the various sampling times (see Definition 3) ranging from 5 ms to 20 ms.

Figures 5.10 and 5.11 show the comparison of the difference trajectory \mathbf{z}_i for the sampling time 5 ms with those corresponding to 10 ms and 20 ms, respectively. It is observed that for each sampling time, a_i tracks a_0 and the difference trajectory \mathbf{z}_i enters inside a small neighbourhood around the origin, (see Figures 5.10 and 5.11). The RMS deviation between the 5 ms and 20 ms difference trajectories (see Figure 5.10) is found to be 3.98 units, whereas that between the 5 ms and 10 ms (see Figure 5.11) is 1.68 units. Hence, it can be inferred that as the sampling time increases from 10 ms to 20 ms, the deviation of the difference trajectory \mathbf{z}_i from the nominal trajectory (corresponding to 5 ms) increases. The time required to achieve consensus tracking (\bar{t}) under the various sampling times is given in Table 5.4. It is observed that the time required to achieve consensus tracking does not change significantly with the change in the sampling time. This is an interesting observation because, a comparison of the trajectories reveals that the performance has degraded. It is our hypothesis that if the sampling time is increased further, then there might be a significant change in the time required for consensus tracking. However, experiments could not be carried out for larger sampling times as the quadrotor system becomes unstable and trajectory tracking is not possible when the sampling time is increased beyond 20 ms.

T_s (ms)	$z_i(0)$ (m)	$\dot{z}_i(0)$ (m/s)	\bar{t} (s)	$\bar{t} - t_{mean}$ (s)
5.00	2.30	0.02	2.47	-0.04
10.00	2.29	-0.01	2.47	-0.04
11.11	2.31	0.01	2.63	0.12
12.50	2.31	0.02	2.50	-0.01
14.29	2.32	0.00	2.48	-0.03
16.67	2.31	0.00	2.50	-0.01
20.00	2.29	0.02	2.50	-0.01
Mean (t_{mean})			2.51	

Table 5.4: Time taken by follower to track stationary leader for different sampling times

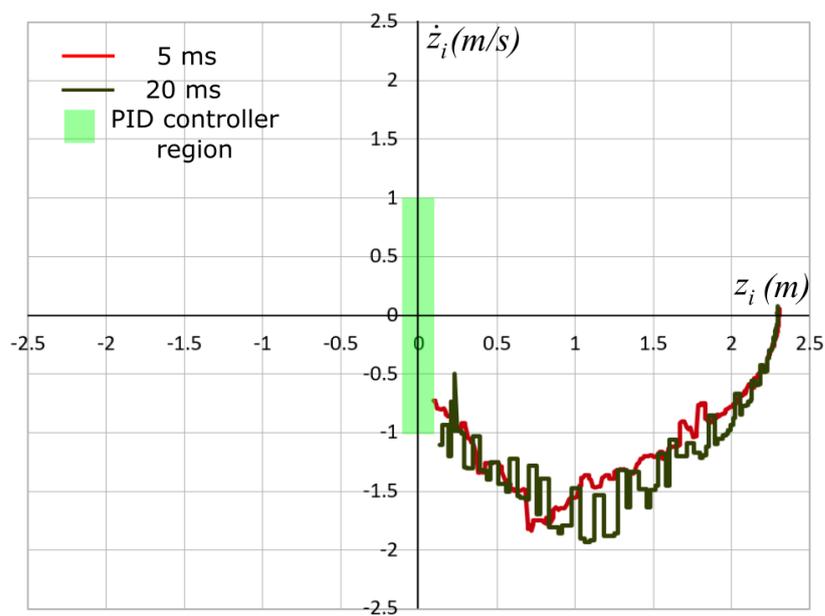


Figure 5.10: Comparison between the difference trajectory of the leader-follower pair for the sampling time of 5 ms and 20 ms

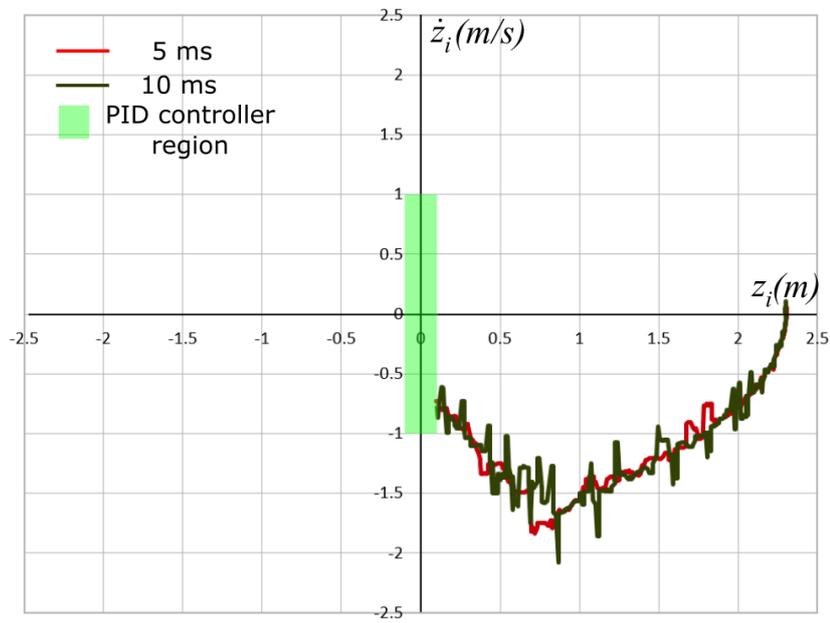


Figure 5.11: Comparison between the difference trajectory of the leader-follower pair for the sampling time of 5 ms and 10 ms

Chapter 6

Consensus

In this chapter a decentralized consensus law to drive multiple quadrotors to a common point, is proposed in Section 6.1. A novel communication protocol suitable for mobile agents is proposed in Section 6.2. Results of implementation of the consensus law and communication protocol on a multi-quadrotor testbed are presented in Section 6.3.

6.1 Proposed consensus law

The team of n quadrotors is said to achieve consensus if, for all $\mathbf{p}_i^E(0)$ and $\mathbf{v}_i^E(0)$ and all $i, j = 1, \dots, n$, $\|\mathbf{p}_i^E(t) - \mathbf{p}_j^E(t)\| \rightarrow 0$ and $\mathbf{v}_i^E(t) \rightarrow 0$ as $t \rightarrow \infty$.

Let each quadrotor represent a node in the set $\mathcal{V} = \{1, \dots, n\}$ and a communication link between two quadrotors at time t , an edge $e_{ij}^\sigma(t) \in \mathcal{E}^\sigma(t) \subseteq (\mathcal{V} \times \mathcal{V})$. A communication link between nodes is formed whenever they are within each others' communication range. If a communication link exists between nodes i and j , then they are said to be neighbours. Let a set \mathcal{S} index all possible undirected interaction topologies between the n nodes. Thus, a time-varying undirected graph, $\mathcal{G}^\sigma(t) := (\mathcal{V}, \mathcal{E}^\sigma(t))$, where a piecewise constant switching signal $\sigma(t) : [0, \infty) \rightarrow \mathcal{S}$ with switching times t_0, t_1, \dots , is used to model the switching interaction topology. The time between two consecutive switches is called the dwell time [Mor96]. Let τ be the minimum dwell time between consecutive switches such that $t_{k+1} - t_k \geq \tau$ for all non-negative integers, k . Let $\mathcal{N}_i^\sigma(t) := \{j \in \mathcal{V} : (i, j) \in \mathcal{E}^\sigma(t)\}$ denote the set of neighbours for a node i at time t . The adjacency matrix $\mathcal{A}^\sigma(t)$ of the undirected graph $\mathcal{G}^\sigma(t)$ represents communication network between nodes at time instant t and is defined as $\mathcal{A}^\sigma(t) := [a_{ij}^\sigma(t)] \in \mathbb{R}^{n \times n}$; $a_{ij}^\sigma(t) = 1$, if communication link exists between agents i and j at time t ($e_{ij}^\sigma(t) \in \mathcal{E}^\sigma(t)$), else $a_{ij}^\sigma(t) = 0$. The Laplacian matrix $\mathcal{L}^\sigma(t)$ of the undirected graph $\mathcal{G}^\sigma(t)$ is defined as $\mathcal{L}^\sigma(t) := [l_{ij}^\sigma(t)] \in \mathbb{R}^{n \times n}$; $l_{ij}^\sigma(t) = -a_{ij}^\sigma(t), i \neq j, l_{ii}^\sigma(t) = \sum_{j=1, j \neq i}^n a_{ij}^\sigma(t)$.

As shown in Section 3.3.2, the forces acting on the quadrotor can be controlled by manipulating θ_r and θ_p . The waypoint navigation law for a single quadrotor given in (3.15) is extended to formulate a consensus law to drive the n nodes to consensus. The proposed consensus law is

given by

$$\mathbf{f}_i^E(t) = \sum_{j \in \mathcal{N}_i^\sigma(t)} a_{ij}^\sigma(t) (\mathbf{p}_j^E(t) - \mathbf{p}_i^E(t)) - \beta \mathbf{v}_i^E(t), \quad i = 1, \dots, n \quad (6.1)$$

where β is a positive constant to be tuned by the designer. Under this consensus law, the closed-loop dynamics of the system in matrix form becomes

$$\begin{bmatrix} \dot{\mathbf{p}} \\ \dot{\mathbf{v}} \end{bmatrix} = (\boldsymbol{\alpha}^\sigma(t) \otimes I_2) \begin{bmatrix} \mathbf{p} \\ \mathbf{v} \end{bmatrix} \text{ where } \boldsymbol{\alpha}^\sigma(t) = \begin{bmatrix} 0_{n \times n} & I_n \\ -\mathcal{L}^\sigma(t) & -\beta I_n \end{bmatrix}, \quad (6.2)$$

where $I_n \in \mathbb{R}^{n \times n}$ is an identity matrix, $\mathbf{p} := [\mathbf{p}_1^E \ \dots \ \mathbf{p}_n^E]^T$ and $\mathbf{v} := [\mathbf{v}_1^E \ \dots \ \mathbf{v}_n^E]^T$.

Lemma 3. *If $\beta > 0$ and the undirected communication graph is connected at each time t , then $\boldsymbol{\alpha}^\sigma$ has $2n - 1$ negative real eigenvalues and one zero eigenvalue for each $\sigma \in \mathcal{S}$.*

Proof. The eigenvalues, $\lambda_{i\pm}^\sigma$ of $\boldsymbol{\alpha}^\sigma$ are given by [RA07]

$$\lambda_{i\pm}^\sigma = \frac{-\beta \pm \sqrt{\beta^2 + 4\mu_i^\sigma}}{2}, \quad (6.3)$$

where μ_i^σ is the i^{th} eigenvalue of $-\mathcal{L}^\sigma$. For an undirected graph, all non-zero eigenvalues of \mathcal{L}^σ are real, positive and zero is a simple eigenvalue if and only if the underlying graph is connected [RB08]. Then, without loss of generality, say $\mu_1^\sigma = 0$. Hence $\lambda_{1+}^\sigma = 0$. Choosing $\beta > 0$ ensures that the remaining $2n - 1$ eigenvalues of $\boldsymbol{\alpha}^\sigma$ have negative real parts as $\sqrt{\beta^2 + 4\mu_i^\sigma} \leq \beta$ for all μ_i^σ \square

If the communication graph remains static, that is $\sigma(t) = s$ (say s), for all t , then the following theorem predicts the final consensus value that the agents reach.

Theorem 4. *Suppose a system is governed by dynamics given by (6.2) with constant $\sigma(t) = s$. Then, the control law in (6.1) with $\beta > 0$ achieves consensus asymptotically if the communication graph \mathcal{G}^s is connected.*

Proof. (Sufficiency) Given that the communication graph is connected, from Lemma 3, it is known that $\boldsymbol{\alpha}^s$ has a simple zero eigenvalue. Let $\mathbf{x}_1 = [\mathbf{x}_a^T \ \mathbf{x}_b^T]^T$ be the eigenvector of $\boldsymbol{\alpha}^s$ associated with the zero eigenvalue. Then,

$$\boldsymbol{\alpha}^s \mathbf{x}_1 = \begin{bmatrix} 0_{n \times n} & I_n \\ -\mathcal{L}^s & -\beta I_n \end{bmatrix} \begin{bmatrix} \mathbf{x}_a \\ \mathbf{x}_b \end{bmatrix} = \begin{bmatrix} \mathbf{0}_n \\ \mathbf{0}_n \end{bmatrix},$$

where $\mathbf{0}_n$ is the $n \times 1$ vector of all zeros. From the above equation, $\mathbf{x}_b = \mathbf{0}_n$ and $-\mathcal{L}^s \mathbf{x}_a = \mathbf{0}_n$. Then \mathbf{x}_a , the eigenvector of \mathcal{L}^s associated with the zero eigenvalue equals $\mathbf{1}_n$, the $n \times 1$ vector of all ones.

Now $\boldsymbol{\alpha}^s$ can be written in the Jordan canonical form as

$$\boldsymbol{\alpha}^s = T J T^{-1} = \begin{bmatrix} \mathbf{x}_1 & \dots & \mathbf{x}_{2n} \end{bmatrix} \begin{bmatrix} 0_{1 \times 1} & 0_{1 \times 2n-1} \\ 0_{2n-1 \times 1} & J' \end{bmatrix} \begin{bmatrix} \mathbf{y}_1^T \\ \vdots \\ \mathbf{y}_{2n}^T \end{bmatrix},$$

where $\mathbf{x}_i \in \mathbb{R}^{2n}$, $i = 1, \dots, 2n$ can be chosen to be the right eigenvectors or the generalized right eigenvectors of α^s , $\mathbf{y}_i \in \mathbb{R}^{2n}$, $i = 1, \dots, 2n$ can be chosen to be the left eigenvectors or the generalized left eigenvectors of α^s , and J' is the Jordan canonical matrix corresponding to the non-zero eigenvalues of α^s .

To get the above form, $\mathbf{x}_1 = \begin{bmatrix} \mathbf{1}_n^T & \mathbf{0}_n^T \end{bmatrix}^T$ is chosen as the right eigenvector and $\mathbf{y}_1 = \begin{bmatrix} \beta \mathbf{1}_n^T & \mathbf{1}_n^T \end{bmatrix}^T$ as the left eigenvector of α^s corresponding to the zero eigenvalue. Following the analysis given in [RA07, Lemma 4.1], it is possible to show that $\mathbf{p}(t) \rightarrow (\beta \mathbf{1}_n \mathbf{1}_n^T \otimes I_2) \mathbf{p}(0) + (\mathbf{1}_n \mathbf{1}_n^T \otimes I_2) \mathbf{v}(0)$ and $\mathbf{v}(t) \rightarrow 0$ as $t \rightarrow \infty$. Hence $\|\mathbf{p}_i^E(t) - \mathbf{p}_j^E(t)\| \rightarrow 0$ and $\mathbf{v}_i^E \rightarrow 0$ as $t \rightarrow \infty$ for all $i, j = 1, \dots, n$.

The proof for necessity is similar to the necessity proof of [RA07, Lemma 4.1]. \square

Let us define the consensus error vector as $\tilde{\mathbf{p}} = \begin{bmatrix} \mathbf{p}_{12}^E & \dots & \mathbf{p}_{1n}^E \end{bmatrix}^T$ and $\tilde{\mathbf{v}} = \begin{bmatrix} \mathbf{v}_{12}^E & \dots & \mathbf{v}_{1n}^E \end{bmatrix}^T$, where $\mathbf{p}_{ij}^E = \mathbf{p}_i^E - \mathbf{p}_j^E$ and $\mathbf{v}_{ij}^E = \mathbf{v}_i^E - \mathbf{v}_j^E$. Then (6.2) can be rewritten in relative dynamics form as

$$\begin{bmatrix} \dot{\tilde{\mathbf{p}}} \\ \dot{\tilde{\mathbf{v}}} \end{bmatrix} = (\tilde{\alpha}^\sigma \otimes I_2) \begin{bmatrix} \tilde{\mathbf{p}} \\ \tilde{\mathbf{v}} \end{bmatrix}, \quad (6.4)$$

where $\tilde{\alpha}^\sigma \in \mathbb{R}^{2(n-1) \times 2(n-1)}$. If, for fixed $\sigma \in \mathcal{S}$, $\tilde{\alpha}^\sigma$ is stable, then there exist $p^\sigma \geq 0$ and $q^\sigma \geq 0$ such that $\|e^{\tilde{\alpha}^\sigma t}\| \leq e^{(p^\sigma - q^\sigma t)}$, $t \geq 0$ [RB08].

Theorem 5. *If $\beta > 0$, the communication graph is connected for all $\sigma \in \mathcal{S}$ and the dwell time, τ satisfies $\tau > \sup_{\sigma \in \mathcal{S}} \frac{p^\sigma}{q^\sigma}$ for some $p^\sigma, q^\sigma \geq 0$, then consensus is asymptotically achieved using (6.1).*

Proof. Given $\beta > 0$ and that the communication graph is connected for each $\sigma \in \mathcal{S}$, Theorem 4 holds and ensures that consensus is achieved asymptotically for each $\sigma \in \mathcal{S}$. Hence, $\tilde{\mathbf{p}} \rightarrow 0$ and $\tilde{\mathbf{v}} \rightarrow 0$ asymptotically and the switched system (6.4) is stable for each $\sigma \in \mathcal{S}$. If the dwell time τ satisfies $\tau > \sup_{\sigma \in \mathcal{S}} \frac{p^\sigma}{q^\sigma}$, then using [Mor96], it follows that the switched system (6.4) is globally exponentially stable. This implies that consensus can be achieved asymptotically. \square

This consensus law is implemented on the multi-quadrotor testbed in an outdoor environment. The control architecture for cooperative control using the consensus law is shown in Fig. 6.1. Observe that the control architecture is similar to that depicted in Figure 3.3. Again, the inner control loops are used to control the attitude and altitude of the quadrotor and the outer loops, for control of position. The quadrotors receive position data from neighboring agents and calculate the desired position value according to the consensus law. as shown in Figure 6.1.

6.2 Communication Protocol

For successful execution of cooperative control strategies, inter-agent communication plays a vital role. Carrier sense multiple access with collision avoidance protocols (CSMA/CA) are

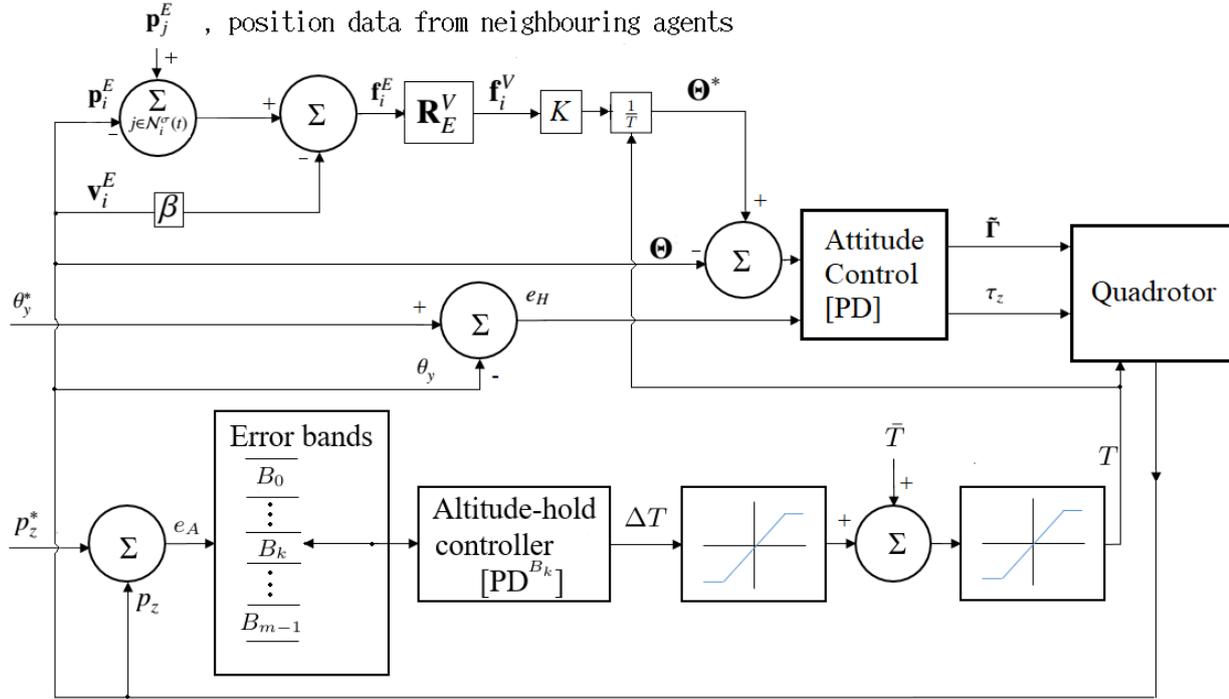


Figure 6.1: A block diagram of the quadrotor control loops

commonly used for communication in multi-agent systems [GGC13], [VVS⁺14]. However, communication outages due to data packet collisions for up to 2 s have been observed in preliminary experiments with multiple quadrotors [JLA⁺16]. With additional computational overheads, [VVS⁺14] reports outages in the range of seconds. Thus, the standard CSMA-CA protocol is not suitable for real-time communication in this context as there is no guaranteed upper-bound on the delay in transmission [GGC13].

In this work a communication protocol based on time division multiplexing (TDM) is developed wherein agents with limited communication range can reliably transmit data without collisions. The protocol allows synchronized real-time communication between agents without the need for a ground station. Synchronization is achieved using an additional reference node mounted on-board one of the quadrotors, thus making the communications system capable of being fully airborne. It is also capable of handling dynamic communication topologies and robust to link breakages with the reference node. It provides guaranteed throughput (of upto 3 data packets per second per node for data packets of 15 bytes) and maintains connectivity between agents.

6.2.1 Steps involved

Broadly, the protocol comprises of the following steps:

1. Time synchronization
2. Slot allotment

- 3. Data transfer
- 4. Re-synchronization

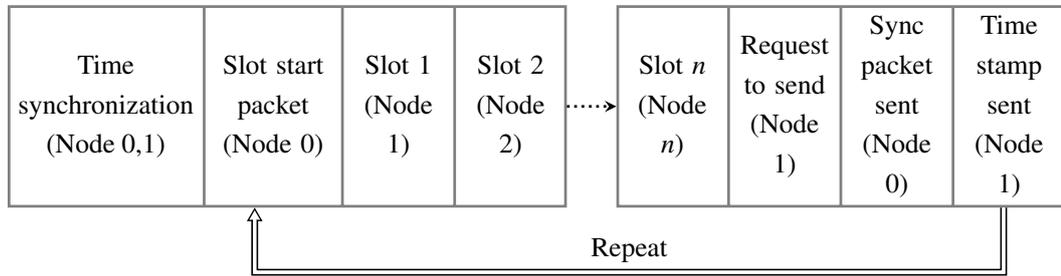


Figure 6.2: Workflow of the communication protocol

Suppose that the communication network needs to be set up between n agents. Then $n + 1$ nodes are required for the implementation of this protocol. A reference node, say Node 0, is used for synchronization. Nodes $1, \dots, n$ correspond to the n agents. Node 0 and Node 1 are physically set up adjacent to each other on one agent. Fig. 6.2 depicts the various steps from left to right in chronological order. Each step is discussed in detail below.

Time synchronization

To achieve time-based slot allotment, the local clocks of all nodes need to be synchronized during the initialization phase. All the nodes are synchronized using reference broadcast synchronization (RBS) [EGE02]. Fig. 6.3 explains the time synchronization process.

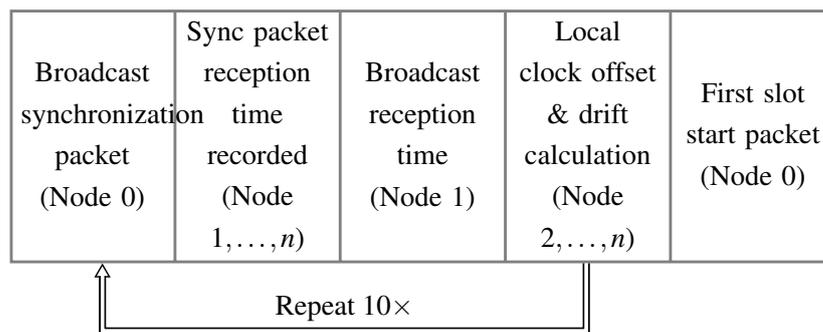


Figure 6.3: Steps involved in the initial time synchronization

Example

Consider time synchronization between Node 1 and Node 2. Initially, Node 0 broadcasts a synchronization packet which is received by Node 1 at (say) time t_1 and Node 2 at t_2 . Node 1 now sends its packet reception time, t_1 . This process is repeated ten times. Node 2 uses these ten time-stamps to calculate its local clock offset ($t_2 - t_1$) with respect to local clock of Node 1 and correct for any clock drift.

Experiments show that the worst case synchronization error between the clocks of Node 1 and any other node is less than 2 ms.

Slot allotment

Now that local clocks of all the nodes are synchronized, the time slots during which each node can transmit its data, can be allotted. All nodes are assigned a number based on which the time slots are allotted in an ascending order. It is observed that given the worst case of synchronization error, a node is successfully able to transmit a data packet of 15 bytes in a time duration of less than 20 ms. Hence, each node is given 20 ms for data transmission. The end of slot allotment is marked by a broadcast packet by Node 0 which signals that the nodes can now start data transfer. Fig 6.2 depicts the manner in which Node i transmits data in Slot i .

Data transfer

Data transfer commences once Node 0 broadcasts a slot start packet. Node 1 is the first to transmit data. The other nodes follow based on local clock times which are synchronized with respect to Node 1. The Xbee API mode is used for data transfer. Fig. 6.4 shows the data packet structure used for transmission. The packet consists the standard start delimiter (0x7E), which marks start of new packet and length, the position coordinates (15 bytes) of the node and checksum which is used to validate the received data.

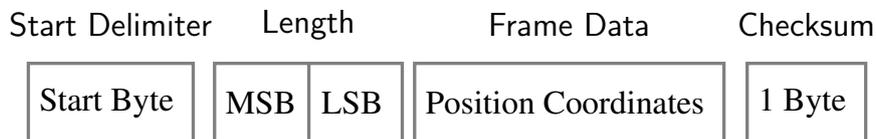


Figure 6.4: Data packet structure of Xbee in API mode

Re-synchronization

This step helps maintain the synchronization achieved during the initialization phase. At the end of the last slot of every frame, Node 1 requests Node 0 to transmit a synchronization packet followed by the time-stamp sent by Node 1 as shown in Fig. 6.2. New coefficients are calculated using a sliding window average of the last ten consecutive time differences for drift compensation. This process is similar to the time synchronization step. Node 0 broadcasts a special data packet that marks the end of time synchronization and the whole data transfer process is repeated.

6.2.2 Addressing contingencies

The main application of this protocol is in the case where the agents are in motion. As the agents move in and out each others' communication range, links are dynamically created and

dropped. In cluttered environments, there can be situations where obstacles affect communication between agents.

Link break with Node 0, 1

An obstacle appearing in between Node 0, 1 and any other nodes, can result in a loss of reception of the synchronizing and time stamped data packets. In such a case, the node receives data packets from other nodes in its vicinity and records their time of arrival. As each node knows its assigned slot number, it can calculate its relative slot start time based on the recorded time of arrival. This process is explained below with the help of an example.

Example

Consider the situation presented in Fig. 6.5. Observe that Node 3 has lost connection with Node 0, 1 due to the presence of an obstacle. However, it can still form a communication link with nodes 2 and/or 4. Suppose the total frame cycle has a duration of T milliseconds. If Node 3 receives a data packet from Node 2 at time t , then it calculates its own slot start time to be $(t + 20)$ milliseconds. If Node 3 receives a data packet from Node 4 at t , then it calculates its own slot start time to be $(t + T - 20)$ milliseconds. This process continues till the broken link with the Node 0, 1 is re-established and synchronization takes place.

Complete loss of reception

There can be a situation wherein a node is completely isolated and does not receive any data from the other nodes. In such a case, the node keeps sending its data packets in the allotted time slot for up to 15 s after complete loss of reception and then switches to only receiving data from the other nodes. A time duration of 15 s is chosen as it is experimentally observed that nodes maintain acceptable synchronization of up to 20 ms (the time duration of one slot), after which data packet collisions tend to occur due to time slots overlapping. When a node returns to the network, it gets synchronized again and starts participating in the normal data transfer process.

6.3 Experiments

This section contains the results of the experiments performed to test the performance of the communication protocol, the consensus algorithm and the effect of communication rate on consensus performance.

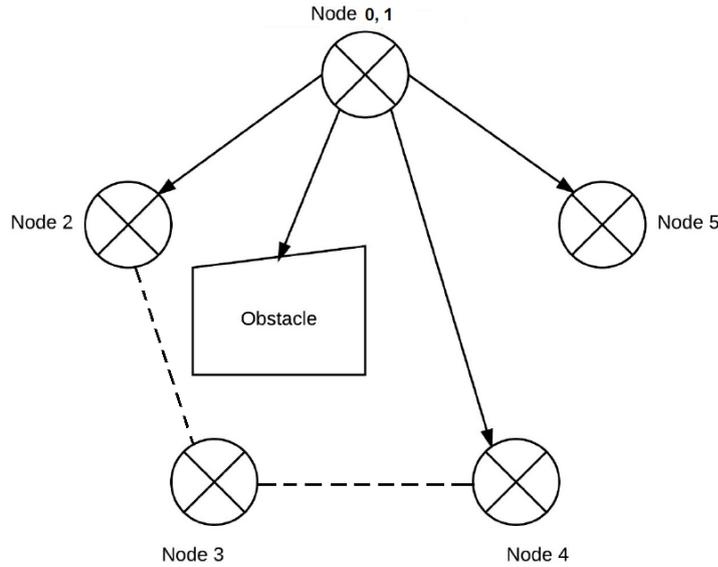


Figure 6.5: An instance of case of link break between Nodes 1 and 3 due to the presence of an obstacle

6.3.1 Communication protocol experiments

Experiment 1: Verification of communication protocol

(a) *Setup time*: Setup time is the time required for the completion of the initialization phase which comprises of the time synchronization and slot allotment stages. Table 6.1 contains sample data regarding the times at which Node 0 broadcasts synchronization and slot start packets. It broadcasts ten synchronization packets between 0 - 2.07 s. This is followed by the slot allotment phase after which the slot start packet is broadcast at 3.1 s.

Table 6.1: Sample packet transmission times of Node 0 during initialization phase

Packet type	Packet no.	Time (s)
Broadcast sync packet	1	0.000000
	2	0.228948
	3	0.462112
	4	0.686640
	5	0.916764
	6	1.146924
	7	1.379964
	8	1.607384
	9	1.837680
	10	2.068088
Slot start packet	1	3.098908

(b) *Communication efficiency*: The efficiency of communication can be inferred from the transmission efficiency and reception efficiency. The transmission efficiency is characterized by the number of slots a node participates in. The number of slots a node participates in is equivalent to the number of data packets it transmits. As stated earlier, local clocks of all nodes are synchronized with respect to the clock of Node 1. Node 1 always receives the slot start packet sent by Node 0 as they are physically placed next to each other. Thus, it is assumed that Node 1 participates in all the slots. Based on this assumption, the efficiency of transmission of Node i is calculated as

$$\eta_{t,i} = \frac{D_{t,i}}{D_{t,1}}, \quad (6.5)$$

where $D_{t,i}$ is the total number of packets transmitted by Node i over the entire duration of the experiment. Reception efficiency is defined as

$$\eta_{r,i} = \frac{1}{n-1} \sum_{\substack{j=1 \\ j \neq i}}^n \frac{D_{r,i}}{D_{t,j}}, \quad (6.6)$$

where $D_{r,i}$ is the total number of packets received by Node i over the entire duration of the experiment. Then, the average transmission and reception efficiency over R test runs is calculated as

$$\eta_t^{avg} = \frac{1}{R} \frac{\sum_{i=1}^n \eta_{t,i}}{n} \quad (6.7)$$

and

$$\eta_r^{avg} = \frac{1}{R} \frac{\sum_{i=1}^n \eta_{r,i}}{n} \quad (6.8)$$

respectively.

A total of 18 test runs of duration 300 s each are carried out in indoor and outdoor environments with three to six nodes in the communication network. Using (6.7), the average transmission efficiency is found to be 98.04 %. The reception efficiency of the protocol is found using (6.8) to be 98.52%. Experiments are performed to observe the effect of addition of nodes to the network, on the communication transmission and reception efficiency. These experiments are repeated for different number of nodes. From Table 6.2, we can infer that the addition of communication nodes to the network does not decrease the transmission and reception efficiency significantly.

Table 6.2: A comparison of number of nodes in the network and communication efficiency

No. of nodes	No. of test runs	η_t^{avg}	η_r^{avg}
3	6	99.09 %	99.06 %
4	4	98.29 %	99.60 %
5	3	98.82 %	98.96 %

Table 6.3 provides a comparison of communication efficiency in indoor and outdoor environment. The data in this table is for communication between 6 nodes. The indoor test runs are

carried out in an area of $10 \text{ m} \times 10 \text{ m}$ and outdoor area has dimensions $55 \text{ m} \times 46 \text{ m}$. It can be seen that there is a reduction (approximately 2 %) in transmission and reception communication efficiency. This loss can be attributed to environmental factors like interference and attenuation.

Table 6.3: A comparison of communication efficiency in indoor and outdoor environment

Environment	No. of test runs	η_t^{avg}	η_r^{avg}
Indoor	2	97.38 %	97.88 %
Outdoor	3	95.04 %	95.58 %

(c) *Link break*: In this experiment, the performance of the protocol in case a node is unable to receive synchronization packets from Node 0, is tested. The experiment is performed with six nodes. Fig. 6.6 depicts, along the y-axis, the data packets that Node 3 transmits (in blue) and receives from other nodes (in grey, maroon, green, red and black). The different slots in which communication occurs are plot along the x-axis. It can be seen that Node 3 is unable to receive synchronization packets from Node 0 starting from slot number 260. However, it is still able to receive data packets from Node 2. Using the time of arrival of this data, Node 3 calculates its own time slot and successfully transmits data as seen in the figure.

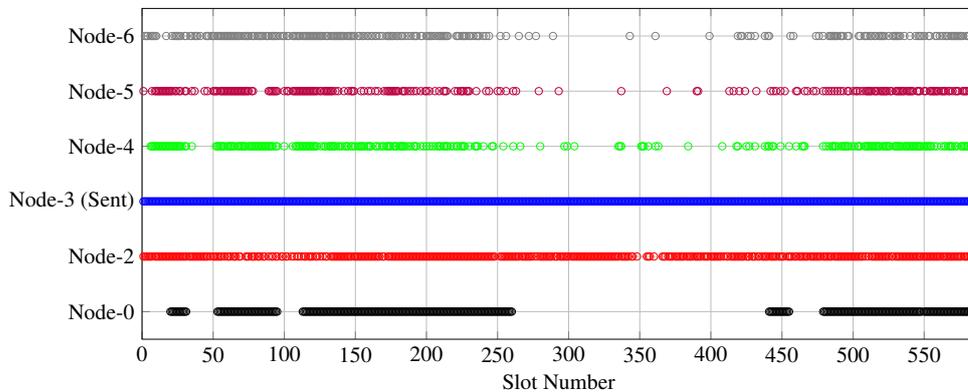


Figure 6.6: Data transfer during the link break mode of operation

(d) *Sample communication cycle*: A snapshot of nine complete time-frames of the communication protocol can be seen in Fig. 6.7. Each circle in the figure depicts the time at which the corresponding node transmits data. The first four packets (marked along the y-axis) correspond to the resynchronization phase which is followed by the data exchange between the six nodes. It can be verified that each node transmits data in the allotted slot of 20 ms and that there are no data collisions during this time period.

6.3.2 Cooperative control experiments

In this section the results of the cooperative control experiments performed with the physical and virtual quadrotors are presented. The experiments are performed on an outdoor field with

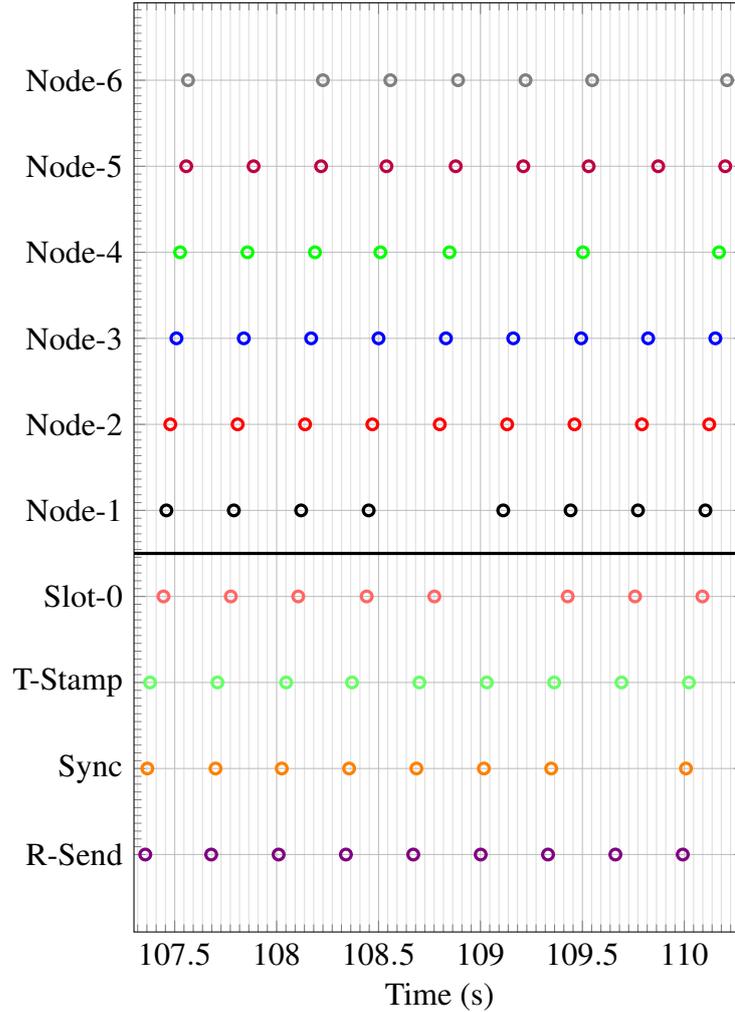


Figure 6.7: A snapshot of nine time-frames of the communication protocol. Each circle corresponds to the time at which the node transmits data

dimensions $90 \text{ m} \times 50 \text{ m}$. The ublox LEA-6H GPS receiver used for localization by the quadrotors has an accuracy of 2.5 m CEP [ubl17] (the measured position is within a circle of radius 2.5 m of the true position 50% of the time [Nel88]). Hence, for practical purposes the quadrotors are said to achieve consensus when $\|\mathbf{p}_i^E(t) - \mathbf{p}_j^E(t)\| \leq 5 \text{ m}$ for all $i, j = 1, \dots, 6$.

Experiment 2: Static graph

(a) *Path graph*: This experiment is performed with the communication graph between the three physical quadrotors as shown in Fig 6.8. Table 6.4 provides details about the initial and final conditions of the three physical quadrotors with respect to a reference frame $\{R\}$ shown in Fig. 6.10. A GPS plot of the trajectories that the quadrotors follow to reach consensus is given in Fig. 6.10.

A simulation for consensus of three double integrator agents with the same communication graph and initial conditions as the physical quadrotors is also performed. The resulting trajectories are depicted by the solid lines in Fig 6.9. The corresponding trajectories from the physical

Table 6.4: Position data of the quadrotors for Experiment 2(a)

Node	Type	Initial position (m)	Final position (m)
1	Physical	(14.39, 0.57)	(20.62, 15.17)
2	Physical	(26.21, 49.11)	(20.25, 20.88)
3	Physical	(52.44, 5.18)	(22.66, 21.31)

Time to consensus = 4 s

implementation, depicted by the circles have been overlaid on the simulated trajectories.

In both cases the time taken by the agents to reach consensus is 4 s. The average absolute tracking error along the X -axis and Y -axis is 6.16 m and 6.54 m respectively. The cumulative RMS error

$$e_{RMS} = \frac{1}{n} \sum_{i=1}^n \sum_{k=1}^T \frac{\sqrt{(x_i^E(k) - x_i^S(k))^2 + (y_i^E(k) - y_i^S(k))^2}}{T},$$

where x_i^E and y_i^E are the coordinates of the physical quadrotor and x_i^S and y_i^S are the coordinates of the simulated quadrotor i along the X and Y axis respectively, and T is the time duration of the experiment, is found to be 9.83 m.

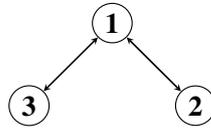


Figure 6.8: Static communication graph for consensus experiment with three physical nodes

(b) *Complete graph*: This experiment is performed with a complete communication graph between the three physical and three virtual quadrotors, i.e. every quadrotor can communicate with every other quadrotor. Table 6.5 provides details about the initial and final conditions of the three physical quadrotors with respect to a reference frame $\{R\}$ shown in Fig. 6.11. From the GPS plot of trajectories and the data in Table 6.5, it can be seen that the six nodes reach consensus, inspite of wind disturbances (in the range of 5 - 10 km/h). The time taken to reach consensus in this case is also 4 s.

Experiment 3: Dynamic graphs

In these experiments, the communication graph between the quadrotors changes based on the distance between them.

(a) *Simulation results*: This experiment is performed with three virtual quadrotors. The communication range of each node is 50 m. Initially, the three quadrotors are placed such that the communication graph between them is as shown in Fig. 6.12a. Each node moves towards the centroid of its neighbours as per the consensus law (6.1). Consider the motion of Node 1. As

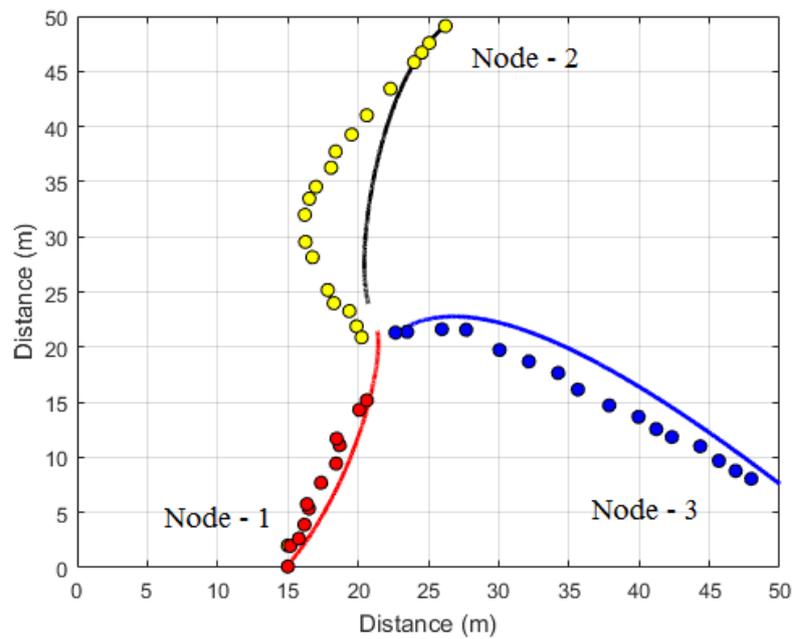


Figure 6.9: Comparison between simulated and experimental trajectories of three agents under the consensus law. Trajectories from the experiment are depicted by the circles and simulated trajectories are by solid lines



Figure 6.10: GPS plot of consensus between three physical quadrotors

Node 1 is connected to only Node 2, it starts moving towards the centroid of Nodes 1 and 2, $\frac{(\mathbf{p}_2 + \mathbf{p}_1)}{2}$. As soon as the distance between Nodes 1 and 3 reduces to below 50 m (as depicted by the vertical blue line in Fig. 6.13), a communication link between them is formed. The new communication graph is now as shown in Fig. 6.12b. Thus, Node 1 now starts moving towards the updated centroid, $\frac{(\mathbf{p}_1 + \mathbf{p}_2 + \mathbf{p}_3)}{3}$, of all its neighbours i.e. Nodes 1, 2 and 3. In spite of the shift

Table 6.5: Position data of the quadrotors for Experiment 2(b)

Node	Type	Initial position (m)	Final position (m)
1	Physical	(0.45, 22.78)	(27.29, 28.07)
2	Physical	(15.93, 10.98)	(27.67, 28.56)
3	Physical	(1.66, 10.50)	(29.17, 30.28)
4	Virtual	(49.97, 10.05)	(27.48, 28.49)
5	Virtual	(49.78, 52.74)	(28.26, 29.72)
6	Virtual	(23.65, 51.01)	(27.33, 27.96)

Time to consensus = 4 s

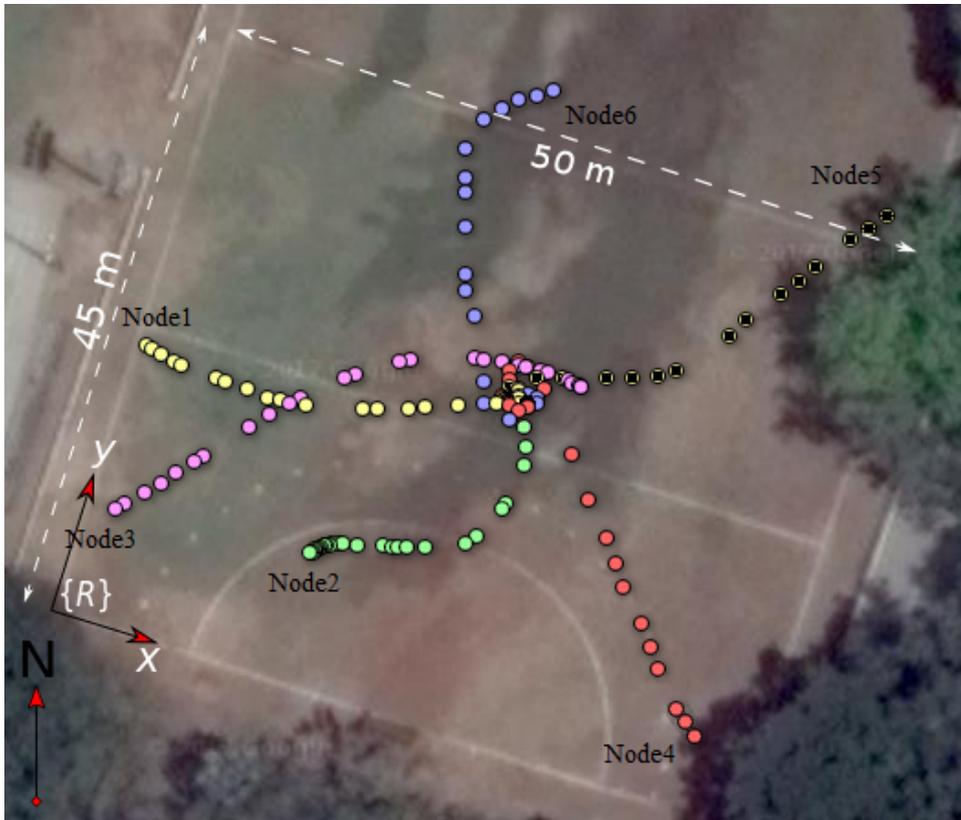


Figure 6.11: GPS plot of consensus between six nodes: three physical and three virtual units

in consensus point due to the change in the communication graph, as depicted by the kink in Fig. 6.13, the nodes finally reach consensus.

(b) *On-field implementation:* The same experiment is performed using physical quadrotors. Initially, the communication graph between the nodes is as shown in Fig. 6.12a and Nodes 1 and 3 move towards Node 2. As soon as Nodes 1 and 3 are within range, switching occurs and the communication graph changes to Fig. 6.12b. Table 6.6 provides details about the initial and final conditions of the three physical quadrotors with respect to a reference frame $\{R\}$ shown in Fig. 6.14. The quadrotors follow trajectories as shown in Fig. 6.14 and reach consensus. The

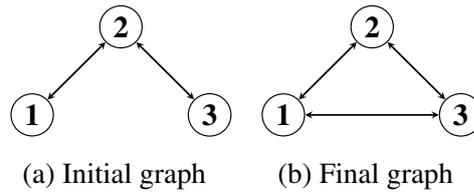


Figure 6.12: A case of switching communication topology

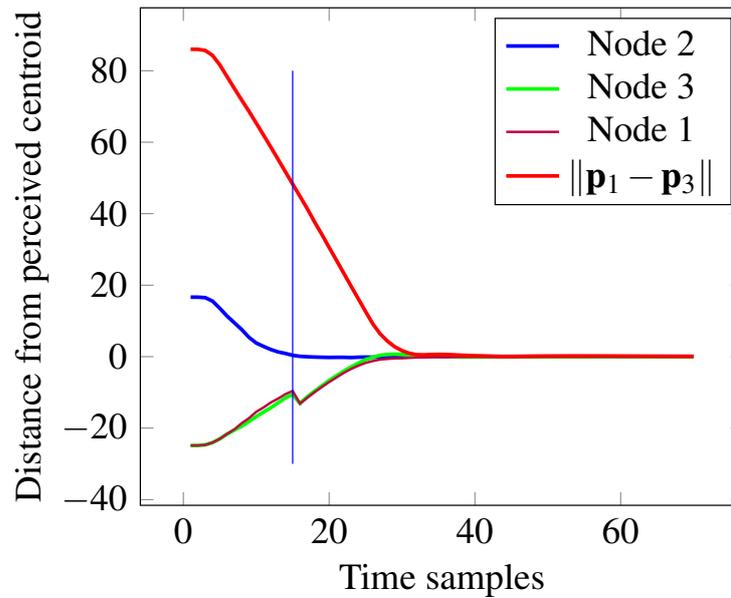


Figure 6.13: Centroid tracking in the case of switching communication topology. The blue vertical line denotes the time at which a link between Nodes 1 and 3 is formed

time taken to reach consensus in this case is 8 s.

Table 6.6: Position data of the quadrotors for Experiment 3(b)

Node	Type	Initial position (m)	Final position (m)
1	Physical	(29.61, -29.39)	(6.84, -5.87)
2	Physical	(-1.87, 3.91)	(7.52, -10.22)
3	Physical	(27.05, 34.58)	(7.22, -10.33)

Time to consensus = 8 s

Experiment 5: Consensus with different data exchange rates

In this experiment, the rate at which data is exchanged between the nodes is altered to study effect of communication on consensus performance. The consensus law is executed on the virtual units for various data exchange rates ranging from 3 Hz to 0.1 Hz. The speed of the

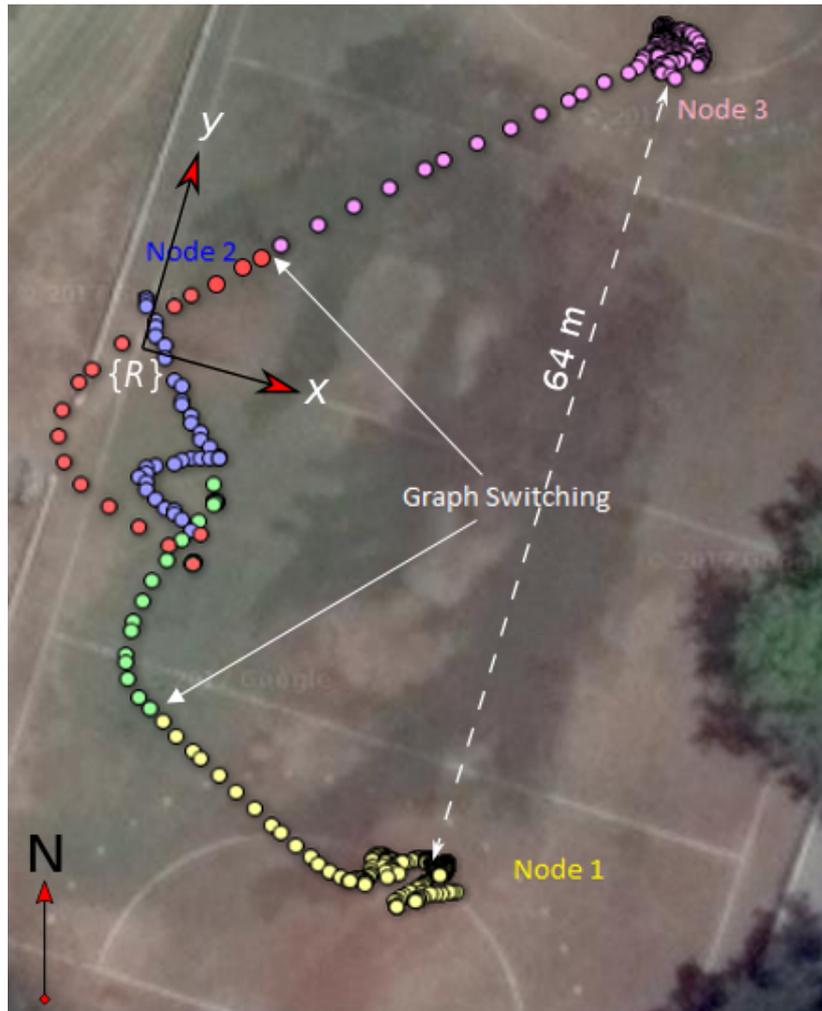


Figure 6.14: GPS plot of consensus between three physical quadrotors with a dynamic communication graph

quadrotors is fixed to 5 m/s. In all the following experiments, the quadrotors are given the same initial conditions.

The consensus law guarantees that the agents will asymptotically reach consensus. For communication rates of 3 Hz, 1 Hz and 0.3 Hz, it is observed in Figure 6.15 that the quadrotors reach the same consensus point at $(10.56, 0.04)$, in approximately the same time of 19 s.

However for the communication rates of 0.25 Hz, 0.2 Hz and 0.1 Hz, it can be seen in Figure 6.16 that consensus is not reached. The quadrotors are seen to deviate from the actual consensus point at $(10.56, 0.04)$ and oscillate.

It should be noted that in [VVS⁺14], the authors report communication outages even in the range of seconds. From these experiments, it is evident that in close range operations, such as consensus, performance degrades in such situations.

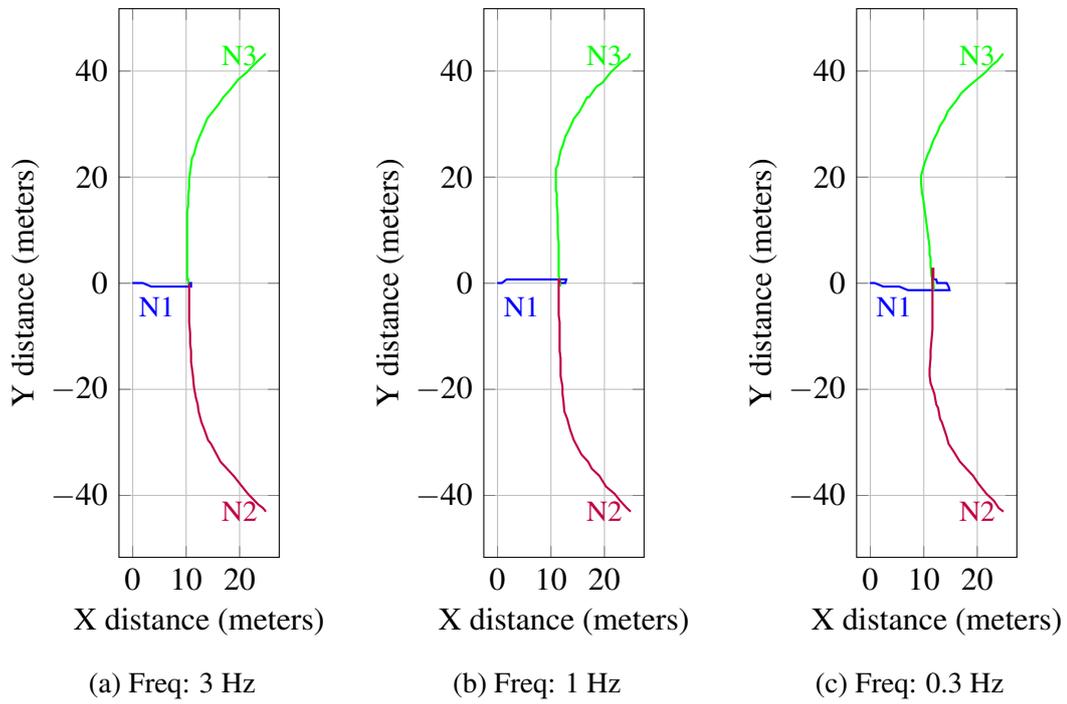


Figure 6.15: Plot of quadrotor trajectories for different data exchange rates

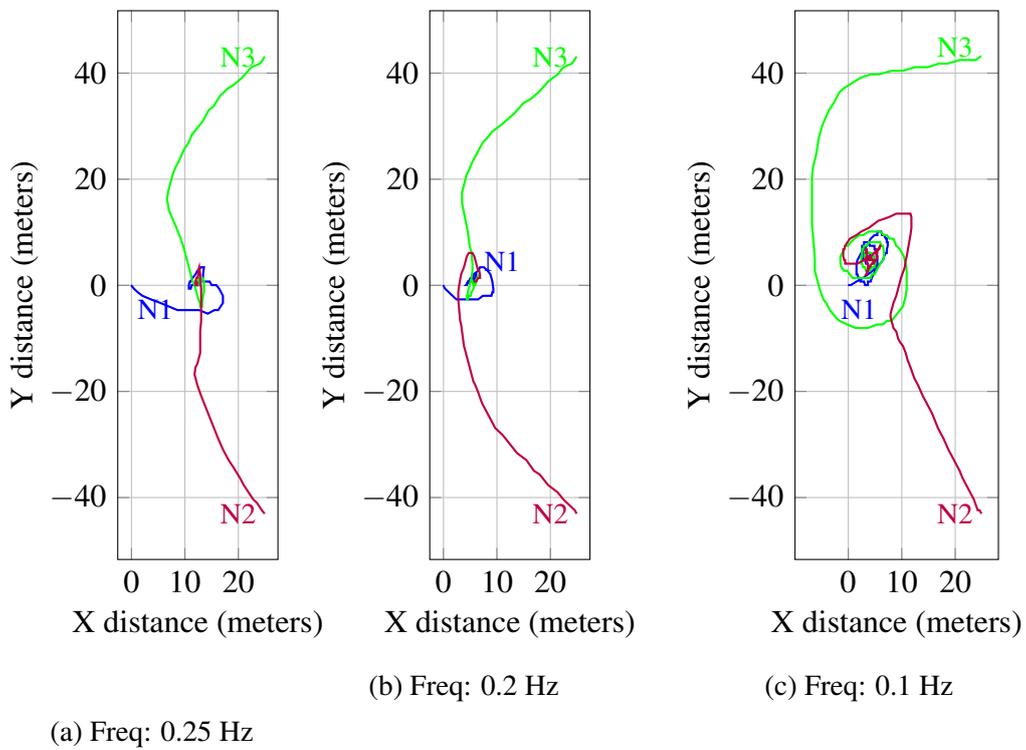


Figure 6.16: Plot of quadrotor trajectories for different data exchange rates

Part II

Modeling of multi-agent systems

Chapter 7

Review of a model for lane-less traffic

The focus in this part of the thesis is to infer the laws of interaction through observation of vehicular traffic in conditions where the human drivers do not follow lane discipline. The chapter reviews a mathematical model developed in [MJC⁺18] for lane-less traffic, where the interaction topology between vehicles remains fixed. This model predicts the motion of vehicles in dense traffic conditions, where all drivers are forced to behave homogeneously. In this thesis (Chapter 8), this model is extended to predict behavior of heterogeneous traffic in sparse traffic conditions and then experiments are carried out for the verification of these models.

We start this chapter by revisiting some notation from graph theory in Section 7.1. The modeling assumptions are stated in Section 7.2. In Section 7.3, the model for homogeneous traffic is presented.

7.1 Notation from graph theory

Like in the case of quadrotors considered in the first part of the thesis, the interaction between vehicles on roadways is modeled using graph theory. However, as we shall see in Section 7.2, each driver assigns weights to neighboring vehicles, which influence his/her control actions. In this section, we shall modify the notation introduced in Section 5.1 to make it applicable to *weighted graphs*.

An undirected *graph* $\mathcal{G} = (\mathcal{A}, \mathcal{E}, w)$ is a finite set of nodes \mathcal{A} connected by a set of edges $\mathcal{E} \subset \mathcal{A} \times \mathcal{A}$ along with a function $w : \mathcal{E} \rightarrow \mathbb{R}^+$ that denotes the *weights* of the edges. When two nodes $a_i \in \mathcal{A}$ and $a_j \in \mathcal{A}$ are connected to each other the graph \mathcal{G} is said to have an edge between a_i and a_j , denoted by $(a_i, a_j) \in \mathcal{E}$. A *spanning tree* is a connected graph $\mathcal{G}_{tree} = (\mathcal{A}_{tree}, \mathcal{E}_{tree}, w)$ having no cycles in the graph. If the edges of a graph $\vec{\mathcal{G}} = (\mathcal{A}, \vec{\mathcal{E}}, w)$ are directed i.e., $(a_i, a_j) \in \vec{\mathcal{E}} \not\Rightarrow (a_j, a_i) \in \vec{\mathcal{E}}$, the graph is called a *directed graph* (or a *digraph*). A *directed path* is a sequence of nodes $a_{i_1}, a_{i_2}, \dots, a_{i_r} \in \mathcal{A}$ such that $\forall k = 1, \dots, r-1, (a_{i_k}, a_{i_{k+1}}) \in \vec{\mathcal{E}}$. A *directed cycle* is a path such that $a_{i_r} = a_{i_1}$. A digraph is said to be *acyclic*, when it does not contain a cycle. A *rooted directed tree* is an acyclic digraph such that there exists a node (called *root*) and a directed path from that node to all other nodes in the digraph. A digraph $\vec{\mathcal{G}}$ is said to contain

a *directed spanning tree*, if there exists a rooted directed tree $\vec{\mathcal{G}}_{tree} = (\mathcal{A}_{tree}, \vec{\mathcal{E}}_{tree}, w)$ such that $\mathcal{A}_{tree} = \mathcal{A}$ and $\vec{\mathcal{E}}_{tree} \subseteq \vec{\mathcal{E}}$. In a digraph, a node with no incoming edge is called the *leader node*. The remaining nodes are called *follower nodes*. When a digraph has more than one leader, it is said to contain *united directed spanning tree* [CR09], if for each of the follower nodes, there exist at least one leader, from which there is a directed path to the follower node. The Laplacian (\mathcal{L}) for a directed graph with weights w_{ij} is defined as follows: $l_{ij} := -w_{ij}$ if $(a_j, a_i) \in \vec{\mathcal{E}}$, $l_{ij} := 0$ if $(a_j, a_i) \notin \vec{\mathcal{E}}$, and $l_{ii} := \sum_{j=1}^n w_{ij} :=$ cumulative weight of incoming edges.

Next, we recall definitions of some terms related to *layered graphs* from [Tam13]. Layered graphs are usually defined for visual representation of the graphs by proper positioning of nodes. However, in the context of traffic, since the influence graphs arise from the spatial arrangement of nodes (vehicles), this definition becomes especially relevant. For a directed graph $\vec{\mathcal{G}} = (\mathcal{A}, \vec{\mathcal{E}}, w)$, let $\mathbf{L} = \{L_0, L_1, L_2, \dots, L_k\}$, $k \geq 1$ be a partition of \mathcal{A} such that if $(a_i, a_j) \in \vec{\mathcal{E}}$ with $u_i \in L_p$ and $u_j \in L_q$, then $q < p$. Such an \mathbf{L} is called a *layering* of $\vec{\mathcal{G}}$ and L_0, L_1, \dots, L_k are referred to as *layers*. A digraph with layering is called a *layered digraph* [Tam13]. The index of a layer that contains a node a_i is denoted by $l(a_i, \mathbf{L})$, where $l(a_i, \mathbf{L}) := p$ if and only if $a_i \in L_p$. The *span* of an edge $e = (a_i, a_j)$ in layering \mathbf{L} is given by $s(e, \mathbf{L}) = l(a_i, \mathbf{L}) - l(a_j, \mathbf{L})$. For an edge $e \in \vec{\mathcal{E}}$, if $s(e, \mathbf{L}) = 1$, the edge is called a *tight edge* and if $s(e, \mathbf{L}) > 1$, then the edge is called a *long edge*. A layering \mathbf{L} is *proper* if all edges of $\vec{\mathcal{G}}$ are tight. In a digraph $\vec{\mathcal{G}}$ with proper layering \mathbf{L} , a layer with index $l(a_i, \mathbf{L}) - 1$ is called the *leading layer* of a_i .

7.2 Modeling assumptions

In this section, we state the assumptions made in [MJC⁺18] for development of the model. When drivers do not adhere to lane discipline, they avoid collision based on visual feedback about the positions and velocities of other vehicles. It is assumed that the driver control in longitudinal (along the road traffic or Y) direction and lateral (perpendicular to the road traffic or X) direction, are based on visual feedback from different sets of neighboring vehicles. Hence the influence graphs used in Y and X directional models are different. Moreover it is assumed that the drivers can accelerate independently in the two orthogonal directions. These simplifying assumptions allow us to write separate decoupled models for Y and X directional motions.

7.2.1 Assumptions for longitudinal (Y) direction

In this subsection, we review the model developed in [MJC⁺18] for the driver-neighboring vehicle interaction for motion in the longitudinal direction.

The Pseudo-Leader

Consider n (possibly heterogeneous) vehicles traveling along a multi-lane road. Of these, assume k vehicles belong to a particular homogeneous set \mathcal{M} (e.g., passenger cars), whereas

the remaining $n - k$ vehicles consists of all vehicles not belonging to the class \mathcal{M} . The motion of the vehicles in the set \mathcal{M} is modeled. The Y position of the i^{th} vehicle is denoted by y_i and the velocity by v_{yi} . For any such set \mathcal{M} , the drivers prefer a particular speed based on various factors such as congestion, road surface conditions, safety factors, etc. This preferred longitudinal speed is modeled here by a fictitious vehicle (a_0^y) leading the modeled convoy with fixed Y -velocity. Denote $\mathcal{M} = \{0, 1, 2, \dots, k\}$, and $\bar{\mathcal{M}} = \{k + 1, \dots, n\}$ and let $y = [y_0 \ y_1 \ \dots \ y_k]^T \in \mathbb{R}^{k+1}$ and $v_y = [v_{y0} \ v_{y1} \ \dots \ v_{yk}] \in \mathbb{R}^{k+1}$ represent the absolute Y -positions and Y -velocities of the $k + 1$ modeled (\mathcal{M} set) vehicles (including a_0^y as leader) respectively. Whereas the position and velocity vectors for the unmodeled ($\bar{\mathcal{M}}$ set) vehicles are denoted by $\bar{y} = [y_{k+1} \ y_{k+2} \ \dots \ y_n]^T \in \mathbb{R}^{n-k}$ and $\bar{v}_y = [v_{y(k+1)} \ v_{y(k+2)} \ \dots \ v_{yn}] \in \mathbb{R}^{n-k}$.

Note: The reason for segregation of vehicles into two sets \mathcal{M} and $\bar{\mathcal{M}}$ will be made clear in Chapter 8. The notation is introduced here for the sake of completeness.

The Influence Graph

Each driver is considered to have a fixed viewing angle ($aov < 180^\circ$), with the visibility cone aligned with the Y -axis. Only those vehicles that are within this viewing angle of the driver can affect the vehicle (see Figure 7.1). The vehicles in this viewing angle are considered as *visible* vehicles. A digraph $\mathcal{G}_{cone}^y = (\mathcal{A}, \vec{\mathcal{E}}_{cone}^y, w)$ with a set of nodes \mathcal{A} and a set of edges $\vec{\mathcal{E}}_{cone}^y$ is used to represent the *communication graph* between the vehicles. Each node $a_i \in \mathcal{A}$ represents a vehicle. With a slight abuse of notation, $\mathcal{A} = \mathcal{M} \cup \bar{\mathcal{M}}$. If a_j is *visible* from a_i ($i \neq j$), then there exists an edge from a_j to a_i i.e., $(a_j, a_i) \in \vec{\mathcal{E}}_{cone}^y$. The set of vehicles visible from a_i is represented by $\mathcal{V}_i := \{a_j : (a_j, a_i) \in \vec{\mathcal{E}}_{cone}^y\}$. The connections from all the vehicles in \mathcal{V}_i are shown by thick arrows (both solid and dashed) in Figure 7.1.

Dense Traffic - Homogeneous Behavior with Proper Layered Influence

In congested conditions, all vehicles (irrespective of their capabilities) are compelled to behave homogeneously due to lack of available maneuvering space. Hence by the formulation presented above, in such cases, $|\mathcal{M}| = n + 1$ (including the pseudo leader) and $|\bar{\mathcal{M}}| = 0$. In such dense traffic, it is assumed that drivers do not get influenced by all the vehicles in \mathcal{V}_i . In fact they tend to track only the vehicles in the immediate vicinity and ignore the (visible) vehicles that are not in the leading *layer* (defined later) of vehicles. The vehicles that are not ignored by a_i are called the *neighbors* of a_i and their set is denoted by $\mathcal{N}_i \subseteq \mathcal{V}_i$. For example, in the group of 6 vehicles shown in Figure 7.1a, vehicle a_3 ignores a_0 even though $a_0 \in \mathcal{V}_i$. This is because, a_3 typically considers a_1 and a_2 to be immediately important: both, for avoiding possible collisions as well as for its relative positioning and velocity. The ignored edges are shown by thick dashed arrows. On the other hand, if a_3 moves relatively ahead in Y -direction, and/or a_1 and a_2 move laterally, the cones of influence are shifted to the situation shown in Figure 7.1b. Here a_1 and a_2 are outside the cone of influence for a_3 and hence a_3 is only influenced by a_0 . Clearly, in both situations a_4 and a_5 being outside the cone of a_3 , do not influence a_3 . The graph

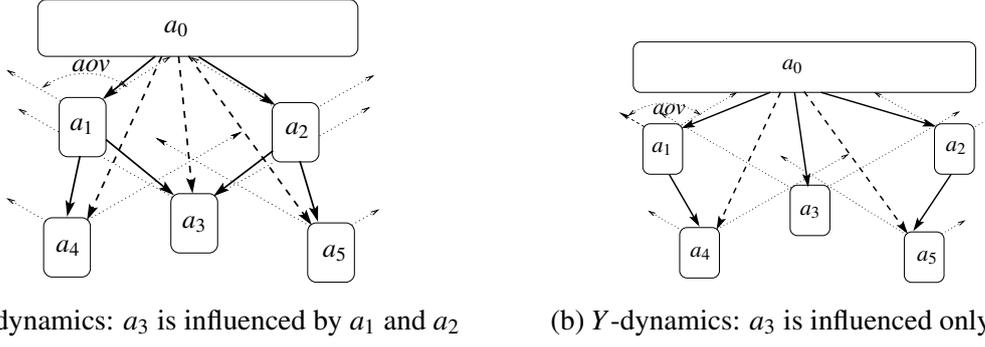


Figure 7.1: Different influence graphs for Y direction depending on the angle of view (aov) of the vehicles

that contains edges only from \mathcal{N}_i to a_i is referred as the Y -influence graph and is denoted by $\vec{\mathcal{G}}^y = (\mathcal{A}, \vec{\mathcal{E}}^y, w)$, where w denotes the set of relative importances (weights) the drivers attach to different neighbors. This importance may be based on the various factors like inter-vehicle distances, relative speeds, drivers objectives/intentions etc.

While the influence graph forms and updates itself automatically in traffic, for modeling as well as simulation purposes a rigorous definition of $\vec{\mathcal{G}}^y$ is needed. The following Algorithm 7.2.1 is proposed in [MJC⁺18] to define the influence graph and its levels/layers.

Algorithm 7.2.1 Extracting $\vec{\mathcal{G}}^y$ from $\vec{\mathcal{G}}_{cone}^y$

Assumption: $\vec{\mathcal{G}}_{cone}^y$ contains a directed spanning tree rooted at a_0^y

1. a_0^y is the leader node.
 2. Number other vehicles as per their Y -coordinates, i.e., for two vehicles a_i and a_j , $i < j$ if $y_i \geq y_j$. This implies $y_0 > y_1 \geq y_2 \geq \dots \geq y_n > 0$.
 3. For vehicle a_k , $k \in \{1, \dots, n\}$
 - (a) Calculate maximum path length l from a_0^y
 - (b) Assign Level L_l for a_k
 4. Remove all long edges from $\vec{\mathcal{E}}_{cone}^y$.
-

Lemma 6. If $\vec{\mathcal{G}}_{cone}^y$ contains a directed spanning tree rooted at a_0^y , Algorithm 1 generates a proper layered graph $\vec{\mathcal{G}}^y$ with layers L_l , $l = 0, 1, 2, \dots, m$.

Remark 7. Observe that, due to the assumption on viewing angle i.e., ($aov < 180^\circ$), and since all visibility cones are aligned along the Y -axis, the influence graph $\vec{\mathcal{G}}_{cone}^y$ and hence $\vec{\mathcal{G}}^y$ is acyclic.

7.2.2 Assumptions for lateral (X) direction

Consider the n vehicles moving along a road. The lateral maneuver of a vehicle depends on the other vehicles in the vicinity and the boundaries of the road. As before the set \mathcal{M} denotes the set of vehicles with modeled X motion, while $\bar{\mathcal{M}}$ denotes the unmodeled vehicles. For most practical purposes (as in our experiments) both the Y and X motions are modeled for the same set of vehicles i.e., except for the pseudo-vehicles, the set \mathcal{M} and $\bar{\mathcal{M}}$ are identical for both the Y and X motion.

Road Boundaries

Let the X position of the i^{th} vehicle be denoted by x_i and the velocity by v_{xi} . First, the boundaries of the road are modeled as two infinitely long pseudo vehicles a_0^x and a_{n+1}^x having fixed X positions (This assumption can be easily modified for road curves, narrowing or widening roads). Denote $\mathcal{M} = \{0, 1, 2, \dots, k, n+1\}$, and $\bar{\mathcal{M}} = \{k+1, \dots, n\}$ and let $x = [x_0 \ x_1 \ \dots \ x_k \ x_{n+1}]^T \in \mathbb{R}^{k+2}$ and $v_x = [v_{x0} \ v_{x1} \ \dots \ v_{xk} \ v_{x(n+1)}] \in \mathbb{R}^{k+2}$ represent the absolute X -positions and X -velocities of the $k+2$ modeled (set \mathcal{M}) vehicles (including a_0^x and a_{n+1}^x as the road boundaries) respectively. Whereas the position and velocity vectors for the unmodeled (set $\bar{\mathcal{M}}$) vehicles are denoted by $\bar{x} = [x_{k+1} \ x_{k+2} \ \dots \ x_n]^T \in \mathbb{R}^{n-k}$ and $\bar{v}_x = [v_{x(k+1)} \ v_{x(k+2)} \ \dots \ v_{xn}] \in \mathbb{R}^{n-k}$. The lateral velocities of a_0^x and a_{n+1}^x are $v_{x0} = v_{x(n+1)} = 0$ and their positions are $x_0 = 0$ and $x_{n+1} = d$, where d is the width of the road.

Bidirectional Influence

Unlike longitudinal motion, the lateral motion of a vehicle depends on the other vehicles present both to the left and to the right. That is, a driver has to track other vehicles on both sides of the vehicle and therefore, the angle of viewing (aov) in lateral direction is modeled as two sideways symmetric cones as shown in Figure 7.2. Due to this assumption, the edges in the lateral information exchange graph become bidirectional. Note, however, that the edges starting from a_0^x and a_{n+1}^x are always unidirectional since the road boundaries are not influenced by other vehicles. For a vehicle a_i , there are two associated cones of visions, one on the left and other on the right. Let $\vec{\mathcal{G}}_{cone}^x = (\mathcal{V}, \vec{\mathcal{O}}_{cone}^x, w)$ represent the communication graph between the vehicles. Unlike $\vec{\mathcal{G}}_{cone}^y$, the lateral graph $\vec{\mathcal{G}}_{cone}^x$ has two leaders, a_0^x and a_{n+1}^x . Hence, $\vec{\mathcal{G}}_{cone}^x$ is constructed as a union of two graphs $\vec{\mathcal{G}}_{lcone}^x$ and $\vec{\mathcal{G}}_{rcone}^x$. The graph $\vec{\mathcal{G}}_{lcone}^x$ ($\vec{\mathcal{G}}_{rcone}^x$) is formed similarly to $\vec{\mathcal{G}}_{cone}^y$ by considering a_0^x (a_{n+1}^x) as the leader and only left (right) aov of vehicles and ignoring a_{n+1}^x (a_0^x). Using these definitions, two sets of visible vehicles are defined for each a_i , $\mathcal{V}_i^{left} := \{a_j : (a_j, a_i) \in \vec{\mathcal{O}}_{lcone}^x \text{ i.e., } x_j < x_i\}$ and $\mathcal{V}_i^{right} := \{a_j : (a_j, a_i) \in \vec{\mathcal{O}}_{rcone}^x \text{ i.e., } x_j > x_i\}$.

Dense Traffic: Homogeneous Behavior

For congested conditions, homogeneity in driver behavior is assumed according to the logic introduced for the Y direction. Consequently $|\mathcal{M}| = n+2$ (including the road boundaries) and

$|\bar{\mathcal{N}}| = 0$. As above, the neighbors are selected according to the immediate neighborhood of visible vehicles and the resulting influence graphs are assumed to be time invariant. The lateral time-invariant influence graph $\vec{\mathcal{G}}^x$ is the union of two subgraphs $\vec{\mathcal{G}}_{left}^x$ and $\vec{\mathcal{G}}_{right}^x$. These subgraphs are obtained using an algorithm similar to Algorithm 7.2.1 on $\vec{\mathcal{G}}_{lcone}^x$ and $\vec{\mathcal{G}}_{rcone}^x$ respectively. The set of left (right) neighbors \mathcal{N}_i^{left} (\mathcal{N}_i^{right}) of a vehicle a_i is the set of nodes a_j of $\vec{\mathcal{G}}_{left}^x$ ($\vec{\mathcal{G}}_{right}^x$) such that $(a_j, a_i) \in \vec{\mathcal{E}}_{left}^x$ ($(a_j, a_i) \in \vec{\mathcal{E}}_{right}^x$). Observe that this construction does not change the uni/bi-directional nature of the edges of $\vec{\mathcal{G}}^x$. When there are no vehicles in the *aov* of vehicle a_i , the corresponding (either left or right) road boundary influences a_i . An example of lateral influence graph is shown in Figure 7.2.

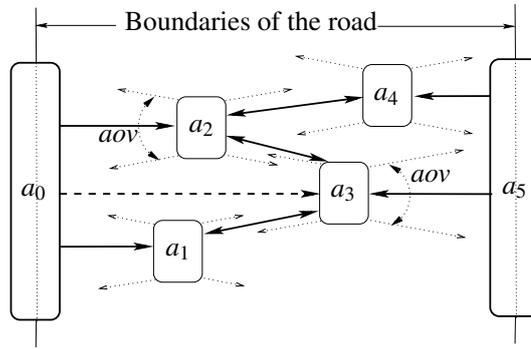


Figure 7.2: Influence graph for X direction depending on the angle of view (*aov*) of the vehicles

7.3 Homogeneous Traffic Model: Fixed Graph

In this section it is assumed that the influence graphs $\vec{\mathcal{G}}^y$ and $\vec{\mathcal{G}}^x$, do not change with time. Moreover, all vehicles under consideration are assumed to behave homogeneously. A model of driver behavior under these assumptions is reviewed in this section.

7.3.1 Driver model for longitudinal direction

Consider the convoy of n vehicles. Recall that node a_0^y , the fictitious leader, moves at a fixed velocity (say v_{y0}). Let the dynamics of the leader a_0^y be given by $\dot{y}_0 = v_{y0}$ and $\dot{v}_{y0} = 0$. In practical scenarios, each driver targets a inter-vehicle spacing based on two factors [Liu06, Chapter 2]:

1. The distance needed to decelerate to zero, which is proportional to the vehicle's current velocity and is given by $k_i v_i$. Since all the vehicles are assumed to be identical, it is considered that $k_i = \tilde{k}$ for $i = 1, \dots, n$ where $\tilde{k} > 0$ is constant.
2. The desired constant spacing, g_y , during stopped traffic.

Hence, the Y -direction control law for vehicle a_i , $i = 1, \dots, n$ is

$$\begin{aligned} \dot{y}_i &= v_{yi} \\ \dot{v}_{yi} &= \sum_{j \in \mathcal{N}_i} - \left(b_y w_{ij} (v_{yj} - v_{yi}) - k_y \left(w_{ij} (y_j - y_i) + \frac{1}{|\mathcal{N}_i|} (g_y - \tilde{k} v_i) \right) \right) \end{aligned} \quad (7.1)$$

Recall that, \mathcal{N}_i is the set of neighbors of a_i and w_{ij} are the edge weights. The constants b_y and k_y denote the influence of relative velocities and spacings between the vehicles respectively on acceleration.

The closed loop system then can be written as follows:

$$\begin{bmatrix} \dot{y} \\ \dot{v}_y \end{bmatrix} = \begin{bmatrix} \mathbf{0}_{(n+1) \times (n+1)} & I_{(n+1) \times (n+1)} \\ -k_y \mathcal{L}^y & -b_y \mathcal{L}^y - k_y \tilde{k} \tilde{I} \end{bmatrix} \begin{bmatrix} y \\ v_y \end{bmatrix} - k_y g_y \begin{bmatrix} \mathbf{0}_{(n+2) \times 1} \\ \mathbf{1}_{n \times 1} \end{bmatrix}, \quad (7.2)$$

where $\mathbf{1}_{p \times q}$ and $\mathbf{0}_{p \times q}$ are $p \times q$ matrices with all 1s and 0s respectively. The matrix \tilde{I} is obtained by setting the 1st diagonal entry of $I_{(n+1) \times (n+1)}$ to 0. \mathcal{L}^y denotes the weighted Laplacian of the directed influence graph $\vec{\mathcal{G}}^y$ with entries $l_{ij} = -w_{ij}$ and $l_{ii} = \sum_{j=1}^n w_{ij} = 1$.

7.3.2 Analysis of longitudinal dynamics

As mentioned in Remark 7, $\vec{\mathcal{G}}^y$ is acyclic and hence the corresponding Laplacian (\mathcal{L}^y) has a lower triangular structure [Bap10] with $l_{ii} = 1$ as the diagonal entries. That is, \mathcal{L}^y has one 0 eigenvalue and all its other eigenvalues are 1. Similarly, \tilde{I} has one 0 eigenvalue and the remaining eigenvalues are 1.

Let $\Gamma = \begin{bmatrix} \mathbf{0}_{(n+1) \times (n+1)} & I_{(n+1) \times (n+1)} \\ -k_y \mathcal{L}^y & -b_y \mathcal{L}^y - k_y \tilde{k} \tilde{I} \end{bmatrix}$. As $k_y, b_y, \tilde{k} > 0$, the rank of Γ is $2n + 1$. The eigenvalues of Γ are

$$\lambda_{i\pm} = \frac{(b_y \mu_i - k_y \tilde{k} v_i) \pm \sqrt{(b_y \mu_i - k_y \tilde{k} v_i)^2 + 4k_y \mu_i}}{2},$$

where μ_i and v_i are the eigenvalues of \mathcal{L}^y and \tilde{I} respectively. Observe that both \mathcal{L}^y and \tilde{I} have a zero row corresponding to the leader a_0 . Due to the structure of \mathcal{L}^y and \tilde{I} , we can write $\mu_0 = v_0 = 0$. Hence $\lambda_{0\pm} = 0$. For $i = 1, \dots, n$, the eigenvalues $\lambda_{i\pm}$ have negative real part.

Theorem 8. [MJC⁺18] Consider a time invariant, connected, proper, layered graph obtained using Algorithm 7.2.1. Assume that the total weight across all incoming edges is 1 for each vehicle except a_0 i.e., $W_i = W_j (= 1)$ for all $i, j \in \{1, \dots, n\}$. If $\tilde{k} > 0$ and the velocity v_{y0} of a_0 is constant, then:

1. $v_{yi} \rightarrow v_{y0}$ as $t \rightarrow \infty \forall i \in \{1, \dots, n\}$
2. $|y_i(t) - y_j(t)| \rightarrow 0$ as $t \rightarrow \infty$ for all $a_i, a_j \in L_k, k = 1, 2, \dots, m$.
3. For any two vehicles a_i and a_j such that $a_i \in L_{k-1}$ and $a_j \in L_k, k = 1, 2, \dots, m$, as $t \rightarrow \infty$, $|y_i(t) - y_j(t)| \rightarrow g_y + \tilde{k} v_{y0}$.

7.3.3 Driver model for lateral direction

Consider the convoy of n vehicles along with two fictitious vehicles representing the boundaries of the road. The dynamics of the fictitious vehicles representing the road boundary is given by $\dot{x}_k = v_{xk} = 0$ and $\dot{v}_{xk} = 0$, for $k = 0, n+1$. Let the X -position of a_0^x is $x_0 = 0$, while the X -position of a_{n+1} is $x_{n+1} = d$ where $d > 0$ is the width of the road.

It is assumed that the objective of each vehicle is to locally adjust the distance from the influencing vehicles based on the width of the road and traffic density. Specifically, each vehicle tries to go to the midpoint of the weighted average of X -positions of its left and right neighbors. This assumption models the tendency of the drivers to maintain the safe distance from vehicles both on the left and the right. The proposed model dynamics for a_i in X -direction is

$$\begin{aligned} \dot{x}_i &= v_{xi} \\ \dot{v}_{xi} &= \sum_{j \in \mathcal{N}_i} \{b_x w_{ij}(v_{xj} - v_{xi}) + k_x w_{ij}(x_j - x_i)\}, \end{aligned} \quad (7.3)$$

where, $\mathcal{N}_i = \mathcal{N}_i^{left} \cup \mathcal{N}_i^{right}$. The variable w_{ij} denotes the weight given to the link edge (a_j, a_i) for each $a_j \in \mathcal{N}_i$. Similar to the Y -dynamics, the constants b_x and k_x respectively denote the influence of relative velocities and spacings between the vehicles on acceleration.

Without loss of generality, it is assumed that the cumulative weight of all incoming edges at a vehicle is 1. The sets \mathcal{N}_i^{left} and \mathcal{N}_i^{right} denote the set of left and right influencing vehicles of a_i respectively. Moreover it is assumed that the cumulative weights of edges incident on a_i from \mathcal{N}_i^{right} and from \mathcal{N}_i^{left} are the same and equal to $\frac{1}{2}$. This can be interpreted as each driver giving equal importance to left and right maneuvers. Among the vehicles in either \mathcal{N}_i^{right} or \mathcal{N}_i^{left} , the weights are assigned locally by a_i based on the importance given by the driver to the corresponding vehicles. With this weight assignment, the X -dynamics of the convoy can be written as:

$$\begin{bmatrix} \dot{x} \\ \dot{v}_x \end{bmatrix} = \begin{bmatrix} \mathbf{0}_{(n+2) \times (n+2)} & I_{(n+2) \times (n+2)} \\ -k_x \mathcal{L}^x & -b_x \mathcal{L}^x \end{bmatrix} \begin{bmatrix} x \\ v_x \end{bmatrix}, \quad (7.4)$$

where, \mathcal{L}^x is the Laplacian of $\vec{\mathcal{G}}^x$ with entries $l_{ij} = -w_{ij}$ & $l_{ii} = -\sum_{j=1}^n w_{ij} = 1$.

Note that, in $\vec{\mathcal{G}}^x$, there are two leaders a_0^x and a_{n+1}^x i.e., the pseudo boundary vehicles. Hence, there are two zero rows in \mathcal{L}^x . However, as $\vec{\mathcal{G}}^x$ contains an united directed spanning tree, both the algebraic and geometric multiplicity of the zero eigenvalue of \mathcal{L}^x is two [CR09, CV06].

7.3.4 Analysis of lateral dynamics

The following theorem describes the implications of the driver model (7.3).

Theorem 9. [MJC⁺18] Consider a convoy of n vehicles with directed time invariant influence graph $\vec{\mathcal{G}}^x$ and system dynamics (7.3). If the positions and velocities of the boundary vehicles a_0^x and a_{n+1}^x are $x_0 = 0$, $x_{n+1} = d$ and $v_{x0} = v_{x(n+1)} = 0$ respectively, then:

1. $v_{xi} \rightarrow 0$ as $t \rightarrow \infty \forall i \in \{1, \dots, n\}$
2. Position x_i of each vehicle a_i , $i = 1, \dots, n$ converge to the weighted average of the X -positions of its neighbors

7.3.5 Coupling between longitudinal and lateral motion

For the sake of simplicity, the vehicles are assumed to have independent acceleration in the two orthogonal directions. In [MJC⁺18], it is shown that, even when partial coupling between the dynamics of the vehicles in X and Y directions is considered, the results presented above still hold.

A way to introduce this coupling is to observe that, lateral (X -direction) motion decisions are usually based on human behavioral considerations, e.g. due to specific objectives such as obstacles/potholes avoidance, overtaking, future left/right turns etc. Based on these X -directional requirements, the Y -directional acceleration and speed are adjusted so as to safely allow the lateral maneuvers. Hence, during each lateral movement within the formation, the velocity (v_y) along the road direction of the car decreases to account for the horizontal movement. The exact reduction in velocity depends on a multitude of factors like radius of curvature, velocity along the road, the capabilities of the vehicle and skill level of the driver. Note that, irrespective of sign of lateral velocity v_x , the longitudinal acceleration of the vehicle is reduced. For simplicity, it is assumed that, the acceleration (a_y) in Y -direction is linearly dependent on the magnitude of velocity (v_x) in the X -direction. With this notion in mind, the longitudinal dynamics (7.1) can be modified as

$$\begin{aligned} \dot{y}_i &= v_{yi} \\ \dot{v}_{yi} &= - \sum_{j \in \mathcal{N}_i} \left(b_y w_{ij} (v_{yj} - v_{yi}) + k_y \left(w_{ij} (y_j - y_i) + \frac{1}{|\mathcal{N}_i|} g_y \right) \right) - m \text{sign}(v_{xi}) v_{xi} \end{aligned}$$

for some $m > 0$. The dynamics of α_0^y , α_0^x and α_{n+1}^x are unchanged. Define a $(n+1) \times (n+2)$ matrix M_{v_x} such that

$$\begin{aligned} M_{v_x}(i, i) &= m \text{sign}(v_{xi}) \text{ for } i = 2, \dots, n+1 \\ &= 0 \quad \text{for } i = 1 \\ M_{v_x}(i, j) &= 0 \quad \text{for } i \neq j \end{aligned}$$

The collective dynamics in both Y and X direction given by (7.2) and (7.4), thus can be rewritten as follows:

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{v}_x \\ \dot{v}_y \end{bmatrix} = \Gamma_c \begin{bmatrix} x \\ y \\ v_x \\ v_y \end{bmatrix} - k_y g_y \begin{bmatrix} \mathbf{0}_{n \times 2} \\ \mathbf{0}_{n \times 1} \\ \mathbf{0}_{n \times 2} \\ \mathbf{1}_{n \times 1} \end{bmatrix} \quad (7.5)$$

where

$$\Gamma_c = \left[\begin{array}{cc|cc} \mathbf{0}_{(n+2) \times (n+2)} & \mathbf{0}_{(n+2) \times (n+1)} & I_{(n+2) \times (n+2)} & \mathbf{0}_{(n+2) \times (n+1)} \\ \mathbf{0}_{(n+1) \times (n+2)} & \mathbf{0}_{(n+1) \times (n+1)} & \mathbf{0}_{(n+1) \times (n+2)} & I_{(n+1) \times (n+1)} \\ \hline -k_x \mathcal{L}^x & \mathbf{0}_{(n+2) \times (n+1)} & -b_x \mathcal{L}^x & \mathbf{0}_{(n+2) \times (n+1)} \\ \mathbf{0}_{(n+1) \times (n+1)} & -k_y \mathcal{L}^y & -M_{v_x} & -b_y \mathcal{L}^y \end{array} \right].$$

Since the sign of elements of M_{v_x} switches with the sign of state component vector v_x , the system in (7.5) is a state (v_x) dependent switching system. Recall that when the influence graphs are connected, \mathcal{L}^y has one eigenvalue at 0 while \mathcal{L}^x has two 0 eigenvalues with geometric multiplicity 2. Also, the non-zero eigenvalues of both \mathcal{L}^y and \mathcal{L}^x are real and positive.

The main result determining the spacing of the vehicles as $t \rightarrow \infty$ can be formally stated as:

Theorem 10. [MJC⁺18] Consider a convoy of n vehicles with three pseudo vehicles a_0^y , a_0^x and a_{n+1}^x representing the virtual leader in Y -direction, left and right road boundaries respectively. Let the influence graphs in X and Y direction, $\vec{\mathcal{G}}^x$ and $\vec{\mathcal{G}}^y$ respectively, be time invariant with $W = 1$. Assume that $\vec{\mathcal{G}}^y$ is connected proper layered graph, $\vec{\mathcal{G}}^x$ has a united directed tree rooted at a_0^x and a_{n+1}^x . If v_{y0} , x_0 and x_{n+1} are fixed, then for the closed loop system (7.5),

1. $v_{yi} \rightarrow v_{y0}$ as $t \rightarrow \infty \forall i \in \{1, \dots, n\}$
2. $|y_i(t) - y_j(t)| \rightarrow 0$ as $t \rightarrow \infty$ for all $a_i, a_j \in L_k$, $k = 1, 2, \dots, m$.
3. For any two vehicles a_i and a_j such that $a_i \in L_{k-1}$ and $a_j \in L_k$, $k = 1, 2, \dots, m$, as $t \rightarrow \infty$, $|y_i(t) - y_j(t)| \rightarrow g_y$.
4. $v_{xi} \rightarrow 0$ as $t \rightarrow \infty \forall i \in \{1, \dots, n\}$
5. Position x_i of each vehicle a_i $i = 1, \dots, n$ converge to the weighted average of positions of its neighbors in $\vec{\mathcal{G}}^x$.

This analysis shows that, this dependence of longitudinal acceleration on lateral dynamics of the vehicles does not alter the properties of the convoy dynamics significantly and the results still hold.

Chapter 8

Proposed traffic model and verification

The theory proposed in [MJC⁺18] (and reviewed in Chapter 7) assumes that the traffic is homogeneous, i.e. the vehicles under consideration have identical capabilities and driving behavior. This assumption holds when the roads are congested. However, this assumption needs to be modified in order to account for observations made for sparse traffic conditions. In this chapter, a model for heterogeneous traffic is presented in Section 8.2. Then, the traffic models presented in Chapter 7 and Section 8.2, are validated in Section 8.4.

8.1 Modeling assumptions for heterogeneous traffic

In this section we introduce the assumptions made to model heterogeneous traffic. The assumptions made here are an extension of those outlined in Section 7.2.

8.1.1 Assumptions for longitudinal (Y) direction

In India (and in our experiments) any fleet of vehicular traffic typically includes extremely disparate vehicles (e.g., a typical truck in India has maximum safe velocity of 40-60 km/hr while most cars range between 80-120 km/hr on the average). In uncongested traffic situations, available road space enables the disparity between the vehicles to affect driver behavior. Such heterogeneous capabilities lead to large changes in inter-vehicle distances thereby resulting in frequently changing influence graphs between the vehicles. For application of the model equations, any homogeneous set (denoted by \mathcal{M}) can be selected from the fleet (e.g., passenger cars) and all other types of vehicles (e.g., heavy vehicles, motor cycles, scooters etc.) in the fleet are denoted by $\bar{\mathcal{M}}$. For simplicity, consider the collection at an arbitrary but fixed instant of time. Assume that each driver in the set \mathcal{M} gets influenced by vehicles (both from sets \mathcal{M} and $\bar{\mathcal{M}}$) in his *visibility cone*. However, the drivers in the set $\bar{\mathcal{M}}$, follow unknown autonomous trajectories based on unmodeled influences, and have no incoming edges from other vehicles. As described above, the visibility graph is denoted by $\vec{\mathcal{G}}_{cone}^y = (\mathcal{A}, \vec{\mathcal{E}}_{cone}^y, w)$. Among the edges in $\vec{\mathcal{E}}_{cone}^y$, all the edges incident on the set of $\bar{\mathcal{M}}$ vehicles are ignored. In addition, some of the

drivers in the set \mathcal{M} vehicles can also ignore some edges incident on itself from $\vec{\mathcal{E}}_{cone}^y$ for various reasons (e.g., the influencing vehicle is visible but too far away). After deleting all such edges the *neighbors* of $a_i \in \mathcal{M}$ are denoted by the set $\mathcal{N}_i \subseteq \mathcal{V}_i$ and the corresponding Y -influence graph is denoted by $\vec{\mathcal{G}}^y = (\mathcal{A}, \vec{\mathcal{E}}^y, w)$. Let \mathcal{M}_i denote the set of neighbors of vehicle a_i which are modeled and $\bar{\mathcal{M}}_i$ denote the set of neighbors which are unmodeled. Note that $\mathcal{A} = \mathcal{M} \cup \bar{\mathcal{M}}$ and $\mathcal{N}_i = \mathcal{M}_i \cup \bar{\mathcal{M}}_i$. Clearly this notation must be modified to denote the effects of changing influence graphs. This is done later in Section 8.2 along with the driver model.

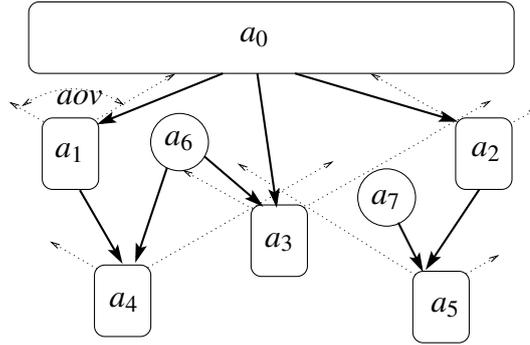


Figure 8.1: Influence graph for Y direction with unmodeled vehicles

8.1.2 Assumptions for lateral (X) direction

The assumptions are similar to those in Y -direction model for sparse traffic. Briefly, the incoming edges to the set of $\bar{\mathcal{M}}$ vehicles are deleted from $\vec{\mathcal{E}}_{cone}^x$ and a subset of the remaining edges are selected based on driver preferences to create $\vec{\mathcal{G}}^x$. Various real life situations such as stationary obstacles, road narrowing etc. can be modeled under this framework. Clearly, $\vec{\mathcal{G}}^x$ is time varying.

8.2 Heterogeneous Traffic Model: Time Varying Graph

When traffic is heterogeneous and sparse, or when there are obstacles on the road, the influence graph changes with time. For example, consider a scenario with 5 vehicles, having some fixed viewing angles. Let the influence graphs be as shown in Figure 8.2, with the edges of Y -graph denoted by solid arrows and the edges of the X -graph denoted by dashed arrows. In Y -direction, only a_0^y affects the dynamics of a_5 . Hence, according to (7.1), a_5 moves relatively ahead causing changes in the X -influence graph as shown in Figure 8.3a. Due to this change in the influence graphs, the dynamics of the vehicles according to the proposed model causes, in turn, changes in the Y -influence graphs as shown in Figure 8.3b. In addition to such autonomous graph change scenarios, influence graphs can also change due to obstacles (permanent and temporary), rogue driving, overtaking, slower/faster vehicles and road conditions.

Clearly, these changes result in the systems given by (7.2) and (7.4) becoming switched systems. It is quite reasonable to assume that changes in the influence graph do not occur

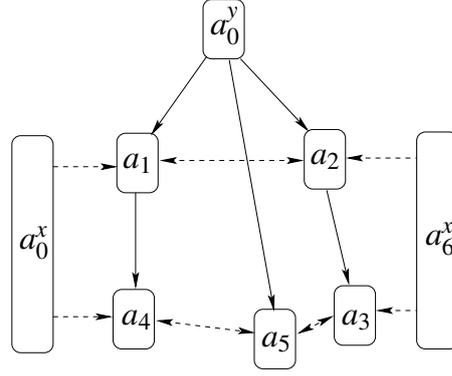


Figure 8.2: The convoy of vehicles with the influence graphs

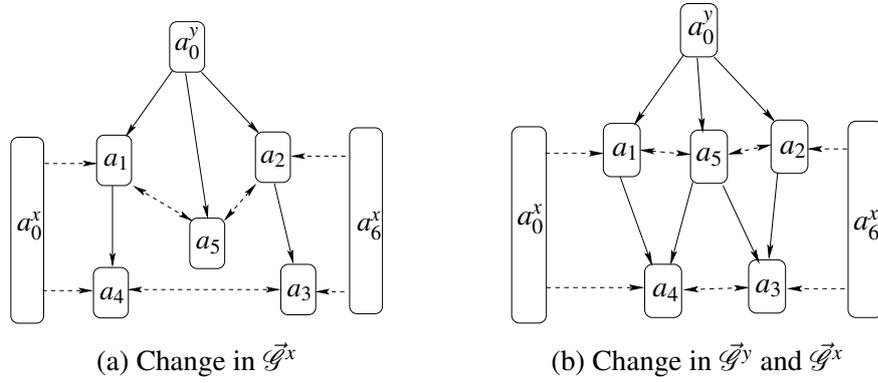


Figure 8.3: Switching influence graphs

arbitrarily fast. The influence graph between vehicles at each instant of time corresponding to the value of the switching function $\sigma(t)$ is denoted by $\mathcal{G}_\sigma^y = (\mathcal{A}, \mathcal{E}_\sigma^y, w_\sigma)$ (slightly modifying the notation of Section 7.1). Recall that, the set of neighbors of the i^{th} vehicles corresponding to $\sigma(t)$ is denoted by $\mathcal{N}_i^\sigma = \mathcal{M}_i^\sigma \cup \bar{\mathcal{M}}_i^\sigma$. Further for simplicity of presentation, we modify our target inter-vehicle distance to be a constant, rather than dependent on velocity as in Section 7.3.1. Then, the Y -direction equations for the modeled vehicles a_i , $i = 1, \dots, k$ in the set \mathcal{M} are

$$\begin{aligned}
 \dot{y}_i &= v_{yi} \\
 \dot{v}_{yi} &= \sum_{j \in \mathcal{N}_i^\sigma} \left(b_y w_{ij}^\sigma (v_{yj} - v_{yi}) + k_y \left(w_{ij}^\sigma (y_j - y_i) + \frac{1}{|\mathcal{N}_i^\sigma|} g_y \right) \right) \\
 &= \sum_{j \in \mathcal{M}_i^\sigma} \left(b_y w_{ij}^\sigma (v_{yj} - v_{yi}) + k_y \left(w_{ij}^\sigma (y_j - y_i) + \frac{1}{|\mathcal{M}_i^\sigma|} g_y \right) \right) \\
 &\quad + \sum_{l \in \bar{\mathcal{M}}_i^\sigma} \left(b_y w_{il}^\sigma (v_{yl} - v_{yi}) + k_y \left(w_{il}^\sigma (y_l - y_i) + \frac{1}{|\bar{\mathcal{M}}_i^\sigma|} g_y \right) \right) \tag{8.1}
 \end{aligned}$$

for $t_i < t \leq t_{i+1}$ and $\sigma(t) \in \mathcal{S}$. Here the dynamics of unmodeled vehicles are not known. However, the position and velocities (v_{yl}, v_l) of the unmodeled vehicles can still be measured by their corresponding neighbors. The constants b_y and k_y denote the influence of relative velocities

and spacings between the vehicles respectively on acceleration. Parameter g_y is proportional to the (fixed, as opposed to velocity dependent as in Section 7.3.1) target inter-vehicle distance that each driver wishes to achieve. This switched version of (7.1) captures the observation that, if due to surrounding influences the set of influencing neighbors change, then the following vehicle has no choice but to try to create the same offset from the new set of influencing vehicles. The combined k -vehicle dynamics can be written as follows:

$$\begin{aligned} \begin{bmatrix} \dot{y}(t) \\ \dot{v}_y(t) \end{bmatrix} &= \begin{bmatrix} \mathbf{0}_{(k+1) \times (k+1)} & I_{(k+1) \times (k+1)} \\ -k_y \mathcal{L}_\sigma^y - k_y \bar{D}_\sigma & -b_y \mathcal{L}_\sigma^y - b_y \bar{D}_\sigma \end{bmatrix} \begin{bmatrix} y(t) \\ v_y(t) \end{bmatrix} - k_y g_y \begin{bmatrix} \mathbf{0}_{(k+2)} \\ \mathbf{1}_k \end{bmatrix} \\ &+ \begin{bmatrix} \mathbf{0}_{(k+1) \times (n-k)} & \mathbf{0}_{(k+1) \times (n-k)} \\ \mathbf{P}_\sigma & \mathbf{Q}_\sigma \end{bmatrix} \begin{bmatrix} \bar{y} \\ \bar{v}_y \end{bmatrix} \end{aligned} \quad (8.2)$$

for $t_i < t \leq t_{i+1}$ and $\sigma(t) \in \mathcal{S}$. Here $\mathcal{L}_\sigma^y \in \mathbb{R}^{(k+1) \times (k+1)}$ is the Laplacian of the influence graph with only the modeled vehicles as vertices. In this Laplacian, the vertices of the influence graph denoting the unmodeled vehicles as well as all edges to and from such nodes are ignored. Further, \bar{D}_σ denotes a diagonal matrix with $\bar{D}_\sigma(i, i) = \sum_{l \in \bar{\mathcal{M}}_i^\sigma} w_{il}^\sigma$ and thirdly, $\mathbf{P}_\sigma, \mathbf{Q}_\sigma \in \mathbb{R}^{(k+1) \times (n-k)}$ together model the interaction between the $k+1$ modeled vehicles and $n-k$

modeled vehicles. The entries of $\mathbf{P}_\sigma \in \mathbb{R}^{(k+1) \times (n-k)}$, are given by $p_{ij}^\sigma = \begin{cases} -b_y w_{ij}^\sigma & \text{if } j \in \bar{\mathcal{M}} \\ 0 & \text{if } j \notin \bar{\mathcal{M}} \end{cases}$, while the entries of $\mathbf{Q}_\sigma \in \mathbb{R}^{(k+1) \times (n-k)}$, are $q_{ij}^\sigma = \begin{cases} -k_y w_{ij}^\sigma & \text{if } j \in \bar{\mathcal{M}} \\ 0 & \text{if } j \notin \bar{\mathcal{M}} \end{cases}$.

Define $y^{ji} := y_i - y_j$ and $v_y^{ji} := v_{yi} - v_{yj}$. For the modeled vehicles, define

$$\bar{y} = \begin{bmatrix} y^{10} & y^{20} & \dots & y^{k0} \end{bmatrix}^T \quad \text{and} \quad \bar{v}_y = \begin{bmatrix} v_y^{10} & v_y^{20} & \dots & v_y^{k0} \end{bmatrix}^T.$$

Similarly, for the unmodeled vehicles let

$$\bar{\bar{y}} = \begin{bmatrix} y^{(k+1)0} & y^{(k+2)0} & \dots & y^{n0} \end{bmatrix}^T \quad \text{and} \quad \bar{\bar{v}}_y = \begin{bmatrix} v_y^{(k+1)0} & v_y^{(k+2)0} & \dots & v_y^{n0} \end{bmatrix}^T.$$

This translates (8.2) to

$$\begin{aligned} \begin{bmatrix} \dot{\bar{y}} \\ \dot{\bar{v}}_y \end{bmatrix} &= \begin{bmatrix} \mathbf{0}_{k \times k} & I_{k \times k} \\ -k_y \tilde{\mathcal{L}}_\sigma^y - k_y \tilde{D}_\sigma & -b_y \tilde{\mathcal{L}}_\sigma^y - b_y \tilde{D}_\sigma \end{bmatrix} \begin{bmatrix} \bar{y} \\ \bar{v}_y \end{bmatrix} - k_y g_y \begin{bmatrix} \mathbf{0}_k \\ \mathbf{1}_k \end{bmatrix} \\ &+ \begin{bmatrix} \mathbf{0}_{k \times (n-k)} & \mathbf{0}_{k \times (n-k)} \\ \tilde{\mathbf{P}}_\sigma & \tilde{\mathbf{Q}}_\sigma \end{bmatrix} \begin{bmatrix} \bar{\bar{y}} \\ \bar{\bar{v}}_y \end{bmatrix} \end{aligned} \quad (8.3)$$

where $\tilde{\mathcal{L}}_\sigma^y$ and \tilde{D}_σ are obtained by deleting the first row and column of \mathcal{L}_σ^y and \bar{D}_σ respectively, while $\tilde{\mathbf{P}}_\sigma$ and $\tilde{\mathbf{Q}}_\sigma$ are obtained by deleting the first rows of \mathbf{P}_σ and \mathbf{Q}_σ . Let $\tilde{\Gamma}_\sigma^y$ denote the system

matrix $\begin{bmatrix} \mathbf{0}_{k \times k} & I_{k \times k} \\ -k_y \tilde{\mathcal{L}}_\sigma^y - k_y \tilde{D}_\sigma & -b_y \tilde{\mathcal{L}}_\sigma^y - b_y \tilde{D}_\sigma \end{bmatrix}$.

Using results from [DV75], the following statement holds.

Lemma 11. [RB08] Consider system (8.3). For each σ , if the system matrix $\tilde{\Gamma}_\sigma^y$ is stable, then there exists $\alpha_\sigma \geq 0$ and $\chi_\sigma > 0$ such that $\|e^{\tilde{\Gamma}_\sigma^y t}\| \leq e^{\alpha_\sigma - \chi_\sigma t}$ for $t \geq 0$.

Additional assumption about unmodeled vehicles: We assume that the unmodeled vehicles cease to affect the modeled vehicles if the unmodeled vehicles overtake the pseudo-leader. This translates to an additional condition on \mathbf{P}_σ and \mathbf{Q}_σ : p_{ij}^σ (or q_{ij}^σ) = 0 if $y^{j0} = y_0 - y_j > 0$. On the other hand, by our earlier assumptions about forward pointed visibility cones, the effect of unmodeled vehicles which falls behind the modeled collection automatically get ignored. In effect, by this assumption, $\tilde{\mathbf{P}}_\sigma \tilde{y} + \tilde{\mathbf{Q}}_\sigma \tilde{v}_y$ get bounded uniformly and hence the system (8.3) can be considered to have bounded input.

Theorem 12. Consider the closed loop dynamics of the convoy of k vehicles with fictitious leader a_0 and $n - k$ unmodeled vehicles given by (8.3). Let t_0, t_1, \dots be the time instants at which the communication graph \mathcal{G}_σ^y switches. Further, let τ be the dwell time such that $t_{i+1} - t_i \geq \tau$ for all $i \geq 0$. For each σ , assume that \mathcal{G}_σ^y contains a spanning tree rooted at a_0 . Then, for $\tau > \sup_\sigma \left\{ \frac{\alpha_\sigma}{\chi_\sigma} \right\}$, the states of the switched system (8.3) are uniformly bounded.

Proof. Note that, for each σ , \mathcal{G}_σ^y is obtained using Algorithm 7.2.1, without renumbering the vehicles each time. Due to the layered structure of \mathcal{G}_σ^y , there exists a similarity transform of the Laplacian \mathcal{L}_σ^y that has lower triangular structure for all $\sigma(t)$ and all non-zero eigenvalues are equal to 1. Further, as each \mathcal{G}_σ^y contains a spanning tree rooted at a_0 , the first row of \mathcal{L}_σ^y (row corresponding to a_0) is zero. From the properties of \mathcal{L}_σ^y , it is clear that $\tilde{\mathcal{L}}_\sigma^y$ has $\lambda_i(\tilde{\mathcal{L}}_\sigma^y) = 1$ for $i = 1, \dots, n$. Moreover subtracting the term \tilde{D}_σ makes the eigenvalues of the resultant matrix $(-\tilde{\mathcal{L}}_\sigma^y - \tilde{D}_\sigma)$ more negative. Thus, the system matrix $\tilde{\Gamma}_\sigma^y$ is stable for each σ [RB08].

Thus, using Lemma 11, according to [RB08, Theorem 4.17], the autonomous part system (8.3) is globally exponentially stable if $\tau > \sup_\sigma \left\{ \frac{\alpha_\sigma}{\chi_\sigma} \right\}$. Hence (8.3) is BIBO stable and when bounded input $u(t) = -k_y g_y \begin{bmatrix} \mathbf{0}_k \\ \mathbf{1}_k \end{bmatrix} + \begin{bmatrix} \mathbf{0}_{k \times (n-k)} & \mathbf{0}_{k \times (n-k)} \\ \tilde{\mathbf{P}}_\sigma & \tilde{\mathbf{Q}}_\sigma \end{bmatrix} \begin{bmatrix} \tilde{y} \\ \tilde{v}_y \end{bmatrix}$ is used, the states of (8.3) are uniformly bounded for any dwell time $\tau > \sup_\sigma \left\{ \frac{\alpha_\sigma}{\chi_\sigma} \right\}$. \square

Remark 13. Clearly the solution of (8.3) is dependent on the influence graph \mathcal{G}_σ^y and the unmodeled vehicles. Hence the inter vehicle distance and relative velocities keep oscillating while remaining uniformly bounded. This is a common observation on Indian roads, especially in sparse traffic.

Remark 14. The analysis for X -dynamics is similar to that of Y -dynamics with a few modifications.

We shall now validate the proposed lane-less traffic model with real vehicle-level trajectories extracted from videos of typical Indian traffic recorded from the top a high-rise building adjoining a sample road. Vehicle trajectories over a length of approximately 150-230 m of the road were recorded. The road under consideration is 12 m wide. As the average speed of the cars is in the range of 40-50 km/h, the duration of the simulation is restricted to 15 s. Clearly,

an experimental validation performed on a set of cars over a longer road stretch would have increased the confidence in the proposed model. This was however impossible due to infrastructural constraints.

To compensate for the short window length, we repeat the tracking experiment for seven sets of five to six cars each over the 230 m road stretch, thereby characterizing the statistical properties of the error propagation. In all, the trajectories of 38 cars are compared with those generated by the model and RMS prediction errors are computed as functions of time. Cumulative values of prediction errors over time are also calculated to check the performance of the model. The low values of these errors demonstrate the ability of our model to predict trajectories, overtaking and lane changing behavior.

8.3 Calibration of the model

In this section, transformations that convert the observed data (pixel values) from the camera's frame of reference into an orthogonal frame in \mathbb{R}^2 for comparison with the model, are proposed. It is essential that these frames of reference are well-calibrated in order to check the goodness-of-fit of the model. The following frames of reference are involved in the process of image transformation: (1) The camera frame, $\{C\}$: The camera frame is set up as shown in Figure 8.4, where each vehicle a_i under consideration is represented by a pixel with coordinates (u_i^C, v_i^C) in the camera frame $\{C\}$. (2) The local-level frame, $\{L\}$: The local-level frame is set up as shown in Figure 8.5 using satellite images (Google Maps) of the same road segment. Coordinates of vehicle a_i are denoted by (u_i^L, v_i^L) in the local level frame $\{L\}$. It can be fairly assumed that the distance between any points in frame is preserved linearly as seen in Figure 8.5. (3) X - Y frame: This is the simulation frame of reference, where the position of vehicle a_i is denoted by (x_i, y_i) and the unit of distance measurement is meters.

8.3.1 Transformation from camera frame to local level frame

For transforming $(u_i^C, v_i^C) \mapsto (u_i^L, v_i^L)$ for each vehicle $a_i, i = 1, \dots, n$, we use static markers like street-lights and road dividers that are common to both the images; see Figures 8.4 and 8.5. For mapping along the u -axis, static markers in the form of lamp-posts are marked along the central divider D_1 , and the bottom divider D_2 . Let $(u_{j_1}^C, v_{j_1}^C)$, where $j_1 = 1, \dots, m_1$, be the positions of m_1 static markers along D_1 and $(u_{j_2}^C, v_{j_2}^C)$, where $j_2 = 1, \dots, m_2$ be the positions of m_2 static markers along D_2 in $\{C\}$. We fit two polynomials, p_1 and p_2 which map $p_1 : u_{j_1}^C \mapsto u_{j_1}^L$ and $p_2 : u_{j_2}^C \mapsto u_{j_2}^L$ respectively as shown in Figure 8.6. Here $u_{j_1}^L$ and $u_{j_2}^L$ are the u -axis components of the m_1 and m_2 static markers along D_1 and D_2 in $\{L\}$ respectively. Note that the v -axis components of static markers in $\{L\}$ are fixed constants due to the geometry of the image. We can assume that there are no errors in the marking of these common features as they are marked manually. Polynomials, p_1 and p_2 are the best fit between the markers in the least square sense. Next, we use a combination of these polynomials to map the u -axis components

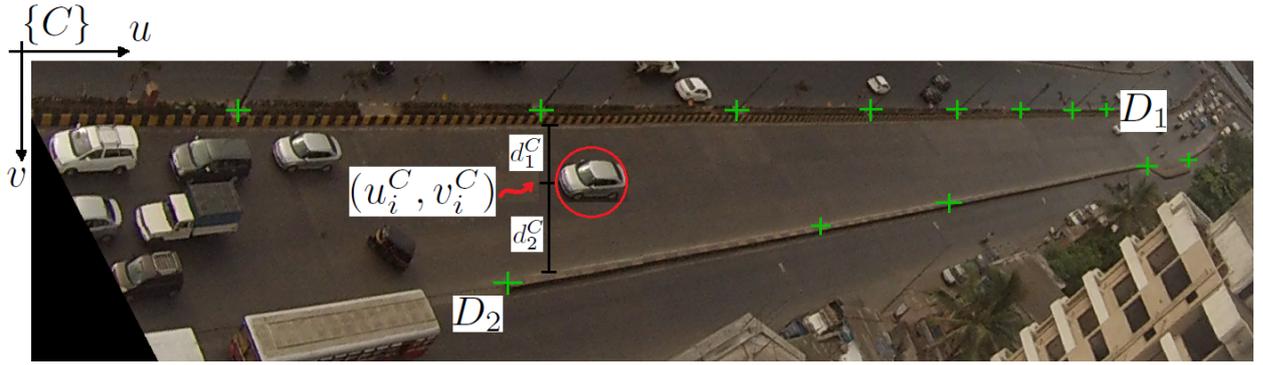


Figure 8.4: The camera frame $\{C\}$

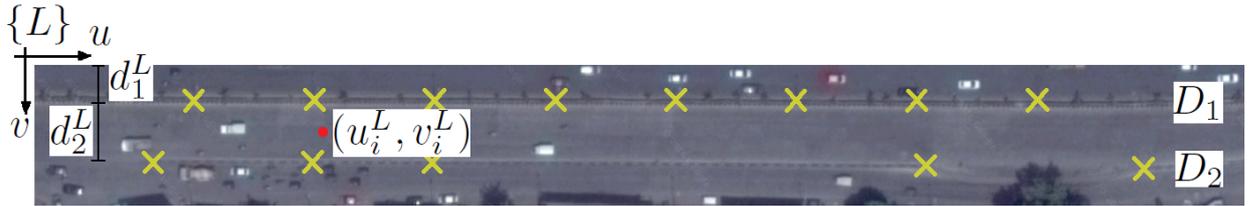


Figure 8.5: The local level frame $\{L\}$:

of the vehicles. For each vehicle a_i with coordinates (u_i^C, v_i^C) , we first calculate its distance d_1^C from D_1 and d_2^C from D_2 . Then the transformation from $\{C\}$ to $\{L\}$ along u -axis is given by $u_i^L = \frac{d_2^C p_1(u_i^C) + d_1^C p_2(u_i^C)}{d_1^C + d_2^C}$.

For transformation along the v -axis, due to the geometry of the image, we use the ratio of the distance of the car from the divider to the total length of the road. Hence, the errors in transformation along the v -axis are negligible. The v -axis component, v_i^L of vehicle a_i in the $\{L\}$ frame is found using the transformation $v_i^L = d_1^L + \left(\frac{d_1^C}{d_1^C + d_2^C}\right) d_2^L$, where d_1^C and d_2^C are as defined earlier, d_1^L is the distance of D_1 from u -axis in the $\{L\}$ frame and d_2^L is the width of the road in the $\{L\}$ frame.

Errors are introduced in this process of transformation from $\{C\}$ to $\{L\}$ due to camera shake. The standard deviation from the base image obtained from the camera is found out to be

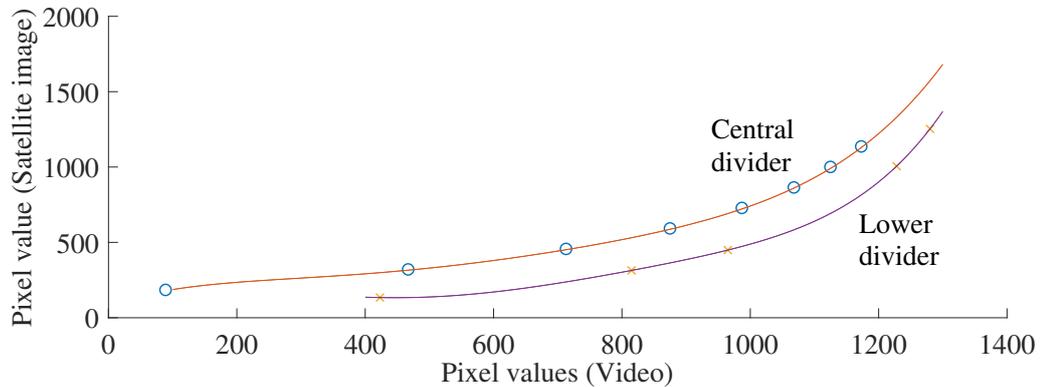


Figure 8.6: Boundary polynomials

10 pixels along the u -axis and 12 pixels along the v -axis.

8.3.2 Conversion to X - Y axes

Let $\kappa = 1, 2, \dots$ be the discrete time instants over which the images are captured. The $\{L\}$ frame coordinates are used to find the actual positions (in meters) of the vehicles $(x_i(\kappa), y_i(\kappa))$ along the X - and Y - axes for comparison with the simulation model. The X -axis component is found using $x_i(\kappa) = (u_i^W(\kappa) - d_{off}) \frac{w_{act}}{w_{img}}$, where, d_{off} is the offset of D_2 , w_{act} is the actual width of the road in meters and w_{img} is the number of pixels corresponding to the width of the road. Similarly, $y_i(\kappa) = (l_{img} - v_i^W(\kappa)) \frac{l_{act}}{l_{img}}$, where l_{act} is the actual length of the road in meters and l_{img} is the number of pixels corresponding to the length of the road. The distances, w_{act} and l_{act} used in these transformation are obtained from Google maps. Conversion from pixels to distances (SI units) is done using measurements off Google maps using the “distance measurement tool”. However, information regarding the accuracy of these measurements are not publicly available from Google. From the base image used for calibration, we calculate that 1 pixel corresponds to 0.2 m. Hence, the camera shake corresponds to ± 2 m error along X -axis and ± 2.4 m error along Y -axis.

8.4 Model Validation

In this section we discuss the tuning of model parameters in order to generate valid vehicle trajectories. The parameters in the model that require tuning are:

1. Weights (w_{ij}) that the driver i of vehicle a_i , assigns to the other vehicles, $j \in \mathcal{N}_i$ in their cone of vision: these are chosen inversely proportional to distance of the other vehicles from the driver of a_i ; vehicles closer to the driver are assigned more weight than the ones farther away. Also, for all time, t , $\sum_{j \in \mathcal{N}_i} w_{ij}(t) = 1$.
2. Influence of inter-vehicle spacing (k_x, k_y) and relative velocities (b_x, b_y) on acceleration: these are chosen manually (by trial and error) such that the average RMS errors are minimized.
3. Safe inter-vehicle distance (g_y): this is chosen as the actual average of the inter-vehicle distance between all neighboring vehicles, as calculated from the real data.

The videos are analyzed at discrete intervals $\kappa = 1, \dots, T_c$, where T_c is the duration of the video clip in each case, c . We present the analysis of 7 cases (named S_c , where $c = 1, \dots, 7$). These seven cases are further classified into two regimes: (1) congested regime (cases S_1, S_2): dense traffic with fixed influence graphs and homogeneous behavior and (2) sparse regime (cases S_3 - S_7): sparse heterogeneous traffic, (time varying influence graphs) with overtaking and lane changing. In all, 38 vehicles are tracked and their coordinates, $(u_i^C(\kappa), v_i^C(\kappa))$ at each discrete time interval κ , are transformed to $(x_i(\kappa), y_i(\kappa))$ along the X and Y directions for comparison

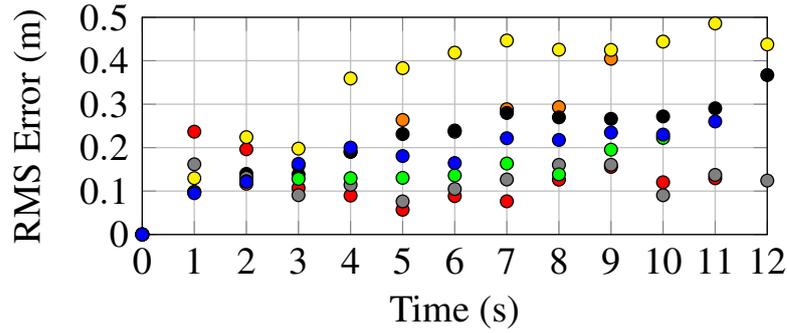


Figure 8.7: Plot of tracking errors (RMS) for the 7 cases along X-axis. Each case is represented by a different colour: S_1 : red, S_2 : purple, S_3 : black, S_4 : yellow, S_5 : grey, S_6 : green, S_7 : blue.

with the proposed model. The model, which is initialized with the initial conditions as obtained from these transformations, simulates the trajectories of the vehicles over the same time interval as the video clip. The output of the simulation is compared with actual data. The RMS errors in prediction at each discrete time interval κ , is calculated as

$$e_{RMS}(\kappa) = \frac{\sqrt{\sum_{i=1}^{n_c} (x_i^s(\kappa) - x_i^o(\kappa))^2}}{n_c} \quad (8.4)$$

where $x_i^s(\kappa)$ is the position of a vehicle a_i at time instant κ as obtained from the simulation and $x_i^o(\kappa)$ is its observed position in the X-Y frame. The total number of cars in case S_c is denoted by n_c , $c = 1, \dots, 7$. Plots of these RMS errors along the lateral and longitudinal directions for cases S_1 - S_7 are given in Figures 8.7 and 8.8. It can be seen from Figures 8.7 and 8.8 that the RMS errors remain within 0.5 m in the lateral direction and 3 m in the longitudinal direction. It is also clear from Figures 8.7 and 8.8 that within the window of 230 m and 15 s, the errors are not diverging. In fact, the errors oscillate within the same bounds or in some cases even “settle” to lower values within the first 10 s.

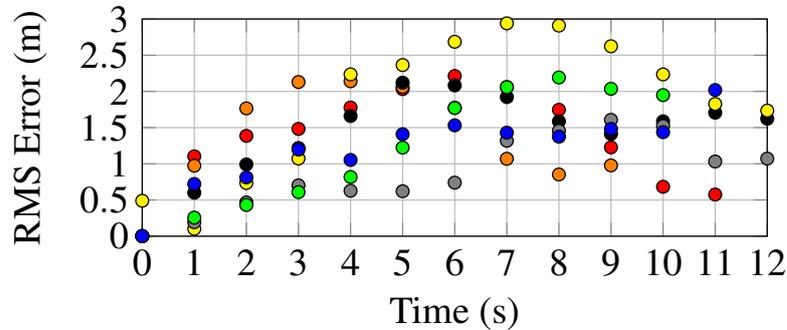


Figure 8.8: Tracking errors (RMS) for the 7 cases along Y-axis. Colours for the cases: S_1 —red, S_2 —orange, S_3 —black, S_4 —yellow, S_5 —grey, S_6 —green, S_7 —blue.

In addition to instantaneous errors, cumulative (over time) errors are also calculated:

- cumulative RMS error

$$e_{RMSC} = \frac{1}{7} \sum_{c=1}^7 \left[\sum_{\kappa=0}^{T_c} \frac{1}{T_c} \sqrt{\frac{\sum_{i=1}^{n_c} (x_i^s(\kappa) - x_i^o(\kappa))^2}{n_c}} \right] \quad (8.5)$$

- average absolute error

$$|e|_{avg} = \frac{1}{7} \sum_{c=1}^7 \left[\sum_{\kappa=0}^{T_c} \frac{1}{T_c} \frac{\sum_{i=1}^{n_c} |x_i^s(\kappa) - x_i^o(\kappa)|}{n_c} \right] \quad (8.6)$$

- maximum absolute error

$$|e|_{max} = \max_{c, 0 \leq \kappa \leq T_c, 1 \leq i \leq n_c} |x_i^s(\kappa) - x_i^o(\kappa)| \quad (8.7)$$

The values of these cumulative errors presented in Table 8.1 and the RMS error plots in Figures 8.7 and 8.8, verify our claims about the validity of our theoretical predictions.

Next, we present two cases in detail to demonstrate that our model can replicate micro-behavior such as layering, lane-changing and overtaking.

Parameter	X (m)	Y (m)
e_{RMSC}	0.1878	1.3248
$ e _{avg}$	0.3808	2.7598
$ e _{max}$	1.6973	13.2220

Table 8.1: Consolidated cumulative errors for all cases.

8.4.1 Congested regime: Time invariant graph (case S_1)

In this case of a time invariant influence graph, a fleet of 5 vehicles move along a 200 m stretch of the road in 12 s, maintaining a fixed distance between them. Their positions at select time intervals are depicted in Figure 8.9(a). Each of the five vehicles is represented by a different color. We observe that the vehicles converge to the desired inter-vehicle spacing.

The RMS errors at each time instant along the X and Y directions are depicted by the red dots in Figures 8.7 and 8.8 respectively. The evolution of absolute errors of each vehicle along the longitudinal direction of motion at each time instant is plotted in Figure 8.10. It can be seen in Figure 8.10 that the errors settle down to low values (< 3 m) for each vehicle within the first 10-12 s. The average and maximum of the absolute value of prediction errors of the model along the X direction are found to be 0.23 m and 0.60 m respectively. Similarly, the average and maximum of the absolute values of prediction errors of the model along the Y direction are found to be 2.83 m and 5.60 m respectively. Hence we see that for the time invariant case, the proposed model successfully predicts the motion of the vehicles in this case. We further observe from Figure 8.9(a) that the model predicts correctly the fixed layered structure of the observed traffic. The cumulative tracking errors for each vehicle in this case are given in Table 8.2.

	Average absolute error		Maximum absolute error	
	X (m)	Y (m)	X (m)	Y (m)
Car - 1	0.1809	3.1107	0.4703	5.4577
Car - 2	0.1520	2.0736	0.4097	3.3613
Car - 3	0.3116	3.1174	0.620	5.4678
Car - 4	0.1725	2.4025	0.6005	5.0473
Car - 5	0.3128	3.4348	0.5595	5.5987

Table 8.2: Cumulative tracking errors for each vehicle: time invariant graph, case S_1 .

8.4.2 Sparse regime: Time varying graph, overtaking and lane changing (case S_3)

In this case there are 5 vehicles which are tracked for a period of 13 s over a distance of 210 m. As seen in Figure 8.9(b), the vehicle marked in black observes the space present between itself and the leader vehicle marked in green. It shifts laterally to the left and overtakes the vehicles marked in blue and red. This behavior can be seen in both, the actual and simulated trajectories in Figure 8.9(b).

The RMS errors for this case, S_3 at each time instant along the X and Y directions are depicted by the black dots in Figures 8.7 and 8.8 respectively. The average and maximum of the absolute value of the prediction errors of the model along the X direction are found to be 0.41 m and 1.33 m respectively. Similarly, the average and maximum of the absolute value of the prediction errors of the model along the Y direction are found to be 2.77 m and 7.48 m respectively. We can thus conclude that the model does indeed predict the observed actual behavior to a great extent in this case too. The cumulative tracking errors for each vehicle in this case are given in Table 8.3.

	Average absolute error		Maximum absolute error	
	X (m)	Y (m)	X (m)	Y (m)
Car - 1	0.6600	2.8901	1.0804	6.3608
Car - 2	0.2357	2.6309	0.7423	5.4041
Car - 3	0.3461	1.8152	0.5179	3.9728
Car - 4	0.4938	3.4919	0.9680	6.1694
Car - 5	0.3204	3.0045	1.3283	7.4843

Table 8.3: Cumulative tracking errors for each vehicle: time varying graph, case S_3 .

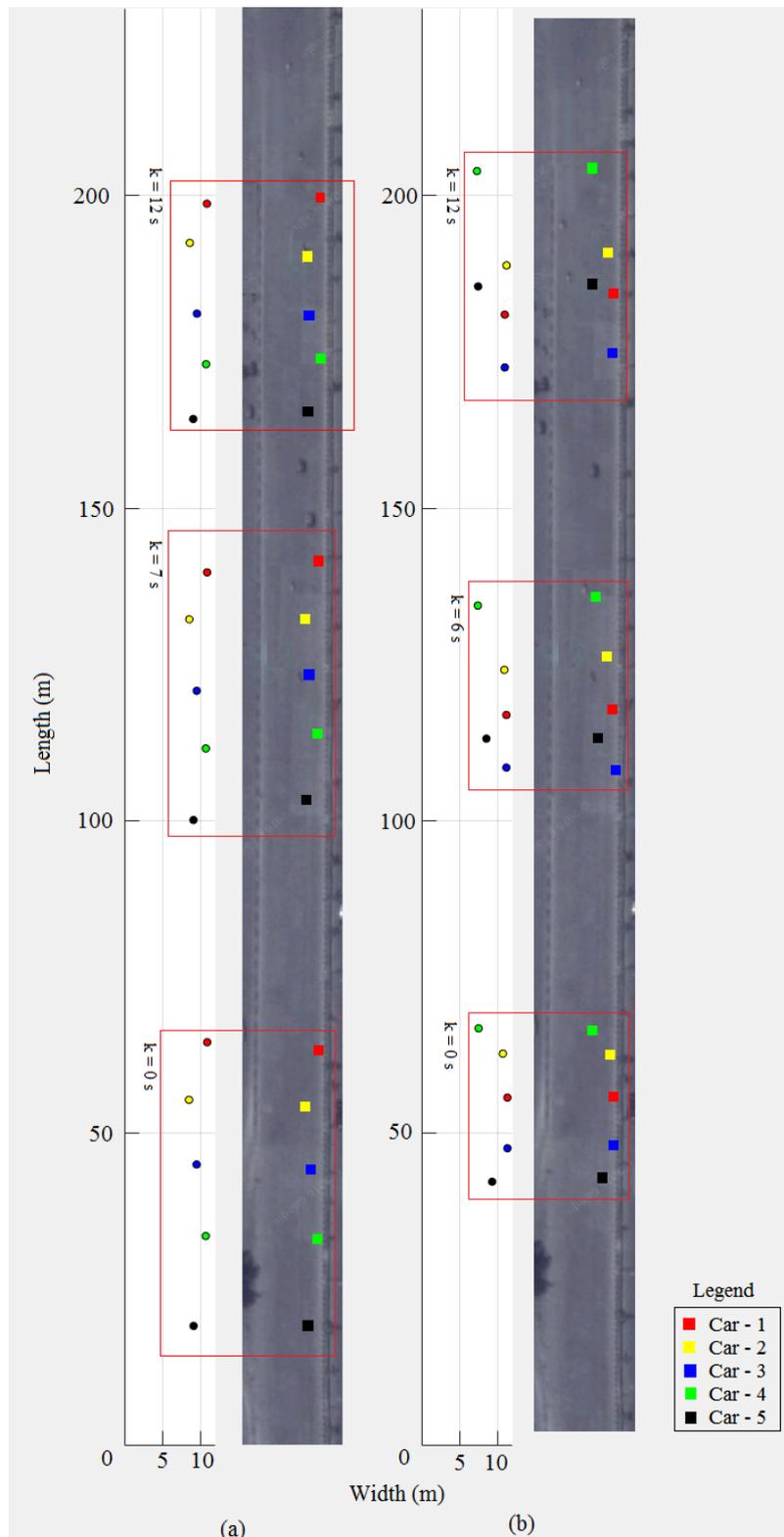


Figure 8.9: Comparison between actual and simulated trajectories for (a) case S_1 , time invariant influence graph and (b) case S_3 , time varying graph. For each case, the plot on the left denotes the predictions made by the model and the one on the right denotes actual positions at different time instants.

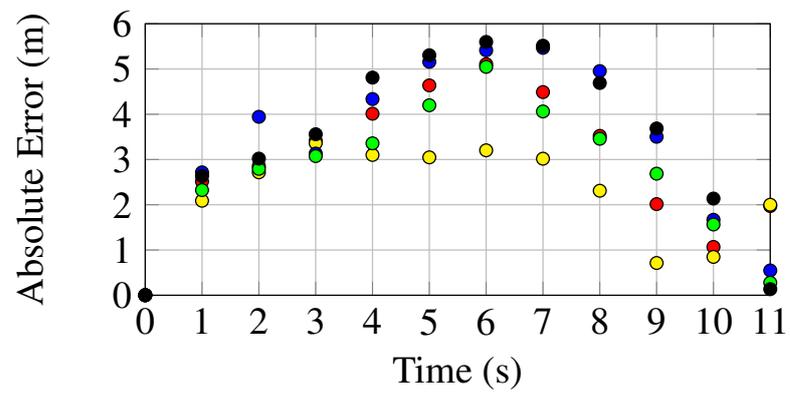


Figure 8.10: Absolute errors of each vehicle along the longitudinal direction for case S_1 . The colour of each trajectory corresponds to the colour of the vehicle marked in Figure 8.9(a).

Chapter 9

Conclusions and Future Work

In this work, we have addressed the engineering challenges in implementation of cooperative control algorithms on physical systems and understand the complex interactions between human vehicle drivers in conditions where lane discipline is not followed. In the first part of this thesis, the implementation of decentralized cooperative control laws on a fleet of quadrotors are presented. In the second part, a model developed for vehicular traffic applicable to cases where lane following is not adhered to, is validated with data recorded using a camera mounted atop a building adjoining a sample road in Mumbai, India.

A summary of our achievements now follows. Potential directions of future work are discussed in Section 9.1.

1. Cooperative control of multi-quadrotor systems

In this work we have successfully achieved autonomous cooperative flight on a quadrotor fleet in indoor and outdoor conditions. It is shown that the quadrotor, for small roll and pitch angles, with heading (yaw) held constant, can be approximated as a pair of double integrators. This allows us to modify and implement consensus and consensus tracking algorithms for double integrator dynamics with provable convergence properties, on a quadrotor fleet. From the experiments, a match between theoretical results and practice is observed. Let us look at the two cooperative control problems addressed in this thesis:

Consensus tracking

A min-max time consensus tracking algorithm has been implemented on a multi-quadrotor testbed with four quadrotors in an indoor environment. The follower quadrotors successfully converge onto the reference trajectory generated by the leader quadrotor in min-max time using local feedback control strategy. The implemented algorithm is decentralized and is computationally cheap due to the availability of closed form expressions of the feedback law. The theoretical claims made in [JSCC19] regarding bounds on the deviations of the trajectories due to finite communication/measurement rate, are verified through experiments.

Consensus

In this work, a decentralized consensus algorithm is implemented on testbed of quadrotors in an outdoor environment and a real-time communication protocol for inter-agent data transfer is developed. Consensus between quadrotors is achieved using minimum possible information exchange between agents (position only). Experiments are performed with the proposed communication protocol to verify its efficacy. It is verified that the protocol is capable of transmitting data without collisions in real-time and handling link breakages and additions. Consensus experiments with different data exchange rates are performed to show the degradation of performance as communication rate is reduced.

The following challenges were faced while implementing cooperative control algorithms on practical quadrotor systems:

- Creating a stable system of systems: To achieve cooperation among multiple agents, each agent is required to perform its task efficiently. Hence, rigorous tests were performed on each quadrotor to ensure basic manual and autonomous flight (see Section 3.3 for details).
- Establishing a communication network: Information exchange is vital for cooperation among agents. Hence rigorous testing of the communication systems was carried out prior to the controlled motion experiments with the quadrotors.
- Each quadrotor had limited hardware capabilities. Hence, the system architecture was carefully designed for seamless performance (see Chapter 4 for details).
- Monitoring flight paths of multiple quadrotors during testing phase is a daunting task. Necessary safety precautions were taken by setting up safety nets and providing flight safety training to all involved personnel.

2. Modeling of lane-less traffic

In this work, the model proposed in [MJC⁺18] is extended for predicting motion of vehicles in sparse traffic. Verification of the models for sparse and dense traffic has been done by comparing model-predicted data with data observed on Indian roads. Video footage of traffic is recorded using a camera mounted atop a building adjoining a sample road. Information regarding vehicle trajectories has been extracted using image processing techniques. These trajectories are then compared with the trajectories generated by the model. It is shown that the theoretical model proposed in [MJC⁺18] is able to successfully capture some of the complex behavior of drivers observed on Indian roads.

The challenges in this work were primarily in the collection of vehicle data, due to infrastructural constraints and safety regulations. Collection of more data would have resulted in an increased confidence in the traffic models. Also, analysis of the data collected for verification of the model for dense traffic, lead to observations which helped in the extension of the model for sparse traffic.

9.1 Future Work

This section contains potential directions of future work.

9.1.1 Cooperative control of multi-agent systems

From the cooperative control experiments it is seen that cooperation can be achieved to perform low-level tasks like formation flight and rendezvous using flying robots. The next step would be the execution of missions (which implicitly require low-level tasks) to achieve more complex/abstract tasks. The agents, possibly heterogeneous, would be required to switch between cooperative behaviors and still be able to successfully complete the mission. Such missions would require human supervision and interaction.

The following areas are worth exploring in the context of cooperative robotic systems of the future:

- Development of supervisory human-in-loop controllers: A supervisory control framework needs to be developed wherein humans monitor the system remotely and take top-level control decision. The remote monitoring station should be capable of fusing mission data and generating low-level control tasks for the multi-robot system to perform.
- Development of mobile ad-hoc networks: This is an active research area in communication systems engineering and there is a push towards development of communication protocols for mobile robots/vehicles. However, given environmental and hardware limitations, it would be impossible to have a practical system without any latency or packet drops. Hence it is essential to characterize effects of communication delays and outages on performance of cooperative behavior.

9.1.2 Modeling of lane-less traffic

Laws of interaction for humans driving in conditions where lane discipline is not followed have been understood and verified through the work described in this thesis. As a result, the following directions are worth exploring:

- The proposed model can be used to build simulators [SM12], like SUMO [KEBB12], SiMTraM [KS⁺16], that more realistically reflect the Indian traffic scenario.
- This model can potentially be used to numerically compute macroscopic quantities like road capacities, flow rate etc. Then, a comparison with data available for Indian conditions needs to be done.
- This model can be the basis of understanding of possible future semi-autonomous robot drivers for Indian conditions. It can form the basis for a simulator to test the co-existence of autonomous vehicles with human drivers who do not follow lane discipline.

Publications from this thesis

Journal papers

- Mulla, A.K., **Joshi, A.**, Chavan, R., Chakraborty, D. and Manjunath, D., 2018. A Microscopic Model for Lane-Less Traffic. *IEEE Transactions on Control of Network Systems*, 6(1), pp.415-428.
- **Joshi, A.**, Wala, A., Ludhiyani, M., Chakraborty, D., Manjunath, D., Chung, H., 2018. Outdoor cooperative flight using decentralized consensus algorithm and a guaranteed real-time communication protocol. *Control Engineering Practice*, (minor revision).
- **Joshi, A.**, Sawant, V., Chakraborty, D., Chung, H., 2018. Min-max time consensus tracking on a multi-quadrotor testbed. *Control Engineering Practice*, (under review)

Conference papers

- **Joshi, A.**, Limbu, N., Ahuja, I., Mulla, A.K., Chung, H. and Chakraborty, D., 2016, June. Implementation of distributed consensus on an outdoor testbed. In *European Control Conference (ECC), 2016 (pp. 2146-2151)*. IEEE.
- **Joshi, A.**, Chakraborty, D., 2017, July. Experimental verification of a dynamic model for lane-less (Indian) traffic. In *IFAC-PapersOnLine 50, no. 1 (2017): 7493-7498*..
- **Joshi, A.**, Wala, A., Ludhiyani, M., Singh, S., Gagrani, M., Hazra, S., Chakraborty, D., Manjunath, D., Chung, H. 2016, June. Implementation of distributed consensus with guaranteed real-time communication on an outdoor quadrotor testbed. In *Decision and Control (CDC), 2017 IEEE 56th Annual Conference on, pp. 2158-2163*. IEEE, 2017..
- **Joshi, A.**, Sawant, V., Chakraborty, D., Chung, H., 2018. Implementation of min-max time consensus tracking on a multi-agent quadrotor testbed. In *European Control Conference (ECC), 2019*. IEEE. (accepted)

Bibliography

- [AAMO13] Jose Joaquin Acevedo, Begoña C Arrue, Ivan Maza, and Anibal Ollero. Cooperative large area surveillance with a team of aerial mobile robots for long endurance missions. *Journal of Intelligent & Robotic Systems*, pages 1–17, 2013.
- [Adm17] Federal Highway Administration. Next generation simulation (NGSIM). <https://ops.fhwa.dot.gov/trafficanalysistools/ngsim.htm>, 2017.
- [aer15] Aeroquad - an Arduino based autopilot. <https://code.google.com/p/aeroquad/downloads/list>, 2015.
- [AKSS16] G Asaithambi, V Kanagaraj, K.K. Srinivasan, and R. Sivanandan. Study of traffic flow characteristics using different vehicle-following models under mixed traffic conditions. *Transportation Letters*, pages 1–12, 2016.
- [AKT16] Gowri Asaithambi, Venkatesan Kanagaraj, and Tomer Toledo. Driving behaviors: Models and challenges for non-lane based mixed traffic. *Transportation in Developing Economies*, 2(2):19, 2016.
- [ANT10] Kostas Alexis, George Nikolakopoulos, and Anthony Tzes. Design and experimental verification of a constrained finite time optimal control scheme for the attitude control of a quadrotor helicopter subject to wind gusts. In *Robotics and Automation (ICRA), 2010 IEEE International Conference on*, pages 1636–1641. IEEE, 2010.
- [apm15] Ardupilot Mega 2.6 hardware documentation. <http://copter.ardupilot.com/wiki/common-apm25-and-26-overview/>, 2015.
- [APVvA14] Lasmini Ambarwati, Adam J Pel, Robert Verhaeghe, and Bart van Arem. Empirical analysis of heterogeneous traffic flow and calibration of porous flow model. *Transportation Research Part C: Emerging Technologies*, 48:418–436, 2014.
- [AR00] A Aw and M Rasclé. Resurrection of "second order" models of traffic flow. *SIAM Journal on Applied Mathematics*, 60(3):916–938, 2000.

- [AT10] Abdelkader Abdessameud and Abdelhamid Tayebi. On consensus algorithms for double-integrator dynamics without velocity measurements and with input constraints. *Systems & Control Letters*, 59(12):812–821, 2010.
- [Bap10] Ravindra B Bapat. *Graphs and matrices*. Springer, 2010.
- [BCM09] Francesco Bullo, Jorge Cortes, and Sonia Martinez. *Distributed control of robotic networks: a mathematical approach to motion coordination algorithms*, volume 27. Princeton University Press, 2009.
- [Bex68] Sten Bexelius. An extended model for car-following. *Transportation Research*, 2(1):13–21, 1968.
- [BGC03] Sylvie Benzoni-Gavage and Rinaldo M Colombo. An n -populations model for traffic flow. *European Journal of Applied Mathematics*, 14(5):587–612, 2003.
- [BM12] B Bhavathrathan and C Mallikarjuna. Evolution of macroscopic models for modeling the heterogeneous traffic: an Indian perspective. *Transportation Letters*, 4(1):29–39, 2012.
- [BSK11] Axel Bürkle, Florian Segor, and Matthias Kollmann. Towards autonomous micro UAV swarms. *Journal of intelligent & robotic systems*, 61(1-4):339–353, 2011.
- [BST13] Ilker Bekmezci, Ozgur Koray Sahingoz, and Şamil Temel. Flying ad-hoc networks (fanets): A survey. *Ad Hoc Networks*, 11(3):1254–1270, 2013.
- [CC17] Lien Wu Chen and Chia Chen Chang. Cooperative traffic control with green wave coordination for multiple intersections based on the internet of vehicles. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 47(7):1321–1335, 2017.
- [CKBM06] David W Casbeer, Derek B Kingston, Randal W Beard, and Timothy W McLain. Cooperative forest fire surveillance using a team of small unmanned air vehicles. *International Journal of Systems Science*, 37(6):351–360, 2006.
- [Cla03] S Clark. Traffic prediction using multivariate nonparametric regression. *Journal of Transportation Engineering*, 129(2):161–168, 2003.
- [Cly02] James R Clynych. Radius of the earth-radii used in geodesy. *Naval Postgraduate School*, 2002.
- [CLZA12] Danjue Chen, Jorge Laval, Zuduo Zheng, and Soyoung Ahn. A behavioral car-following model that captures traffic oscillations. *Transportation Research Part B: Methodological*, 46(6):744–761, 2012.

- [CMCM16] Rakesh Chavan, Ameer K Mulla, Debraj Chakraborty, and D Manjunath. A model for lane-less traffic with local control laws. In *Control Conference (ECC), 2016 European*, pages 2447–2452. IEEE, 2016.
- [Cor06] Jorge Cortés. Finite-time convergent gradient flows with applications to network consensus. *Automatica*, 42(11):1993–2000, 2006.
- [Cor11] Peter Corke. *Robotics, vision and control: fundamental algorithms in MATLAB*, volume 73. Springer Science & Business Media, 2011.
- [COSS11] Hoam Chung, Songhwai Oh, David Hyunchul Shim, and S Shankar Sastry. Toward robotic sensor webs: Algorithms, systems, and experiments. *Proceedings of the IEEE*, 99(9):1562–1586, 2011.
- [CPD⁺18] Soon-Jo Chung, Aditya Avinash Paranjape, Philip Dames, Shaojie Shen, and Vijay Kumar. A survey on aerial swarm robotics. *IEEE Transactions on Robotics*, 34(4):837–855, 2018.
- [CR09] Yongcan Cao and Wei Ren. Containment control with multiple stationary or dynamic leaders under a directed interaction graph. In *48th IEEE Conference on Decision and Control*, pages 3014–3019, 2009.
- [CR12] Yongcan Cao and Wei Ren. Distributed coordinated tracking with reduced interaction via a variable structure approach. *IEEE Transactions on Automatic Control*, 57(1):33–48, 2012.
- [cra18] Crazyflie 2.0. <https://www.bitcraze.io/crazyflie-2/>, Last accessed 22nd Oct, 2018.
- [CRM10] Yongcan Cao, Wei Ren, and Ziyang Meng. Decentralized finite-time sliding mode estimators and their applications in decentralized finite-time formation tracking. *Systems & Control Letters*, 59(9):522–529, 2010.
- [CV06] John S Caughman and JJP Veerman. Kernels of directed graph laplacians. *The Electronic Journal of Combinatorics*, 13(1):R39, 2006.
- [CWLL15] CL Philip Chen, Guo-Xing Wen, Yan-Jun Liu, and Zhi Liu. Observer-based adaptive backstepping consensus tracking control for high-order nonlinear semi-strict-feedback multiagent systems. *IEEE transactions on cybernetics*, 46(7):1591–1601, 2015.
- [CYRC13] Yongcan Cao, Wenwu Yu, Wei Ren, and Guanrong Chen. An overview of recent progress in the study of distributed multi-agent coordination. *IEEE Transactions on Industrial Informatics*, 9(1):427–438, 2013.

- [DV75] Charles A Desoer and Mathukumalli Vidyasagar. *Feedback systems: input-output properties*, volume 55. SIAM, 1975.
- [DYSZ15] Xiwang Dong, Bocheng Yu, Zongying Shi, and Yisheng Zhong. Time-varying formation control for unmanned aerial vehicles: Theories and applications. *IEEE Transactions on Control Systems Technology*, 23(1):340–348, 2015.
- [DZRZ16] Xiwang Dong, Yan Zhou, Zhang Ren, and Yisheng Zhong. Time-varying formation control for unmanned aerial vehicles with switching interaction topologies. *Control Engineering Practice*, 46:26–36, 2016.
- [EGE02] Jeremy Elson, Lewis Girod, and Deborah Estrin. Fine-grained network time synchronization using reference broadcasts. *ACM SIGOPS Operating Systems Review*, 36(SI):147–163, 2002.
- [EGPS01] Deborah Estrin, Lewis Girod, Greg Pottie, and Mani Srivastava. Instrumenting the world with wireless sensor networks. In *ICASSP*, pages 2033–2036. IEEE, 2001.
- [fle18] Flex 13 motion capture cameras by Optitrack. <https://optitrack.com/products/flex-13/>, Last accessed 22nd Oct 2018.
- [FM04] J Alexander Fax and Richard M Murray. Information flow and cooperative control of vehicle formations. *IEEE Transactions on Automatic Control*, 49(9):1465–1476, 2004.
- [FSC14] Yaling Fang, Zhongke Shi, and Jinliang Cao. Calibration of an interrupted traffic flow system using ngsim trajectory data sets. In *Intelligent Control and Automation (WCICA), 2014 11th World Congress on*, pages 4887–4892. IEEE, 2014.
- [FTB03] Terrence Fong, Charles Thorpe, and Charles Baur. Multi-robot remote driving with collaborative control. *IEEE Transactions on Industrial Electronics*, 50(4):699–704, 2003.
- [FW15] Shimao Fan and Daniel B Work. A heterogeneous multiclass traffic flow model with creeping. *SIAM Journal on Applied Mathematics*, 75(2):813–835, 2015.
- [Gaz17] Gazebo: Robot simulation made easy. <http://gazebosim.org/>, Last accessed 10th March, 2017.
- [GFS⁺16] Jason Gregory, Jonathan Fink, Ethan Stump, Jeffrey Twigg, John Rogers, David Baran, Nicholas Fung, and Stuart Young. Application of multi-robot systems to disaster-relief scenarios with limited communication. In *Field and Service Robotics*, pages 639–653. Springer, 2016.

- [GGC13] Jose Alfredo Guerrero, Pedro Castillo Garcia, and Yacine Challal. Quadrotors formation control. *Journal of Intelligent & Robotic Systems*, 70(1-4):221–231, 2013.
- [GHR61] Denos C Gazis, Robert Herman, and Richard W Rothery. Nonlinear follow-the-leader models of traffic flow. *Operations research*, 9(4):545–567, 1961.
- [GJV16] Lav Gupta, Raj Jain, and Gabor Vaszkun. Survey of important issues in UAV communication networks. *IEEE Communications Surveys & Tutorials*, 18(2):1123–1152, 2016.
- [GWJ09] Yanping Gao, Long Wang, and Yingmin Jia. Consensus of multiple second-order agents without velocity measurements. In *American Control Conference, 2009. ACC'09.*, pages 4464–4469. IEEE, 2009.
- [HA17] Wolfgang Hönig and Nora Ayanian. Flying multiple UAVs using ROS. In *Robot Operating System (ROS)*, pages 83–118. Springer, 2017.
- [Hel95] Dirk Helbing. Improved fluid-dynamic model for vehicular traffic. *Physical Review E*, 51(4):3164, 1995.
- [HHWT11] Gabriel M Hoffmann, Haomiao Huang, Steven L Waslander, and Claire J Tomlin. Precision flight control for a multi-vehicle quadrotor helicopter testbed. *Control engineering practice*, 19(9):1023–1036, 2011.
- [HÖR⁺03] Cem Hatipoglu, Ümit Özgüner, Keith Redmill, et al. Automated lane change controller design. *Intelligent Transportation Systems, IEEE Transactions on*, 4(1):13–22, 2003.
- [HSL⁺15] Ke Han, Yuqi Sun, Hongcheng Liu, Terry L. Friesz, and Tao Yao. A bi-level model of dynamic traffic signal control with continuum approximation. *Transportation Research Part C: Emerging Technologies*, 55:409–431, 2015.
- [HSM18] Victoria Hsiao, Anandh Swaminathan, and Richard M Murray. Control theory for synthetic biology: Recent advances in system characterization, control design, and controller implementation for synthetic biology. *IEEE Control Systems*, 38(3):32–62, 2018.
- [JGL15] Wen-Long Jin, Qi-Jian Gan, and Jean-Patrick Lebacque. A kinematic wave theory of capacity drop. *Transportation Research Part B: Methodological*, 81:316–329, 2015.
- [JLA⁺16] Apurva Joshi, Narendra Limbu, Indrajit Ahuja, Ameer K Mulla, Hoam Chung, and Debraj Chakraborty. Implementation of distributed consensus on an outdoor testbed. In *European Control Conference (ECC), 2016*, pages 2146–2151. IEEE, 2016.

- [JLM03] Ali Jadbabaie, Jie Lin, and A Stephen Morse. Coordination of groups of mobile autonomous agents using nearest neighbor rules. *IEEE Transactions on Automatic Control*, 48(6):988–1001, 2003.
- [Jos18] Apurva Joshi. Implementation of consensus tracking on a Crazyflie 2.0 fleet. https://github.com/zapurva/crazygames_ws, 2018.
- [JRT⁺15] Niels Joubert, Mike Roberts, Anh Truong, Floraine Berthouzoz, and Pat Hanrahan. An interactive tool for designing quadrotor camera shots. *ACM Transactions on Graphics (TOG)*, 34(6):238, 2015.
- [JSCC19] Apurva Joshi, Vishal Sawant, Debraj Chakraborty, and Hoam Chung. Implementation of min-max time consensus tracking on a multi-quadrotor testbed. In *European Control Conference*. IEEE, IFAC, (accepted) 2019.
- [KEBB12] Daniel Krajzewicz, Jakob Erdmann, Michael Behrisch, and Laura Bieker. Recent development and applications of SUMO - Simulation of Urban MObility. *International Journal On Advances in Systems and Measurements*, 5(3&4):128–138, December 2012.
- [Kir12] Donald E Kirk. *Optimal control theory: an introduction*. Courier Corporation, 2012.
- [KM11] Jiwon Kim and Hani Mahmassani. Correlated parameters in driving behavior models. *Transportation Research Record: Journal of the Transportation Research Board*, 2249:62–77, 2011.
- [KMPK13] Alex Kushleyev, Daniel Mellinger, Caitlin Powers, and Vijay Kumar. Towards a swarm of agile micro quadrotors. *Autonomous Robots*, 35(4):287–300, 2013.
- [KPA17] Sung-Mo Kang, Myoung-Chul Park, and Hyo-Sung Ahn. Distance-based cycle-free persistent formation: Global convergence and experimental test with a group of quadcopters. *IEEE Transactions on Industrial Electronics*, 64(1):380–389, 2017.
- [KS⁺16] Abhilash Kumar, Partha Pratim Sarkar, et al. Studying and simulating mix traffic using simtram. *Traffic Engineering & Control*, 57(3), 2016.
- [KSBY00] Vikram Kapila, Andrew G Sparks, James M Buffington, and Qiguo Yan. Spacecraft formation flying: Dynamics and control. *Journal of Guidance, Control, and Dynamics*, 23(3):561–564, 2000.
- [KW98] Axel Klar and Raimund Wegener. A hierarchy of models for multilane vehicular traffic i: Modeling. *SIAM Journal on Applied Mathematics*, 59(3):983–1001, 1998.

- [Lav14] A parsimonious model for the formation of oscillations in car-following models. *Transportation Research Part B: Methodological*, 70:228–238, 2014.
- [LD06] Jorge A Laval and Carlos F Daganzo. Lane-changing in traffic streams. *Transportation Research Part B: Methodological*, 40(3):251–264, 2006.
- [LDL11] Shihua Li, Haibo Du, and Xiangze Lin. Finite-time consensus algorithm for multi-agent systems with double-integrator dynamics. *Automatica*, 47(8):1706–1712, 2011.
- [LHM⁺14] Sergei Lupashin, Markus Hehn, Mark W Mueller, Angela P Schoellig, Michael Sherback, and Raffaello D’Andrea. A platform for aerial robotics research and demonstration: The flying machine arena. *Mechatronics*, 24(1):41–54, 2014.
- [Lib03] Daniel Liberzon. *Switching in systems and control*. Springer Science & Business Media, 2003.
- [Liu06] John X Liu. *Focus on Robotics research*. Nova Publishers, 2006.
- [LMK11] Quentin Lindsey, Daniel Mellinger, and Vijay Kumar. Construction of cubic structures with quadrotor teams. *Proc. Robotics: Science & Systems VII*, 2011.
- [LPLK12] Hyon Lim, Jaemann Park, Daewon Lee, and H Jin Kim. Build your own quadrotor: Open-source projects on unmanned aerial vehicles. *IEEE Robotics & Automation Magazine*, 19(3):33–45, 2012.
- [LRLF12] Zhongkui Li, Wei Ren, Xiangdong Liu, and Mengyin Fu. Consensus of multi-agent systems with general linear and lipschitz nonlinear dynamics using distributed adaptive protocols. *IEEE Transactions on Automatic Control*, 58(7):1786–1791, 2012.
- [LSK13] Taeyoung Lee, Koushil Sreenath, and Vijay Kumar. Geometric control of cooperating multiple quadrotor UAVs with a suspended payload. In *52nd IEEE Conference on Decision and Control*, pages 5510–5515. IEEE, 2013.
- [LVH09] X. Y. Lu, P. Varaiya, and R. Horowitz. Fundamental diagram modelling and analysis based NGSIM data. *IFAC Proceedings Volumes*, 42(15):367–374, 2009.
- [LW55] M. J. Lighthill and G. B. Whitham. On kinematic waves. ii. a theory of traffic flow on long crowded roads. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 229(1178):317–345, 1955.
- [May90] Adolf D May. *Traffic flow fundamentals*. 1990.

- [MBK13] Mahendrakumar Metkari, Anuj Budhkar, and Akhilesh Kumar Maurya. Development of simulation model for heterogeneous traffic with no lane discipline. *Procedia - Social and Behavioral Sciences*, 104:360–369, 2013.
- [MC18] Ameer K Mulla and Debraj Chakraborty. Min–max time consensus tracking with communication guarantee. *IEEE Transactions on Automatic Control*, 63(1):132–144, 2018.
- [ME10] Mehran Mesbahi and Magnus Egerstedt. *Graph theoretic methods in multiagent networks*, volume 33. Princeton University Press, 2010.
- [MJC⁺18] Ameer K Mulla, Apurva Joshi, Rakesh Chavan, Debraj Chakraborty, and D Manjunath. A microscopic model for lane-less traffic. *IEEE Transactions on Control of Network Systems*, 6(1):415–428, 2018.
- [ML07] Kevin L Moore and Dennis Lucarelli. Decentralized adaptive scheduling using consensus variables. *International Journal of Robust and Nonlinear Control: IFAC-Affiliated Journal*, 17(10-11):921–940, 2007.
- [MM16] Caleb Ronald Munigety and Tom V. Mathew. Towards behavioral modeling of drivers in mixed traffic conditions. *Transportation in Developing Economies*, 2(1):6, 2016.
- [Mor96] A. Stephen Morse. Supervisory control of families of linear set-point controllers - Part I: Exact matching. *IEEE Transactions on Automatic Control*, 41(10):1413–1431, 1996.
- [MR13] Ranju Mohan and Gitakrishnan Ramadurai. Heterogeneous traffic flow modelling using macroscopic continuum model. *Procedia - Social and Behavioral Sciences*, 104:402–411, 2013.
- [MRM13] Jie Mei, Wei Ren, and Guangfu Ma. Distributed coordination for second-order multi-agent systems with nonlinear dynamics using only relative position measurements. *Automatica*, 49(5):1419–1427, 2013.
- [MSK⁺12] Johannes Meyer, Alexander Sendobry, Stefan Kohlbrecher, Uwe Klingauf, and Oskar von Stryk. Comprehensive simulation of quadrotor UAVs using ROS and Gazebo. In *3rd Int. Conf. on Simulation, Modeling and Programming for Autonomous Robots (SIMPAN)*, pages 400–411, 2012.
- [MSKR14] Nathan Michael, Mac Schwager, Vijay Kumar, and Daniela Rus. An experimental study of time scales and stability in networked multi-robot systems. In *Experimental Robotics*, pages 631–643. Springer, 2014.

- [MSM⁺12] Nathan Michael, Shaojie Shen, Kartik Mohta, Yash Mulgaonkar, Vijay Kumar, Keiji Nagatani, Yoshito Okada, Seiga Kiribayashi, Kazuki Otake, Kazuya Yoshida, et al. Collaborative mapping of an earthquake-damaged building via ground and aerial robots. *Journal of Field Robotics*, 29(5):832–841, 2012.
- [NCKS17] Kamesh Namuduri, Serge Chaumette, Jae H Kim, and James PG Sterbenz. *UAV Networks and Communications*. Cambridge University Press, 2017.
- [Nel88] William Nelson. Use of circular error probability in target detection. Technical report, MITRE Corporation, Bedford, MA, 1988.
- [NKG12] Aboelmagd Noureldin, Tashfeen B Karamat, and Jacques Geogy. *Fundamentals of inertial navigation, satellite-based positioning and their integration*. Springer Science & Business Media, 2012.
- [NMMH11] Rahul Nair, Hani S Mahmassani, and Elise Miller-Hooks. A porous flow approach to modeling heterogeneous traffic in disordered systems. *Transportation Research Part B: Methodological*, 45(9):1331–1345, 2011.
- [ÖDF⁺18] Kayhan Özcimder, Biswadip Dey, Alessio Franci, Rebecca Lazier, Daniel True-man, and Naomi Ehrich Leonard. Social decision-making driven by artistic explore–exploit tension. *Interdisciplinary Science Reviews*, pages 1–27, 2018.
- [OH08] Saskia Ossen and Serge Hoogendoorn. Validity of trajectory-based calibration approach of car-following models in presence of measurement errors. *Transportation Research Record: Journal of the Transportation Research Board*, (2088):117–125, 2008.
- [OMS01] Esben H Ostergaard, Maja J Mataric, and Gaurav S Sukhatme. Distributed multi-robot task allocation for emergency handling. In *Intelligent Robots and Systems, 2001. Proceedings. 2001 IEEE/RSJ International Conference on*, volume 2, pages 821–826. IEEE, 2001.
- [OPA15] Kwang-Kyo Oh, Myoung-Chul Park, and Hyo-Sung Ahn. A survey of multi-agent formation control. *Automatica*, 53:424–440, 2015.
- [OS06] Reza Olfati-Saber. Flocking for multi-agent dynamic systems: algorithms and theory. *IEEE Transactions on Automatic Control*, 51(3):401–420, 2006.
- [OSM04a] Reza Olfati-Saber and Richard M Murray. Consensus problems in networks of agents with switching topology and time-delays. *IEEE Transactions on automatic control*, 49(9):1520–1533, 2004.
- [OSM04b] Reza Olfati-Saber and Richard M Murray. Consensus problems in networks of agents with switching topology and time-delays. *IEEE Transactions on Automatic Control*, 49(9):1520–1533, 2004.

- [PHSA17] James A Preiss, Wolfgang Honig, Gaurav S Sukhatme, and Nora Ayanian. CrazySwarm: A large nano-quadcopter swarm. In *Robotics and Automation (ICRA), 2017 IEEE International Conference on*, pages 3299–3304. IEEE, 2017.
- [PS15] Nicholas Polson and Vadim Sokolov. Bayesian analysis of traffic flow on interstate I-55: The LWR model. *Annals of Applied Statistics*, 9(4):1864–1888, 2015.
- [PSH02] Aniruddha Pant, Pete Seiler, and Karl Hedrick. Mesh stability of look-ahead interconnected systems. *IEEE Transactions on Automatic Control*, 47(2):403–407, 2002.
- [Pur05] Anuj Puri. A survey of unmanned aerial vehicles (UAV) for traffic surveillance. *Department of computer science and engineering, University of South Florida*, pages 1–29, 2005.
- [QMSW16] Jiahu Qin, Qichao Ma, Yang Shi, and Long Wang. Recent advances in consensus of multi-agent systems: A brief survey. *IEEE Transactions on Industrial Electronics*, 64(6):4972–4983, 2016.
- [Qual7] Quan Quan. *Introduction to multicopter design and control*. Springer, 2017.
- [RA07] Wei Ren and Ella Atkins. Distributed multi-vehicle coordinated control via local information exchange. *International Journal of Robust and Nonlinear Control*, 17(10-11):1002–1033, 2007.
- [RB08] Wei Ren and Randal W Beard. *Distributed consensus in multi-vehicle cooperative control*. Springer, 2008.
- [RBA05] Wei Ren, Randal W Beard, and Ella M Atkins. A survey of consensus problems in multi-agent coordination. In *Proceedings of the 2005, American Control Conference, 2005.*, pages 1859–1864. IEEE, 2005.
- [RC10] Wei Ren and Yongcan Cao. *Distributed coordination of multi-agent networks: emergent problems, models, and issues*. Springer Science & Business Media, 2010.
- [Ren08] Wei Ren. Consensus tracking under directed interaction topologies: Algorithms and experiments. In *American Control Conference, 2008*, pages 742–747. IEEE, 2008.
- [Rey87] Craig W Reynolds. Flocks, herds and schools: A distributed behavioral model. In *ACM SIGGRAPH Computer Graphics*, volume 21, pages 25–34. ACM, 1987.
- [RN16] Stuart J Russell and Peter Norvig. *Artificial intelligence: a modern approach*. Malaysia; Pearson Education Limited, 2016.

- [Roc68] A Rockwell, T. H., Ernst, R. L., & Hanken. A sensitivity analysis of empirically derived car-following models. *Transportation Research*, 2(4):363–373, 1968.
- [ROS17] Robot Operating System (ROS), About. <http://www.ros.org/about-ros/>, Last accessed 10th March, 2017.
- [RWB⁺05] Marc D Richards, Darrell Whitley, J Ross Beveridge, Todd Mytkowicz, Duong Nguyen, and David Rome. Evolving cooperative strategies for UAV teams. In *Proceedings of the 7th Annual Conference on Genetic and Evolutionary Computation*, pages 1721–1728. ACM, 2005.
- [SCHS07] Elaine Shaw, Hoam Chung, J Karl Hedrick, and Shankar Sastry. Unmanned helicopter formation flight experiment for the study of mesh stability. In *Cooperative Systems*, volume 588, pages 37–56. Springer, 2007.
- [SFZR15] Osamah Saif, Isabelle Fantoni, and Arturo Zavala-Río. Real-time flocking of multiple-quadrotor system of systems. In *System of Systems Engineering Conference (SoSE), 2015 10th*, pages 286–291. IEEE, 2015.
- [SG17] Fabrizio Schiano and Paolo Robuffo Giordano. Bearing rigidity maintenance for formations of quadrotor uavs. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1467–1474. IEEE, 2017.
- [SH96] D Swaroop and JK Hedrick. String stability of interconnected systems. *Automatic Control, IEEE Transactions on*, 41(3):349–357, 1996.
- [SHP04] Daniel P Scharf, Fred Y Hadaegh, and Scott R Ploen. A survey of spacecraft formation flying guidance and control (Part II): Control. *Pasadena, CA: Jet Propulsion Laboratory, NASA*, 2004.
- [SHUS14] Yasuhiro Shiomi, Teruaki Hanamori, Nobuhiro Uno, and Hiroshi Shimamoto. Modeling mixed traffic flow with motorcycles based on discrete choice approach. In *93rd Annual Meeting Transportation Research Board*, 2014.
- [SL14] Vaibhav Srivastava and Naomi Ehrich Leonard. Collective decision-making in ideal networks: The speed-accuracy tradeoff. *IEEE Transactions on Control of Network Systems*, 1(1):121–132, 2014.
- [SL18] William Lewis Scott and Naomi Ehrich Leonard. Optimal evasive strategies for multiple interacting agents with motion constraints. *Automatica*, 94:26–34, 2018.
- [SM03] Reza Olfati Saber and Richard M Murray. Flocking with obstacle avoidance: cooperation with limited communication in mobile networks. In *Decision and Control, 2003. Proceedings. 42nd IEEE Conference on*, volume 2, pages 2022–2028. IEEE, 2003.

- [SM12] M Sreekumar and Akhilesh Kumar Maurya. Need for a comprehensive traffic simulation model in indian context. In *IJCA proceedings on international conference on emerging frontiers in technology for rural area (EFITRA-2012)*, volume 5, pages 13–18, 2012.
- [SMW⁺14] Daniel Strömbom, Richard P Mann, Alan M Wilson, Stephen Hailes, A Jennifer Morton, David JT Sumpter, and Andrew J King. Solving the shepherding problem: heuristics for herding autonomous, interacting agents. *Journal of the Royal Society Interface*, 11(100):20140719, 2014.
- [SPH99] Pete Seiler, Aniruddha Pant, and JK Hedrick. Preliminary investigation of mesh stability for linear systems. *Proc. of ASME: DSC Div*, pages 359–364, 1999.
- [ST94] Sandra L Scrivener and Roger C Thompson. Survey of time-optimal attitude maneuvers. *Journal of Guidance, Control, and Dynamics*, 17(2):225–233, 1994.
- [SZ12] Qiankun Song and Jiye Zhang. Novel results on mesh stability for a class of vehicle following system with time delays. In *Advances in Neural Networks–ISNN 2012*, pages 336–342. Springer, 2012.
- [Tam13] Roberto Tamassia. *Handbook of graph drawing and visualization*. CRC press, 2013.
- [TFG07] Geetam Tiwari, Joseph Fazio, and Sushant Gaurav. Traffic planning for non-homogeneous traffic. *Sadhana*, 32(4):309–328, 2007.
- [TKP02] Herbert Tanner, Vijay Kumar, and George Pappas. Stability properties of interconnected vehicles. In *15th International Symposium on Mathematical Theory of Networks and Systems*, 2002.
- [TKT08] Martin Treiber, Arne Kesting, and Christian Thiemann. How much does traffic congestion increase fuel consumption and emissions? applying a fuel consumption model to the ngsim trajectory data. In *87th Annual Meeting of the Transportation Research Board, Washington, DC*, 2008.
- [TLHY14] T. Q. Tang, J. G. Li, H. J. Huang, and X. B. Yang. A car-following model with real-time road conditions and numerical tests. *Measurement*, 48(1):63–76, 2014.
- [TLR10] Antoine Tordeux, Sylvain Lassarre, and Michel Roussignol. An adaptive time gap car-following model. *Transportation Research Part B: Methodological*, 44(8-9):1115–1131, 2010.
- [TMH15] Alireza Talebpour, Hani S. Mahmassani, and Samer H. Hamdar. Modeling lane-changing behavior in a connected environment: A game theory approach. *Transportation Research Procedia*, 7:420–440, 2015.

- [TMK12] Matthew Turpin, Nathan Michael, and Vijay Kumar. Decentralized formation control with variable shapes for aerial robots. In *Robotics and Automation (ICRA), 2012 IEEE International Conference on*, pages 23–30. IEEE, 2012.
- [Tol74] J. E. Tolle. Composite car following models. *Transportation Research*, 8(2):91–96, 1974.
- [Tre75] Joseph Treiterer. Investigation of traffic dynamics by aerial photogrammetry techniques. Technical report, 1975.
- [ubl17] ublox LEA-4 series GPS modules. <https://www.u-blox.com/en/product/lea-6-series#product-information>, Last accessed 10th March, 2017.
- [VCBJ⁺95] Tamás Vicsek, András Czirók, Eshel Ben-Jacob, Inon Cohen, and Ofer Shochet. Novel type of phase transition in a system of self-driven particles. *Physical Review Letters*, 75(6):1226, 1995.
- [vrp18] Virtual Reality Peripheral Network. <https://github.com/vrpn/vrpn/wiki>, Last accessed 22nd Oct 2018.
- [VSB⁺11] João Valente, David Sanz, Antonio Barrientos, Jaime del Cerro, Ángela Ribeiro, and Claudio Rossi. An air-ground wireless sensor network for crop monitoring. *Sensors*, 11(6):6088–6108, 2011.
- [VVS⁺14] Gábor Vásárhelyi, Csaba Virágh, Gergő Somorjai, Norbert Tarcai, T Szorenyi, Tamás Nepusz, and Tamás Vicsek. Outdoor flocking and formation flight with autonomous aerial robots. In *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2014)*, pages 3866–3873, 2014.
- [W⁺06] Kong Wai Weng et al. Design and control of a quad-rotor flying robot for aerial surveillance. In *Research and Development, 2006. SCOREd 2006. 4th Student Conference on*, pages 173–177. IEEE, 2006.
- [WCLK18] Aaron Weinstein, Adam Cho, Giuseppe Loianno, and Vijay Kumar. Visual inertial odometry swarm: An autonomous swarm of vision-based quadrotors. *IEEE Robotics and Automation Letters*, 3(3):1801–1807, 2018.
- [WDCY13] Guanghui Wen, Zhisheng Duan, Guanrong Chen, and Wenwu Yu. Consensus tracking of multi-agent systems with lipschitz-type node dynamics and switching topologies. *IEEE Transactions on Circuits and Systems I: Regular Papers*, 61(2):499–511, 2013.
- [WH08] Xiaoli Wang and Yiguang Hong. Finite-time consensus for multi-agent networks with second-order agent dynamics. *IFAC Proceedings Volumes*, 41(2):15185–15190, 2008.

- [WSCW17] Fangyu Wu, Raphael Stern, Miles Churchill, and Daniel Work. Measuring trajectories and fuel consumption in oscillatory traffic: Experimental results. In *96th Annual Meeting of Transportation Research Board*, 2017.
- [WSD⁺05] R. H. White, J. D. Spengler, K. M. Dilwali, B. E. Barry, and J. M. Samet. Report of workshop on traffic, health, and infrastructure planning. *Archives of Environmental & Occupational Health*, 60(2):70–76, 2005.
- [YCCK09] Wenwu Yu, Guanrong Chen, Ming Cao, and Jürgen Kurths. Second-order consensus for multiagent systems with directed topologies and nonlinear dynamics. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 40(3):881–891, 2009.
- [YSS⁺13] Ye Yuan, G-B Stan, Ling Shi, Mauricio Barahona, and Jorge Goncalves. Decentralised minimum-time consensus. *Automatica*, 49(5):1227–1235, 2013.
- [ZDWC15] Yu Zhao, Zhisheng Duan, Guanghui Wen, and Guanrong Chen. Distributed finite-time tracking for a multi-agent system under a leader with bounded unknown acceleration. *Systems & Control Letters*, 81:8–13, 2015.
- [ZDWZ13] Yu Zhao, Zhisheng Duan, Guanghui Wen, and Yanjiao Zhang. Distributed finite-time tracking control for multi-agent systems: an observer-based approach. *Systems & Control Letters*, 62(1):22–28, 2013.
- [Zhe14] Zuduo Zheng. Recent developments and research needs in modeling lane changing. *Transportation Research Part B: Methodological*, 60:16–32, 2014.
- [ZL12] Hongwei Zhang and Frank L Lewis. Adaptive cooperative tracking control of higher-order nonlinear systems with unknown dynamics. *Automatica*, 48(7):1432–1439, 2012.
- [ZLD13] Yu Zhao, Zhongkui Li, and Zhisheng Duan. Distributed consensus tracking of multi-agent systems with nonlinear dynamics under a reference leader. *International Journal of Control*, 86(10):1859–1869, 2013.
- [ZW12] Yuanshi Zheng and Long Wang. Finite-time consensus of heterogeneous multi-agent systems with and without velocity measurements. *Systems & Control Letters*, 61(8):871–878, 2012.
- [ZWC16] Yu Zhao, Guanghui Wen, and Guanrong Chen. Distributed finite-time cooperative control of multi-agent systems. In *Complex Systems and Networks*, pages 163–206. Springer, 2016.

Acknowledgements

I would like to thank my supervisors, Prof. Debraj Chakraborty and Dr. Hoam Chung for their advice, teachings and encouragement. This journey has been truly enriching because of their guidance and support. Many thanks to Prof. Ameer Mulla and Prof. D. Manjunath for all the technical discussions and inputs which helped shape this thesis. I am also grateful to Prof. Madhu Belur for his reviews, comments and suggestions for improving our research.

I am grateful to Mohit L., Ankit, Subhadeep, Mohit G., Pankaj P., Saksham, Indrajeet and Narendra for their contributions. I hope you guys cherish the memories of those early mornings and late nights on the flying grounds and in the lab as much as I do. A special thanks to my lab-mates Vishal, Aditya, Shana, Mousumi and Imrul at IIT Bombay, and Calvin, Rihab, Phillip and Sherry at Monash University. You guys always welcomed me with my technical doubts.

I would also like to thank the faculty at the Dept. of Electrical Engineering, IIT Bombay, especially Prof. S.D. Agashe, Prof. Debasattam Pal, Prof. Harish Pillai, Prof. Madhu Belur and Prof. Jaykrishnan Nair. Their enthusiasm, insight and command over the subject has been very inspiring and motivating. Thanks are due to my mentors Prof. Deepak Patil, Dr. Chayan Bhawal and Dr. Aditya Joshi. They have always been my go-to people for excellent technical and non-technical advice.

Along the way, I managed to make some amazing friends and IIT Bombay and Monash University. Thanks to Deepak A, Shivam, Pankaj S, Sachchit, Anmol, Mostafa, Yathi, Phan, Jason for nerding out and being fun to hang out with. Also, a shout out to all my Ultimate friends from all over the world, who helped relax and get my mind off research problems. A special thanks to the *Shvain* group for always being there.

This journey would not have been possible without the love and support from *Aai*, *Baba*, *Nanu* and *Amu*. Thanks for putting up with me and pushing me to work hard.