Coordination and control of autonomous vehicles at urban intersections

Politecnico di Bari

LIBERATORIA PER L'ARCHIVIAZIONE DELLA TESI DI DOTTORATO

Al Magnifico Rettore
del Politecnico di Bari

Il sottoscritto DIFILIPPO GIANVITO nato a ACQUAVIVA DELLE F. il 30/03/1995

residente a SANTERAMO in via ALESSANDRO MANZONI 25 e-mail gianvito.difilippo@gmail.com

iscritto al 3° anno di Corso di Dottorato di Ricerca in INGEGNERIA ELETTRICA E DELL'INFORMAZIONE ciclo XXXVI

ed essendo stato ammesso a sostenere l'esame finale con la prevista discussione della tesi dal titolo:

COORDINATION AND CONTROL OF AUTONOMOUS VEHICLES AT URBAN INTERSECTIONS

DICHIARA

1) di essere consapevole che, ai sensi del D.P.R. n. 445 del 28.12.2000, le dichiarazioni mendaci, la falsità negli atti e l'uso di atti falsi sono puniti ai sensi del codice penale e delle Leggi speciali in materia, e che nel caso ricorressero dette ipotesi, decade fin dall'inizio e senza necessità di nessuna formalità dai benefici conseguenti al provvedimento emanato sulla base di tali dichiarazioni;
2) di essere iscritto al Corso di Dottorato di ricerca INGEGNERIA ELETTRICA E DELL'INFORMAZIONE ciclo XXXVI, corso attivato ai sensi del "*Regolamento dei Corsi di Dottorato di ricerca del Politecnico di Bari*", emanato con D.R. n.286 del 01.07.2013;
3) di essere pienamente a conoscenza delle disposizioni contenute nel predetto Regolamento in merito alla procedura di deposito, pubblicazione e autoarchiviazione della tesi di dottorato nell'Archivio Istituzionale ad accesso aperto alla letteratura scientifica;
4) di essere consapevole che attraverso l'autoarchiviazione delle tesi nell'Archivio Istituzionale ad accesso aperto alla letteratura scientifica del Politecnico di Bari (IRIS-POLIBA), l'Ateneo archivierà e renderà consultabile in rete (nel rispetto della Policy di Ateneo di cui al D.R. 642 del 13.11.2015) il testo completo della tesi di dottorato, fatta salva la possibilità di sottoscrizione di apposite licenze per le relative condizioni di utilizzo (di cui al sito http://www.creativecommons.it/Licenze), e fatte salve, altresì, le eventuali esigenze di "embargo", legate a strette considerazioni sulla tutelabilità e sfruttamento industriale/commerciale dei contenuti della tesi, da rappresentarsi mediante compilazione e sottoscrizione del modulo in calce (Richiesta di embargo);
5) che la tesi da depositare in IRIS-POLIBA, in formato digitale (PDF/A) sarà del tutto identica a quelle **consegnate**/inviate/da inviarsi ai componenti della commissione per l'esame finale e a qualsiasi altra copia depositata presso gli Uffici del Politecnico di Bari in forma cartacea o digitale, ovvero a quella da discutere in sede di esame finale, a quella da depositare, a cura dell'Ateneo, presso le Biblioteche Nazionali Centrali di Roma e Firenze e presso tutti gli Uffici competenti per legge al momento del deposito stesso, e che di conseguenza va esclusa qualsiasi responsabilità del Politecnico di Bari per quanto riguarda eventuali errori, imprecisioni o omissioni nei contenuti della tesi;
6) che il contenuto e l'organizzazione della tesi è opera originale realizzata dal sottoscritto e non compromette in alcun modo i diritti di terzi, ivi compresi quelli relativi alla sicurezza dei dati personali; che pertanto il Politecnico di Bari ed i suoi funzionari sono in ogni caso esenti da responsabilità di qualsivoglia natura: civile, amministrativa e penale e saranno dal sottoscritto tenuti indenni da qualsiasi richiesta o rivendicazione da parte di terzi;
7) che il contenuto della tesi non infrange in alcun modo il diritto d'Autore né gli obblighi connessi alla salvaguardia di diritti morali od economici di altri autori o di altri aventi diritto, sia per testi, immagini, foto, tabelle, o altre parti di cui la tesi è composta.

Luogo e data Bari, 29/03/1995 Firma Gianvito Difilippo

Il sottoscritto, con l'autoarchiviazione della propria tesi di dottorato nell'Archivio Istituzionale ad accesso aperto del Politecnico di Bari (POLIBA-IRIS), pur mantenendo su di essa tutti i diritti d'autore, morali ed economici, ai sensi della normativa vigente (Legge 633/1941 e ss.mm.ii.),

CONCEDE

- al Politecnico di Bari il permesso di trasferire l'opera su qualsiasi supporto e di convertirla in qualsiasi formato al fine di una corretta conservazione nel tempo. Il Politecnico di Bari garantisce che non verrà effettuata alcuna modifica al contenuto e alla struttura dell'opera.
- al Politecnico di Bari la possibilità di riprodurre l'opera in più di una copia per fini di sicurezza, back-up e conservazione.

Luogo e data Bari, 29/03/1995 Firma Gianvito Difilippo

Politecnico di Bari

Department of Electrical and Information Engineering

Electrical and Information Engineering
Ph.D. Program

SSD: ING-INF/04: Systems and Control Engineering

Final dissertation

Coordination and Control of Autonomous Vehicles at Urban Intersections

---

by

Gianvito Difilippo

Referee:
Prof. Silvia Siri, Ph.D.

Supervisor:
Prof. Maria Pia Fanti, Ph.D.

Coordinator of Ph.D. Program:
Prof. Mario Carpentieri, Ph.D.

Course n° 36, 01/11/2020 - 31/01/2024

Mais je n'ai pas le temps et mes idées ne sont pas encore bien développées sur ce terrain qui est immense.

Évariste Galois

Abstract

This thesis is devoted to the investigation of multi-agent systems theory within the context of autonomous intersection management. It is divided into two main parts: the first and main part addresses the problem of autonomous vehicles crossing an intersection; the second part focuses on the control of multi-agent systems in the scenario of a group of autonomous agents pursuing the objective of forming a uniformly spaced string.

In the section regarding autonomous intersection management, a geometric approach is employed to establish the foundations of a framework for modeling autonomous intersections. After defining the geometry of the trajectories that vehicles follow, the collision conditions are rigorously modeled to enable the definition of a general optimization problem for describing the intersection scenario: the Intersection Crossing Problem (ICP). The ICP is a functional optimization problem and its solution gives the trajectories vehicles must follow in order to cross the intersection without crashing, while optimizing some predefined quantity of interest. Convex analysis and optimization theory are used to derive results stemming from the proposed theory, which are then used to develop an algorithm that is proven to solve a specialization of the ICP in a finite number of iterations. Practical considerations about the computation time are given to derive some approximation methods that are used to speed up the collision avoidance phase of the algorithm while ensuring safety.

The other section deals with the consensus problem in a leaderless network of agents that have to reach a common velocity while forming a uniformly spaced string. Moreover, the final common velocity (reference velocity) is determined by the agents in a distributed and leaderless way. Then, the consensus protocol parameters are optimized for networks characterized by a communication topology described by a class of directed graphs having a directed spanning tree, in order to maximize the convergence rate and avoid oscillations. Finally, necessary and sufficient conditions are derived to guarantee consensus in the presence of a constant delay. The advantages of the optimized consensus protocol are enlightened by some simulation results and comparisons with a protocol proposed in the related literature.

This thesis is organized into six chapters. Chapter 1 introduces the work and outlines its main contributions. Chapter 2 is the review of the available literature on the topic of autonomous intersection management and multi-agent systems. Chapter 3 covers the mathematical tools used to develop the proposed results. Chapter 4 presents the ICP and the proposed distributed algorithm for its solution. Chapter 5 exposes the contributions in the field of leaderless multi-agent systems. Chapter 6 contains remarks and considerations for future work.

Contents

Abstract ii

Contents v

List of figures vii

List of tables viii

List of publications ix

Acknowledgments x

1 Introduction 1
- 1.1 Research objectives . . . 3
- 1.2 Contributions . . . 3
- 1.3 Thesis outline . . . 4
- 1.4 Notation . . . 4

2 Literature review 6
- 2.1 Autonomous intersection management . . . 6
  - 2.1.1 Centralized approaches . . . 7
  - 2.1.2 Decentralized approaches . . . 10
  - 2.1.3 Related works . . . 12
- 2.2 Control of multi-agent systems . . . 14
  - 2.2.1 Platooning . . . 15

3 Preliminaries 16
- 3.1 Convex analysis . . . 16
  - 3.1.1 Support functions . . . 17
- 3.2 Bézier curves . . . 19
- 3.3 Optimization theory . . . 20
  - 3.3.1 Constrained optimization . . . 20
  - 3.3.2 Algorithms for constrained optimization . . . 21
- 3.4 Dynamical systems theory . . . 23
  - 3.4.1 Axiomatic definition of a dynamical system . . . 23
  - 3.4.2 Stability . . . 24
  - 3.4.3 The Jordan canonical form . . . 25
  - 3.4.4 Lyapunov's method for stability . . . 26

3.4.5 Controllability . . . 27
3.5 Graph theory . . . 27
3.5.1 Definition of a graph . . . 28
3.5.2 Properties of graphs . . . 28
3.6 Consensus algorithms . . . 30
3.6.1 First-order algorithms . . . 31
3.6.2 Average consensus . . . 32

4 Cooperative intersection management **33**
4.1 The Intersection Crossing Problem . . . 33
4.1.1 Vehicle model . . . 33
4.1.2 Problem formulation . . . 34
4.2 Intersection model . . . 35
4.2.1 Trajectory model . . . 35
4.2.2 Oblique intersections . . . 36
4.2.2.1 Straight path . . . 38
4.2.2.2 Left and right turns . . . 38
4.2.2.3 Other inbound lanes . . . 39
4.2.3 Collision model . . . 40
4.2.3.1 Bounding circle . . . 40
4.2.3.2 Straight envelope of circles . . . 41
4.2.3.3 Radial envelope of circles . . . 43
4.2.4 Intersection zones . . . 44
4.2.5 Decentralized solution approach . . . 46
4.3 Kinematic laws . . . 46
4.3.1 Minimum energy control . . . 48
4.3.1.1 Second-order MEC . . . 48
4.3.1.2 Third-order MEC . . . 51
4.3.2 Stop and go . . . 53
4.4 Collision avoidance . . . 55
4.4.1 Lateral collision detection . . . 56
4.4.2 Lateral collision avoidance . . . 59
4.5 Computation of the support function . . . 60
4.5.1 Characterization of the collision set . . . 60
4.5.2 Computation algorithm . . . 63
4.5.2.1 Approximation by bounding . . . 65
4.5.2.2 Approximation by Bézier curves . . . 67
4.6 Distributed scheduling . . . 68
4.7 Simulations and numerical results . . . 71

5 Leaderless control of multi-agent systems **74**
5.1 A second-order consensus problem . . . 74
5.1.1 Consensus problem statement . . . 74
5.1.2 Control law synthesis via Lyapunov's method . . . 75
5.1.3 Consensus protocol . . . 78
5.2 Convergence . . . 79
5.3 Eigenvalue allocation . . . 83

5.4 Communication delays 87
5.5 Simulations and numerical results 90
5.5.1 Delay-free scenario 90
5.5.2 Delayed communication scenario 94

6 Conclusions **97**

Appendixes **109**

A Proofs **110**
A.1 Chapter 3 110
A.2 Chapter 4 112
A.3 Chapter 5 113

List of figures

2.1 Research goals and sub-goals. 6
2.2 AIM scheme. 8
2.3 Examples of successful and rejected reservations. 8
2.4 Obstacle regions and their closures. 13

3.1 The function curves of logarithmic barrier function under different values of μ. 22
3.2 Representation of a system with its input, states, and outputs. 23
3.3 Representation of a graph with nodes and arcs. 28
3.4 Centralized (left) and decentralized (right) control. 31

4.1 Position of a vehicle halfway through a left turn. 35
4.2 Intersection of straight roads meeting at an angle $\alpha = 70^\circ$. 37
4.3 Bounding circles of rectangles with different w/h ratio. 41
4.4 Straight envelope of circles. 42
4.5 Radial envelope of circles. 43
4.6 Zones of the intersection. 44
4.7 Domain of $\boldsymbol{\lambda}$ produced by the second-order MEC. 50
4.8 Position, speed, and acceleration profiles produced by the second-order MEC. 51
4.9 Domain of $\boldsymbol{\lambda}$ produced by the third-order MEC. 52
4.10 Position, speed, and acceleration profiles produced by the third-order MEC. 53
4.11 Position, speed, and acceleration profiles produced by the SnG. 54
4.12 Domain of $\boldsymbol{\lambda}$ produced by the SnG. 55
4.13 Collision course displayed inside the configuration space. 57
4.14 Possible shapes of the collision set. 62
4.15 Gradient descent for computation of the starting points. 65
4.16 Construction of $\boldsymbol{\eta}'_a$ and $\boldsymbol{\eta}'_b$. 66
4.17 Approximation of ∂C via a quadratic Bézier curve. 67
4.18 Average cost over time. 72
4.19 Distribution of delays caused by the intersection. 73

5.1 Two critical hyperbolae determining the stability region. 83
5.2 The (γ, κ)-plane divided into regions A, B, C, and D. 85
5.3 Chain communication topology corresponding to the digraph $\mathcal{G}_C$. 90
5.4 Positions over time for the network topology $\mathcal{G}_C$. 91
5.5 Velocities over time for the network topology $\mathcal{G}_C$. 92
5.6 Representation of $\mathcal{G}_{C'}$. 92
5.7 Positions over time for the network topology $\mathcal{G}_{C'}$. 93
5.8 Velocities over time for the network topology $\mathcal{G}_{C'}$. 93

5.9 Velocities over time for the network topology $\mathcal{G}_C$ with delay. 95
5.10 Velocities over time for the network topology $\mathcal{G}_C$ with critical delay. 96

List of tables

4.1 Pairs of trajectories that can produce a collision. 61

5.1 Simulation values used in the delay-free scenario. 91
5.2 Simulation values used in delayed communication scenario. 95

List of publications

1. **Gianvito Difilippo**, Maria Pia Fanti, Giambattista Fiume, Agostino Marcello Mangini, and Nicolas Monsel, "A cloud optimizer for eco route planning of heavy duty vehicles," in 2018 IEEE Conference on Decision and Control (CDC), pp. 7142–7147, 2018.

2. Maria Pia Fanti, Agostino Marcello Mangini, Alfredo Favenza, and **Gianvito Difilippo**, "An eco-route planner for heavy duty vehicles," IEEE/CAA Journal of Automatica Sinica, vol. 8, no. 1, pp. 37–51, 2021.

3. **Gianvito Difilippo**, Maria Pia Fanti, and Agostino Marcello Mangini, "Maximizing convergence speed for second order consensus in leaderless multi-agent systems," IEEE/CAA Journal of Automatica Sinica, vol. 9, no. 2, pp. 259–269, 2022.

4. **Gianvito Difilippo**, Maria Pia Fanti, and Agostino Marcello Mangini, "A leaderless consensus algorithm in networks of vehicles with time delays," in 2022 European Control Conference (ECC), pp. 169–174, 2022.

5. **Gianvito Difilippo**, Maria Pia Fanti, and Agostino Marcello Mangini, "A consensus protocol for connecting automated vehicles at signal-free intersection," in 2022 IEEE 61st Conference on Decision and Control (CDC), pp. 6568–6573, 2022.

Acknowledgments

The writing of this thesis is the culmination of a tortuous path that I wasn't sure about undertaking to begin with. Pursuing a Ph.D. degree while working full-time is no joke, and on countless occasions, I was on the verge of giving up. Inspiration for continuing my research eluded me for the bulk of my Ph.D. years. Until, at the very last attempt, with a spark of intuition, I came up with this theory with support functions and stuff.

Why didn't I fail (completely)?

Behind works like this there is much more than a single student/researcher trying to do something original in their life. In my case, there are many people that directly or indirectly contributed to this. First and foremost, my family. It would be pointless to list the reasons why I should thank them, so I will indicate just one: they have believed in me when even I did not. But they are not alone in this. Michaela has walked beside me every step of the way. Her unwavering support has been fundamental in keeping me motivated all along. I'm incredibly lucky to have such a wonderful person by my side. Then, I want to mention my lifetime friends: Leo, Maria Luisa, Francesca, Rossella, Filippo, Michele, Alessandro, and all those I've shared a piece of my path with. Together, we have grown, celebrated successes, and weathered failures. Knowing that they've always been there fills me with gratitude.
Beyond this invaluable emotional support, I want to give credit to other individuals who contributed to this work on various fronts. I am deeply grateful to my academic advisor, Prof. Fanti, who guided me with helpful insights, and whose patience and wisdom pushed me through the obstacles. I extend my thanks to Agostino and Walter for their valuable opinions shared during our weekly meetings. Finally, I also want to thank my boss, Giuseppe, for allowing me to continue my studies while working for our company.
Well, that's it. Hope I didn't miss anyone, but if I did, blame it on my last-minute rush. If you're carefully reading the acknowledgments in the hope of spotting an Easter egg, I'm afraid that - this time - there are none. But only because I didn't have enough time!

Chapter 1

Introduction

For more than a decade, most of the world's population has been living in urban areas. In 2018 this percentage was 55%, which is expected to increase to 68% in 2050 [1]. This phenomenon is called urbanism and is a process that began in the nineteenth century, in conjunction with industrialization. Understanding why more and more people feel encouraged to move towards cities is quite easy: cities host a great wealth of services, infrastructure, business, and job demand. Therefore, the causes of urbanism can be summed up in a single motive: the desire for a better life. Increasing urbanization is a great opportunity for growth, but also a serious risk. Well-planned urbanization can maximize the benefits of high population density, such as the reduction of fuel and energy needed for transport, or the lower environmental impact in terms of occupied area. However, unorganized urbanization can result in unsustainable development, pollution, and urban degradation. A surge in the number of vehicles on the road is the natural consequence of the urban migration trend. Worldwide car sales grew to around 67.2 million automobiles in 2022, up from around 66.7 million units in 2021. In recent years, the sector experienced a hit due to world-changing events, such as the COVID-19 pandemic and the Russian war on Ukraine. Despite this, vehicle sales are expected to keep rising in the next years [2].

Urban planning and traffic management is a major challenge in reducing road congestion, which is forecasted to increase by 60% by 2030, compared to the levels observed in 2017 [3]. In addition to creating obvious inconveniences, traffic jams reduce people's productivity and cause waste of fuel, with consequent pollution. A study by The Texas A&M Transportation Institute [4] showed how in 2021, the average American commuter has spent 54 hours a year stuck in traffic - more than a working week! This translates into a total of 8.7 billion hours lost each year and 13.3 billion liters of fuel burned, for a total cost of 190 billion dollars.

Intersections play a critical role in influencing the efficiency of traffic management systems within urban areas. Beyond the significant contribution of intersections to congestion, crashes at crossroads constitute over 40% of all accidents, making it the second-largest category of incidents on the road [5]. Evidence from various studies highlights the impact of driver error on both traffic congestion and accidents, as human errors are accountable for 75% of all roadway crashes [6]. Other estimates indicate that more than 50% of all accidents are caused by actions taken voluntarily because of evaluation errors; these percentages include accidents due to driving while drunk or under the influence of drugs, fatigue, and nervousness [7].

Without significant improvements in driving behavior, the primary solution is reducing traffic density. Remarkably, in conditions of high traffic density, a decrease in the number of vehicles yields more than a proportional enhancement in traffic flow. Two approaches emerge

for diminishing traffic density: building wider roads and reducing the number of circulating vehicles. Enlarging roads offers short-term benefits, but an efficient road also attracts more motorists, ultimately reinstating congestion levels. Therefore, the more sustainable option lies in reducing the number of vehicles by establishing an efficient public transport system, which is not the direction the majority of people and most governments are heading towards.

Relying on the spontaneous improvement of human driving skills seems unlikely, but artificial systems have proven to be a promising alternative. Computers can already fly a passenger jet better than a trained human pilot, so it is natural to expect the same thing to happen with cars. Self-driving cars, also known as autonomous vehicles (AVs), are becoming increasingly common, promising to revolutionize the realm of transportation. These vehicles use advanced technologies, including artificial intelligence and sensor systems, to navigate and operate without direct human intervention. Research in this sector has been making great strides in recent years: companies like Waymo, Tesla, and others have been pioneering autonomous driving, finally bringing this staggering technology to the public; today, dozens of automotive companies are testing their AVs. With such technology available, people should no longer deal with the tedious task of driving in traffic. Furthermore, the efficiency of the road network will no longer be determined by the presence of human drivers, but by the automatic driver coordination mechanisms. The concept of using a fully automated vehicle system dates back to 1939 when General Motors presented their idea of vehicles without a human pilot at the World's Fair in New York; their concept cars were supposed to be controlled by mechanical actuators via radio waves. In 2003, Griffith University's Intelligent Control Systems Laboratory completed what, according to them, is the first on-the-road demonstration of AVs crossing intersections without traffic lights or road signs, although at speeds much lower than normal. This was achieved cooperatively, on a first-come, first-served strategy [8].

When AVs become connected, they exchange information and are known as connected autonomous vehicles (CAVs). Vehicle-to-vehicle (V2V) and vehicle-to-infrastructure (V2I) communication refer to the ways vehicles can connect to the surroundings to exchange data and, as of today, there are two major standards: ITS-G5 in Europe [9] and DSRC in US [10]. With this technology available, a lot of effort has been put into investigating the possibilities of Autonomous Intersection Management (AIM) systems. In contrast to traditional traffic control systems, which mostly consist of signalized intersections and timed semaphores, AIM leverages the real-time communication and coordination capabilities of CAVs. These vehicles exchange information about their trajectories, speeds, and intentions, allowing them to navigate intersections efficiently without the need for traffic signals, significantly improving overall traffic efficiency and drastically minimizing the risk of accidents and fatalities.

The distributed control problem of multi-agent systems (MASs) also received tremendous attention in the last decades due to its applications in different areas [11], including control of CAVs at autonomous intersections. MAS control involves the study of systems of independent agents that operate, collaboratively or antagonistically, in a shared environment; in the context of AIM, these agents are the CAVs and effective coordination among them is pivotal for reaching the desired objectives. In particular, distributed approaches to the coordination problems of multi-agent systems hold numerous advantages: low operational costs, fewer system requirements, high robustness, strong adaptivity, and flexible scalability [11]. In the context of CAVs, they aim to leverage the computational power, sensors, and other features of autonomous vehicles.

As technology continues to advance and regulatory frameworks evolve, the integration of

AVs into transportation systems holds the promise of becoming the future of mobility.

1.1 Research objectives

The available literature on AIM and MASs is very extensive. In the AIM field, various surveys [12, 13, 14, 15] have categorized the approaches to AIM, differentiating them based on their methodologies, main objectives, and the mathematical and engineering tools employed to tackle this complex problem. Despite this, very few works have given importance to the design of solid mathematical foundations to the modeling of intersections and to the problem of crossing the intersection. Although there are some noteworthy attempts to standardize the concept of collision, most works rely on pessimistic or overapproximative approaches, often focusing on other aspects of the model. Instead, a common approach is to model of the intersection crossing as an optimization problem. In the context of distributed control of MAS, leader-following control strategies are more common than leaderless algorithms. Moreover, while the problem of maximizing the convergence speed has been resolved for first-order systems, challenges have persisted for second-order systems.

The primary objective of this thesis is to propose a robust mathematical framework to model the problem of autonomous intersection management and collision detection and avoidance. A fundamental aspect of this research involves validating the proposed model through comprehensive numerical simulations. A secondary objective is to delve into a specific scenario within the domain of MASs, specifically related to AIM. The intention is to contribute to the existing body of literature by addressing the problem of maximizing the convergence speed of second-order leaderless systems and evaluating their stability in the presence of time delays.

1.2 Contributions

In the context of AIM, the main contributions of this thesis are reported hereafter.

1. The primary contribution is the formulation of the intersection crossing problem. This is a functional optimization problem that models the crossing of intersections with a focus on optimizing vehicle objectives while ensuring the absence of collisions.

2. The thesis investigates how the kinematics of vehicles approaching intersections influence the design of an algorithm candidate to solve the intersection crossing problem.

3. A significant contribution is the definition of a general collision model along with analytical collision conditions. Furthermore, reasonable approximations are proposed to mitigate the analytic complexity of the subject.

4. A novel contribution is in the introduction of convex analysis tools to enhance collision detection and avoidance. This approach is inspired by robotic trajectory planning and provides a more rigorous and efficient methodology for addressing collision-related challenges.

5. Based on the previous results, a distributed algorithm is designed to tackle the intersection crossing problem.

In the field of MASs, the following contributions are given.

1. A second-order leaderless protocol to allow agents to form a uniformly spaced string without the need for a leader is proposed.
2. The parameters of this protocol are optimized to obtain the fastest convergence speed, while avoiding oscillations, in the absence of delays.
3. The effect of delays on the system is studied and the conditions of stability are assessed.

1.3 Thesis outline

This thesis is organized into six chapters. This chapter introduces the context of my work, outlining the research objectives and the main contributions. Chapter 2 gives an overview of the related literature by analyzing the state-of-the-art approaches to both AIM and MASs. Chapter 3 delves into the mathematical tools needed to develop the results exposed in the following chapters. It contains definitions and explanations of convex analysis, optimization theory, dynamical systems theory, graph theory, and consensus algorithms. Additionally, the chapter introduces some useful results that I was not able to find in the mathematical literature. Chapter 4 introduces the optimization crossing problem and establishes the mathematical framework proposed for its resolution. Based on the introduced problem, the chapter ends with the description of a distributed algorithm for its solution, validated through numerical simulations. Chapter 5 is about the leaderless control of multi-agent systems: it introduces the main results of optimizing the convergence speed of the proposed second-order protocol and its sensitivity to delays, validating the obtained results via numerical simulations. Chapter 6 contains the conclusions and highlights topics for future exploration and development. Appendix A at the end of the thesis consists of proofs of some results presented during the exposition.

1.4 Notation

Vectors and matrices are denoted in boldface, such as $\boldsymbol{a}$ and $\boldsymbol{A}$, respectively. The vector $\mathbf{0}_n$ represents a vector of n components composed entirely of zeros, whereas $\mathbf{1}_n$ denotes a vector of n components composed entirely of ones. Similarly, $\boldsymbol{O}_{n,m}$ represents an n by m matrix consisting of zeros, and $\boldsymbol{I}_{n,m}$ represents an n by m matrix consisting of ones. When referring to square matrices, $\boldsymbol{O}_n$ and $\boldsymbol{I}_n$ denote n by n matrices composed of zeros and ones, respectively. The inverse of a matrix $\boldsymbol{A}$ is written as $\boldsymbol{A}^{-1}$. The transposed of a matrix $\boldsymbol{A}$ is denoted by $\boldsymbol{A}^\mathsf{T}$. By combining these notations, $\boldsymbol{A}^{-\mathsf{T}}$ indicates the transposed of the inverse of $\boldsymbol{A}$. The inner product of vectors $\boldsymbol{a}$ and $\boldsymbol{b}$ is denoted as $\langle \boldsymbol{a}, \boldsymbol{b} \rangle$ or $\boldsymbol{a}^\mathsf{T}\boldsymbol{b}$. $\|\cdot\|$ denotes the Euclidean norm, i.e., $\|\boldsymbol{a}\| = \sqrt{\langle \boldsymbol{a}, \boldsymbol{a} \rangle}$. Block matrices are denoted by stacking vectors or matrices horizontally and/or vertically, i.e.,

$$\boldsymbol{A} = \begin{bmatrix} \boldsymbol{A}_{1,1} & \boldsymbol{A}_{1,2} \\ \boldsymbol{A}_{2,1} & \boldsymbol{A}_{2,2} \end{bmatrix}, \quad \boldsymbol{P} = \begin{bmatrix} \boldsymbol{p}_1 & \boldsymbol{p}_2 \end{bmatrix}.$$

The i-th component of a vector is denoted by the subscript i; often, when reporting the expression of a vector on a single line, the notation $\boldsymbol{a} = \begin{bmatrix} a_1 & \cdots & a_n \end{bmatrix}^\mathsf{T}$ is used, where

the transposition operator means transforming a row vector into a column vector. When operations classically defined for scalar values are applied to vectors, they are considered to be applied component-wise; in particular, $|\boldsymbol{a}| = \begin{bmatrix}|a_1| & \cdots & |a_n|\end{bmatrix}^\mathsf{T}$ and the writing $\boldsymbol{a} > \boldsymbol{b}$ means $a_i > b_i$, $\forall i = 1, ..., n$. Sets are normally named with capital letters and, occasionally, with calligraphic capital letters, e.g., $\mathcal{N}$. $\mathcal{S}(\boldsymbol{A})$ and $\mathcal{K}(\boldsymbol{A})$ indicate the span and the kernel of matrix $\boldsymbol{A}$, respectively. $\mathcal{P}(S)$ denotes the power set of S. $\mathbb{N}$, $\mathbb{R}$, and $\mathbb{C}$ denote the sets of natural, real, and complex numbers, respectively. $\bar{\mathbb{R}}$ is the extended set of reals, that is, $\bar{\mathbb{R}} = \mathbb{R} \cup \{+\infty\}$. $\mathbb{R}_{>a}$ indicates the set of the real numbers strictly greater than a; $\mathbb{R}_{<a}$, $\mathbb{R}_{\geq a}$, and $\mathbb{R}_{\leq a}$ have analogous meaning. $\mathbb{R}_+$ denotes the set of strictly positive numbers, hence $\mathbb{R}_+ = \mathbb{R}_{>0}$. $\mathbb{R}^n$ and $\mathbb{R}^{n\times m}$ are the sets of n-dimentional real vectors and the set of n by m real matrices. Pairs of numbers enclosed in brackets indicate real intervals, with the direction of the brackets meaning openness or closedness on that interval endpoint, i.e., $[a, b]$ indicates a closed interval and $]a, b[$ indicates an open interval. Integer intervals are denoted by $[\![a, b]\!]$, i.e., $[\![a, b]\!] = [a, b] \cap \mathbb{N}$. The writing $\{c_i\}_{i\in[\![a,b]\!]}$ compactly indicates $\{c_i \,:\, i \in [\![a, b]\!]\}$. The restriction of a function f to the $[a, b]$ interval is denoted by $f_{[a,b]}$. Given a vector $\boldsymbol{a} \in \mathbb{R}^n$ and a set $S \subset \mathbb{R}^n$, with abuse of notation, I write $\boldsymbol{a} + S$ meaning $\{\boldsymbol{x} + \boldsymbol{a} \,:\, \boldsymbol{x} \in S\}$. Analogously, given a matrix $\boldsymbol{A} \in \mathbb{R}^{m\times n}$, the notation $\boldsymbol{A}S$ indicates the set $\{\boldsymbol{A}\boldsymbol{x} \,:\, \boldsymbol{x} \in S\}$. Similarly, the image of a set S under a function g is indicated by $g(S)$. The derivative of a function $g(x)$ of real variable x is denoted by $g'(x)$ or by $\dot{g}(x)$ when the variable represents time. $\nabla f(\boldsymbol{x})$ is the gradient of function f and $\nabla^2 f(\boldsymbol{x})$ is its Hessian matrix. $\partial f(\boldsymbol{x})$ denotes the subdifferential of f. For $c \in \mathbb{C}$, $\Re(c)$ and $\Im(c)$ are its real and imaginary parts, respectively; $\jmath$ is the imaginary unit, therefore $c = \Re(c) + \jmath\Im(c)$. The magnitude of a complex number c is denoted by $|c|$.

Chapter 2

Literature review

2.1 Autonomous intersection management

AIM is a relatively new research topic, but the related literature is already vast. One of the most recent surveys [12] analyzed more than 1200 relevant publications including surveys, reviews, and short articles taking into account main aspects of intersection management for heterogeneous vehicles at the signalized, non-signalized, and hybrid intersections. Both [12] and [13] identify the main goals set by the research as safety, efficiency, passenger comfort, and ecology, as shown in Figure 2.1.

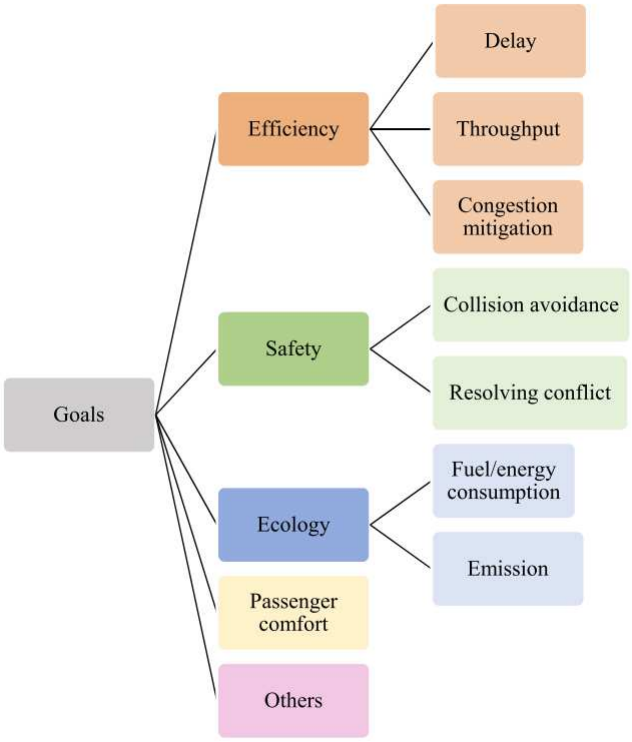


Figure 2.1: *Research goals and sub-goals. [13]*

Safety is the tenet of any AIM system: it is unacceptable for an intersection management protocol to introduce new hazards. Collision avoidance is the primary safety topic, as any

AIM algorithm must be able to anticipate collisions and avert them by controlling the vehicles appropriately. The other main objective is to improve the efficiency of intersections. This can be done by increasing its throughput, which is the number of vehicles passing through it in a given unit of time, or by reducing the delay introduced by it or, moreover, by reducing the possibilities of congestion. A secondary objective, of ecological scope, is to increase energy efficiency, for example by trying to make the speed of AVs as constant as possible. Furthermore, ensuring the comfort of the passengers, who cannot be subjected to intense accelerations, is also desirable.

Intersection traffic coordination methodologies are typically classified into two primary groups: centralized and decentralized. Centralized approaches require Vehicle-to-Infrastructure (V2I) communication. Centralized control strategies typically rely on an intersection manager for traffic control. In contrast, decentralized strategies use Vehicle-to-Vehicle (V2V) communication and operate in a distributed fashion, with vehicles coordinating cooperatively by exchanging information and making decisions locally. Additionally, hybrid methodologies may occasionally emerge, wherein the infrastructure is less involved, or a vehicle is temporarily elected as a leader and assumes responsibility for intersection coordination.

With respect to intersection modeling, two main approaches are used: Spatio-Temporal (ST) reservation and Trajectory Planning (TP). The former is based on the idea of dividing the intersection center, into a square grid. Each square, at a particular interval of time, is a resource shared among the AVs, which have to exclusively reserve it for passing the intersection without causing collisions. The granularity of the subdivision depends on the intensity of the traffic: when the flow of vehicles is low, a coarse grid is sufficient to ensure both safety and efficiency; conversely, a high volume of traffic requires finer subdivisions, with consequent increase in computational complexity. The latter is based on analyzing the fixed trajectories that AVs follow and collision avoidance is performed with various approaches, specifically safe pattern detection and priority-based.

2.1.1 Centralized approaches

Centralized approaches are characterized by great control and management capabilities over vehicles, the capacity to handle high computation loads, and less network overhead with respect to the alternatives.

Starting from 2004, with a series of publications, K. Dresner and P. Stone became pioneers in the research of the AVs cooperation algorithms, introducing a centralized intersection management system called AIM (Autonomous Intersection Manager), presented in 2008 [16]. Their work is certainly a milestone in the literature and many authors based their algorithms on it, showing improvements or variations.

The idea behind this system is to place a centralized controller, called Intersection Manager (IM), at the intersection, able to communicate with all the vehicles that approach it. The vehicles, called driver agents, make a request (call ahead) to the IM before reaching the intersection to reserve the part of the road of interest in the time interval of interest. The controller can accept or reject the request based on the intersection management policy. A simplified diagram of the communications between the two types of agents is shown in Figure 2.2.

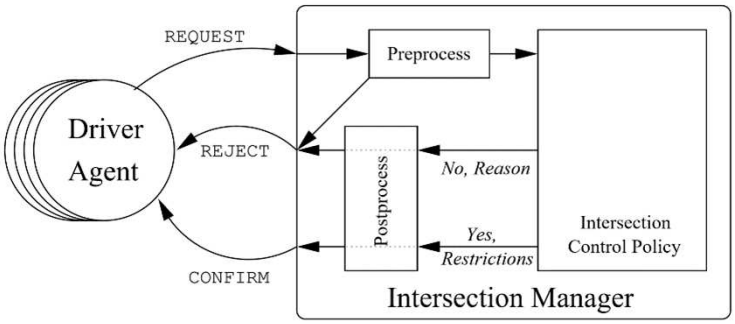
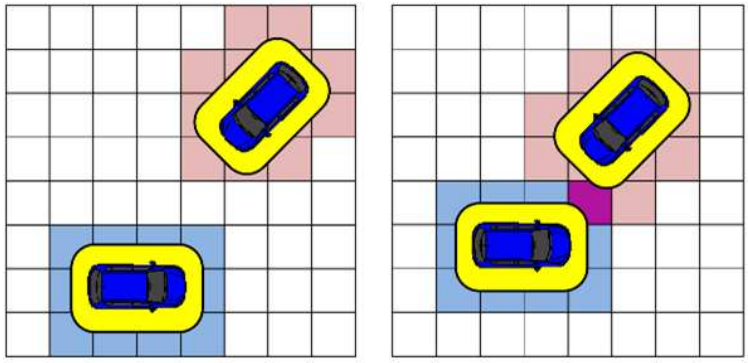


Figure 2.2: *AIM scheme. [16]*

One of these policies is called First Come First Served (FCFS) policy, in which the intersection space is divided into a $n \times n$ grid; at the request, the controller simulates the trajectory of the vehicle, using the parameters provided in the request by the vehicle, and verifies possible collisions with other vehicles that have reserved space in those moments. Figure 2.3 shows two possible cases: a successful reservation and a rejected reservation.



(a) *Successful* **(b)** *Rejected*

Figure 2.3: *Examples of successful and rejected reservations. [17]*

The proposed FCFS-Light strategy takes into account the presence of human drivers: the controller reserves special corridors for those, in coordination with the green light duty cycle. Furthermore, it is shown how the assignment of special priorities to emergency vehicles does not appreciably reduce the efficiency of the intersection. Finally, the authors state that this protocol, combined with simple safety rules, would reduce accidents by 80% [16]. This work has had a massive impact on the literature: many authors have studied and analyzed this system, pointing out its strengths and weaknesses. [18] proved that AIM risks penalizing some of the vehicles in unbalanced traffic conditions, leading to long waiting times. Besides AIM by Dresner and Stone, there is a good number of centralized intersection management algorithms. A rule-based system inspired by the work of Dresner and Stone is the Batch-Light policy, introduced by Wei *et al.* in [19], where they propose a strategy to solve the problem of vehicle starvation caused by unbalanced traffic. The main difference with [16] is that requests are processed in batches: the time is divided into slots and the IM collects requests within a time slot and processes them together at the end of the slot. Dahlberg *et al.* [20] propose a rule-based algorithm with trajectory planning considering the dynamical model of the vehicle and PID control. Vasirani *et al.* [21] suggest the idea of a computational

market: vehicles buy permission to cross the intersection from the IM; the price is not fixed and depends on the traffic conditions, with vehicles bidding for acquiring the spatio-temporal reservation. The same authors extend their work to multiple intersections in [22]. Wang *et al.* [23] determine the possibility of collisions at T-shaped intersections based on data produced by GPS, IMU, and other vehicle sensors; real-time state information are fed to an algorithm that detects collisions between the vehicles approaching the intersection. Zhang *et al.* [24] introduce csPriorFIFO, a V2I priority-based service-oriented reservation scheme mechanism, considering heterogeneous traffic participants, while ensuring Quality of Service (QoS) in line with vehicle priorities. Morales Medina *et al.* [25] optimize the crossing sequence by designing an intersection access management methodology; Cooperative Intersection Control (CIC) is employed to regulate the safety and efficiency of vehicles in the intersection, whereas a high-level hybrid queuing model describes the dynamics of the vehicle queues at each lane. A two-stage optimization algorithm is designed by Feng *et al.* [26], with signal optimization at the first stage and vehicle trajectory control at the second stage. Their objectives contemplate reducing fuel consumption and travel time; moreover, sensitivity analysis is carried out with respect to the velocities and accelerations of vehicles. Lu *et al.* [27] describe a trajectory discretization method named Discrete-Time Occupancies Trajectory (DTOT) to model the movement of vehicles inside the intersection. The same authors [28] then study a genetic algorithm that prioritizes emergency vehicles while reducing the delay caused to other vehicles; later, [29] they propose the usage of Mixed Integer Programming (MIP) to generate vehicle trajectories for reducing delays while ensuring safety. Critical situations are also considered in [30], where Chang *et al.* present a reservation method for facilitating evacuation in case of an emergency occurring at the intersection. Xu *et al.* [31] applied optimal control in a cooperative V2I scenario with existing traffic signals to reduce travel time and fuel consumption. The same technique is employed by Meng *et al.* [32] to achieve optimal acceleration and speed profiles for vehicles approaching an intersection regulated by traffic lights. The proposed strategy aims at avoiding stops at the intersection, achieving stop-free traffic flow. Crossroads+ by Khayatian *et al.* [33] is a time-sensitive centralized approach for autonomous intersections that encompass practical concepts, such as synchronization, safety margins, communication delays, and acceleration saturation. Models that account for human drivers are also present in literature: Katriniok *et al.* [34] use Model Predictive Control (MPC) to issue speed advice to the drivers and assess the problem of collision avoidance considering uncertainties and driver reaction times.

Centralized methods are often easier to work with and to reason about because of the control logic being confined within a single self-consistent unit; however, they also present shortcomings and criticalities.

- Poor scalability: this kind of system are not scalable due to the high computational capabilities that are required to coordinate dozens of vehicles at the same time.
- Poor reliability: a central infrastructure agent is both a bottleneck and a single point of failure of the system.
- High cost: by removing traffic lights and road signs, almost all intersections should be regulated by an IM. This would produce high costs, and a waste of resources if the intersection is scarcely frequented.

To mitigate these weaknesses, hybrid approaches with V2X (V2I + V2V) communication. Katriniok *et al.* [35] investigate the role of V2X communication in enhancing the capabilities

of automated vehicles, focusing on an intersection priority management (IPM) system empowered by MPC that aims to streamline vehicle passage without traditional traffic control measures. Liu *et al.* [36] present TP-AIM, a cooperative scheduling mechanism whose task is to assign vehicle priorities and to plan trajectories to ensure safe and efficient passage. A window-searching algorithm identifies collision-free trajectories with minimal delay, while also considering backup trajectories in the case of failures. Azimi *et al.* [37] introduce the Ballroom Intersection Protocol (BRIP), a spatio-temporal technique designed to facilitate the safe and efficient passage of autonomous vehicles through intersections. BRIP prioritizes maximizing intersection throughput, eliminating the need for stopping behind or inside the intersection area. Kowshik *et al.* [38] propose a hybrid architecture for intelligent intersections, integrating centralized coordination with distributed decision-making to ensure safety and high performance. The approach involves each car maintaining an updated contingency plan and a dynamically changing partial-order relation to guard against worst-case behaviors. Wang *et al.* [39] propose a vehicle trajectory collision warning system, with trajectory prediction based on V2V and V2I communication. Real-time collision detection is achieved through a proposed collision detection algorithm, with time to collision (TTC) serving as a risk indicator. The paper of Wu *et al.* [40] proposes a novel approach to traffic control at intersections utilizing V2I and V2V communications, with vehicles competing for passage rights. The problem is modeled as a variant of the classic mutual exclusion problem, with both centralized and distributed algorithms designed to solve it.

2.1.2 Decentralized approaches

In contrast to centralized approaches, decentralized strategies can help assess these points by leveraging by leveraging the capabilities inherent in individual vehicles, such as computing power, sensors, communication devices, etc. The computation burden is distributed among many agents, eliminating the need for expensive centralized controllers. Besides, the overall robustness of the system is improved as failures occurring on individual vehicles are less likely to lead to system-wide breakdowns. Overall, decentralized approaches hold the potential to address scalability and cost concerns while improving the resilience and efficiency of intersection management systems.

Khoury *et al.* [41] propose two new distributed algorithms: the first one still partly relies on the presence of an agent at the intersection that provides vehicles with some information, while the second version eliminates this agent and relies only on the sensors of vehicles. The result is a good improvement in intersection efficiency and a guarantee of safety if the protocol is respected. Bazzi *et al.* [42] define a distributed algorithm for smart vehicles (with human driver) to simulate the presence of traffic lights at the intersections that are lacking. The algorithm is based on a platooning strategy and the consequent election of a leader, who will be responsible for negotiating the crossing time and space with other leaders. An example of optimal control applied to distributed scenarios is given by Jiang *et al.* [43], which suggests a Least Square Regulator (LQR) that is parallelized and distributed and is based on the ALADIN method (Augmented Lagrangian-based Algorithm for DIstributed Non-convex optimization). Each vehicle solves its own optimization problem and exchanges some useful information with neighboring vehicles to achieve coordination and avoid collisions. The work of Hassan *et al.* [7] concerns a heuristic method where vehicles within 200 meters of the intersection elect a leader for each lane. Then, the heads alternatively assume the role of schedulers. De Campos *et al.* [44] propose a system in which conflicts are solved

in a cooperative and distributed way, via a heuristic algorithm based on the intersection modeling, resulting in a fast algorithm that is suitable for online computation. Tu *et al.* [45] introduce Forwards, a map-free intersection collision-warning system utilizing a triple Kalman filter-based estimator for calibrated motion state information. The system predicts short-term trajectories of vehicles and issues hierarchical warnings based on collision-detection algorithms. Tomas-Gabarron *et al.* [46] address collision avoidance in scenarios requiring high-speed vehicles to generate evasive maneuvers within short time intervals, reconducting the issue to a trajectory generation problem and optimizing routes based on maximizing lateral distances between vehicles and obstacles; trajectories are computed using a gradient-descent-based methodology. Lu *et al.* [47] present a set of rules for determining the sequence of vehicles passing through uncontrolled intersections, derived from road traffic safety laws. Based on these rules, vehicles make decisions to preempt or yield to others using information from vehicle-vehicle communication. Rahmati *et al.* [48] explore the potential of connectivity and automation to improve safety and reduce congestion. The study considers both human and robot drivers and focuses on developing a game theory-based framework able to achieve high accuracy in predicting real drivers' choices. Murgowski *et al.* [49] address optimal control of autonomous vehicles for intersection crossing by modeling the optimal crossing sequences as a convex program. The method transforms the problem into the space domain, enhancing effectiveness. Constrained optimization is utilized by Malikopoulos *et al.* [50] for coordinating connected and automated vehicles, and analytical solutions are derived in closed form using Hamiltonian analysis to minimize a cost function associated with the control input. Joerer *et al.* [51] present a metric to estimate vehicle collision probability based on the exchange of beacon messages, with respect to their trajectories. De Campos *et al.* [52] introduce an active safety system for frontal collision detection and prevention/mitigation in complex traffic scenarios, including a probabilistic motion prediction algorithm, a threat assessment method, and a reachability-based decision-making protocol for emergency intervention. A consensus-based cooperative control algorithm is developed by Mirheli *et al.* [53], formulating the trajectory planning as a Mixed-Integer Non-Linear Program (MINLP) to minimize travel time and speed variations while avoiding near-crash conditions. Azimi *et al.* [54] design efficient and reliable intersection protocols to prevent vehicle collisions and increase traffic throughput, demonstrating the benefits of the proposed V2V intersection protocols, and highlighting significant improvements in throughput while avoiding deadlocks. Some works make use of the navigation function, which is an idea coming from the realm of robotics, whose goal is creating feasible paths and avoiding obstacles; in particular, its gradient is used as a control input for the AVs. For example, Makarem *et al.* [55] present a novel decentralized navigation function for coordinating autonomous vehicles at intersections, aiming to avoid deadlock situations and optimize energy consumption, and considering the expected time of arrival to the intersection for collision avoidance, accounting for different inertias of vehicles, and prioritizing heavier vehicles for smoother motion and energy optimization. Katriniok *et al.* [56] introduce a distributed motion planning scheme for coordinating highly automated vehicles at road intersections, addressing the inherently nonconvex nature of the problem by employing Proximal Averaged Newton method for Optimal Control (PANOC). Each agent solves a real-time nonlinear model predictive control problem and communicates its trajectory to other agents, even with conflicting objectives. Conditional constraints enable agents to decide whether to wait at a stopping line if safe crossing is not possible.

2.1.3 Related works

This thesis is focused on decentralized scenarios, with TP-based collision detection and avoidance conditions. Among the analyzed papers, there emerge some works that share similar concepts and ideas.

Hafner *et al.* [57, 58] tackle the collision avoidance problem by defining the concept of *capture set*. The capture set corresponds to the set of vehicle displacements, such that given the current vehicle velocities, no control signal can prevent an eventual collision [57]. This idea is similar to the concept of *collision set* that is described later in this thesis, although with substantial differences. In particular, in [57] they present experimental results for an active control Intersection Collision Avoidance (ICA) system implemented. Safety is ensured in potential collision scenarios by controlling vehicle velocities with automatic brake and throttle commands, ensuring they adhere to predefined upper and lower speed limits. Furthermore, model uncertainty and communication delays are explicitly addressed in [58]. An analogous route is taken by Colombo *et al.* [59], where the *maximally controlled invariant set* is described as the largest set of states for which there exists a control that avoids collisions. This set is complementary to the capture set and in Colombo's paper, it is used in the synthesis of the least restrictive controller for collision avoidance, proving that checking membership in the maximal controlled invariant set is NP-hard for a general model of vehicle dynamics. At the same time, they present an approximate algorithm with provable error bounds, utilized to design a collision avoidance supervisor whose complexity scales polynomially with the number of vehicles.

This work proposes a framework for modeling the problem of autonomous intersections, and so did Yu *et al.* in [60], where they consider in detail multiple kinds of collisions, addressing the challenges of diverse agent geometries and capabilities. The proposed framework is claimed to ensure accurate collision prediction and decision-making in various collision configurations.

Wang *et al.* [61] introduce a temporal-spatial collision warning method tailored for intersection accidents. By incorporating both time and spatial coefficients, the algorithm calculates the probability of accidents, considering safety braking distances. In this paper, attention is directed to straight paths only, with conflicting trajectories meeting at an angle. An exact geometric approach is undertaken by Belkhouche in [62], where a technique known as Virtual Configuration Space transformation (VCS) is employed to derive equations describing unfeasibility conditions inside the space of vehicle velocities. Moreover, the usage of a Lagrangian function enables the analysis of collaboration in terms of cost and time. The derived conditions do not consider turning at the intersection, with applicability to straight trajectories only. The same author [63] had introduced a collaborative approach for collision avoidance among autonomous vehicles at unsignalized intersections, framing the problem as an optimization task with safety constraints based on speed ratios, with explicit resolution of conflicts in a three vehicles-scenario.

The collision set approach I employ is used in robotics for path coordination of multiple robots. Leroy *et al.* [64] introduce a geometric approach to coordinate the motion of multiple mobile robots independently computing their paths to avoid collisions. Utilizing a bounding box representation of obstacles in a coordination diagram, the resolution-complete algorithm ensures collision-free trajectories. The study of collisions is conducted inside the *configuration space* of the robots: a point of the configuration space uniquely determines the position of robots and the collision set is deemed as the region of that space where a collision between

robots or fixed obstacles occur. The paper by Gregoire *et al.* [65] presents a priority-based framework for coordinating multiple robots with positive velocity along fixed paths, utilizing a configuration space approach. The framework is applied to intersection management, demonstrating optimality properties and achieving good performance in numerical experiments. The collision set inside the configuration space is the section of a cylinder, i.e., an ellipse, or a larger set derived from it, deemed as its closure under a certain priority, as depicted in Figure 2.4.

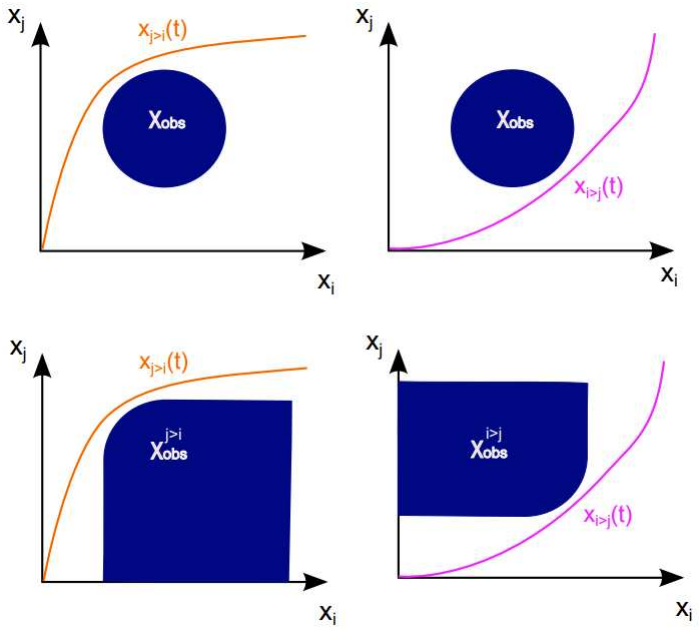


Figure 2.4: *Obstacle regions and their closures. [65]*

This corresponds to conflicts being addressed to straight trajectories only. The same ideas were used by Qian *et al.* [66], with the application of MPC and the study of particular maneuvers such as emergency braking. They also consider efficiency, comfort, and fuel economy requirements while maintaining system-wide safety.

Kneissl *et al.* [67] propose a distributed control methodology for coordinating AVs, where each vehicle computes its local trajectory using a model predictive control (MPC) law and communicates with relevant vehicles to optimize planned trajectories iteratively. They tailor the Jacobi over-relaxation (JOR) algorithm to vehicle coordination, guaranteeing collision-free trajectories with any-time feasibility and scalability. The work uses a configuration space approach, with the collision set referring to the whole intersection, resulting in a mutual exclusion policy. Another strategy leveraging collision sets is by Mo *et al.* : in [68] they introduce novel multiple-collision-set coordination strategies to enhance traffic throughput, proposing low-complexity methods based on multi-agent Q-learning frameworks.

The paper by Steinmetz *et al.* [69] introduces a Collision-Aware Resource Allocation (CARA) strategy for coordinating automated and connected vehicles at intersections via wireless communication. Compared to collision-agnostic strategies, CARA significantly reduces communication requirements without compromising safety, offering flexibility to adjust control performance and communication load based on specific scenarios. The collision conditions are studied inside the configuration space, with the collision set (named *bad set*) in the paper, evaluated for the whole shared region. They even consider probabilities of collisions

in the presence of uncertainties in the model.

Ultimately, Malikopoulos *et al.* [70] propose the subdivision of the intersection into zones and determine the *merging zone* (MZ) as the region of the intersection where conflicts between trajectories may happen. Their model considers that vehicles travel with constant speed while crossing the MZ.

2.2 Control of multi-agent systems

The distributed control problem of multi-agent networks received tremendous attention in the last decades due to its applications in different areas [11]. Each agent is a dynamical system and the problem of reaching an agreement on all or some components of their statuses is known as consensus problem. The consensus problem for first-order multi-agent systems has been largely studied considering different aspects: switching topology and time-delays [71, 72, 73]; nonuniform time-varying delays [74]; diverse time-delays and jointly-connected topologies [75]; stabilization problem with time delay controlled by a distributed PID regulator [76]; leaderless and leader-following consensus [77]. Moreover, several types of nonlinear vehicle dynamics can not be feedback linearized into single, but into double integrators and third order multi-agent systems [78]. Therefore, formal analysis of consensus problems for second-order systems is provided by [79, 80, 81, 82, 83, 84, 85, 86]. In particular, Yu *et al.* [84] demonstrate that the real and imaginary parts of the eigenvalues of the Laplacian matrix of the corresponding network play key roles in reaching consensus. In addition, in systems modeled by double-integrator dynamics, Qin *et al.* [85] investigate two kinds of different consensus strategies for multi-vehicle systems with a time-varying reference velocity under a directed communication topology. In [86], Santini *et al.* study a decentralized control action for platooning maneuvers in vehicular networks embedding the spacing policy information as well as all the time-varying communication delays. Recently, a more general class of high-power multi-agent systems described by an extension of second-order nonlinear models are studied in [87] and [88]. Consensus and distributed control of multi-agent systems also find applications in combination with adaptive neural networks, as shown in [89] where the authors simultaneously guarantee practical finite-time stability and asymptotic convergence. In the literature, both the leaderless consensus and the leader-following consensus problems have been studied, depending on whether or not a virtual leader specifies the global information [90]. More precisely, in a leaderless consensus problem, there does not exist a virtual leader, whereas in a leader-following consensus problem, there exists a virtual leader that specifies the objective for the whole group [77]. For example, Meng *et al.* [90] study a leader-follower consensus problem for a set of agents subject to control input saturation. In addition, Fu *et al.* [91] consider a distributed leader-following consensus for second-order multi-agent systems with non-convex velocity and control input constraints. On the contrary, a leader is not required in the approach proposed by Jafarian *et al.* [92], but it is mandatory that at least one agent of the network knows the reference velocity. For a leader-following multi-agent system, Wei *et al.* [93] study the consensus control of such systems with heterogeneous disturbances generated by the Brownian motion, developing an adaptive protocol based on Riccati inequalities. In addition, Yao *et al.* [94] integrate the distributed sliding-mode control algorithm to investigate the tracking control issue for second-order leader–follower multi-agent systems subject to nonlinearities. Some advantages of leaderless consensus with respect to the leader-following approach have been enlightened in the related literature. In particular,

leaderless consensus typically scales better and is more fault-tolerant than leader-following consensus [95]. Moreover, the presence of leaders also decreases the degree of autonomy of the network [96], since the leaders generate global desired trajectories of agents, whereas, in many practical missions, the agents need to reach autonomous agreement on an a priori unknown quantity. Furthermore, an important problem concerning consensus algorithms is their convergence speed in order to implement them in real applications. In linear systems, a measure of the convergence speed is the smallest non-zero real part (for the continuous-time case) or magnitude (for the discrete-time case) of the system eigenvalues. Some efforts in this field are performed also in [97, 98, 99, 100, 101, 102]. Yoonsoo *et al.* [97] propose an iterative algorithm for maximizing the second smallest eigenvalue of a state-dependent graph Laplacian. In [98] and [99], communication time delay is considered in the optimal consensus problem. Furthermore, Jiandong provides a closed form for the optimal gains of some consensus protocols in [100]. The work of Yu *et al.* [101] is concerned with the consensus convergence rate for second-order multi-agent systems: the fastest consensus convergence rate under the protocol is derived based on the assumption that all the eigenvalues of the Laplacian matrix are real. In [102], Xing *et al.* consider the explicit expression of the maximum convergence rate and analyze the effects of control parameters on the convergence rate. The convergence speed problems in discrete-time systems are recently analyzed in [103] and [104]. In [103] Eichler *et al.* maximize the convergence speed of multi-agent systems with discrete-time double-integrator dynamics, optimally choosing the free parameters of the consensus protocol. The same authors in [104] optimize the consensus protocol speed subject to a lower bound on damping.

2.2.1 Platooning

Multi-agent systems theory is also used in the context of autonomous intersection management. Some works evaluate the applicability of platoon-based algorithms for efficient scheduling and intersection crossing. The term *platooning* refers to a collection of vehicles that travel together [105]. The most recent works in the study of the control of such kind of formation consider a platoon a string made up of an infinite number of vehicles [106]. Although, for any platoon, the number of vehicles is finite, an infinite string is considered in the literature to study concepts like stability independently of the size of the platoon. The concept of *string stability* is used to describe the resiliency of the platoon to perturbations: intuitively, string stability implies uniform boundedness of all the states of the interconnected system for all time if the initial states of the interconnected system are uniformly bounded [107]. String stability is a tenet of platoon control: its use is fundamental in ensuring that the formation of many vehicles stays coordinated and is robust to disturbances for long periods. According to [108], the platoon formation, that is the formation of strings of vehicles moving at the same speed and in the same direction, would improve the efficiency of any intersection scheduling algorithm, besides reducing the communication overhead. If a group of vehicles moves in coordination as they are a single vehicle, the platoon leader alone can communicate with other leaders to schedule the intersection crossing. Timmerman *et al.* [109] investigate different platoon algorithms for increasing efficiency at the intersection, reporting interesting results in decreasing the mean delay introduced by the crossing; the authors also acknowledge the risk of platoon formations penalizing fairness in the crossing schedule.

Chapter 3

Preliminaries

The modeling and development of effective coordination and control strategies for autonomous vehicles requires some mathematical foundation to be discussed. In this chapter, I present key mathematical concepts and tools that form the backbone of the subsequent theoretical analysis.

3.1 Convex analysis

Convexity has been increasingly important in recent years in the study of extremum problems in many areas of applied mathematics [110]. Convex analysis is a branch of mathematics that focuses on the study of convex sets and convex functions and is significant in optimization because of its capacity to address a wide range of practical problems efficiently. I start by introducing some basic concepts of set theory and convex analysis. Most of the definitions and statements of this section are taken from [110].

A subset A of $\mathbb{R}^n$ is called an *affine set* if $(1-\lambda)\boldsymbol{x}+\lambda\boldsymbol{y} \in A$ for every $\boldsymbol{x}, \boldsymbol{y} \in A$ and $\lambda \in \mathbb{R}$. An $(n-1)$-dimensional affine set in $\mathbb{R}^n$ is called a *hyperplane*. For any non-zero $\boldsymbol{b} \in \mathbb{R}^n$ and any $d \in \mathbb{R}$, the set

$$H(\boldsymbol{b}, d) = \{\boldsymbol{x} \in \mathbb{R}^n \ : \ \langle \boldsymbol{b}, \boldsymbol{x} \rangle = d\}$$

is a hyperplane.

A subset C of $\mathbb{R}^n$ is said to be *convex* if $(1-\lambda)\boldsymbol{x}+\lambda\boldsymbol{y} \in C$ whenever $\boldsymbol{x}, \boldsymbol{y} \in C$ and $0 < \lambda < 1$. Given a vector $\boldsymbol{\zeta} \in \mathbb{R}^n$, with $\boldsymbol{\zeta} \neq \mathbf{0}_n$, a subset C of $\mathbb{R}^n$ is said to be $\boldsymbol{\zeta}$-*convex* if $(1-\lambda)\boldsymbol{x}+\lambda\boldsymbol{y} \in C$ whenever $\boldsymbol{x}, \boldsymbol{y} \in C$, $\boldsymbol{\zeta} = \boldsymbol{y} - \boldsymbol{x}$, and $0 < \lambda < 1$. A convex set is also $\boldsymbol{\zeta}$-convex for all non-null $\boldsymbol{\zeta} \in \mathbb{R}^n$.

Half-spaces are important examples of convex sets. For any non-zero $\boldsymbol{b} \in \mathbb{R}^n$ and any $d \in \mathbb{R}$, the sets

$$H_{\leq}(\boldsymbol{b}, d) = \{\boldsymbol{x} \in \mathbb{R}^n \ : \ \langle \boldsymbol{b}, \boldsymbol{x} \rangle \leq d\} \quad \text{and} \quad H_{\geq}(\boldsymbol{b}, d) = \{\boldsymbol{x} \in \mathbb{R}^n \ : \ \langle \boldsymbol{b}, \boldsymbol{x} \rangle \geq d\}$$

are called *closed half-spaces*. The sets

$$H_{<}(\boldsymbol{b}, d) = \{\boldsymbol{x} \in \mathbb{R}^n \ : \ \langle \boldsymbol{b}, \boldsymbol{x} \rangle < d\} \quad \text{and} \quad H_{>}(\boldsymbol{b}, d) = \{\boldsymbol{x} \in \mathbb{R}^n \ : \ \langle \boldsymbol{b}, \boldsymbol{x} \rangle > d\}$$

are called *open half-spaces*. These half-spaces are associated to the hyperplane $H(\boldsymbol{b}, d)$, so that one may speak unambiguously of the open and closed half-spaces corresponding to a given hyperplane.

The intersection of all the convex sets containing a given subset C of $\mathbb{R}^n$ is called the *convex hull* of C and is denoted by $\text{conv}(C)$. It is a convex set and, more precisely, the unique smallest one containing C.

A subset K of $\mathbb{R}^n$ is called a *cone* if it is closed under positive scalar multiplication, i.e. $\lambda\boldsymbol{x} \in K$ when $\boldsymbol{x} \in K$ and $\lambda > 0$. Such a set is a union of half-lines emanating from the origin. The *polar* of a cone K is defined to be the set of vectors that form an obtuse angle with all other vectors of K:

$$K^{\circ} = \{\boldsymbol{y} \in \mathbb{R}^n \ : \ \langle \boldsymbol{x}, \boldsymbol{y} \rangle \leq 0, \ \forall \boldsymbol{x} \in K\}.$$

Let C be a non-empty convex set in $\mathbb{R}^n$. We say that C *recedes* in the direction $\boldsymbol{\eta}$ if C includes all the half-lines in the direction $\boldsymbol{\eta}$ which start at points of C. Equivalently, $\boldsymbol{\eta}$ is a *direction of recession* of C. A convex set C admits a direction of recession if and only if it is unbounded. In other words, C recedes in the direction of $\boldsymbol{\eta}$, where $\boldsymbol{\eta} \neq \boldsymbol{0}_n$, if and only if $\boldsymbol{x} + \lambda\boldsymbol{\eta} \in C$ for every $\lambda > 0$ and $\boldsymbol{x} \in C$. The set of all vectors $\boldsymbol{\eta} \in \mathbb{R}^n$ satisfying the latter condition, with the inclusion of $\boldsymbol{\eta} = \boldsymbol{0}_n$, is be called the *recession cone* of C, indicated by K_C:

$$K_C = \{\boldsymbol{\eta} \in \mathbb{R}^n \ : \ \boldsymbol{x} + \lambda\boldsymbol{\eta} \in C, \ \forall \boldsymbol{x} \in C, \forall \lambda > 0\}.$$

The *Minkowski sum* (or *vector sum*) of two subsets A and B of $\mathbb{R}^n$ is denoted by $A + B$ and is defined as

$$A + B = \{\boldsymbol{a} + \boldsymbol{b} \ : \ \boldsymbol{a} \in A, \boldsymbol{b} \in B\}.$$

Let A be a convex subset of $\mathbb{R}^n$. A function $f : A \to \mathbb{R}$ is said to be convex whenever

$$f((1-\lambda)\boldsymbol{x} + \lambda\boldsymbol{y}) \leq (1-\lambda)f(\boldsymbol{x}) + \lambda f(\boldsymbol{y}), \quad \forall \boldsymbol{x}, \boldsymbol{y} \in A.$$

Moreover, if f is differentiable, it holds

$$f(\boldsymbol{y}) \geq f(\boldsymbol{x}) + \langle \nabla f(\boldsymbol{x}), \boldsymbol{y} - \boldsymbol{x} \rangle.$$

The *indicator function* is a function that indicates whether an element belongs to a specific set. The indicator function of a set C is defined as follows:

$$\chi_C(\boldsymbol{x}) = \begin{cases} 0, & \text{if } \boldsymbol{x} \in C, \\ +\infty, & \text{otherwise.} \end{cases}$$

3.1.1 Support functions

The conjugacy transformation is a functional that associates with any function f, a convex function f^*, called the conjugate of f. Consider an extended real-valued function $f : A \subseteq \mathbb{R}^n \to \bar{\mathbb{R}}$. The *conjugate function* of f is the convex function $f^* : A^* \subseteq \mathbb{R}^n \to \bar{\mathbb{R}}$ defined by

$$f^*(\boldsymbol{y}) = \sup_{\boldsymbol{x} \in A} \left(\langle \boldsymbol{\eta}, \boldsymbol{x} \rangle - f(\boldsymbol{x}) \right).$$

The *support function* σ_C of a set $C \subset \mathbb{R}^n$ is defined to be the conjugate of the indicator set χ_C and its expression is given by

$$\sigma_C(\boldsymbol{\eta}) = \sup_{\boldsymbol{x} \in C} \langle \boldsymbol{\eta}, \boldsymbol{x} \rangle .$$

A set C and its convex hull have the same support function. The support function of C describes all the closed half-spaces that contain C. Indeed, one has

$$C \subset H_{\leq}(\boldsymbol{\eta}, d) \iff d \geq \sigma_C(\boldsymbol{\eta}).$$

If C is open, the same applies with respect to open half-spaces, i.e.,

$$C \subset H_{<}(\boldsymbol{\eta}, d) \iff d \geq \sigma_C(\boldsymbol{\eta}). \tag{3.1}$$

The support function can be used to determine whether a hyperplane intersects an open set.

Lemma 3.1. *Given $\boldsymbol{\eta} \in \mathbb{R}^n$ and $d \in \mathbb{R}$ and an open set $C \subset \mathbb{R}^n$, it holds*

$$d \leq -\sigma_C(-\boldsymbol{\eta}) \vee d \geq \sigma_C(\boldsymbol{\eta}) \Rightarrow H(\boldsymbol{\eta}, d) \cap C = \emptyset. \tag{3.2}$$

Proof. The proof is in Appendix A. □

Support functions are *positively homogeneous*, i.e., given a support function σ_C of a set C, it holds

$$\sigma_C(a\boldsymbol{\eta}) = a\sigma_C(\boldsymbol{\eta}),$$

for any $a > 0$. This trivially follows from the definition by the linearity of the inner product and the positive homogeneity of the supremum operator.

The *effective domain* of σ_C, that is, the set of $\boldsymbol{\eta} \in \mathbb{R}^n$ such that $\sigma_C(\boldsymbol{\eta}) < +\infty$, indicated by $\mathrm{dom}(\sigma_C)$ is the polar of the recession cone of C, i.e, K_C°. This means that $\mathrm{dom}(\sigma_C) = \mathbb{R}^n$ if and only if C does not have a direction of recession or, equivalently, a subset C of $\mathbb{R}^n$ is bounded if and only if $\sigma_C(\boldsymbol{\eta})$ is finite $\forall \boldsymbol{\eta} \in \mathbb{R}^n$.

The support function of the Minkowski sum of two subsets A and B of $\mathbb{R}^n$ is the sum of the respective support functions:

$$\sigma_{A+B}(\boldsymbol{\eta}) = \sigma_A(\boldsymbol{\eta}) + \sigma_B(\boldsymbol{\eta}).$$

The subgradients of a support function σ_C are related to the extremum points of C; in particular, it is verified that

$$\partial\sigma_C(\boldsymbol{\eta}) = H(\boldsymbol{\eta}, \sigma_C(\boldsymbol{\eta})) \cap C = \arg\max_{\boldsymbol{x} \in C} \langle \boldsymbol{\eta}, \boldsymbol{x} \rangle$$

and

$$\partial\sigma_C(\boldsymbol{\eta}) \subseteq \partial C,$$

where ∂C denotes the boundary of C.

When $\arg\max_{\boldsymbol{x} \in C} \langle \boldsymbol{\eta}, \boldsymbol{x} \rangle$ is a singleton, with abuse of notation we can write [111]:

$$\nabla\sigma_C(\boldsymbol{\eta}) = \arg\max_{\boldsymbol{x}\in C} \langle \boldsymbol{\eta},\, \boldsymbol{x}\rangle\,.$$

The notion of support functions can be extended to hypersurfaces. Consider a hypersurface determined by $\boldsymbol{\psi} : A \subseteq \mathbb{R}^{n-1} \to \mathbb{R}^n$. Its support function $\sigma_{\boldsymbol{\psi}} : \mathbb{R}^n \to \mathbb{R}$ is defined as

$$\sigma_{\boldsymbol{\psi}}(\boldsymbol{\eta}) = \sup_{\boldsymbol{x}\in A} \langle \boldsymbol{\eta},\, \boldsymbol{\psi}(\boldsymbol{x})\rangle\,.$$

This kind of support function is closely related to the support functions of sets, as $\sigma_{\boldsymbol{\psi}}$ is the support function of the convex hull of the hypersurface.

Here I write some support functions that are used later in this work. Let B be the open unit ball, i.e., the set $B = \{\boldsymbol{x} \in \mathbb{R}^n : \|\boldsymbol{x}\| < 1\}$. The support function of B is

$$\sigma_B(\boldsymbol{\eta}) = \|\boldsymbol{\eta}\|\,. \tag{3.3}$$

Let $\boldsymbol{r} \in \mathbb{R}^n$, with $r_i > 0$ for all $i \in [\![1, n]\!]$, and consider the open hyperrectangle $R = \{\boldsymbol{x} \in \mathbb{R}^n : |\boldsymbol{x}| < \boldsymbol{r}\}$. The support function of R is

$$\sigma_R(\boldsymbol{\eta}) = \langle \boldsymbol{r},\, |\boldsymbol{\eta}|\rangle\,. \tag{3.4}$$

This section ends with the analysis of the effect of affine transformations on support functions. Consider the non-singular matrix $\boldsymbol{A} \in \mathbb{R}^{n\times n}$ and the vector $\boldsymbol{b} \in \mathbb{R}^n$. The image of a set S under the affine transformation $\boldsymbol{x} \to \boldsymbol{A}\boldsymbol{x} + \boldsymbol{b}$ is $\boldsymbol{A}S + \boldsymbol{b}$ and its support function is

$$\sigma_{\boldsymbol{A}S+\boldsymbol{b}}(\boldsymbol{\eta}) = \sigma_S(\boldsymbol{A}^{\mathsf{T}}\boldsymbol{\eta}) + \langle \boldsymbol{\eta},\, \boldsymbol{b}\rangle\,. \tag{3.5}$$

3.2 Bézier curves

A *Bézier curve* of degree n is a parametric curve defined by a set of $n+1$ points $\boldsymbol{p}_0, ..., \boldsymbol{p}_n \in \mathbb{R}^2$, called *control points*. Given the control points, the resulting curve is

$$\boldsymbol{p}(\lambda) = \sum_{k=0}^{n} \binom{n}{k} \lambda^k (1-\lambda)^{n-k} \boldsymbol{p}_k, \quad \lambda \in [0, 1]\,.$$

Because of their simplicity, powerful control over shape, and efficient computability, Bézier curves are widely used in graphic design software, computer-aided design (CAD), animation, and even font creation.

Consider now a quadratic ($n = 2$) Bézier curve, given by

$$\boldsymbol{q}(\lambda) = (1-\lambda)^2\boldsymbol{a} + 2\lambda(1-\lambda)\boldsymbol{o} + \lambda^2\boldsymbol{b},$$

where the control points are $\boldsymbol{a}$, $\boldsymbol{o}$, and $\boldsymbol{b}$. I refer to control points $\boldsymbol{a}$ and $\boldsymbol{b}$ as *endpoints* of the curve and to control point $\boldsymbol{o}$ as its *control handle*.

Proposition 3.1. *Provided that $\boldsymbol{a}$, $\boldsymbol{o}$, and $\boldsymbol{b}$ are not collinear, the support function $\sigma_{\boldsymbol{q}}$ of $\boldsymbol{q}$ is*

$$\sigma_{\boldsymbol{q}}(\boldsymbol{\eta}) = \frac{\langle \boldsymbol{\eta},\, \boldsymbol{a}\rangle \langle \boldsymbol{\eta},\, \boldsymbol{b}\rangle - \langle \boldsymbol{\eta},\, \boldsymbol{o}\rangle^2}{\langle \boldsymbol{\eta},\, \boldsymbol{a} - 2\boldsymbol{o} + \boldsymbol{b}\rangle}, \tag{3.6}$$

with $\boldsymbol{\eta} \in H_{\geq}(\boldsymbol{a}-\boldsymbol{o},0) \cap H_{\leq}(\boldsymbol{b}-\boldsymbol{o},0)$. *Moreover,*

$$\nabla\sigma_{\boldsymbol{q}}(\boldsymbol{\eta}) = \frac{\langle \boldsymbol{\eta},\, \boldsymbol{b}-\boldsymbol{o}\rangle^2 \boldsymbol{a} + 2\langle \boldsymbol{\eta},\, \boldsymbol{a}-\boldsymbol{o}\rangle \langle \boldsymbol{\eta},\, \boldsymbol{b}-\boldsymbol{o}\rangle \boldsymbol{o} + \langle \boldsymbol{\eta},\, \boldsymbol{a}-\boldsymbol{o}\rangle^2 \boldsymbol{b}}{\langle \boldsymbol{\eta},\, \boldsymbol{a}-2\boldsymbol{o}+\boldsymbol{b}\rangle^2} = \boldsymbol{q}(\lambda^*),$$

with

$$\lambda^* = \frac{\langle \boldsymbol{\eta},\, \boldsymbol{a}-\boldsymbol{o}\rangle}{\langle \boldsymbol{\eta},\, \boldsymbol{a}-2\boldsymbol{o}+\boldsymbol{b}\rangle} \tag{3.7}$$

Proof. The proof is in Appendix A. □

3.3 Optimization theory

Optimization theory is a mathematical discipline that addresses the challenge of finding the best possible solution to a problem from a set of feasible solutions. Its applications span a multitude of fields, including engineering, economics, and machine learning. The area of optimization has received enormous attention in recent years, primarily because of the rapid progress in computer technology, including the development and availability of user-friendly software, high-speed and parallel processors, and artificial neural networks. A clear example of this phenomenon is the wide accessibility of optimization software tools such as the Optimization Toolbox of MATLAB and many other commercial software packages [112].

3.3.1 Constrained optimization

Constrained optimization arises from the need to make optimal decisions or find the best solution to a problem while considering certain limitations or constraints. A *constrained optimization problem* has the form

$$\begin{aligned} &\text{minimize} \quad f(\boldsymbol{x}) \\ &\text{subject to} \quad \boldsymbol{x} \in \Omega. \end{aligned} \tag{3.8}$$

The function $f : \mathbb{R}^n \to \bar{\mathbb{R}}$ that we wish to minimize is called the *objective function*, or *cost function*. The vector $\boldsymbol{x} \in \mathbb{R}^n$ is a vector of independent variables, $x_1, ..., x_n$, which are often referred to as *decision variables*. The set $\Omega \subseteq \mathbb{R}^n$ is called the *constraint set* or *feasible set*. A point $\boldsymbol{x}^* \in \Omega$ is a *global minimizer* of f over Ω if $f(\boldsymbol{x}) \geq f(\boldsymbol{x}^*)$, $\forall \boldsymbol{x} \in \Omega$. Note that a problem of maximization can be represented in the above form because maximizing f is equivalent to minimizing $-f$. Often, the constraint set Ω takes the form $\Omega = \{\boldsymbol{x} \in \mathbb{R}^n : \boldsymbol{h}(\boldsymbol{x}) = \boldsymbol{0}_n,\, \boldsymbol{g}(\boldsymbol{x}) \leq \boldsymbol{0}_n\}$, where $\boldsymbol{h} : \mathbb{R}^n \to \mathbb{R}^{m_h}$ and $\boldsymbol{g} : \mathbb{R}^n \to \mathbb{R}^{m_g}$ are given functions. Such constraints are referred to as functional constraints [112].

In general, this kind of problem can be very hard to solve, especially when the cost function and the constraint set do not have nice properties, such as convexity. Moreover, algorithms for solving constrained optimization problems can be computationally expensive, especially for large-scale problems.

A special class of constrained optimization problems are convex problems. In a convex optimization problem, the objective function is a convex function, and the constraint set is a convex set. An interesting and useful property of convex problems is that local minimizers are also global [112].

3.3.2 Algorithms for constrained optimization

For most of non-trivial optimization problems, obtaining a closed-form solution is not possible; fortunately, there are many possible strategies and numerical algorithms for tackling them.

Gradient descent is a widely-used optimization algorithm that seeks to minimize a function iteratively. The method relies on the gradient $\nabla f(\boldsymbol{x})$ at a given point $\boldsymbol{x}$ to determine the direction of steepest ascent, therefore it requires the function to be differentiable at that point. The idea is to iteratively adjust the current point in the opposite direction of the gradient to move towards the minimum of the function. To formulate an algorithm that implements the above idea, suppose that at iteration k we are given a point $\boldsymbol{x}^{(k)}$. To find the next point $\boldsymbol{x}^{(k+1)}$, we start at $\boldsymbol{x}^{(k)}$ and move by an amount $-\alpha^{(k)}\nabla f(\boldsymbol{x}^{(k)})$, where $\alpha^{(k)}$ is a positive scalar called the *step size*. The above procedure leads to the following iterative algorithm [112]:

$$\boldsymbol{x}^{(k+1)} = \boldsymbol{x}^{(k)} - \alpha^{(k)}\nabla f(\boldsymbol{x}^{(k)}). \tag{3.9}$$

The choice of $\alpha^{(k)}$ is a delicate matter because, with a step size that is too large, the descent might skip the minimizer; conversely, a step size that is too small leads to a high number of iterations. *Newton's method* is an efficient gradient descent method with quadratic convergence when some conditions apply. In Newton's method, $\alpha^{(k)} = \nabla^2 f(\boldsymbol{x}^{(k)})^{-1}$.

The process defined by (3.9) is usually repeated until a stopping criterion is met, such as achieving a sufficiently small change in the objective function. This corresponds to $\nabla f(\boldsymbol{x})$ approaching $\mathbf{0}_n$, indicating that the current point is a minimizer of f. However, for constrained optimization, $\boldsymbol{x}^*$ being a global minimizer of f over Ω does not imply $\nabla f(\boldsymbol{x}^*) = \mathbf{0}_n$.

A class of well-studied and tested algorithms that address this issue is *penalty algorithms*. The idea behind all penalty algorithms is to replace problem (3.8) with the equivalent unconstrained one:

$$\text{minimize} \quad f(\boldsymbol{x}) + \chi_\Omega(\boldsymbol{x}). \tag{3.10}$$

χ_Ω ensures that a solution to this reformulated problem is feasible, but, on the other hand, is non-differentiable, discontinuous, and not even finite. Considering this, χ_Ω is usually replaced by numerically better-behaving functions [113]. A breakthrough in this area of mathematics was the development of *interior-point methods* (or *barrier methods*). Barrier methods assess this issue by approximating χ_Ω inside the feasible set with a function called *barrier function*. These methods are methods to efficiently solve convex optimization problems, as their time complexity is polynomial [114].

Consider the case when Ω is defined only by inequality constraints given by the functions g_i, for $i = 1, ..., m_g$. One of the simplest, yet most popular, barrier functions is the *logarithmic barrier function*, defined by

$$B(\boldsymbol{x}, \mu) = -\mu \sum_{i=1}^{m_g} \ln\left(-g_i(\boldsymbol{x})\right),$$

where the *barrier parameter* μ is strictly positive. Since the logarithm is undefined for nonpositive arguments, the logarithmic barrier function is defined only in Ω [115]. As μ approaches 0, the barrier function B becomes a better approximation of χ_Ω, as shown in Figure 3.1.

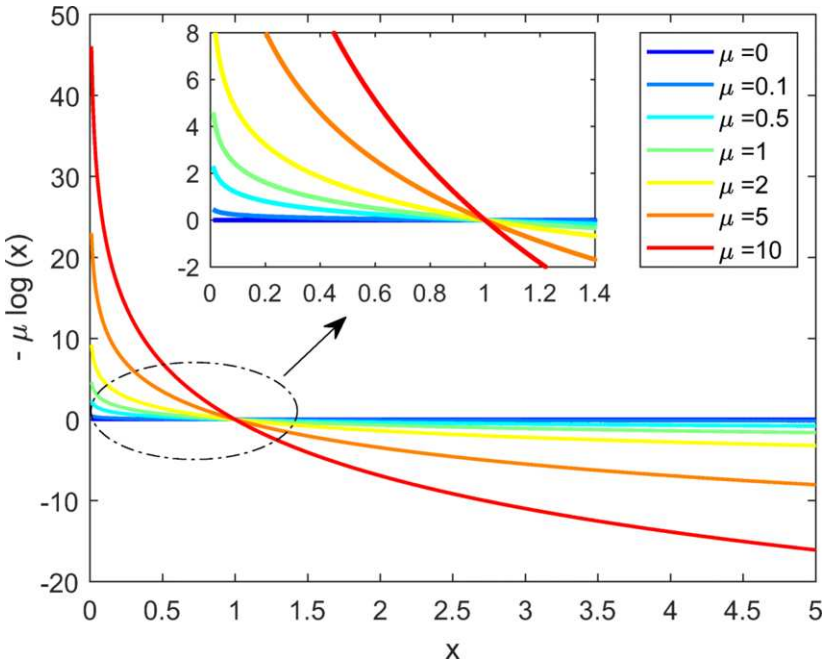


Figure 3.1: *The function curves of logarithmic barrier function under different values of μ [116].*

When in (3.10) the indicator function is replaced by the barrier function, we obtain an optimization problem that depends on μ

$$\text{minimize} \quad f(\boldsymbol{x}) - \mu \sum_{i=1}^{m_g} \ln\left(-g_i(\boldsymbol{x})\right). \tag{3.11}$$

The solution to this unconstrained problem $\boldsymbol{x}^*(\mu)$ also depends on the value of μ. The idea behind barrier methods is to solve this unconstrained problem for values of μ that progressively go to zero and use the last solution as the starting point for solving the next problem. As $\mu \to 0$, $\boldsymbol{x}^*(\mu)$ should converge to the minimizer $\boldsymbol{x}^*$ of (3.8). The classical interior-point algorithms have the following form [115]:

0. Set the starting value $\boldsymbol{x}^{(0)}$ to a strictly feasible point, so that $g(\boldsymbol{x}^{(0)}) \leq \mathbf{0}_n$, and set $\mu^{(0)}$ to a positive value.

1. Check whether $\boldsymbol{x}^{(k)}$ qualifies as an approximate local constrained minimizer for the original problem (3.8). If so, stop with $\boldsymbol{x}^{(k)}$ as the solution.

2. Compute an unconstrained minimizer $\boldsymbol{x}^*(\mu^{(k)})$ of (3.11).

3. Set $\boldsymbol{x}^{(k+1)} \leftarrow \boldsymbol{x}^*(\mu^{(k)})$, $k \to k+1$, and choose $\mu^{(k+1)} < \mu^{(k)}$. Go back to Step 1.

For convex problems, it is proven that $\boldsymbol{x}^{(k)} \to \boldsymbol{x}^*$ as $k \to +\infty$, provided that $\mu^{(k)} \to 0$ [115].

3.4 Dynamical systems theory

Dynamical system theory is an interdisciplinary field focused on the analysis of dynamic systems. While there isn't a universally accepted definition of a dynamical system, it is typically described as an entity that exhibits a reaction, called output, as a response to an outer stimulus, called input; since the system evolves through time, it is called dynamical. The INCOSE (International Council on Systems Engineering) has provided a very broad and inclusive definition: "A system is an arrangement of parts or elements that together exhibit behavior or meaning that the individual constituents do not." [117]. Generally, a system is characterized by a state, which is a set of values defining its current conditions, a law that describes how the state evolves through time as a consequence of external inputs, and a relationship between the state and the observed output.

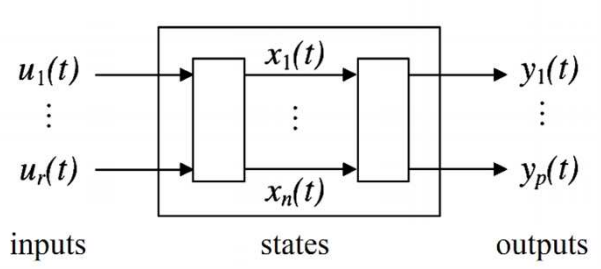


Figure 3.2: *Representation of a system with its input, states, and outputs.*

3.4.1 Axiomatic definition of a dynamical system

A general and axiomatic definition of dynamical system is the following [118]. A *dynamical system* is the 8-tuple:

$$\mathcal{S} = (T, U, \Omega, X, Y, \Gamma, \boldsymbol{\varphi}, \boldsymbol{\eta}),$$

where:

- T is the ordered set of times, which can be continuous, ($\subseteq \mathbb{R}$) or discrete ($\subseteq \mathbb{Z}$);
- U is the set of the feasible input values;
- Ω is the set of the feasible input functions;
- X is the set of the feasible state values;
- Y is the set of the feasible output values;
- Γ is the set of the feasible output functions;
- $\boldsymbol{\varphi} : T \times T \times X \times \Omega \to X$ is the state transition function, which describes how the state $\boldsymbol{x} \in X$ moves from one value to another;
- $\boldsymbol{\eta} : T \times X \times \Omega \to Y$ is the output function, which describes how the output $\boldsymbol{y} \in Y$ evolves.

$\boldsymbol{\varphi}$ describes the evolution of the state $\boldsymbol{x} \in X$ given the current time $t \in T$, provided that the system starts its evolution from the initial state $\boldsymbol{x}_\tau \in X$ at time $\tau \in T$ and the input function $\boldsymbol{u} \in \Omega$ is applied to it; therefore, one can write $\boldsymbol{x} = \boldsymbol{\varphi}(t, \tau, \boldsymbol{x}_\tau, \boldsymbol{u})$. The output $\boldsymbol{y}$ depends on the state, the input function, and the current time: $\boldsymbol{y} = \boldsymbol{\eta}(t, \boldsymbol{x}, \boldsymbol{u})$.

For this work, I will consider continuous-time dynamical systems, i.e., $T = \mathbb{R}$, for which the sets of feasible states, inputs and output are the vector spaces $X = \mathbb{R}^n$, $U = \mathbb{R}^m$ and $Y = \mathbb{R}^p$, with $n, m, p \in \mathbb{N}$, and the evolution of the state and the output are described by the following set of equations, called the *state space representation* of the system:

$$\begin{cases} \dot{\boldsymbol{x}}(t) = \boldsymbol{f}(\boldsymbol{x}(t), \boldsymbol{u}(t)), \\ \boldsymbol{y}(t) = \boldsymbol{g}(\boldsymbol{x}(t), \boldsymbol{u}(t)). \end{cases}$$

When $\boldsymbol{f}$ and $\boldsymbol{g}$ are linear functions, the system can also be described by the following equations:

$$\begin{cases} \dot{\boldsymbol{x}} = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{B}\boldsymbol{u} \\ \boldsymbol{y} = \boldsymbol{C}\boldsymbol{x} + \boldsymbol{D}\boldsymbol{u} \end{cases}, \tag{3.12}$$

where the dependency on time was dropped for simplicity, and

- $\boldsymbol{A} \in \mathbb{R}^{n \times n}$ is the *state matrix*,
- $\boldsymbol{B} \in \mathbb{R}^{n \times m}$ is the *input matrix*,
- $\boldsymbol{C} \in \mathbb{R}^{p \times n}$ is the *output matrix*, and
- $\boldsymbol{D} \in \mathbb{R}^{p \times m}$ is the *feedthrough matrix*.

A system described by (3.12) is called *linear time-invariant* (LTI) system.

3.4.2 Stability

The evolution of the state of an LTI system is given by the sum of two terms, named *free response* and *forced response*. The former is the response of the system when the input is zero, so it is only produced by the initial condition of the system. The latter is the response of the system when the initial condition is zero and an input $\boldsymbol{u}$ is applied. Denoting the initial condition by $\boldsymbol{x}_0 = \boldsymbol{x}(0)$, the total response of the system is given by the Lagrange formula:

$$\boldsymbol{x}(t) = e^{\boldsymbol{A}t}\boldsymbol{x}_0 + \int_0^t e^{\boldsymbol{A}(t-\tau)}\boldsymbol{B}\boldsymbol{u}(\tau)\,\mathrm{d}\tau$$

Stability is a concept of fundamental importance in dynamical system theory. The rigorous definition of stability falls outside the scope of this thesis, so only the practical meaning will be presented here. If one considers the free response only and takes the limit of it to infinity, i.e., $\lim_{t\to\infty} e^{\boldsymbol{A}t}\boldsymbol{x}_0$, three cases are possible:

- $e^{\boldsymbol{A}t}$ converges to the null matrix: the system is said to be *asymptotically stable*;
- $e^{\boldsymbol{A}t}$ converges to a non-null matrix: the system is said to be *stable*;

- $e^{\boldsymbol{A}t}$ diverges: the system is said to be *unstable*.

It is well-known that this depends on the position of the eigenvalues of $\boldsymbol{A}$ on the complex plane and, in particular:

- the system is *asymptotically stable* if and only if all eigenvalues have negative real parts;
- The system is *stable* if and only if all eigenvalues have negative or zero real parts and all the eigenvalues on the imaginary axis have unitary index;
- the system is *unstable* if and only if there is an eigenvalue with positive real part or zero real part and index greater than 1.

3.4.3 The Jordan canonical form

The Jordan canonical (or normal) form provides a structured and canonical representation of a matrix, highlighting its eigenvalues and their indexes. When used on $\boldsymbol{A}$, it offers insights into the inherent behavior and stability of the system it describes.

Let me recall the definition of diagonalizable matrix: a square matrix $\boldsymbol{M}$ is *diagonalizable* if it is similar to a diagonal matrix. That is, there exists an invertible matrix $\boldsymbol{P}$ and a diagonal matrix $\boldsymbol{D}$ such that $\boldsymbol{M} = \boldsymbol{P}\boldsymbol{D}\boldsymbol{P}^{-1}$.

A *Jordan block* associated to an eigenvalue $\lambda \in \mathbb{C}$ of index $r \in \mathbb{N}$ is the $r \times r$ matrix of the form

$$\begin{bmatrix} \lambda & 1 & & \\ & \lambda & \ddots & \\ & & \ddots & 1 \\ & & & \lambda \end{bmatrix},$$

i.e., each element on the main diagonal is λ and each element on the upper diagonal is 1, while the rest of the elements are 0. A matrix is in the *Jordan form* if is a block diagonal matrix and each diagonal block is a Jordan block $\boldsymbol{J}_i$ associated with the eigenvalue λ_i:

$$\boldsymbol{J} = \begin{bmatrix} \boldsymbol{J}_1 & & & \\ & \boldsymbol{J}_2 & & \\ & & \ddots & \\ & & & \boldsymbol{J}_q \end{bmatrix}.$$

A square matrix $\boldsymbol{M}$ can always be transformed into a Jordan matrix and the transformation is unique, up to a permutation of the diagonal blocks, i.e., $\boldsymbol{M} = \boldsymbol{P}\boldsymbol{J}\boldsymbol{P}^{-1}$ The Jordan canonical form is particularly useful when it comes to computing the matrix exponential of $\boldsymbol{A}$. The following result holds:

$$e^{\boldsymbol{A}t} = \boldsymbol{P}e^{\boldsymbol{J}t}\boldsymbol{P}^{-1},$$

with

$$e^{\boldsymbol{J}t} = \begin{bmatrix} e^{\boldsymbol{J}_1 t} & & & \\ & e^{\boldsymbol{J}_2 t} & & \\ & & \ddots & \\ & & & e^{\boldsymbol{J}_q t} \end{bmatrix}.$$

The exponential of a Jordan block has a known expression:

$$e^{\boldsymbol{J}_i t} = e^{\lambda_i t} \begin{bmatrix} 1 & t & \frac{t^2}{2} & \cdots & \frac{t^{r-1}}{(r-1)!} \\ & 1 & t & \cdots & \frac{t^{r-2}}{(r-2)!} \\ & & & \ddots & \vdots \\ & & & & 1 \end{bmatrix}.$$

The transformation matrix $\boldsymbol{P}$ is obtained by putting the generalized right eigenvectors $\{\boldsymbol{p}_i\}_{i\in[\![1,n]\!]}$ of $\boldsymbol{A}$ side by side; the same thing applies to its inverse $\boldsymbol{Q} = \boldsymbol{P}^{-1}$ with the generalized left eigenvectors $\{\boldsymbol{q}_i\}_{i\in[\![1,n]\!]}$:

$$\boldsymbol{P} = \begin{bmatrix} \boldsymbol{p}_1 & \cdots & \boldsymbol{p}_n \end{bmatrix}, \quad \boldsymbol{Q} = \begin{bmatrix} \boldsymbol{q}_1^\mathsf{T} \\ \vdots \\ \boldsymbol{q}_n^\mathsf{T} \end{bmatrix}.$$

The eigenvectors must be chosen so that they form a biorthogonal system, i.e., $\boldsymbol{q}_i^\mathsf{T}\boldsymbol{p}_j = \delta_{i,j}$, where $\delta_{i,j}$ is the Kronecker delta because $\boldsymbol{QP} = \boldsymbol{I}_n$.

3.4.4 Lyapunov's method for stability

In the field of dynamical systems theory, equilibrium states hold significant interest as they represent states in which a system can persist indefinitely. An *equilibrium point*, denoted as $\bar{\boldsymbol{x}} \in X$, is a state for which there exists an input function $\bar{\boldsymbol{u}} \in \Omega$ satisfying the condition

$$\boldsymbol{x} = \boldsymbol{\varphi}(t, \tau, \bar{\boldsymbol{x}}, \bar{\boldsymbol{u}}) = \bar{\boldsymbol{x}},\ \forall t > \tau.$$

Since $\boldsymbol{x} = \bar{\boldsymbol{x}}\ \forall t > \tau$, then the derivative of the state with respect to time is zero, i.e., $\dot{\boldsymbol{x}}(t) = \boldsymbol{f}(\bar{\boldsymbol{x}}, \bar{\boldsymbol{u}}(t)) = \boldsymbol{0}_n$. In practical terms, controlling a system often involves ensuring that its state remains at an equilibrium point. While in linear systems, stability is a property of the entire system, in non-linear systems, stability applies to the movement of the state and to equilibrium points.

There exists a sufficient condition that proves the stability of an equilibrium point and it makes use of a function, called the *Lyapunov function*. Let us consider a time-invariant system and an equilibrium state $\bar{\boldsymbol{x}}$. If there exists a function $V(\boldsymbol{x})$ that satisfies the following properties:

- $V(\boldsymbol{x})$ and its gradient are continuous,
- $V(\boldsymbol{x})$ is positive definite in $\bar{\boldsymbol{x}}$,
- its time derivative $\dot{V}(\boldsymbol{x})$ is negative definite in $\bar{\boldsymbol{x}}$,

then $\bar{\boldsymbol{x}}$ is an asymptotically stable equilibrium point. $\dot{V}(\boldsymbol{x})$ is defined as:

$$\dot{V}(\boldsymbol{x}) = \frac{\partial V(\boldsymbol{x})}{\partial \boldsymbol{x}}\dot{\boldsymbol{x}} = \frac{\partial V(\boldsymbol{x})}{\partial \boldsymbol{x}}\boldsymbol{f}(\boldsymbol{x}, \bar{\boldsymbol{u}}),$$

with a fixed equilibrium input $\bar{\boldsymbol{u}} \in \Omega$.

3.4.5 Controllability

Controllability is a concept that addresses the question of whether a system can be directed from one state to another through a suitable control input. In other words, it explores the extent to which an external input can influence the evolution of a dynamic system.

Formally, a state $\boldsymbol{x} \in X$ of a dynamical system is said to be *controllable* in the interval $[t_0, t]$ if there exists an input function $\boldsymbol{u} \in \Omega$ such that $\boldsymbol{\varphi}(t, t_0, \boldsymbol{x}, \boldsymbol{u}) = \boldsymbol{0}$.

For LTI systems, similarly to the stability property, this property applies in general to the whole system, instead of a particular state. For an LTI system, it can be shown that there exists a control input $\boldsymbol{u}$ that transfers the state of the system from $\boldsymbol{x}_0$ at time 0 to state $\boldsymbol{x}$ at time $t > 0$ if and only if $\boldsymbol{x} - e^{\boldsymbol{A}t}\boldsymbol{x}_0$ is in the column space of

$$\boldsymbol{W}(t) = \int_0^t e^{\boldsymbol{A}\tau}\boldsymbol{B}\boldsymbol{B}^{\mathsf{T}}e^{\boldsymbol{A}^{\mathsf{T}}\tau}\,\mathrm{d}\tau.$$

The symmetric matrix $\boldsymbol{W}(t)$ is called the *controllability Gramian* of the system. Provided that this matrix is invertible for every $t > 0$, then a control input that transfers the state of the system from $\boldsymbol{x}_0$ to $\boldsymbol{x}_f$ in the time interval $[0, t_f]$ is given by

$$\boldsymbol{u}^*(t) = \left(e^{\boldsymbol{A}(t_f - t)}\right)^{\mathsf{T}}\boldsymbol{W}^{-1}(t_f)\left(\boldsymbol{x}_f - e^{\boldsymbol{A}t_f}\boldsymbol{x}_0\right). \tag{3.13}$$

It can be shown that of all the control functions $\boldsymbol{u} \in \Omega$ that succeed in transfering the state of the system from $\boldsymbol{x}_0$ to $\boldsymbol{x}_f$ in t_f seconds, $\boldsymbol{u}^*$ is a minimizer of the functional:

$$E[\boldsymbol{u}] = \int_0^{t_f} \|\boldsymbol{u}(t)\|^2\,\mathrm{d}t,$$

which is deemed as the energy associated with the control input $\boldsymbol{u}$. For this reason, the control law obtained by equation (3.13) is referred to as *minimum energy control*.

3.5 Graph theory

Graphs are mathematical structures used to elegantly model connections between objects. They are broadly used in situations where multiple objects have some kind of relationship among them. The objects are represented as dots and they are called vertices (or nodes), while the relationships are drawn as lines that connect two dots and they are called edges (or arcs). Some concepts that can be modeled by graphs are, e.g., social networks, road networks, electrical circuits, etc.

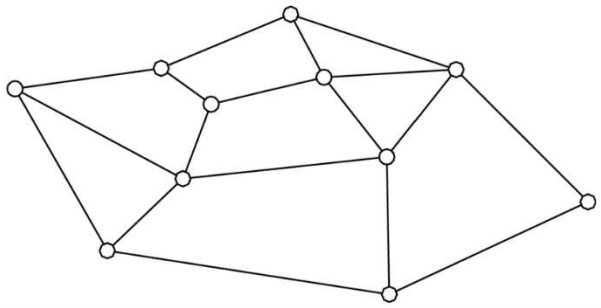


Figure 3.3: *Representation of a graph with nodes and arcs.*

3.5.1 Definition of a graph

A graph is the pair $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ is a set of elements called *vertices* and $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ is a set of pairs of vertices $e_{i,j} = (v_i, v_j)$ called *edges*. If the pair is unordered, i.e., $(v_i, v_j) = (v_j, v_i)$, then the graph is said to be *undirected*; conversely, if the pair is ordered, i.e., $(v_i, v_j) \neq (v_j, v_i)$, the graph is said to be *directed*. Throughout this dissertation, I will assume that a graph cannot contain multiple edges between the same pair of vertices and there are no loops, i.e., edges connecting a vertex to itself.

3.5.2 Properties of graphs

A list of useful properties and definitions is given here.

A directed graph is symmetric if $(v_i, v_j) \in \mathcal{E} \iff (v_j, v_i) \in \mathcal{E}$, that is, each edge has a corresponding reverse edge.

Two vertices $v_i, v_j \in \mathcal{V}$ are *adjacent* if there exists an edge connecting them, i.e., $(v_i, v_j) \in \mathcal{E}$ (or $(v_j, v_i) \in \mathcal{E}$ if the graph is directed).

For directed graphs, the *in-degree* of a vertex v_i is the number of edges ending in v_i, i.e., the number of the elements of the form $(v, v_i) \in \mathcal{E}$. Similarly, the *out-degree* of v_1 is the number of edges starting from v_i, i.e., the number of the elements of the form $(v_i, v) \in \mathcal{E}$. For undirected graphs, the in-degree and the out-degree coincide and will be referred to as the *degree* of a vertex.

A sequence of vertices $(v_{i_1}, ..., v_{i_p})$ is called a *path* of length p between v_{i_1} and v_{i_p} if $(v_{i_k}, v_{i_{k+1}}) \in \mathcal{E}$, $\forall k \in [\![1, p-1]\!]$.

A directed graph is *strongly connected* (or just *connected*), if there exists a path for each pair of vertices of the graph. If a directed graph is not connected but its corresponding undirected graph is, the directed graph is said to be *weakly connected*. A strongly connected component of a digraph is a sub-digraph $\mathcal{G}' = (\mathcal{V}', \mathcal{E}')$, with $\mathcal{V}' \subseteq \mathcal{V}$ and $\mathcal{E}' \subseteq \mathcal{V}' \times \mathcal{V}'$, that is connected. The distance between v_i and v_j is denoted by $\text{dist}(v_i, v_j)$ and is the length of the shortest path between v_i and v_j. If v_i and v_j are not connected, then $\text{dist}(v_i, v_j) = +\infty$.

The set defined as $\mathcal{D}_i(d) = \{v_j \in \mathcal{V} \ : \ \text{dist}(v_i, v_j) = d\}$ is the set of vertices that are at a distance d from v_i. The vertices that are at a distance 1 from v_i are called *neighbors* of v_i; given v_i, the set of its neighbors is denoted by $\mathcal{N}_i = \mathcal{D}(1) = \{v_j \in \mathcal{V} \ : \ (v_i, v_j) \in \mathcal{E}\}$.

The *adjacency matrix* $\boldsymbol{A}$ of a graph is a square matrix representing the connections between vertices. If there are n vertices in $\mathcal{V}$, then $\boldsymbol{A} \in \mathbb{R}^{n \times n}$, whose generic element $a_{i,j}$ is defined as

$$a_{i,j} = \begin{cases} 1 & \text{if } e_{i,j} \in \mathcal{E}, \\ 0 & \text{if } e_{i,j} \notin \mathcal{E}. \end{cases}$$

Note that the absence of loops in the graphs considered here implies that the diagonal elements of the matrix are always 0.

The *degree matrix* $\boldsymbol{D} \in \mathbb{R}^{n \times n}$ is a diagonal matrix in which diagonal elements are the out-degree of the vertices:

$$d_{i,j} = \begin{cases} \sum_{k=1}^{n} a_{i,k} & \text{if } i = j, \\ 0 & \text{if } i \neq j. \end{cases}$$

The *Laplacian matrix* $\boldsymbol{L} \in \mathbb{R}^{n \times n}$ is defined as the difference between the degree matrix and the adjacency matrix, i.e., $\boldsymbol{L} = \boldsymbol{D} - \boldsymbol{A}$; thus, its generic entry $l_{i,j}$ has the following expression:

$$l_{i,j} = \begin{cases} \sum_{k=1}^{n} a_{i,k} & \text{if } i = j, \\ -a_{i,j} & \text{if } i \neq j. \end{cases}$$

The Laplacian matrix plays an important role in a class of algorithms called consensus algorithms. These are algorithms that work on multi-agent systems, where each system is represented by a vertex and edges represent the information flow between two agents. Consensus algorithms are generally based on computing the difference of the states of adjacent agents, which can be represented compactly using the Laplacian matrix. The spectral properties of $\boldsymbol{L}$ are of great importance in this field.

An important property of $\boldsymbol{L}$ is that the sum of the elements of each row is 0:

$$\sum_{k=1}^{n} l_{i,k} = l_{i,i} + \sum_{k=1, k \neq i}^{n} l_{i,k} = \sum_{k=1}^{n} a_{i,k} - \sum_{k=1}^{n} a_{i,k} = 0.$$

This can also be written as $\boldsymbol{L}\mathbf{1}_n = \mathbf{0}_n$, therefore the kernel of $\boldsymbol{L}$ is non-empty: in particular, $\boldsymbol{L}$ has at least one null eigenvalue and the span of $\mathbf{1}_n$ is the eigenspace associated to the 0 eigenvalue.

Let $\mu_k \in \mathbb{C}$ denote the k-th eigenvalue of $\boldsymbol{L}$ and α_k and β_k denote its real and imaginary parts, respectively, i.e., $\alpha_k = \Re(\mu_k)$ and $\beta_k = \Im(\mu_k)$. The eigenvalues are indexed from 0 to $n-1$, to highlight the fact that there is always a null eigenvalue $\mu_0 = 0$. The spectrum of $\boldsymbol{L}$ tells some properties of the graph. One of the most important properties is the following: the Laplacian matrix of a directed graph has a simple zero eigenvalue with an associated eigenvector $\mathbf{1}_n$ and all of the other eigenvalues are in the open right half plane if and only if the directed graph has a directed spanning tree [119]. In this case, $\alpha_k > 0$, $\forall k \in [\![1, n-1]\!]$. Moreover, if the graph is bidirected (or undirected), then $\boldsymbol{L}$ is symmetric and its eigenvalues are real.

Example 3.1. $\boldsymbol{L}$ being symmetric is not a necessary condition for its eigenvalues to be real. Consider a digraph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ composed of two strongly connected components $\mathcal{G} = (\mathcal{V}_1, \mathcal{E}_1)$ and $\mathcal{G}_2 = (\mathcal{V}_2, \mathcal{E}_2)$, with $\mathcal{V} = \mathcal{V}_1 \cup \mathcal{V}_2$ and $\mathcal{V}_1 \cap \mathcal{V}_2 = \emptyset$. The set of edges $\mathcal{E}$ is the union of four disjoint sets, namely, $\mathcal{E} = \mathcal{E}_1 \cup \mathcal{E}_2 \cup \mathcal{E}_{1,2} \cup \mathcal{E}_{2,1}$, with $\mathcal{E}_{1,2} \subseteq \mathcal{V}_1 \times \mathcal{V}_2$ and $\mathcal{E}_{2,1} \subseteq \mathcal{V}_2 \times \mathcal{V}_1$ respectively being the edges connecting vertices of $\mathcal{V}_1$ with vertices of $\mathcal{V}_2$ and vice versa. If either $\mathcal{E}_{1,2} = \emptyset$ or $\mathcal{E}_{2,1} = \emptyset$, then the eigenvalues of the Laplacian matrix of $\boldsymbol{L}$ are real.

To see this, consider $\mathcal{E}_{1,2} = \emptyset$ - the case $\mathcal{E}_{2,1} = \emptyset$ is analogous. If $\mathcal{E}_{1,2} = \emptyset$, then it means $\nexists v_1 \in \mathcal{V}_1$, $v_2 \in \mathcal{V}_2$ such that $(v_1, v_2) \in \mathcal{E}$. As a consequence, the Laplacian matrix has the form

$$\boldsymbol{L} = \begin{bmatrix} \boldsymbol{L}_{1,1} & \boldsymbol{L}_{1,2} \\ \boldsymbol{O} & \boldsymbol{L}_{2,2} \end{bmatrix}.$$

Because $\boldsymbol{L}$ is block triangular, its spectrum is the union of the spectra of $\boldsymbol{L}_{1,1}$ and $\boldsymbol{L}_{2,2}$. If $\mathcal{G}_1$ and $\mathcal{G}_2$ are symmetric, then all the eigenvalues of $\boldsymbol{L}_{1,1}$ and $\boldsymbol{L}_{2,2}$ are real, despite $\mathcal{G}$ not being symmetric.

3.6 Consensus algorithms

A modern problem in control theory is the coordinated control of a set of systems, or agents. Typically, the agents must be controlled in a distributed way: the control algorithm is not implemented on a central controller that directs all agents, but instead, each agent has a simpler decentralized algorithm that uses information from some of the others. Although each agent only has access to incomplete information, those algorithms can efficiently lead the group to reach a common goal. There are multiple reasons to use decentralized algorithms instead of centralized ones [119].

- Limited bandwidth: the connectivity and communication capabilities of the group are often limited a control algorithm relying on the full knowledge of the state of each agent is unrealistic.
- Limited computational resources: cases where a group is composed of many agents are not rare and sometimes a centralized controller just cannot handle the computational load. Instead, most of the time embedded computational resources are abundant in the single agents, that can be exploited by distributed algorithms.
- Scalability: a centralized control algorithm is hardly scalable, while distributed algorithms scale automatically when a new agent is added to the team. Furthermore, centralized control is more sensitive to faults.
- Cost: coordinating a large number of systems usually leads to unacceptable costs to make up for the limitations above.

Information exchange is crucial for cooperation. Coordination problems naturally lead to agreement problems, where the goal is to bring a set of variables, like the states of the agents, to a common value over time. When this happens, we say that *consensus* has been reached [119]. For this to occur, consensus algorithms need to be used: with this class of algorithms, an agent regulates its dynamics using information coming from the other agents it communicates with, trying to reach a common value.

Consensus theory finds application in many fields: transportation, aerial traffic control, military, sensor network coordination, telecommunication, smart grids, and cryptocurrencies.

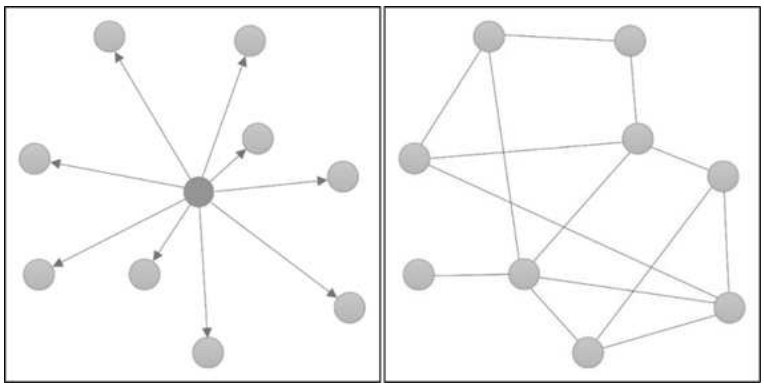


Figure 3.4: *Centralized (left) and decentralized (right) control.*

Now, a mathematical definition of consensus is given. Consider a set of n dynamical systems, having a state $\boldsymbol{x}_i$, with $i \in [\![1, n]\!]$. Consensus is reached if, for any initial condition $\boldsymbol{x}(0)$:

$$\lim_{t\to\infty} \|\boldsymbol{x}_i(t) - \boldsymbol{x}_j(t)\| = 0, \quad \forall i, j \in [\![1, n]\!].$$

The consensus problem is a problem of convergence. There are continuous-time and discrete-time consensus algorithms, but here I will consider the continuous case. The set of agents can be modeled as a directed graph, where nodes represent systems and arcs represent information flow. The arc direction tells the direction of the flow: for convention, arc $e_{i,j}$ represents information flowing from agent j to agent i, although it is visually counterintuitive. An arc between i and j means "agent i receives information from agent j". This convention allows to use the definition of the Laplacian obtained with the out-degree instead of the in-degree of the nodes: with this choice, $\mathbf{1}_n$ is a left eigenvector of λ_0 instead of a right eigenvector.

3.6.1 First-order algorithms

The simplest situation occurs when each agent is described by first-order dynamics, so its state is a scalar and the dynamic equation is:

$$\dot{x}_i = u_i, \quad \forall i \in [\![1, n]\!].$$

We want to choose an appropriate set of inputs $\{u_i(\cdot)\}_{i\in[\![1,n]\!]}$ that guarantees the reaching of consensus. I recall that each agent has only access to the state of its adjacent agents, so u_i can only depend on those states, as well as the state of the agent i. The input is usually chosen to be a weighted sum of the differences in the state values of the adjacent agents and the of agent i:

$$u_i = \sum_{j\in\mathcal{N}_i} k_{i,j}(x_j - x_i), \tag{3.14}$$

where $k_{i,j}$ represents the weight given to agent j by agent i. For simplicity, I will assume that every agent considers each other agent equally important, therefore $k_{i,j} \equiv 1$. With this, equation (3.14) can be written in vector form as

$$\dot{\boldsymbol{x}} = \boldsymbol{u} = -\boldsymbol{L}\boldsymbol{x},$$

with vectors $\boldsymbol{x} = \begin{bmatrix} x_1 & ... & x_n \end{bmatrix}^\mathsf{T}$ and $\boldsymbol{u} = \begin{bmatrix} u_1 & ... & u_n \end{bmatrix}^\mathsf{T}$.

When consensus is reached, then $x_i = x_j = x^*$, $\forall i, j$, i.e., $\boldsymbol{x} = \boldsymbol{x}^* = \mathbf{1}_n x^*$. $\boldsymbol{x}^*$ is an equilibrium state, meaning that the time derivative of the state is zero, as $\dot{\boldsymbol{x}} = -\boldsymbol{L}\boldsymbol{x}^* = -\boldsymbol{L}\mathbf{1}_n x^* = \mathbf{0}_n$.

The evolution of the system depends on the eigenvalues of $-\boldsymbol{L}$. If the graph is connected, these eigenvalues, except for the one null eigenvalue, have negative real parts. Since there is a null eigenvalue, the system is not asymptotically stable, but just stable. This is a desired behavior because the system can be in equilibrium in all consensus states and not just in the origin.

The response of the system is:

$$\boldsymbol{x}(t) = e^{-\boldsymbol{L}t}\boldsymbol{x}_0.$$

When the system is not unstable, then as $t \to +\infty$, the exponential matrix converges to a finite matrix $\boldsymbol{R}$:

$$\boldsymbol{x}^* = \lim_{t\to\infty} \boldsymbol{x}(t) = \lim_{t\to\infty} e^{-\boldsymbol{L}t}\boldsymbol{x}_0 = \boldsymbol{R}\boldsymbol{x}_0.$$

A well-known result states that if $\boldsymbol{L}$ has a simple zero eigenvalue, then $\boldsymbol{R} = \boldsymbol{p}_0\boldsymbol{q}_0^\mathsf{T}$, with $\boldsymbol{p}_0$ and $\boldsymbol{q}_0$ being the right and left eigenvectors of $-\boldsymbol{L}$ associated to the null eigenvalue, chosen so that $\boldsymbol{q}_0^\mathsf{T}\boldsymbol{p}_0 = 1$ [119]. Without loss of generality, let us choose $\boldsymbol{p}_0 = \mathbf{1}_n$ ($\boldsymbol{q}_0$ can still be chosen adequately). We obtain

$$\boldsymbol{x}^* = \boldsymbol{R}\boldsymbol{x}_0 = \mathbf{1}_n\boldsymbol{q}_0^\mathsf{T}\boldsymbol{x}_0 = \mathbf{1}_n x^*.$$

Hence, consensus is reached, and the consensus value is

$$x^* = \boldsymbol{q}_0^\mathsf{T}\boldsymbol{x}_0 \tag{3.15}$$

3.6.2 Average consensus

When the consensus value is the average of the initial state values, we are talking about *average consensus*. This occurs when $\mathbf{1}_n$ is also a right eigenvector of $-\boldsymbol{L}$: indeed, in this case we have $\boldsymbol{q}_0 = n^{-1}\mathbf{1}_n$, satisfying $\boldsymbol{q}_0^\mathsf{T}\boldsymbol{p}_0 = n^{-1}\mathbf{1}_n^\mathsf{T}\mathbf{1}_n = 1$. In this case

$$x^* = n^{-1}\mathbf{1}_n^\mathsf{T}\boldsymbol{x}_0 = \frac{1}{n}\sum_{i=1}^{n} x_i(0).$$

This situation happens depending on the topology of the system: when the in-degree is equal to the out-degree for all nodes of the graph, then the graph is symmetric and also is the Laplacian matrix, i.e., $\boldsymbol{L} = \boldsymbol{L}^\mathsf{T}$. Intuitively, if every node gives the network the same contribution it receives, then its state will be as important as every other one, and if all states are equally important, then the consensus value is the average of those values.

Chapter 4

Cooperative intersection management

In this chapter, I tackle the problem of autonomous intersection crossing using a geometrical and optimization approach. The goal of an intersection management algorithm is to optimize the intersection crossing schedule, in order to minimize the crossing time while making sure that every vehicle safely crosses the junction. Other secondary goals may be ensuring comfort and minimizing energy losses.

4.1 The Intersection Crossing Problem

In this section, I present the Intersection Crossing Problem (ICP), which is an optimization problem modeling the intersection and the vehicles crossing it. The problem is formulated such that a solution to it represents a possible schedule that allows vehicles to safely cross the intersection, as well as possibly do it efficiently.

First, I introduce the geometrical model used throughout the chapter to describe the intersection and the vehicles crossing it. The presented results are mostly obtained by geometrical modeling and minimally involve the physics behind them because the goal is to obtain high-level requirements and constraints about the vehicles' kinematics.

4.1.1 Vehicle model

A vehicle is modeled as a point in the 2D plane representing the position of the vehicle on the ground, with respect to a predetermined origin. Vehicles at an intersection move along a predefined trajectory, therefore I model the position $\boldsymbol{p}(u) = \begin{bmatrix} p_x(u) & p_y(u) \end{bmatrix}^\mathsf{T}$ of each vehicle as a parametrized curve $\boldsymbol{p} : \mathbb{R} \to \mathbb{R}^2$ with u being its curvilinear coordinate, or abscissa.

Trajectories are parametrized such that the norm of their velocity, denoted by $\boldsymbol{p}'$, is a constant, i.e.,

$$\|\boldsymbol{p}'(u)\| \equiv w \in \mathbb{R}_+ \tag{4.1}$$

This does not mean that the speed of the vehicle is always constant: equation (4.1) only refers to the parametrization of its trajectory. Indeed, let us consider the *kinematic function* $\gamma : \mathbb{R} \to \mathbb{R}$ that maps time instants to abscissas, meaning $\gamma(t)$ is the abscissa of the vehicle at time t. γ should be increasing (since a vehicle may not go backward) and $\boldsymbol{p} \circ \gamma$ is a reparametrization of the trajectory with respect to the time variable t. In other words, $\boldsymbol{p}(\gamma(t))$ is the position of the vehicle at time t; moreover,

$$\left\| \frac{\mathrm{d}}{\mathrm{d}t} \boldsymbol{p}(\gamma(t)) \right\| = \|\boldsymbol{p}'(\gamma(t))\gamma'(t)\| = w\,|\gamma'(t)|\,,$$

which, in general, is not constant and depends on t.

Not every kinematic function is feasible since the vehicle must be able to physically follow the trajectory with the given kinematic function. Let us indicate the set of feasible kinematic functions as $\mathbb{\Gamma}$, so we can write $\gamma \in \mathbb{\Gamma}$.

Other than its position, a vehicle is characterized by a region of space it occupies at a given moment. The *occupancy set* of a vehicle is the set $O(u) \subset \mathbb{R}^2$ of the XY plane containing the region of space occupied by a vehicle at position $\boldsymbol{p}(u)$. $O : \mathbb{R} \to \mathcal{P}(\mathbb{R}^2)$ is referred to as the *occupancy function*. Like what was said for the position of the vehicle, $O(\gamma(t))$ is the occupancy set of a vehicle at time t.

Now, I give the definition of a collision between two vehicles. From now on, to distinguish between quantities referred to different vehicles, I include the index of each vehicle as a subscript to the relevant quantity. When unambiguously referring to a single vehicle, subscripts are omitted.

Definition 4.1. *Consider two vehicles with positions $\boldsymbol{p}_1(u_1)$ and $\boldsymbol{p}_2(u_2)$ and occupancy sets $O_1(u_1)$ and $O_2(u_2)$. The vehicles are said to be colliding if*

$$O_1(u_1) \cap O_2(u_2) \neq \emptyset. \tag{4.2}$$

In other words, two vehicles are colliding if the regions of space they occupy overlap. Conversely, two vehicles are said to be not colliding if $O_1(u_1) \cap O_2(u_2) = \emptyset$.

The definition of collision is based on a generic pair of abscissas (u_1, u_2), but when considering a pair of kinematic laws (γ_1, γ_2) the notion of *collision course* can be introduced.

Definition 4.2. *Two vehicles with kinematic laws (γ_1, γ_2) and occupancy sets (O_1, O_2) are said to be on a collision course if*

$$\exists t \in \mathbb{R} \;:\; O_1(\gamma_1(t)) \cap O_2(\gamma_2(t)) \neq \emptyset. \tag{4.3}$$

4.1.2 Problem formulation

The presented model of the vehicles can be already used to define a general optimization problem that models the crossing of the intersection. Even though I have not discussed what an intersection is, intuitively it can be pictured as a region of space where the trajectories of multiple vehicles converge and, possibly, intersect.

Consider a set of n vehicles, where each vehicle $i \in [\![1, n]\!]$ is characterized by its occupancy O_i and feasible kinematic function set $\mathbb{\Gamma}_i$ from which it must choose its kinematic law γ_i. Given a functional $F : \mathbb{\Gamma}_1 \times ... \times \mathbb{\Gamma}_n \to \mathbb{R}$, which represents a cost associated with the given choice of kinematic functions, the *Intersection Crossing Problem* for fixed trajectories is defined as

$$\begin{aligned} \text{minimize} \quad & F[\gamma_1, ..., \gamma_n] \\ \text{subject to} \quad & O_i(\gamma_i(t)) \cap O_j(\gamma_j(t)) = \emptyset \quad \forall t \in \mathbb{R}, \quad \forall i, j \in [\![1, n]\!],\ i \neq j \\ \text{and} \quad & \gamma_i \in \mathbb{\Gamma}_i \quad \forall i \in [\![1, n]\!] \end{aligned} \tag{4.4}$$

The functional F represents the cost associated with the given choice of kinematic laws. Also, note that $\boldsymbol{p}_i$ does not appear in the problem definition, but the occupancy of the vehicle is strictly related to its trajectory; also, the trajectories might as well determine the expression of F. Provided that the trajectories are fixed, the generality of the formulation allows us to consider any kind of cost associated with the motion of the vehicles, as well as all possible shapes vehicles can have - even shapes that change over time! In the following sections, assumptions are made about the trajectories, the occupancies, and the relationships occurring between them, so that this theoretical framework could be specialized for specific kinds of intersections and vehicles.

4.2 Intersection model

4.2.1 Trajectory model

When a vehicle follows a straight path in space, its trajectory is a straight line which can be described by the following parametrization

$$\boldsymbol{p}(u) = \boldsymbol{p}_0 + (\boldsymbol{p}_1 - \boldsymbol{p}_0)u = \boldsymbol{p}_0 + \boldsymbol{v}u, \tag{4.5}$$

with $\boldsymbol{p}_0, \boldsymbol{p}_1 \in \mathbb{R}^2$ being the position of the vehicle when $u = 0$ and $u = 1$, respectively, and $\boldsymbol{v} = \boldsymbol{p}_1 - \boldsymbol{p}_0$ being the velocity vector, since $\boldsymbol{p}'(u) \equiv \boldsymbol{v}$; clearly, (4.1) holds.

When a vehicle curves from one direction to another, it follows an arc of circumference. Let us consider a circumference of radius r that, for simplicity, is centered at the origin. As shown in Figure 4.1, the vehicle approaches the circumference at an *entry angle* $\alpha \in [-\pi, \pi]$, following a straight line which is tangent to it at the point $\boldsymbol{p}_0 = \boldsymbol{p}(0)$; then, it traces an arc and leaves the circumference at an *exit angle* β, such that $|\beta - \alpha| \in]0, \pi[$ carrying on on a straight path again, with this second tangency point being $\boldsymbol{p}_1 = \boldsymbol{p}(1)$. For left turns, we have $\beta > \alpha$, whereas for right turns $\alpha > \beta$.

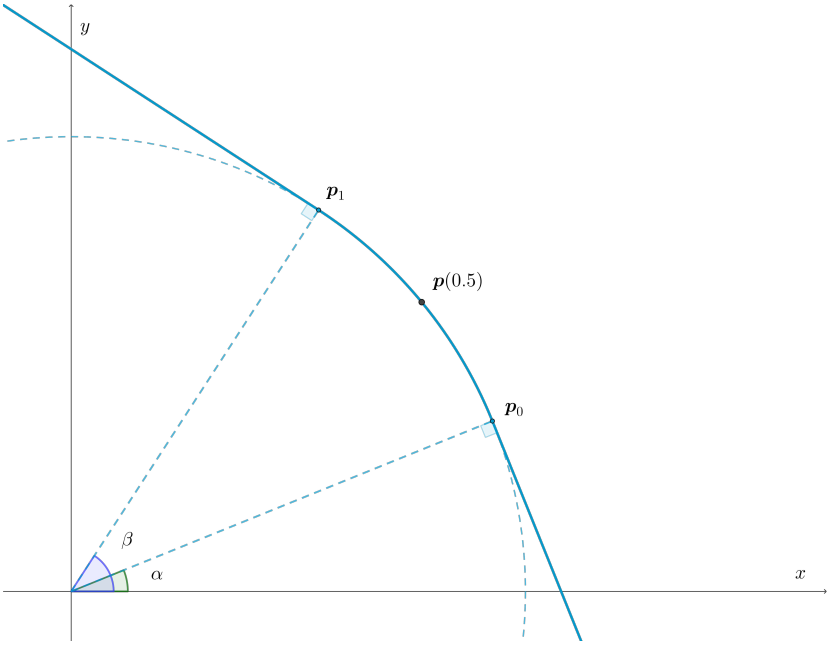


Figure 4.1: *Position of a vehicle halfway through a left turn.*

The coordinates of the points $\boldsymbol{p}_0$ and $\boldsymbol{p}_1$ are

$$\boldsymbol{p}_0 = r \begin{bmatrix} \cos\alpha \\ \sin\alpha \end{bmatrix}, \quad \boldsymbol{p}_1 = r \begin{bmatrix} \cos\beta \\ \sin\beta \end{bmatrix},$$

and the expression of $\boldsymbol{p}(u)$ along the arc, denoted by $\boldsymbol{p}_{[0,1]}(u)$ must interpolate between these two points and respect (4.1). Therefore, the desired expression is

$$\boldsymbol{p}_{[0,1]}(u) = r \begin{bmatrix} \cos\left((\beta-\alpha)u+\alpha\right) \\ \sin\left((\beta-\alpha)u+\alpha\right) \end{bmatrix}.$$

The norm of the velocity along the arc is $\boldsymbol{p}'_{[0,1]}(u) \equiv r(\beta-\alpha)$, which must also hold outside the $[0,1]$ interval, so that (4.1) is satisfied. The unit vectors tangent to the circumference at $\boldsymbol{p}_0$ and $\boldsymbol{p}_1$ are $\begin{bmatrix} -\sin\alpha & \cos\alpha \end{bmatrix}^\mathsf{T}$ and $\begin{bmatrix} -\sin\beta & \cos\beta \end{bmatrix}^\mathsf{T}$, respectively. Hence, the full expression of $\boldsymbol{p}(u)$ is

$$\boldsymbol{p}(u) = r \begin{cases} \begin{bmatrix} \cos\alpha \\ \sin\alpha \end{bmatrix} + (\beta-\alpha) \begin{bmatrix} -\sin\alpha \\ \cos\alpha \end{bmatrix} u & : u < 0, \\ \begin{bmatrix} \cos\left((\beta-\alpha)u+\alpha\right) \\ \sin\left((\beta-\alpha)u+\alpha\right) \end{bmatrix} & : 0 \le u \le 1, \\ \begin{bmatrix} \cos\beta \\ \sin\beta \end{bmatrix} + (\beta-\alpha) \begin{bmatrix} -\sin\beta \\ \cos\beta \end{bmatrix} (u-1) & : u > 1. \end{cases} \tag{4.6}$$

For right turns, the expression is analogous, with β being the entry angle and α being the exit angle:

$$\boldsymbol{p}(u) = r \begin{cases} \begin{bmatrix} \cos\beta \\ \sin\beta \end{bmatrix} + (\beta-\alpha) \begin{bmatrix} -\sin\beta \\ \cos\beta \end{bmatrix} u & : u < 0, \\ \begin{bmatrix} \cos\left((\beta-\alpha)u-\beta\right) \\ \sin\left((\beta-\alpha)u-\beta\right) \end{bmatrix} & : 0 \le u \le 1, \\ \begin{bmatrix} \cos\alpha \\ \sin\alpha \end{bmatrix} + (\beta-\alpha) \begin{bmatrix} -\sin\alpha \\ \cos\alpha \end{bmatrix} (u-1) & : u > 1. \end{cases} \tag{4.7}$$

The length of the trajectory inside the $[0,1]$ interval is indicated with l:

$$l = \int_0^1 \|\boldsymbol{p}'(u)\| \, \mathrm{d}u.$$

4.2.2 Oblique intersections

Equations (4.5), (4.6), and (4.7) can describe the trajectories of a great variety of intersections. However, I will focus on the kind of intersection produced by two straight two-way single-lane roads meeting at an angle[1] $\alpha \in \,]0,\pi[$, as shown in Figure 4.2. I refer to this kind of intersection using the name *α-intersection*. The frame of reference is centered at the

[1] Although α is assumed to belong to the $]0,\pi[$ interval, extreme values of this interval are not realistic.

intersection of the middle lines of the two roads. Each road has a width of 2 units, resulting in a single unit width for each traffic lane. For reference, the road directions are labeled by S, E, N, and W, even though they do not indicate cardinal directions; the angle α is assumed to be determined by directions S and E.

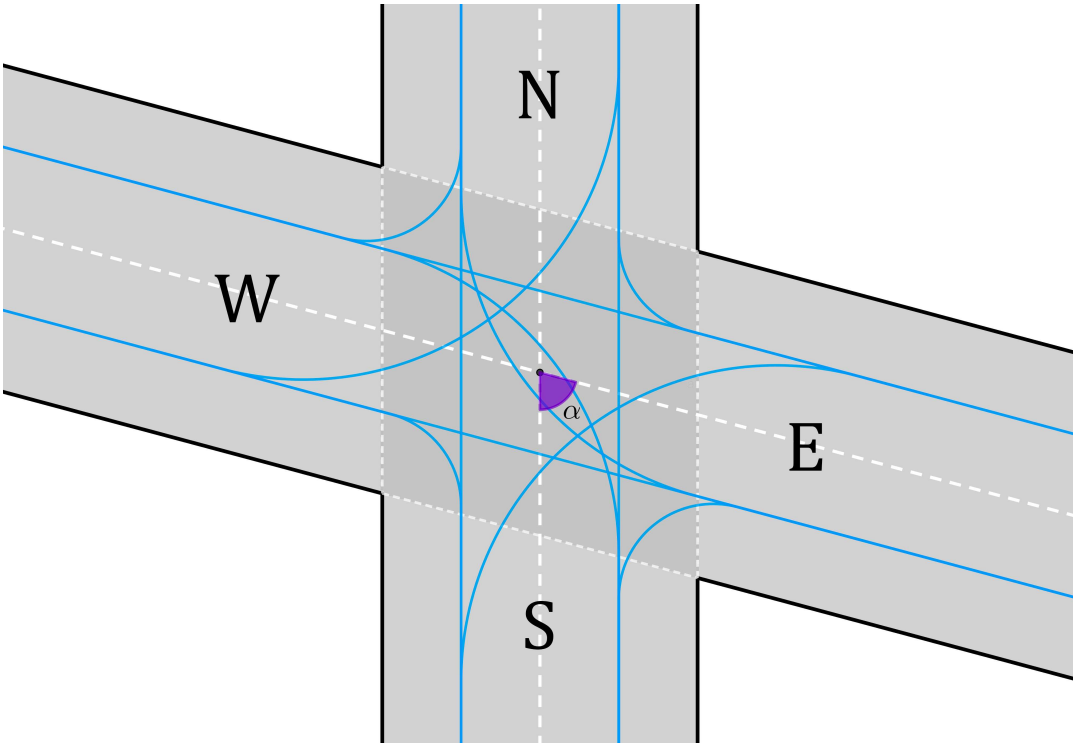


Figure 4.2: *Intersection of straight roads meeting at an angle* $\alpha = 70^\circ$.

The middle lines are described by the following equations

$$\begin{aligned} \text{SN}: &\quad x = 0, \\ \text{EW}: &\quad \cos\alpha\, x + \sin\alpha\, y = 0, \end{aligned}$$

while the borders are given by

$$\begin{aligned} \text{SN}: &\quad x \pm 1 = 0, \\ \text{EW}: &\quad \cos\alpha\, x + \sin\alpha\, y \pm 1 = 0. \end{aligned}$$

All trajectories are determined by their inbound and outbound lanes and can be described using (4.5), (4.6), and (4.7), after determining the entry and exit angles and the centers of the curved parts of the trajectories. The set of all possible trajectories within an α-intersection is denoted by P_α. Now, I formally define the concept of sharing an inbound or an outbound lane.

Definition 4.3. *Two vehicles with trajectories* $\boldsymbol{p}_1, \boldsymbol{p}_2 \in P_\alpha$ *share their inbound lane when*

$$\exists \bar{\boldsymbol{u}} = \begin{bmatrix} \bar{u}_1 & \bar{u}_2 \end{bmatrix}^\mathsf{T} \; : \; \boldsymbol{p}_1(\mathbb{R}_{\leq \bar{u}_1}) = \boldsymbol{p}_2(\mathbb{R}_{\leq \bar{u}_2})$$

and this relation is written as $\boldsymbol{p}_1 \uparrow \boldsymbol{p}_2$.

Definition 4.4. *Two vehicles with trajectories* $\boldsymbol{p}_1, \boldsymbol{p}_2 \in P_\alpha$ *share their outbound lane when*

$$\exists \bar{\boldsymbol{u}} = \begin{bmatrix} \bar{u}_1 & \bar{u}_2 \end{bmatrix}^\mathsf{T} \; : \; \boldsymbol{p}_1(\mathbb{R}_{\geq \bar{u}_1}) = \boldsymbol{p}_2(\mathbb{R}_{\geq \bar{u}_2})$$

and this relation is written as $\boldsymbol{p}_1 \downarrow \boldsymbol{p}_2$.

When the relations of Definitions 4.3 and 4.4 do not hold, I will write $\boldsymbol{p}_1 \not\uparrow \boldsymbol{p}_2$ and $\boldsymbol{p}_1 \not\downarrow \boldsymbol{p}_2$, respectively.

The trajectory from direction X to direction Y, is denoted by $\boldsymbol{p}_{XY}$, where X and Y can be S, E, N, and W; for example, $\boldsymbol{p}_{WN}$ indicates the trajectory of a vehicle that enters the intersection West and leaves it from North.

It is sufficient to determine the expressions of the trajectories from one single direction (S) to all the others (E, N, W); all other trajectories can be obtained by manipulating the expressions of these three.

4.2.2.1 Straight path

The path from S to N, indicated by $\boldsymbol{p}_{SN}$, is a straight line of equation $x = 0.5$, hence (4.5) can be used. The points $\boldsymbol{p}_0$ and $\boldsymbol{p}_1$ of equation (4.5) are determined by the intersections of the trajectory with the border lines of the oblique road.

$$\boldsymbol{p}_0 : \begin{cases} x = 0.5 \\ \cos\alpha\, x + \sin\alpha\, y + 1 = 0 \end{cases} \qquad \boldsymbol{p}_1 : \begin{cases} x = 0.5 \\ \cos\alpha\, x + \sin\alpha\, y - 1 = 0 \end{cases}$$

Hence, we obtain

$$\boldsymbol{p}_0 = \begin{bmatrix} 0.5 \\ -\frac{1+0.5\cos\alpha}{\sin\alpha} \end{bmatrix}, \quad \boldsymbol{p}_1 = \begin{bmatrix} 0.5 \\ \frac{1-0.5\cos\alpha}{\sin\alpha} \end{bmatrix}.$$

Thus, the expression of the trajectory parametrization is

$$\boldsymbol{p}_{SN}(u) = \begin{bmatrix} 0.5 \\ -\frac{1+0.5\cos\alpha}{\sin\alpha} \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{2}{\sin\alpha} \end{bmatrix} u. \tag{4.8}$$

4.2.2.2 Left and right turns

The path from S to W, indicated by $\boldsymbol{p}_{SW}$, is a left turn and its parametrization can be obtained by translating (4.6) after finding the appropriate values for the entry and exit angles. The center of curvature $\boldsymbol{o} = \begin{bmatrix} o_x & o_y \end{bmatrix}^\mathsf{T}$ can be easily found by using the support function of a circle with radius r_{SW}, of which I recall the expression that can be derived from (3.3) and (3.5):

$$\sigma_{SW}(\boldsymbol{s}) = \langle \boldsymbol{o},\, \boldsymbol{s} \rangle + r_{SW} \left\| \boldsymbol{s} \right\|.$$

Since the first and last part of this trajectory are straight lines of equations $x = 0.5$ and $\cos\alpha\, x + \sin\alpha\, y - 0.5 = 0$, and they are tangent to the circle; hence:

$$\begin{cases} \sigma_{SW}\left(\begin{bmatrix}1\\0\end{bmatrix}\right) = r_{SW} + o_x = 0.5, \\ \sigma_{SW}\left(\begin{bmatrix}\cos\alpha\\\sin\alpha\end{bmatrix}\right) = r_{SW} + o_x\cos\alpha + o_y\sin\alpha = 0.5. \end{cases}$$

Solving this system yields

$$\boldsymbol{o} = (0.5 - r_{SW})\begin{bmatrix}1\\\frac{1-\cos\alpha}{\sin\alpha}\end{bmatrix}.$$

Note that, for this trajectory, the entry angle is 0 and the exit angle is α; this allows us to obtain the complete expression of $\boldsymbol{p}_{SW}(u)$:

$$\boldsymbol{p}_{SW}(u) = (0.5-r_{SW})\begin{bmatrix}1\\\frac{1-\cos\alpha}{\sin\alpha}\end{bmatrix} + r_{SW}\begin{cases} \begin{bmatrix}1\\0\end{bmatrix} + \alpha\begin{bmatrix}0\\1\end{bmatrix}u & : u < 0, \\ \begin{bmatrix}\cos(\alpha u)\\\sin(\alpha u)\end{bmatrix} & : 0 \le u \le 1, \\ \begin{bmatrix}\cos\alpha\\\sin\alpha\end{bmatrix} + \alpha\begin{bmatrix}-\sin\alpha\\\cos\alpha\end{bmatrix}(u-1) & : u > 1. \end{cases} \quad (4.9)$$

The expression of $\boldsymbol{p}_{SE}(u)$ is obtained in the same way and by using (4.7):

$$\boldsymbol{p}_{SE}(u) = (0.5+r_{SE})\begin{bmatrix}1\\-\frac{1+\cos\alpha}{\sin\alpha}\end{bmatrix} + r_{SE}\begin{cases} \begin{bmatrix}-1\\0\end{bmatrix} + (\pi-\alpha)\begin{bmatrix}0\\1\end{bmatrix}u & : u < 0, \\ \begin{bmatrix}-\cos((\pi-\alpha)u)\\\sin((\pi-\alpha)u)\end{bmatrix} & : 0 \le u \le 1, \\ \begin{bmatrix}\cos\alpha\\\sin\alpha\end{bmatrix} + (\pi-\alpha)\begin{bmatrix}\sin\alpha\\-\cos\alpha\end{bmatrix}(u-1) & : u > 1. \end{cases} \quad (4.10)$$

4.2.2.3 Other inbound lanes

The expression of the trajectories from the other inbound lanes can be obtained by noting the following facts.

1. By rotating the intersection by α degrees clockwise, we get an oblique intersection of $\pi - \alpha$ degrees. Therefore, the expressions of the trajectories from E, i.e., $\boldsymbol{p}_{EW}(u)$, $\boldsymbol{p}_{ES}(u)$, and $\boldsymbol{p}_{EN}(u)$ can be obtained from (4.8), (4.9), and (4.10) by replacing α with $\pi - \alpha$ and then by rotating the result by α degrees (counterclockwise).

2. Given the symmetry of the crossroad, trajectories from opposite lanes are specular. This means that $\boldsymbol{p}_{NS}(u) = -\boldsymbol{p}_{SN}(u)$, $\boldsymbol{p}_{NE}(u) = -\boldsymbol{p}_{SW}(u)$, and $\boldsymbol{p}_{NW}(u) = -\boldsymbol{p}_{SE}(u)$. Similarly, $\boldsymbol{p}_{WE}(u) = -\boldsymbol{p}_{EW}(u)$, $\boldsymbol{p}_{WN}(u) = -\boldsymbol{p}_{ES}(u)$, and $\boldsymbol{p}_{WS}(u) = -\boldsymbol{p}_{EN}(u)$.

Since all expressions are more or less alike, I choose to not report them here.

4.2.3 Collision model

The absolute priority of an autonomous intersection management strategy is to avoid collision between vehicles; therefore, thoroughly studying the conditions that result in a collision is imperative. Simplifications over the collision model must not be underapproximative because the absence of collisions must be guaranteed under any assumption. In other words, any collision approximation must not be less restrictive than (4.2).

Definition 4.1 suggests that the collision problem should be approached by modeling the occupancy sets and how they change along the trajectory. Assuming that the shape of a vehicle does not change over time, the occupancy set can be obtained with a translation and rotation of a fixed set $S \subset \mathbb{R}^2$, representing the shape of the vehicle.

Consider the tangent vector $\hat{\boldsymbol{v}} = \begin{bmatrix} \hat{v}_x & \hat{v}_y \end{bmatrix}^\mathsf{T}$ to the trajectory $\boldsymbol{p}$:

$$\hat{\boldsymbol{v}}(u) = \frac{\boldsymbol{p}'(u)}{\|\boldsymbol{p}'(u)\|}.$$

The rotation matrix associated with $\hat{\boldsymbol{v}}(u)$ is the matrix $\boldsymbol{R}_{\boldsymbol{p}}(u) \in \mathbb{R}^{2\times 2}$ defined as

$$\boldsymbol{R}_{\boldsymbol{p}}(u) = \begin{bmatrix} \hat{v}_x(u) & -\hat{v}_y(u) \\ \hat{v}_y(u) & \hat{v}_x(u) \end{bmatrix}.$$

Given the shape set S of a vehicle, the expression of the occupancy function is

$$O(u) = \boldsymbol{p}(u) + \boldsymbol{R}_{\boldsymbol{p}}(u)S.$$

Since cars have an approximately rectangular shape, let us consider S to be a rectangle, i.e., $S = \left\{ \boldsymbol{s} = \begin{bmatrix} s_x & s_y \end{bmatrix}^\mathsf{T} \in \mathbb{R}^2 \ : \ |s_x| < w \wedge |s_y| < h \right\}$, with $w \in \mathbb{R}_+$ and $h \in \mathbb{R}_+$ being the half-width and the half-height of the rectangle, respectively, and $w > h$. However, analytically determining whether two rectangles overlap based on their positions and orientations is a hard problem. Therefore, I consider approximations that are easier to work with. A function $\tilde{O} : \mathbb{R} \to \mathcal{P}(\mathbb{R}^2)$ is said to be an *overapproximating occupancy function* of a vehicle if $\tilde{O}(u) \supseteq O(u)$, $\forall u \in \mathbb{R}$. Note that $\tilde{O}_1(u_1) \cap \tilde{O}_2(u_2) = \emptyset \Rightarrow O_1(u_1) \cap O_2(u_2) = \emptyset$.

4.2.3.1 Bounding circle

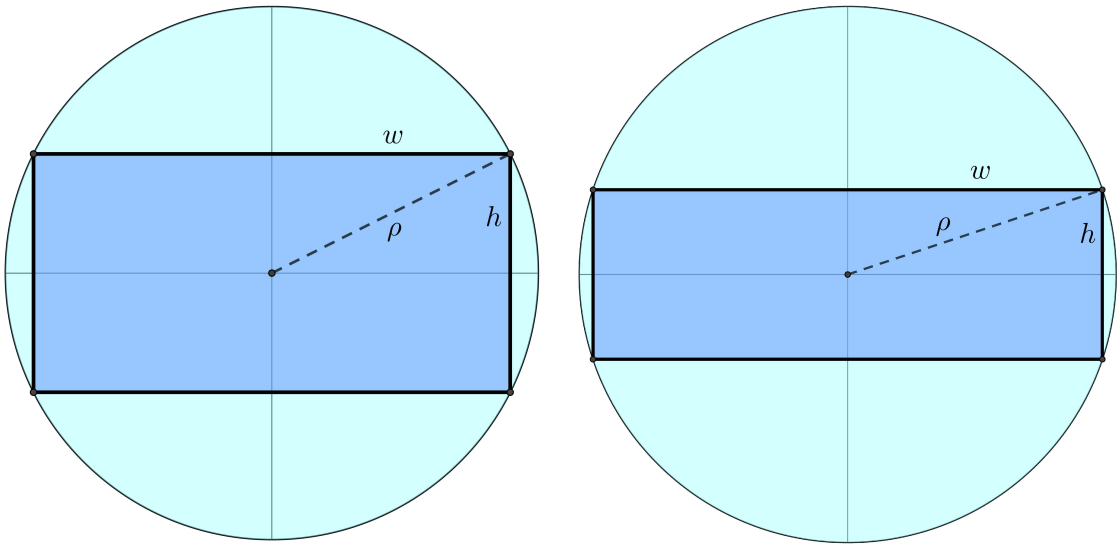
Analytically determining when two circles overlap is much simpler: two circles overlap when the distance between their centers is less than the sum of their radii; moreover, a circle is invariant by rotation. Thus, as a first approximation, let $\tilde{S} = \{\boldsymbol{s} \in \mathbb{R}^2 \ : \ \|\boldsymbol{s}\| < \rho\}$ to be a circle of radius $\rho \geq \sqrt{w^2 + h^2}$. We have $S \subset \tilde{S}$ because for any $\boldsymbol{s} = \begin{bmatrix} s_x & s_y \end{bmatrix}^\mathsf{T} \in S$, $s_x^2 < w^2$ and $s_y^2 < h^2$, hence $s_x^2 + s_y^2 < w^2 + h^2 \leq \rho^2$. Then, I let $\tilde{O}(u) = \boldsymbol{p}(u) + \tilde{S}$.

Given $\tilde{O}_1(u_1)$, $\tilde{O}_2(u_2)$, we have

$$\tilde{O}_1(u_1) \cap \tilde{O}_2(u_2) \neq \emptyset \iff \|\boldsymbol{p}_1(u_1) - \boldsymbol{p}_2(u_2)\| < \rho_1 + \rho_2, \tag{4.11}$$

which is an easier condition to deal with.

The simplicity of the circular approximation comes with another advantage: it provides a safety margin, especially laterally. However, this is also a drawback because, as figure



(a) $w/h = 2$ **(b)** $w/h = 3$

Figure 4.3: *Bounding circles of rectangles with different w/h ratio.*

4.3 shows, the lateral safety margin exasperates when the rectangle's width-to-height ratio increases; when $\rho = \sqrt{w^2 + h^2}$, with $w/h = 2$ the area of the bounding circle is 1.96 times larger than the area of the rectangle, whereas with $w/h = 3$, it is 2.62 times larger.

Since considered roads are two-way, it would be appropriate that the diameter of the circle is kept under the size of the lane, hence $\rho \leq 0.5$, to avoid complications in the collision model, since it would detect collisions between vehicles traveling along the same road with opposite directions. In view of this issue, a more elaborate solution is needed.

4.2.3.2 Straight envelope of circles

As shown in [120], better approximations are achieved when more circles are used to bound a rectangle. I now consider a continuous envelope of circles bounding the rectangle, that is the region spanned by a circle that moves along a segment on the major axis of the rectangle, as shown in figure 4.4. Given $\delta \geq 0$, the points $\begin{bmatrix} 0 & \delta \end{bmatrix}^\mathsf{T}$ and $\begin{bmatrix} 0 & -\delta \end{bmatrix}^\mathsf{T}$ are the centers of the extreme circles of the envelope and ρ is their radius. When $0 \leq \delta < w$, ρ should be grater than $\sqrt{(w-\delta)^2 + h^2}$, whereas when $\delta \geq w$, it should hold $\rho > h$.

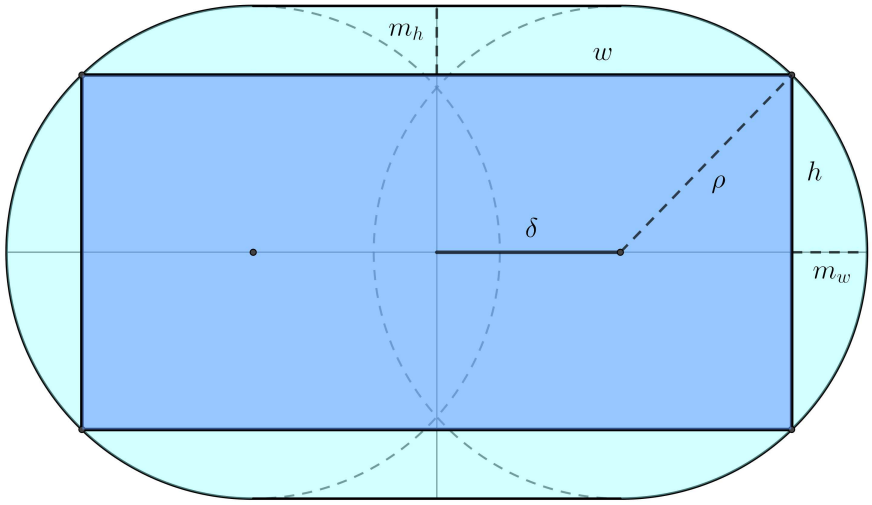


Figure 4.4: *Straight envelope of circles.*

This bounding shape better approximates a rectangle and works with any width-to-height ratio. Given some desired longitudinal and lateral margins, m_w and m_h respectively, δ and ρ can be designed by the following specifications:

$$\delta = m_w - m_h + w - h,$$
$$\rho = m_h + h,$$

subject to the following constraints.

$$\begin{cases} h - w \leq m_w - m_h < h \\ m_w^2 - 2m_w m_h - 2hm_w + h^2 \leq 0 \end{cases} \quad \vee \quad \begin{cases} m_w - m_h \geq h \\ m_h \geq 0 \end{cases}$$

The ratio of the bounding area to the area of the rectangle can be as low as 1.23 for a width-to-height ratio of 2 and 1.16 for a width-to-height ratio of 3, which is more than acceptable.

For this approximation, the expression of $\tilde{S}$ is given by

$$\tilde{S} = \bigcup_{\kappa \in [-\delta, \delta]} \left\{ \boldsymbol{s} \in \mathbb{R}^2 \ : \ \left\| \boldsymbol{s} - \begin{bmatrix} \kappa & 0 \end{bmatrix}^\mathsf{T} \right\| < \rho \right\}$$

and $\tilde{O}(u) = \boldsymbol{p}(u) + \boldsymbol{R}_{\boldsymbol{p}}(u)\tilde{S}$. The collision condition becomes

$$\begin{aligned} \tilde{O}_1(u_1) \cap \tilde{O}_2(u_2) \neq \emptyset \iff & \|\boldsymbol{p}_1(u_1) + \kappa_1 \hat{\boldsymbol{v}}(u_1) - \boldsymbol{p}_2(u_2) - \kappa_2 \hat{\boldsymbol{v}}(u_2)\| < \rho_1 + \rho_2, \\ & \forall\, \kappa_1 \in [-\delta_1, \delta_1]\,,\ \forall\, \kappa_2 \in [-\delta_2, \delta_2]\,. \end{aligned} \tag{4.12}$$

Although in a closed analytical form, this condition is less manageable than condition (4.11), given the presence of κ_1 and κ_2 that vary in continuous intervals. This can be mitigated by checking the condition for a finite number of (κ_1, κ_2) pairs, provided that the union of the obtained circles completely covers the rectangle, as in [120]; obviously, this results in a jagged safety margin.

4.2.3.3 Radial envelope of circles

Another approximation strategy would be to consider the envelope of a circle spanning an arc of trajectory instead of a segment on the axis of the rectangle, as figure (4.5) shows. Firstly, note that for a straight trajectory, this approximation is identical to the previous one. Since the considered trajectories are either straight lines or arcs of circumference, let us consider the latter. In this case, we have an arc of circumference of radius r, and $\delta > 0$ is the half-length of the arc spanned by the circle. Moreover, $\rho \geq \sqrt{(r + h - r\cos(\delta/r))^2 + (w - r\sin(\delta/r))^2}$ when $\delta < r \arctan \frac{w}{r+h}$, and $\rho \geq \sqrt{(r+h)^2 + w^2} - r$ otherwise.

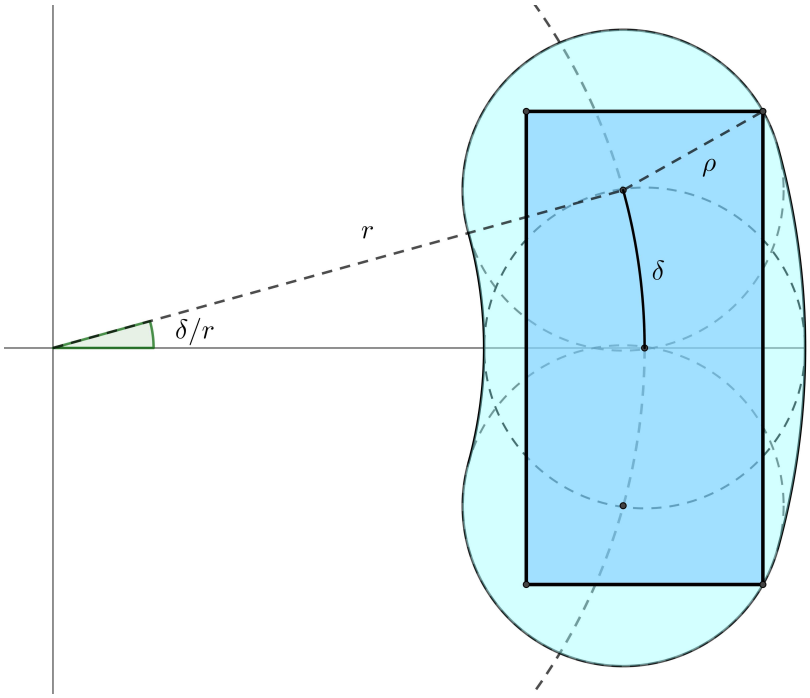


Figure 4.5: *Radial envelope of circles.*

This approximation is not as good as the previous one in terms of the ratio between the bounding area and the area of the rectangle; furthermore, the safety margins are not symmetric, with the inner margin being thicker than the outer margin. Nevertheless, these issues are less evident with larger values of r.

This time, the expression of $\tilde{O}(u)$ is not given by the translation and rotation of a fixed set $\tilde{S}$, but instead by

$$\tilde{O}(u) = \bigcup_{\kappa \in [-\bar{\kappa}, \bar{\kappa}]} \left\{ \boldsymbol{s} \in \mathbb{R}^2 \ : \ \| \boldsymbol{s} - \boldsymbol{p}\,(u - \kappa) \| < \rho \right\},$$

with $\bar{\kappa} = \delta / l$. The collision condition is

$$\begin{gathered} \tilde{O}_1(u_1) \cap \tilde{O}_2(u_2) \neq \emptyset \iff \| \boldsymbol{p}_1\,(u_1 - \kappa_1) - \boldsymbol{p}_2\,(u_2 - \kappa_2) \| < \rho_1 + \rho_2, \\ \forall\, \kappa_1 \in [-\bar{\kappa}_1, \bar{\kappa}_1]\,,\ \forall\, \kappa_2 \in [-\bar{\kappa}_2, \bar{\kappa}_2]\,. \end{gathered} \tag{4.13}$$

This condition is closer to (4.11) in its form, although, like (4.12) it represents an infinite number of conditions due to κ_1 and κ_2 varying in continuous intervals.

4.2.4 Intersection zones

Now, I continue with the exposition of the assumptions about the intersection model, in order to add more details and work with a more concrete version of the problem. I assume the intersection is divided into three zones, named *merging zone* (MZ), *control zone* (CZ), and *scheduling zone* (SZ), as Figure 4.6 shows. This assumption is not new in the literature and, albeit with different variations, was proposed in many works, e.g., [70].

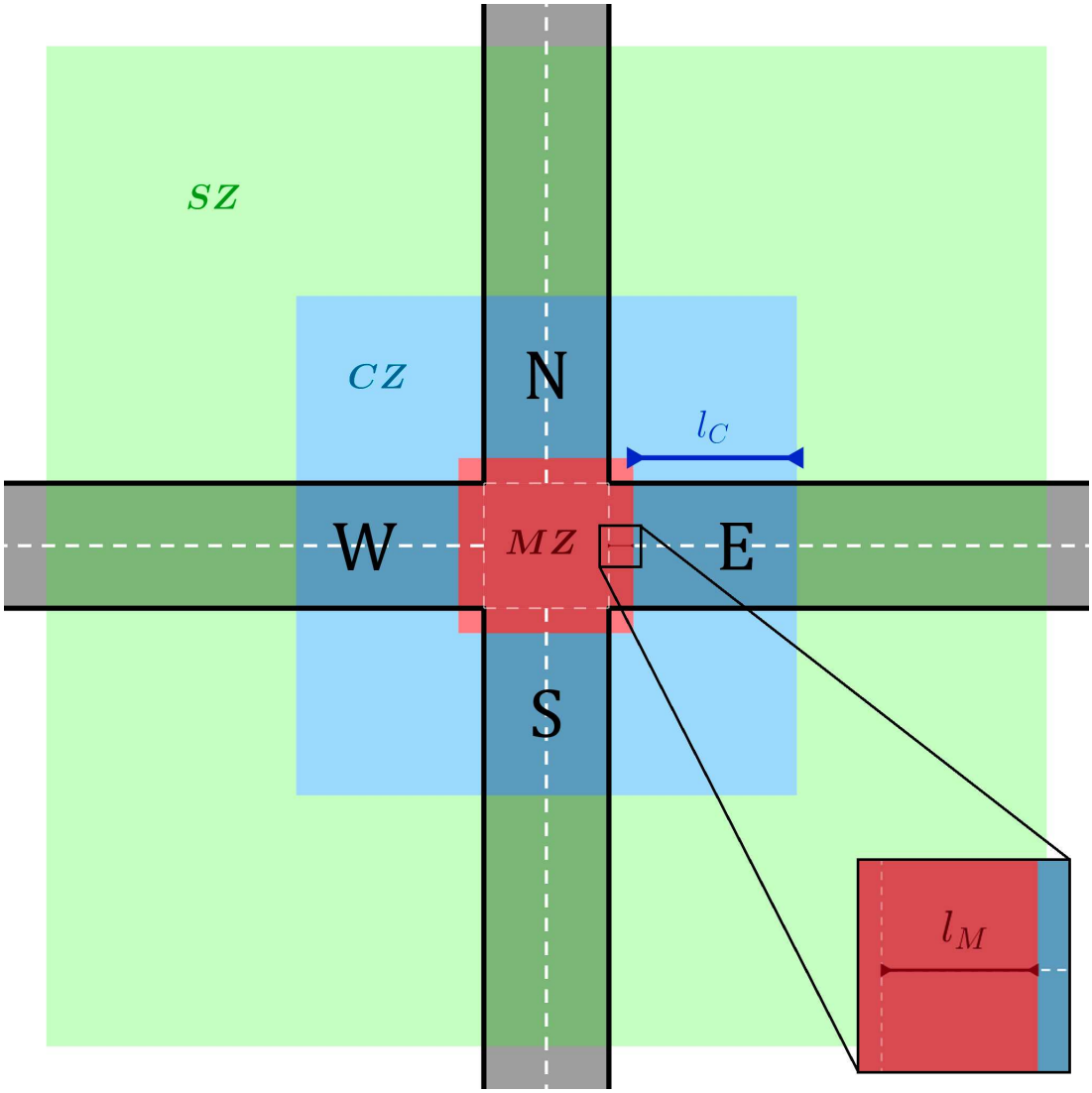


Figure 4.6: *Zones of the intersection.*

Instead of defining the three zones as physical regions of the XY plane, I define them in terms of the abscissa of a vehicle along its trajectory.

The merging zone is the region of potential lateral collisions. Rear-end collisions are possible at any point of the trajectory of a vehicle, whereas lateral collisions are only possible in the intersection center, where trajectories converge. Given vehicle i with trajectory $\boldsymbol{p}_i$ and its associated occupancy function O_i, the set of the occupancy functions of other vehicles it may collide laterally with is indicated with $\mathcal{O}$. The merging zone M_i of this vehicle is the set of abscissas where a collision is possible:

$$M_i = \operatorname{conv}\left(\{u_i \in \mathbb{R} \,:\, (\exists O_j \in \mathcal{O}, u_j \in \mathbb{R} \,:\, O_i(u_i) \cap O_j(u_j) \neq \emptyset)\}\right),$$

where the operation of convex hull was used to ensure that the merging zone is an interval. The merging zone contains a subregion deemed as the *intersection center*, which corresponds to the $[0, 1]$ interval. The length of the road segment that precedes the intersection center within the merging zone is denoted by l_M.

Adjacent to the merging zone there is the control zone, which is the region of a road where vehicles carefully regulate their cruising plan to adhere to a pre-established schedule, thereby preventing collisions. The physical length of the control zone is indicated by l_C. Preceding the control zone there is the scheduling zone, where vehicles elaborate the schedule they will follow within the control zone.

Now, I make an important assumption about the kinematic laws of vehicles.

Assumption 4.1. *The speed of each vehicle is constant inside the merging zone.*

This assumption is not unusual in the literature, as present in [70] and other works. The speed of a vehicle inside the merging zone is denoted by $v_M \in \mathbb{R}_+$, while the arrival time at the same zone is $\tau_M \in \mathbb{R}$. Let τ be the arrival time at the intersection center, that is the instant of time in which $u = 0$. Moreover, let T_M be the time interval in which the vehicle is inside the merging zone. With Assumption 4.1, the expression of $\gamma(t)$ when $t \in T_M$ is

$$\gamma(t) = \frac{v_M}{l}(t - \tau) = \omega(t - \tau), \quad \forall t \in T_M, \tag{4.14}$$

with $\omega = v_M/l$. Therefore, $T_M = \{t \in \mathbb{R} \,:\, \omega(t - \tau) \in M\}$; also, note that $\tau_M = \inf T_M$. For ease of thought, ω is referred to as the arrival speed at the intersection, although not physically being a velocity, because I will work with ω and not with v_M throughout the thesis.

Let us define the vector $\boldsymbol{\lambda} = \begin{bmatrix}\tau & \omega\end{bmatrix}^\mathsf{T}$, which I will refer to as the *arrival state* of the vehicle. Not all values of $\boldsymbol{\lambda}$ are possible, because:

- vehicles are subject to speed limits, therefore ω is also limited;
- as a consequence of speed limits and physical limitations, a vehicle cannot choose to arrive at the intersection at an arbitrary instant of time τ;
- other constraints must be considered depending on the chosen kinematic law that can furtherly narrow the domain of $\boldsymbol{\lambda}$.

$\Lambda_\gamma \subset \mathbb{R} \times \mathbb{R}_+$ denotes the domain of $\boldsymbol{\lambda}$ under kinematic law γ. So, considering all feasible kinematic laws, we can write $\boldsymbol{\lambda} \in \Lambda = \bigcup_{\gamma \in \mathbb{\Gamma}} \Lambda_\gamma$. Note that Λ_γ does not take into account the interaction with other vehicles.

When a vehicle approaches the intersection, its cruising plan has determined its arrival time and speed at the intersection, named τ_0 and ω_0, respectively. In the scheduling zone, the vehicle knows about the presence of other vehicles arriving at the intersection, which have possibly conflicting cruising plans, meaning that with their current arrival time and speed, collisions are possible if no action is taken. Taking the correcting action means adjusting the value of $\boldsymbol{\lambda}$ to avoid lateral collision in the merging zone while also ensuring the absence of rear-end collisions before and after the crossing. Consider a pair of vehicles, their arrival states $\boldsymbol{\lambda}_1$, $\boldsymbol{\lambda}_2$, their kinematic laws γ_1, γ_2, their merging zone M_1, M_2 and the time intervals

$T_{M,1}$, $T_{M,2}$ in which they both are inside their merging zones. Assumption 4.1 allows thinking about the concept of collision with respect to $\boldsymbol{\lambda}_1$ and $\boldsymbol{\lambda}_2$, rather than γ_1 and γ_2, by introducing a new definition of lateral collision course.

Definition 4.5. *Two vehicles with arrival states $\boldsymbol{\lambda}_1$, $\boldsymbol{\lambda}_2$ and occupancy sets O_1, O_2 are said to be on a lateral collision course if*

$$\exists t \in T_{M,1} \cap T_{M,2} \ : \ O_1(\omega_1(t-\tau_1)) \cap O_2(\omega_2(t-\tau_2)) \neq \emptyset.$$

4.2.5 Decentralized solution approach

The ICP can be solved using multiple approaches: with its general formal definition, it is suitable to be tackled using centralized and decentralized strategies, or a mix of the two. The effort of this work is to study the problem to enable the development of a distributed approach, although some information is assumed to be globally known; for example, vehicles are assumed to be aware of the geometry of the intersection they are approaching.

Vehicles entering the scheduling zone establish a connection with other vehicles in range, therefore creating a communication graph. This graph is dynamic, as vehicles approach and leave the intersection and come near each other, creating new communication channels. Considering a wide time window, I assume there are n vehicles approaching the intersection and the communication graph is $\mathcal{G}(t) = (\mathcal{V}, \mathcal{E}(t))$, with $\mathcal{V} = [\![1, n]\!]$. When a connection between two vehicles is established, they exchange their knowledge and thereby information spreads across the network; this process is called *flooding*. Flooding allows vehicles to know data about other vehicles they are not directly connected with, which is crucial for distributed collision avoidance strategies.

I assume each vehicle to be selfish, meaning that its goal is to minimize a certain cost associated with the effort required to deviate from its initial schedule or to the delay introduced by the intersection crossing. Hence, for each vehicle $i \in \mathcal{V}$, problem (4.4) becomes:

$$\begin{aligned} \text{minimize} \quad & F_i[\gamma_i] \\ \text{subject to} \quad & O_i(\gamma_i(t)) \cap O_j(\gamma_j(t)) = \emptyset \quad \forall t \in \mathbb{R}, \quad \forall j \in \mathcal{V},\ i \neq j \\ \text{and} \quad & \gamma_i \in \mathbb{\Gamma}_i \end{aligned} \tag{4.15}$$

As mentioned before, vehicle i enters the scheduling zone with an initial value $\boldsymbol{\lambda}_{i,0}$ and must solve problem (4.15) inside that zone: this means it enters the control zone with the final value of $\boldsymbol{\lambda}_i$, which all other vehicles should acknowledge. Then, it uses the control zone to accelerate or decelerate in order to arrive at the intersection at the agreed time τ_i and with the agreed speed ω_i.

4.3 Kinematic laws

There is an infinite number of possible control laws that a vehicle can choose to meet the agreed schedule. Obviously, the choice of γ has a direct impact on the choice of $\boldsymbol{\lambda}$ since, for a given γ, $\boldsymbol{\lambda} \in \Lambda_\gamma$. I want to stress the fact that, because of Assumption 4.1, the combination of all $\{\boldsymbol{\lambda}_i\}_{i \in [\![1,n]\!]}$ completely determines the presence or absence of collisions; therefore, understanding how the choice of γ influences the domain of $\boldsymbol{\lambda}$ is crucial. Any chosen control algorithm must ensure the vehicle respects the arrival time and speed that it

and the other vehicles agreed upon during the scheduling. In this section, I will analyze a few possible control laws and discuss some of their properties.

I start by introducing some dynamical models that can be used to study the kinematics of a vehicle. This study is not about the control of the dynamics of a vehicle because the objective is to obtain results about its kinematics instead; therefore, I disregard the details on how the control laws are implemented. For the same reason, I neglect the lateral dynamics and focus on the longitudinal dynamics. I describe two (linear) models that are widely used in the literature: second-order and third-order longitudinal models.

A second-order longitudinal model of a vehicle consists of a double integrator that relates the position and the velocity of the vehicle to the acceleration it is subject to:

$$\begin{bmatrix} \dot{s}(t) \\ \dot{v}(t) \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} s(t) \\ v(t) \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} a(t).$$

The acceleration a is given by the total force acting on the vehicle divided by its mass. Those forces are typically the thrust of the engine, the drag of the air resistance, and so on. This is the most simple model and it assumes the acceleration as the control input, implying the possibility of it instantaneously changing. This is certainly not the case, as engines have certain dynamics characterized by a response time; however, for not demanding control algorithms and for reasonably estimating some quantities of interest.

A third-order longitudinal model treats the acceleration as a state variable:

$$\begin{bmatrix} \dot{s}(t) \\ \dot{v}(t) \\ \dot{a}(t) \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} s(t) \\ v(t) \\ a(t) \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} j(t),$$

where j denotes the jerk of the vehicle.

This section focuses on the kinematics of a single vehicle inside the control zone. Let τ_C and v_C be the arrival time and speed at the control zone, respectively. The original value of the arrival state is denoted by $\boldsymbol{\lambda}_0 = \begin{bmatrix} \tau_0 & \omega_0 \end{bmatrix}^\mathsf{T}$, with τ_0 and ω_0 being the original arrival time and speed at the intersection. To simplify the notation, the time axis is shifted so that $t = 0$ when the vehicle enters the control zone. Moreover, I consider $l_M = 0$, so that the merging zone coincides with the intersection center and $\tau = \tau_M$; this is not restrictive since the speed is constant inside the merging zone. I assume that a vehicle has no reason to accelerate or decelerate inside the control zone, except to avoid collisions with other vehicles. Therefore, its speed would be constant inside the control zone if it were not for the schedule change due to the intersection, which means that $v_C = l\omega_0$ and $l_C = l\tau_0\omega_0$; in addition to this, we also have $v_M = l\omega$.

If $v(t) > 0 \ \forall t > 0$, the vehicle does not stop before reaching the merging zone; if this is verified for all vehicles, it results in a stop-free policy, where vehicles efficiently cross the intersection without idling.

I assume the vehicles are subject to speed and acceleration constraints, so the following bounds on v and a must be imposed:

$$\begin{cases} a^- \leq a(t) \leq a^+ \\ 0 \leq v^- \leq v(t) \leq v^+ \end{cases} \quad \forall t > 0; \tag{4.16}$$

a stop-free policy requires $v^- > 0$. For the second-order system, the control input is $a(t)$

and it should transfer the state of the system from the initial state[2] $\begin{bmatrix}0 & v_0\end{bmatrix}^\mathsf{T}$ at time $t = 0$ to the final state $\begin{bmatrix}l_C & v_M\end{bmatrix}^\mathsf{T}$ at time $t = \tau$. For a third-order system, $j(t)$ is the control input, whereas the initial and final states are $\begin{bmatrix}0 & v_0 & 0\end{bmatrix}^\mathsf{T}$ and $\begin{bmatrix}l_C & v_M & 0\end{bmatrix}^\mathsf{T}$, respectively. The expression of the control input will depend on these values and, ultimately, inequalities (4.16) will determine the domain Λ_γ of the decision variables, for a given γ. If the resulting domain is incompatible with the collision avoidance conditions, then the designed control law is unsuitable.

4.3.1 Minimum energy control

As discussed in Section 3.4.5, the minimum energy control (MEC) is a control input that transfers the state of an LTI system from an initial state to a final state in a finite amount of time and minimizes its associated energy.

4.3.1.1 Second-order MEC

In the second-order model, the state and input matrices are, respectively,

$$\boldsymbol{A} = \begin{bmatrix}0 & 1\\ 0 & 0\end{bmatrix}, \quad \boldsymbol{B} = \begin{bmatrix}0\\ 1\end{bmatrix}.$$

Note that the state matrix is a Jordan matrix consisting of a single Jordan block of index 2 associated with the eigenvalue 0. Given this fact, we immediately obtain the exponential of $\boldsymbol{A}$:

$$e^{\boldsymbol{A}t} = \begin{bmatrix}1 & t\\ 0 & 1\end{bmatrix}.$$

The controllability Gramian of the system is

$$\boldsymbol{W}(t) = \int_0^t \begin{bmatrix}1 & \xi\\ 0 & 1\end{bmatrix}\begin{bmatrix}0\\ 1\end{bmatrix}\begin{bmatrix}0 & 1\end{bmatrix}\begin{bmatrix}1 & 0\\ \xi & 1\end{bmatrix}\mathrm{d}\xi = \begin{bmatrix}t^3/3 & t^2/2\\ t^2/2 & t\end{bmatrix},$$

which is invertible for every $t > 0$. Therefore, the desired input is given by

$$\begin{aligned} a(t) &= \left(e^{\boldsymbol{A}(\tau-t)}\boldsymbol{B}\right)^\mathsf{T}\boldsymbol{W}^{-1}(\tau)\left(l\begin{bmatrix}\tau_0\omega_0\\ \omega\end{bmatrix} - e^{\boldsymbol{A}\tau}l\begin{bmatrix}0\\ \omega_0\end{bmatrix}\right) = \\ &= l\left(\frac{6\omega_0(\tau-\tau_0)(2t-\tau)}{\tau^3} + \frac{2(\omega-\omega_0)(3t-\tau)}{\tau^2}\right). \end{aligned} \tag{4.17}$$

The expression of $v(t)$ can be found by integration, with initial condition $v(0) = v_C$:

$$\begin{aligned} v(t) &= v_C + \int_0^t a(\xi)\mathrm{d}\xi = \\ &= l\left(\omega_0 + \frac{6\omega_0(\tau-\tau_0)(t-\tau)t}{\tau^3} + \frac{(\omega-\omega_0)(3t-2\tau)t}{\tau^2}\right). \end{aligned} \tag{4.18}$$

[2]The dynamical state of the system is not to be confused with the arrival state $\boldsymbol{\lambda}$.

Inequalities (4.16) can be solved by finding the extrema of v and a, which will depend on τ, τ_0, ω, and ω_0. The maxima and minima point of a can be only found on the boundary of the $[0, \tau]$ interval, that are:

$$a(0) = l\frac{6\tau_0\omega_0 - 2\tau\left(\omega + 2\omega_0\right)}{\tau^2}, \quad a(\tau) = -l\frac{6\tau_0\omega_0 - 2\tau\left(2\omega + \omega_0\right)}{\tau^2}. \tag{4.19}$$

As for the velocity, assuming that v_C already satisfies the constraints, we are interested in evaluating the extremum of v inside the $[0, \tau]$ interval. This is found at t^* defined by $a(t^*) = 0$, which yields:

$$t^* = \frac{\tau\left(\omega + 2\omega_0\right) - 3\tau_0\omega_0}{\tau\left(\omega + \omega_0\right) - 2\tau_0\omega_0}\frac{\tau}{3}. \tag{4.20}$$

If $0 < t^* < \tau$, then the extremum of v is:

$$v(t^*) = l\frac{\left(\omega^2 + \omega\omega_0 + \omega_0^2\right)\tau^2 - 6\tau_0\omega_0(\omega + \omega_0)\tau + 9\tau_0^2\omega_0^2}{3\tau\left(2\tau_0\omega_0 - (\omega + \omega_0)\tau\right)}. \tag{4.21}$$

Using (4.16) in combination with (4.19), (4.20) and (4.21) yields the following constraints, which define the domain Λ_{MEC}:

$$\Lambda_{\text{MEC}} : \begin{cases} \frac{6\tau_0\omega_0 - 2\tau(\omega+2\omega_0)}{\tau^2} \in [a^-/l, a^+/l]\,, \\ -\frac{6\tau_0\omega_0 - 2\tau(2\omega+\omega_0)}{\tau^2} \in [a^-/l, a^+/l]\,, \\ \frac{\left(\omega_0^2+\omega_0\omega+\omega^2\right)\tau^2 - 6\tau_0\omega_0(\omega_0+\omega)\tau + 9\tau_0^2\omega_0^2}{3\tau(2\tau_0\omega_0 - (\omega_0+\omega)\tau)} \in [v^-/l, v^+/l] \;\vee\; \frac{1}{3}\frac{\tau(2\omega_0+\omega) - 3\tau_0\omega_0}{\tau(\omega_0+\omega) - 2\tau_0\omega_0} \notin [0,1]\,, \\ \omega \in [v^-/l, v^+/l]\,. \end{cases}$$

The energy associated to $a(t)$ is

$$E[a] = \int_0^\tau a^2(t)\mathrm{d}t = 4l^2\left(\frac{3\omega_0^2\left(\tau - \tau_0\right)^2}{\tau^3} + \frac{3\omega_0\left(\tau - \tau_0\right)\left(\omega - \omega_0\right)}{\tau^2} + \frac{\left(\omega - \omega_0\right)^2}{\tau}\right). \tag{4.22}$$

Let us see a numerical example. Suppose the length of the trajectory inside the intersection center is $l = 10\,[\text{m}]$ and the control zone is $l_C = 30\,[\text{m}]$ long. A vehicle is approaching the intersection with a speed $v_C = 28.8\,[\text{km/h}]$, which mean $\tau_0 = 3.75\,[\text{s}]$ and $\omega_0 = 0.8\,[\text{s}^{-1}]$. Figure 4.7 shows Λ_{MEC} obtained with these values of τ_0 and ω_0.

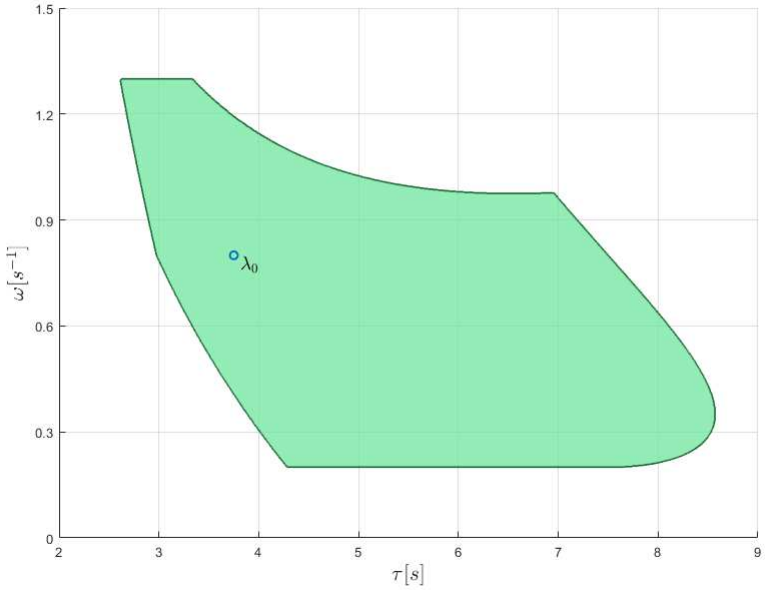


Figure 4.7: *Domain of* $\boldsymbol{\lambda}$ *produced by the second-order MEC.*

During the collision avoidance phase, the vehicle agrees an arrival time $\bar{\tau} = 3.5\,[\mathrm{s}]$ and an arrival speed $\omega = 1.15\,[\mathrm{s}^{-1}]$. The obtained position, speed, and acceleration profiles are depicted in Figure 4.8, where the red dashed line represents the position graph if the car had not taken a correcting action.

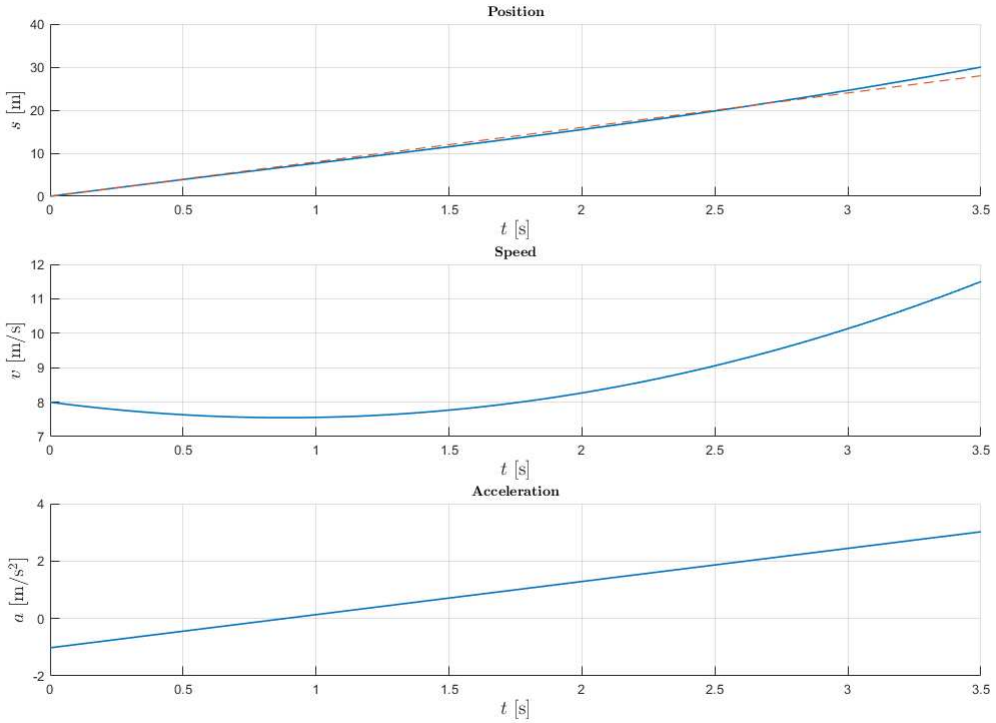


Figure 4.8: *Position, speed, and acceleration profiles produced by the second-order MEC.*

4.3.1.2 Third-order MEC

The minimum energy control for the third-order system is conceptually similar but more complex to analyze. In this case, the state and input matrices are, respectively,

$$\boldsymbol{A} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix}, \quad \boldsymbol{B} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}.$$

Once again, $\boldsymbol{A}$ is a Jordan matrix with a single null eigenvalue of index 3, and its exponential is:

$$e^{\boldsymbol{A}t} = \begin{bmatrix} 1 & t & t^2/2 \\ 0 & 1 & t \\ 0 & 0 & 1 \end{bmatrix}.$$

The controllability Gramian of the system is

$$\boldsymbol{W}(t) = \int_0^t \begin{bmatrix} 1 & \xi & \xi^2/2 \\ 0 & 1 & \xi \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \begin{bmatrix} 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ \xi & 1 & 0 \\ \xi^2/2 & \xi & 1 \end{bmatrix} \mathrm{d}\xi = \begin{bmatrix} t^5/20 & t^4/8 & t^3/6 \\ t^4/8 & t^3/3 & t^2/2 \\ t^3/6 & t^2/2 & t \end{bmatrix},$$

invertible for every $t > 0$. The control input is obtained as before, that is:

$$
\begin{aligned}
j(t) &= \left(e^{\boldsymbol{A}(\tau-t)}\boldsymbol{B}\right)^{\mathsf{T}} \boldsymbol{W}^{-1}(\tau) \left(l \begin{bmatrix} \tau_0\omega_0 \\ \omega \\ 0 \end{bmatrix} - e^{\boldsymbol{A}\tau} l \begin{bmatrix} 0 \\ \omega_0 \\ 0 \end{bmatrix} \right) = \\
&= l \left(-\frac{60\omega_0(\tau-\tau_0)(6t^2-6\tau t+\tau^2)}{\tau^5} - \frac{12(\omega-\omega_0)(15t^2-14\tau t+2\tau^2)}{\tau^4} \right).
\end{aligned} \tag{4.23}
$$

The expressions of $a(t)$ and $v(t)$ can be obtained by integration, with initial conditions $a(0) = 0$ and $v(0) = v_C$:

$$
a(t) = l \left(-\frac{60\omega_0(\tau-\tau_0)(2t^2-3\tau t+\tau^2)t}{\tau^5} - \frac{12(\omega-\omega_0)(5t^2-7\tau t+2\tau^2)t}{\tau^4} \right), \tag{4.24}
$$

$$
v(t) = l \left(\omega_0 - \frac{30\omega_0(\tau-\tau_0)(t^2-2\tau t+\tau^2)t^2}{\tau^5} - \frac{(\omega-\omega_0)(15t^2-28\tau t+12\tau^2)t^2}{\tau^4} \right). \tag{4.25}
$$

Obtaining an analytical expression of (4.16) in this case is burdensome, due to the degrees of the polynomials involved. However, the same graphical results of the second-order version can be obtained numerically, and are shown in Figures 4.9 and 4.10.

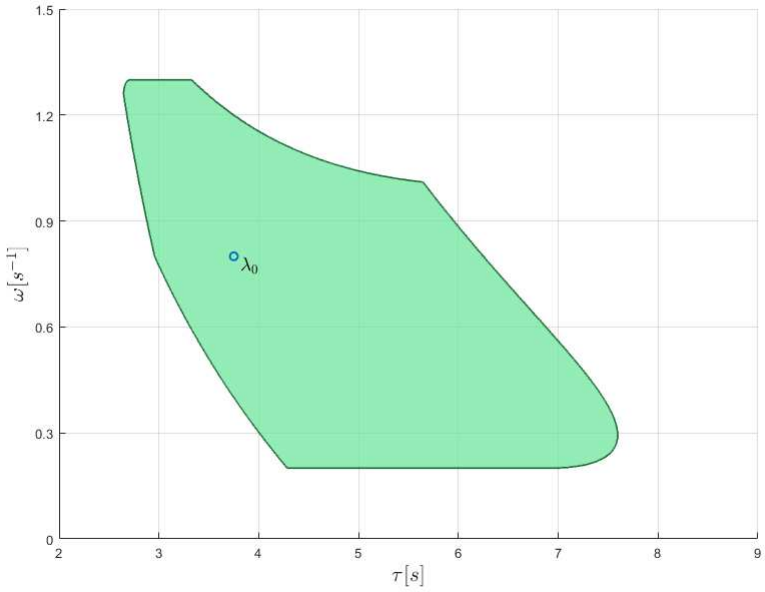


Figure 4.9: *Domain of* $\boldsymbol{\lambda}$ *produced by the third-order MEC.*

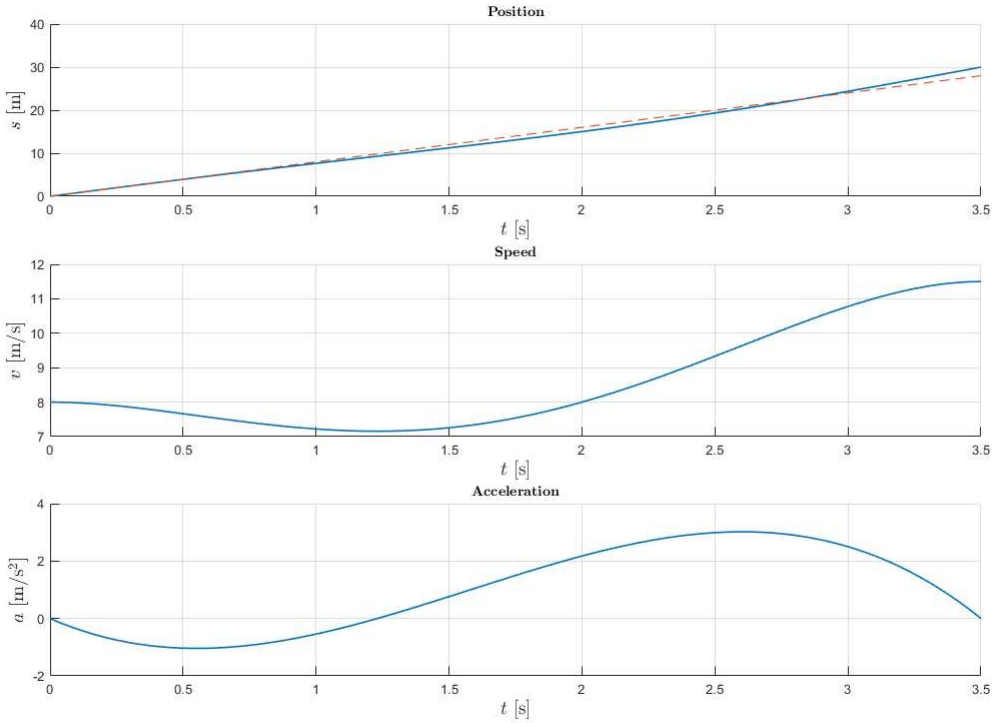


Figure 4.10: *Position, speed, and acceleration profiles produced by the third-order MEC.*

Visually, we can notice the domain Λ_{MEC} is narrower, but the assumptions and the resulting kinematics are more realistic.

Figures 4.7 and 4.9 show that the MEC yields a bounded domain. Intuitively, this is expected, as ω was constrained inside a closed interval and τ cannot be arbitrarily low, since that would mean an unrealistic time of arrival. However, having also an upper bound for τ is a significant limitation, because it means a vehicle cannot queue up at the intersection; moreover, the MEC does not allow a vehicle to stop.

4.3.2 Stop and go

The stop-and-go (SnG) dynamics is a simple model of a vehicle stopping at a certain point before the intersection, waiting for the agreed moment to cross it. Since a vehicle can wait indefinitely, Λ_{SnG} is expected to be unbounded, as τ is allowed to be arbitrarily large. In this situation, I consider a second-order model only, as, in this case, a more complex model does not lead to substantially different conclusions.

The SnG involves 5 phases.

1. The vehicle proceeds with constant speed v_C for t_1 seconds, up to the point when it begins to decelerate.

2. The vehicle decelerates with constant acceleration $a_{stop} < 0$ for t_{stop} seconds, until it stops at $s = l_{stop}$. The distance l_{stop} depends on how many vehicles are already in the queue before the intersection.

3. The vehicle idles at the stopping point for t_{idle} seconds.

4. The vehicle accelerates with constant acceleration $a_{start} > 0$ for t_{start} seconds.

5. The vehicle finishes the acceleration phase and continues with constant speed v_M for t_5 seconds until it reaches the merging zone at time $\tau = t_1 + t_{stop} + t_{idle} + t_{start} + t_5$.

Figure 4.11 pictures the described kinematics of a vehicle approaching the intersection with the same initial arrival state as the MEC scenario, that is, $\tau_0 = 3.75\,[\text{s}]$ and $\omega_0 = 0.6\,[\text{s}^{-1}]$, with final arrival time $\tau = 10\,[\text{s}]$ and arrival speed $\omega_0 = 0.5\,[\text{s}^{-1}]$. This is achieved with by decelerating for $t_{stop} \approx 2.67$ seconds with $a_{stop} = -3\,[\text{m/s}^2]$, idling at $l_{stop} = 20$ meters for $t_{idle} \approx 2.92$ seconds, then accelerating for $t_{start} \approx 2.5$ seconds with $a_{start} = 2\,[\text{m/s}^2]$.

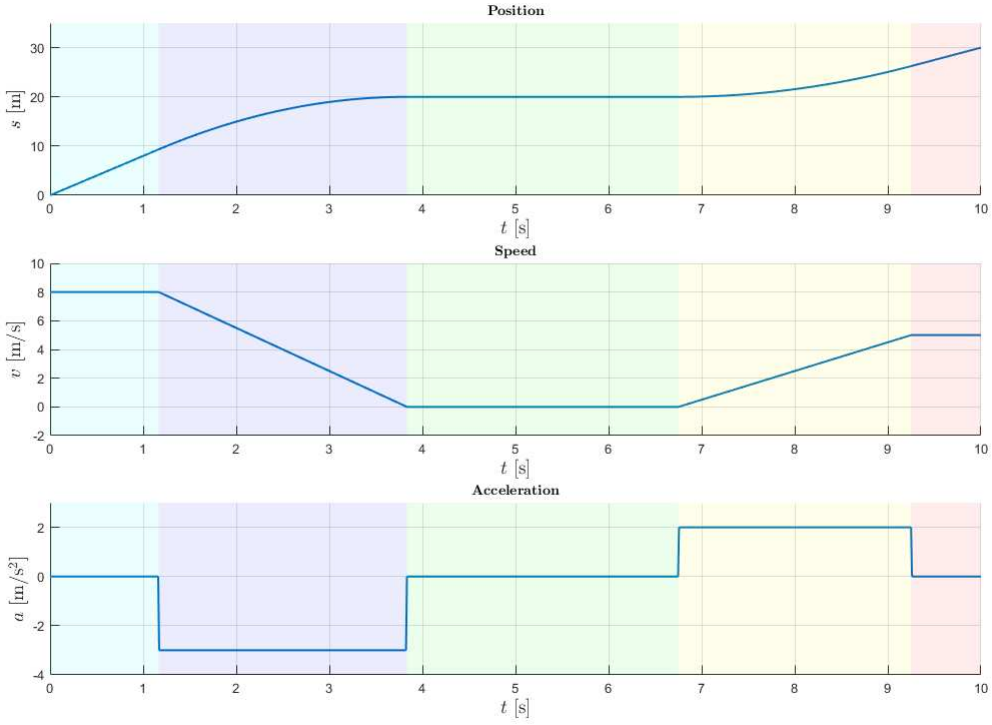


Figure 4.11: *Position, speed, and acceleration profiles produced by the SnG.*

This time, the constraints are computed differently. First, I focus on the fact that the vehicle must accelerate from $v = 0$ to $v = v_M$ inside $l_{start} = l_C - l_{stop}$ meters with acceleration a_{start}. From basic physics, it holds $v_M = \sqrt{2a_{start}l_{start}}$; since $a_{start} \leq a^+$ and $v^- \leq v_M \leq v^+$, the first constraint is

$$v^- \leq v_M \leq \min(v^+, \sqrt{2a^+ l_{start}}). \tag{4.26}$$

The values of the durations of phases 1, 2, 4, and 5 are given by the following formulas

$$t_1 = \frac{l_{stop}}{v_C} + \frac{v_C}{2a_{stop}}, \quad t_{stop} = -\frac{v_C}{a_{stop}}, \quad t_{start} = \frac{v_M}{a_{start}}, \quad t_5 = \frac{l_{start}}{v_M} - \frac{v_M}{2a_{start}},$$

whereas t_{idle} is a free variable, meaning that a vehicle can choose to idle for as long as it needs to. The value of τ is given by the sum of these durations:

$$\tau = \frac{l_{stop}}{v_C} - \frac{v_C}{2a_{stop}} + \frac{l_{start}}{v_M} + \frac{v_M}{2a_{start}} + t_{idle}.$$

For a given value of v_M, τ reaches its minimum value when $a_{stop} = a^-$, $a_{start} = a^+$, and $t_{idle} = 0$; therefore, the second and last constraint is

$$\tau \geq \frac{l_{stop}}{v_C} - \frac{v_C}{2a^-} + \frac{l_{start}}{v_M} + \frac{v_M}{2a^+}. \tag{4.27}$$

Combining constraints (4.26) and (4.27) produces the domain Λ depicted in Figure 4.9.

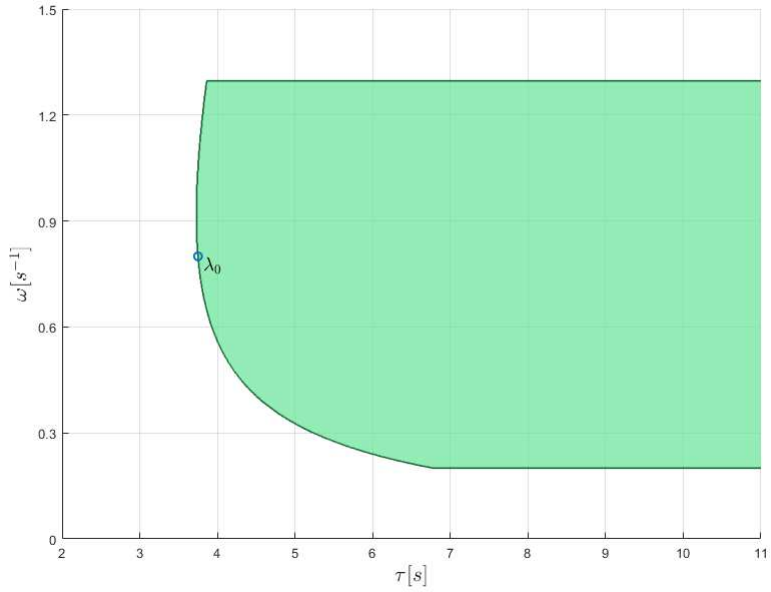


Figure 4.12: *Domain of* $\boldsymbol{\lambda}$ *produced by the SnG.*

This kind of control proves that Λ is unbounded, at least in one direction: indeed, because of (4.27), we can affirm that

$$\forall \omega \in]\omega_{min}, \omega_{max}[\ \exists \tau\ :\ \begin{bmatrix}\tau & \omega\end{bmatrix}^\mathsf{T} \in \Lambda, \tag{4.28}$$

with $\omega_{sup} = \sup\left\{\omega\ :\ \exists\tau\ :\ \begin{bmatrix}\tau & \omega\end{bmatrix}^\mathsf{T} \in \Lambda\right\}$ and $\omega_{inf} = \inf\left\{\omega\ :\ \exists\tau\ :\ \begin{bmatrix}\tau & \omega\end{bmatrix}^\mathsf{T} \in \Lambda\right\}$.

4.4 Collision avoidance

As seen in Chapter 2, there are two main collision avoidance approaches: Spatio-Temporal Reservation (ST) and Trajectory Planning (TP). The presented approach is included within the TP group.

This section is mostly focused on lateral collision avoidance, as it is less trivial than rear-end collision avoidance. The term *lateral collision* includes head-on, side-impact, and

merging collision scenarios. To derive the lateral collision avoidance algorithm that leads to determining feasible solutions, I initially consider a pair of vehicles; after developing a collision-avoiding strategy for the pair, I will extend the result to any number of vehicles.

Given a pair of vehicles and their abscissas u_1 and u_2, the $\boldsymbol{u} = \begin{bmatrix} u_1 & u_2 \end{bmatrix}^\mathsf{T}$ vector represents a configuration of the two vehicles, i.e., a possible combination of their positions. Let us consider the space of the possible values of $\boldsymbol{u}$, named the *configuration space*. I recall that, by definition, lateral collisions are only possible inside the merging zone, which means that $\boldsymbol{u} \in M_1 \times M_2$; since M_1 and M_2 are intervals, $M_1 \times M_2$ is a rectangle inside the configuration space.

In the first instance, I consider the bounding circle as the overapproximating occupancy for the vehicles and, hence, the collision condition is (4.11). After deriving the main idea behind the lateral collision avoidance strategy, I will consider the other discussed approximations.

4.4.1 Lateral collision detection

Let us consider two vehicles, with trajectories $\boldsymbol{p}_1$ and $\boldsymbol{p}_2$, respectively, and bounding radii ρ_1 and ρ_2, respectively. I introduce the *collision function* c:

$$c(\boldsymbol{u}) = \|\boldsymbol{\varphi}(\boldsymbol{u})\|^2 - \rho^2, \tag{4.29}$$

with[3]

$$\boldsymbol{\varphi}(\boldsymbol{u}) = \frac{\boldsymbol{p}_1(u_1) - \boldsymbol{p}_2(u_2)}{2}, \quad \rho = \frac{\rho_1 + \rho_2}{2}.$$

When $c(\boldsymbol{u}) < 0$, condition (4.11) holds. The open set

$$C = \left\{ \boldsymbol{u} \in \mathbb{R}^2 \ : \ c(\boldsymbol{u}) < 0 \right\} \tag{4.30}$$

is called *collision set* and contains all pairs of abscissas that result in a collision. On the other hand, $\mathbb{R}^2 \setminus C = \{\boldsymbol{u} \in \mathbb{R}^2 \ : \ c(\boldsymbol{u}) \geq 0\}$ contains all pais of abscissas that do not result in a collision.

When $C = \emptyset$ no collisions are possible; hence, let us consider the case when $C \neq \emptyset$. By equation (4.14), the point $\boldsymbol{\gamma}(t) = \begin{bmatrix} \gamma_1(t) & \gamma_2(t) \end{bmatrix}^\mathsf{T}$, for $t \in T_{M,1} \cap T_{M,2}$, describes a curve $\Gamma = \boldsymbol{\gamma}(T_{M,1} \cap T_{M,2})$, with the following parametrization:

$$\begin{cases} u_1 = \gamma_1(t) = \omega_1(t - \tau_1), \\ u_2 = \gamma_2(t) = \omega_2(t - \tau_2), \end{cases} \qquad t \in T_{M,1} \cap T_{M,2}.$$

Γ is called *configuration trajectory* and is a segment of the straight line $\bar{\Gamma}$ defined by the equation

$$\bar{\Gamma} : \quad \omega_1^{-1} u_1 - \omega_2^{-1} u_2 - (\tau_2 - \tau_1) = 0, \tag{4.31}$$

which, when put in explicit form with respect to u_2, becomes

$$u_2 = \frac{\omega_2}{\omega_1} u_1 + \omega_2(\tau_1 - \tau_2),$$

[3]The factor of 2 at the denominator was added to consider the mean of the radii instead of their sum. This also results in cleaner coefficients in subsequent results.

with slope $\omega_2/\omega_1 > 0$ and vertical intercept $\omega_2(\tau_1 - \tau_2)$. The speeds of the vehicles are constrained, namely $\omega_1 \in \left[\omega_1^-, \omega_1^+\right]$ and $\omega_2 \in \left[\omega_2^-, \omega_2^+\right]$. Also, note that $\Gamma \subset M_1 \times M_2$ because $\gamma_1(t) \in M_1$ and $\gamma_2(t) \in M_2$, $\forall t \in T_{M,1} \cap T_{M,2}$.

Proposition 4.1. *The two vehicles are not on a collision course if it holds:*

$$\Gamma \cap C \neq \emptyset. \tag{4.32}$$

Proof. Indeed, if (4.32) is satisfied, we get

$$c(\boldsymbol{\gamma}(t)) \geq 0 \quad \forall t \in T_{M,1} \cap T_{M,2},$$

implying $\tilde{O}_1(\gamma_1(t)) \cap \tilde{O}_2(\gamma_2(t)) = \emptyset$ and (4.3) is averted. □

Figure 4.13 shows the configuration space of two vehicles in a perpendicular intersection ($\alpha = 90°$). Specifically, vehicle 1 crosses the intersection in a straight trajectory from South to North, and vehicle 2 enters the intersection from the East lane and performs a left turn towards South. This figure is a snapshot at a specific time t, when the vehicles are not colliding, since $\boldsymbol{\gamma}(t) \notin C$. However, the vehicles are on a collision course, because Γ crosses C and (4.32) is violated: eventually the vehicles will collide.

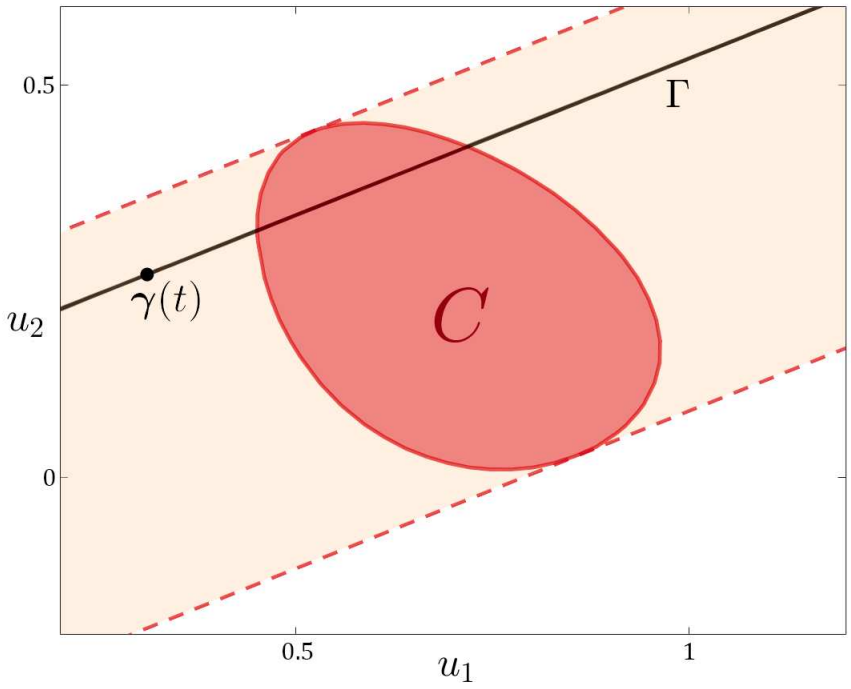


Figure 4.13: *Collision course displayed inside the configuration space.*

My collision detection strategy is based on determining whether Γ intersects C. Similarly, recalling that $\boldsymbol{\lambda}_1$ and $\boldsymbol{\lambda}_2$ determine Γ by (4.31), my collision avoidance strategy consists in adjusting the arrival state values so that they produce a segment Γ that does not intersect C. The key concept behind this approach is based on the fact that a hyperplane (in this case, a line) separates the space it is embedded in (in this case, a plane) into two half-spaces. If C is entirely enclosed within one of the open half-spaces formed by $\bar{\Gamma}$, then Γ does not intersect C, indicating that the vehicles are not on a collision course.

To determine whether $\bar{\Gamma}$ intersects C, the support function is employed. The support function of C is denoted by σ_C and defined by

$$\sigma_C(\boldsymbol{\eta}) = \sup_{\boldsymbol{u} \in C} \langle \boldsymbol{\eta},\, \boldsymbol{u} \rangle \,,$$

for $\boldsymbol{\eta} = \begin{bmatrix} \eta_1 & \eta_2 \end{bmatrix}^\mathsf{T} \in \mathbb{R}^2$. Let $\boldsymbol{\nu} = \begin{bmatrix} \omega_1^{-1} & -\omega_2^{-1} \end{bmatrix}^\mathsf{T}$. Since C is open, (3.2) applies and we obtain

$$\tau_2 - \tau_1 \leq -\sigma_C(-\boldsymbol{\nu}) \;\vee\; \tau_2 - \tau_1 \geq \sigma_C(\boldsymbol{\nu}) \;\Rightarrow\; \bar{\Gamma} \cap C = \emptyset. \tag{4.33}$$

For a given pair (ω_1, ω_2), there are two supporting lines of C with the same slope ω_2/ω_1, given by equations

$$\langle \boldsymbol{\nu},\, \boldsymbol{u} \rangle - \sigma_C(\boldsymbol{\nu}) = 0 \quad \text{and} \quad \langle -\boldsymbol{\nu},\, \boldsymbol{u} \rangle - \sigma_C(-\boldsymbol{\nu}) = 0.$$

The intersection of the open half-spaces that contain C associated with these supporting lines is a strip of the configuration space, called the *critical strip*. As figure 4.13 shows, if Γ is contained within the critical strip, then it intersects C.

It is important to take the boundedness of C into consideration. I recall that $\sigma_C(\boldsymbol{\eta}) < +\infty$, $\forall \boldsymbol{\eta} \in \mathbb{R}^2$ if and only if C is bounded. In this case, for every pair (ω_1, ω_2), there are values of τ_1 and τ_2 that can satisfy both conditions of (4.33). If C is not bounded, there are values of ω_1 and ω_2 such that $\sigma_C(\boldsymbol{\nu}) = +\infty$ and one of the conditions of (4.33), or both, may not hold for any value of τ_1 and τ_2. In other words, there are values of $\boldsymbol{\nu}$ that produce a critical strip that is not bounded; e.g., it degenerates into a half-space.

Another fact worth noticing is that (4.33) does not hold in reverse: if C is not connected, there may exist a line that separates its connected components that do not respect the sufficient condition of (4.33).

Now, I assess another collision model. As discussed in Section 4.2.3, a bounding circle is a poor approximation of a rectangular shape, and it becomes poorer as its width-to-height ratio increases. Envelopes of circles have been shown to better approximate the shape of an elongated vehicle, while still accommodating some safety margins. Condition (4.12) does not blend well with the results obtained so far, because of the presence of the auxiliary variables κ_1 and κ_2 varying in continuous intervals. Condition (4.13) includes the same variables, but, because of its form, can be seamlessly integrated within the proposed approach using the tools developed so far. Let $K = [-\bar{\kappa}_1, \bar{\kappa}_1] \times [-\bar{\kappa}_2, \bar{\kappa}_2]$ and $\boldsymbol{\kappa} = \begin{bmatrix} \kappa_1 & \kappa_2 \end{bmatrix}^\mathsf{T} \in K$. Condition (4.13) is equivalent to $c(\boldsymbol{u} - \boldsymbol{\kappa}) < 0$. With this collision model, the collision set is $C_K = \{\boldsymbol{u} \in \mathbb{R}^2 \;:\; c(\boldsymbol{u} - \boldsymbol{\kappa}) < 0,\, \boldsymbol{\kappa} \in K\}$. The reason why (4.13) can be readily integrated into the presented theory is that C_K can be decomposed as the Minkowski sum of C and K. Indeed, with a simple change of variable $\boldsymbol{z} = \boldsymbol{u} - \boldsymbol{\kappa}$, we obtain

$$C_K = \{\boldsymbol{z} + \boldsymbol{\kappa} \;:\; c(\boldsymbol{z}) < 0,\, \boldsymbol{\kappa} \in K\} = C + K.$$

The set K is the rectangle associated with the vector $\bar{\boldsymbol{\kappa}} = \begin{bmatrix} \bar{\kappa}_1 & \bar{\kappa}_2 \end{bmatrix}^\mathsf{T}$: from (3.4), we immediately obtain the support function of C_K:

$$\sigma_{C_K}(\boldsymbol{\eta}) = \sigma_C(\boldsymbol{\eta}) + \sigma_K(\boldsymbol{\eta}) = \sigma_C(\boldsymbol{\eta}) + \langle \bar{\boldsymbol{\kappa}},\, |\boldsymbol{\eta}| \rangle \,.$$

4.4.2 Lateral collision avoidance

When a pair of vehicles is determined to be on a collision course with their current value of $\boldsymbol{\lambda}_1$ and $\boldsymbol{\lambda}_2$, they should adjust these values to avoid the crash. A simple way to do so is by having one of the two vehicles adapt to the other, based on a predetermined priority; priority-based approach are common in the literature, e.g., [35, 56, 55, 54].

Assumption 4.2. *Each vehicle $i \in \mathcal{V}$ has an associated unique value $\chi_i \in \mathbb{R}$ defined as the priority of vehicle i.*

In a situation of potential collision, the vehicle with the highest priority preserves the value of its decision vector, whereas the other adapts to avoid the collision. I do not propose any particular policy that determines the priorities, but it may be related to the original arrival time and the importance of the vehicle: for example, emergency vehicles should have a higher priority.

A collision avoidance algorithm performs two tasks.

1. Detect if a collision can occur between the vehicles; a sufficient condition to perform this check is given by (4.33).
2. If a collision is detected, modify the arrival state of the vehicle with lower priority so that a collision is averted.

Without loss of generality, I assume vehicle 2 has the lowest priority, i.e., $\chi_2 < \chi_1$.

Since vehicle 2 has the lowest priority, it performs this check by using condition (4.33); this means being able to compute the support function of the collision set. If (4.33) is verified, no action is required. Otherwise, it chooses new value for its arrival state $\boldsymbol{\lambda}_2 \in \Lambda_2$ that verify (4.33). For a given ω_2, τ_2 can be chosen such that

$$\tau_2 \leq \tau_1 - \sigma_C(-\boldsymbol{\nu}) \tag{4.34}$$

or

$$\tau_2 \geq \tau_1 + \sigma_C(\boldsymbol{\nu}). \tag{4.35}$$

While it may not exist a $\boldsymbol{\lambda} \in \Lambda$ that verify (4.34), (4.28) ensures that, for $\omega \in [\omega_{min}, \omega_{max}]$, there always exists a choice of τ that verify (4.35). In other words, a vehicle can always choose to delay its arrival to avoid colliding with another vehicle.

Once the collision avoidance conditions for a pair of vehicles have been assessed, generalizing to more than two vehicles is straightforward.

Theorem 4.1. *Consider $m < n$ different AVs, denoted by indices $1, ..., m$. Then there exists a set of decision vectors $\{\boldsymbol{\lambda}_i\}_{i \in [\![1,m]\!]}$ such that $\forall i, j \in [\![1, m]\!]$, with $i \neq j$, vehicles i and j are not in a collision course.*

Proof. The proof is by induction. Sufficient conditions that proves the case $m = 2$ are (4.34) and (4.35). Without loss of generality, I assume that the vehicles are sorted by priority, i.e., $\chi_i > \chi_{i+1}$, $\forall i \in [\![1, m]\!]$. Suppose that vehicles $1, ..., m-1$ already successfully applied the collision avoidance strategy and vehicle m, last in priority, must adjust its decision variables. The result to prove is that there exists at least one choice of τ_m and ω_m obtained by applying the procedure described in this section and, more specifically, by equations (4.34) and (4.35),

that does not produce a collision with the other vehicles. In our scenario with m vehicles, this must hold for all pairs (i, m), with $i \in [\![1, m-1]\!]$. For an arbitrary value of ω_m, condition (4.34) is satisfied if:

$$\tau_m \leq \min_{i \in [\![1,m-1]\!]} \tau_i - \sigma_C \left(- \begin{bmatrix} \omega_i^{-1} \\ \omega_m^{-1} \end{bmatrix} \right). \tag{4.36}$$

Similarly, condition (4.35) is satisfied if:

$$\tau_m \geq \max_{i \in [\![1,m-1]\!]} \tau_i + \sigma_C \left(\begin{bmatrix} \omega_i^{-1} \\ \omega_m^{-1} \end{bmatrix} \right). \tag{4.37}$$

Like the case with $m = 2$, (4.36) may not produce a feasible value of $\boldsymbol{\lambda}$, but (4.37) always does.

□

4.5 Computation of the support function

The results obtained in Section 4.4.1 and Section 4.4.2 assume each vehicle can compute the support function of the collision set associated with any other vehicle. I underscore that computing the support function is crucial as it enables collision detection by (4.33) and collision avoidance by (4.36) and (4.37). In this section, various strategies for calculating these functions are discussed.

By its definition, the value of $\sigma_C(\boldsymbol{\eta})$ is the result of an optimization problem:

$$\begin{aligned} &\text{maximize} && \langle \boldsymbol{\eta}, \boldsymbol{u} \rangle \\ &\text{subject to} && \boldsymbol{u} \in C. \end{aligned} \tag{4.38}$$

The function $\boldsymbol{u} \to \langle \boldsymbol{\eta}, \boldsymbol{u} \rangle$ is linear in $\boldsymbol{u}$, thus it is convex. The same cannot be said for the set C, which, in general, is not convex. Non-convex optimization problems are hard to solve; nevertheless, the low dimensionality and the limitedness of the search region are an advantage. To solve this problem, I adopt some heuristic ideas that leverage the structure of the problem and the geometry of the setup from which it stems.

4.5.1 Characterization of the collision set

For a given value of α, there is a finite number of possible pairs of trajectories. Not every pair of trajectories can produce a collision: for example, two vehicles performing a right turn will never risk a lateral collision. Table 4.1 shows which pairs of trajectories may produce a collision, called *conflicting trajectories*; the cells representing pairs trajectories with a common inbound or outbound lane are highlighted in gray. The number of trajectories is finite, and so it is the number of conflicting pairs. In particular, straight paths conflict with 9 trajectories, right turns with 6, and left turns with 11, for a total of 26 conflicting pairs per lane.

Depending on α, some of the pairs of conflicting trajectories that do not share the inbound and the outbound lane intersect in either one or two points, or they do not intersect at all. The number of points of intersection - in which c attains a global minimum - and the value of ρ are closely related to the structure of C. By inspection of the 26 conflicting pairs, the

	SE	SN	SW	EN	EW	ES	NW	NS	NE	WS	WE	WN
SE	X	X	X						X		X	X
SN	X	X	X	X	X	X			X		X	X
SW	X	X	X	X	X	X	X	X	X		X	X
EN		X	X	X	X	X						X
EW		X	X	X	X	X	X	X	X			X
ES		X	X	X	X	X	X	X	X	X	X	X
NW			X		X	X	X	X	X			
NS			X		X	X	X	X	X	X	X	X
NE	X	X	X		X	X	X	X	X	X	X	X
WS						X		X	X	X	X	X
WE	X	X	X				X		X	X	X	X
WN	X	X	X	X	X	X		X	X	X	X	X

Table 4.1: *Pairs of trajectories that can produce a collision.*

shapes C can assume can be divided into four categories, which are represented in Figure 4.14:

(a) bounded, connected, and convex, which is the structure most pairs of trajectories produce;

(b) unbounded, obtained in correspondence with merge conflicts;

(c) concave, and

(d) not connected, produced by pairs of specular left turns.

Figure 4.14 also depicts some level curves of c and its global minima. Note that the minimum value that c may assume is $-\rho^2$, obtained when $\boldsymbol{p}_1(u_1) = \boldsymbol{p}_2(u_2)$, i.e., in correspondence with the intersection points of the trajectories.

If the trajectories do not intersect, then its global minimum is greater than $-\rho^2$. The minimizers are characterized as follows:

(a) a single isolated minimizer;

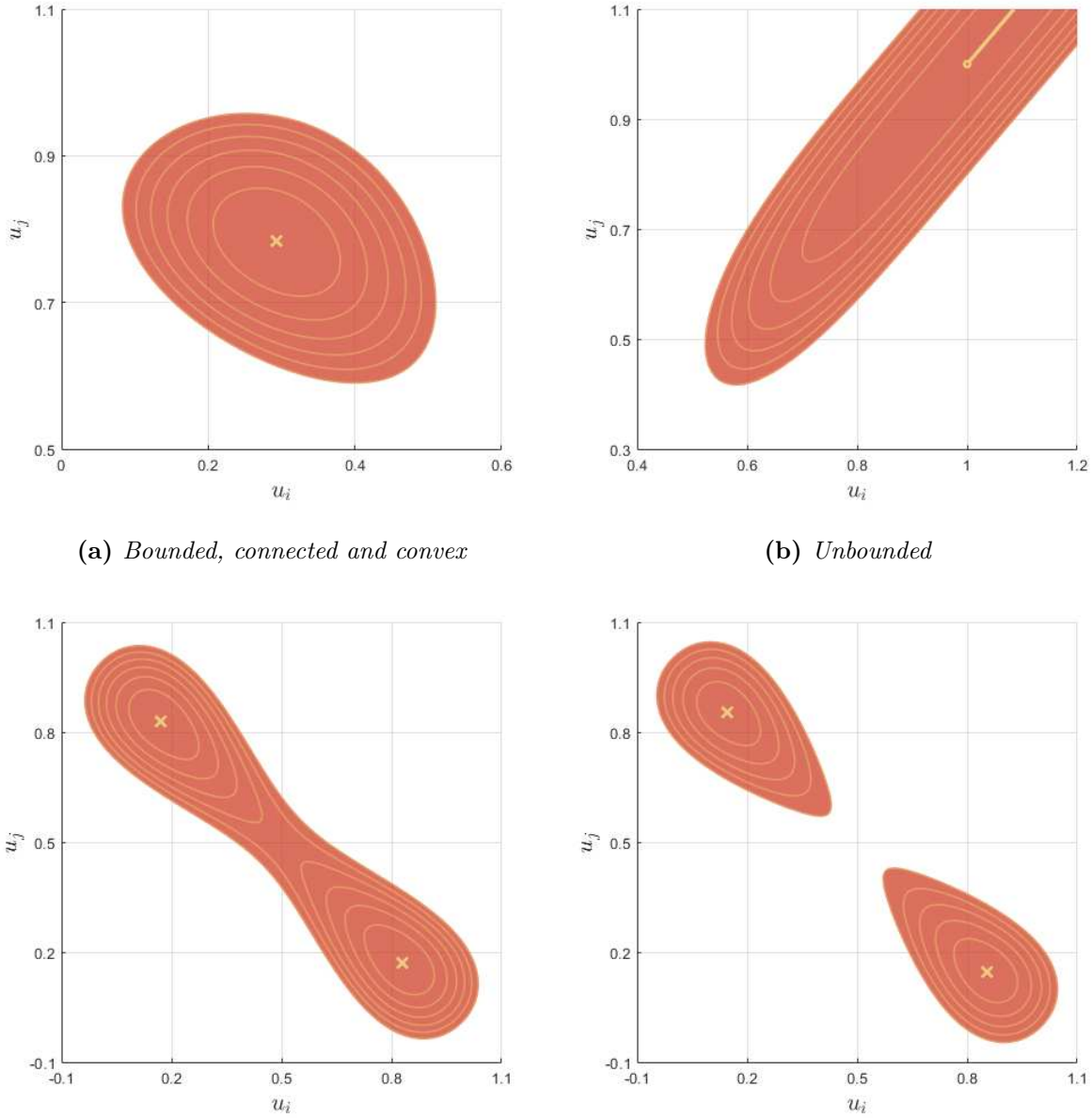
(b) a continuum of minimizers;

(c) either 1 or 2 isolated minimizers;

(d) 2 isolated minimizers.

The number of minimizers in case (c) depends on whether $\boldsymbol{p}_1$ and $\boldsymbol{p}_2$ intercept in two points or do not intercept at all, based on α and the radius of curvature.

The structure of C must be very carefully considered before attempting to compute its support function $\sigma_C(\boldsymbol{\eta})$ by means of (4.38), for a given $\boldsymbol{\eta}$. First of all, I want to remark that C is the constraint set of (4.38), and a constraint set that is not convex - and sometimes not even connected - means the optimization problem is hard to solve. Furthermore, when C is unbounded, $\text{conv}(C)$ has at least one direction of recession, and the effective domain of σ_C



(a) *Bounded, connected and convex*

(b) *Unbounded*

(c) *Non-convex*

(d) *Not connected*

Figure 4.14: *Possible shapes of the collision set.*

is not $\mathbb{R}^2$, meaning $+\infty$ can be a solution of (4.38), for some values of $\boldsymbol{\eta}$; however, this only happens when the trajectories share an inbound or outbound lane. Before quantitatively describing the domain of σ_C with a proposition, I recall that $\boldsymbol{p}_1'(u_1) \equiv w_1$ and $\boldsymbol{p}_2'(u_2) \equiv w_2$.

Proposition 4.2. *Consider two trajectories $\boldsymbol{p}_1, \boldsymbol{p}_2 \in P_\alpha$ that share the outbound direction only, i.e., $\boldsymbol{p}_1 \downarrow \boldsymbol{p}_2$ and $\boldsymbol{p}_1 \not\uparrow \boldsymbol{p}_2$. Let $\boldsymbol{w}' = \begin{bmatrix} w_2 & w_1 \end{bmatrix}^\mathsf{T}$. Then*

$$\operatorname{dom}(\sigma_C) = H_{\leq}(\boldsymbol{w}', 0) = \left\{ \boldsymbol{\eta} \in \mathbb{R}^2 \ : \ \langle \boldsymbol{\eta}, \boldsymbol{w}' \rangle \leq 0 \right\}.$$

When, instead, $\boldsymbol{p}_1 \uparrow \boldsymbol{p}_2$ and $\boldsymbol{p}_1 \not\downarrow \boldsymbol{p}_2$, then

$$\operatorname{dom}(\sigma_C) = H_{\geq}(\boldsymbol{w}', 0) = \left\{ \boldsymbol{\eta} \in \mathbb{R}^2 \ : \ \langle \boldsymbol{\eta}, \boldsymbol{w}' \rangle \geq 0 \right\}.$$

Proof. The proof is in Appendix A. □

Corollary 4.1. *If two trajectories are equal, the domain of σ_C is given by*

$$\operatorname{dom}(\sigma_C) = H(\boldsymbol{w}', 0) = \left\{ \boldsymbol{\eta} \in \mathbb{R}^2 \ : \ \langle \boldsymbol{\eta}, \boldsymbol{w}' \rangle = 0 \right\}.$$

Proof. Follows from Proposition 4.2 by noting that $\boldsymbol{p}_1 = \boldsymbol{p}_2 \iff \boldsymbol{p}_1 \uparrow \boldsymbol{p}_2 \wedge \boldsymbol{p}_1 \downarrow \boldsymbol{p}_2$, hence $\operatorname{dom}(\sigma_C) = H_{\leq}(\boldsymbol{w}', 0) \cap H_{\geq}(\boldsymbol{w}', 0) = H(\boldsymbol{w}', 0)$ □

Remark 4.1. *As a consequence,* $\operatorname{dom}(\sigma_C)$ *is always convex.*

4.5.2 Computation algorithm

With a clearer characterization of C and $\operatorname{dom}(\sigma_C)$, it is now possible to describe a possible algorithm for computing $\sigma_C(\boldsymbol{\eta})$ for a given $\boldsymbol{\eta} \in \operatorname{dom}(\sigma_C)$.

For a pair of straight trajectories, the $\sigma_C(\boldsymbol{\eta})$ can be computed in closed form. To see this, let us first write the expression of the straight paths $\boldsymbol{p}_1$ and $\boldsymbol{p}_2$ in the following form:

$$\boldsymbol{p}_1(u_1) = \bar{\boldsymbol{p}}_1 + \boldsymbol{v}_1 u_1, \quad \boldsymbol{p}_2(u_2) = \bar{\boldsymbol{p}}_2 + \boldsymbol{v}_2 u_2.$$

Then, $\boldsymbol{\varphi}(\boldsymbol{u})$ is

$$\boldsymbol{\varphi}(\boldsymbol{u}) = \frac{\bar{\boldsymbol{p}}_1 + \boldsymbol{v}_1 u_1 - \bar{\boldsymbol{p}}_2 - \boldsymbol{v}_2 u_2}{2} = \frac{\bar{\boldsymbol{p}}_1 - \bar{\boldsymbol{p}}_2 + \boldsymbol{V}\boldsymbol{u}}{2},$$

with $\boldsymbol{V} = \begin{bmatrix} \boldsymbol{v}_1 & -\boldsymbol{v}_2 \end{bmatrix}$. In this case, $\boldsymbol{\varphi}(\boldsymbol{u})$ is an affine transformation and the resulting collision set is an ellipse:

$$C = \left\{ \boldsymbol{u} \in \mathbb{R}^2 \ : \ \left\| \frac{\bar{\boldsymbol{p}}_1 - \bar{\boldsymbol{p}}_2 + \boldsymbol{V}\boldsymbol{u}}{2} \right\|^2 - \rho^2 < 0 \right\}.$$

In this situation, computation of the collision function is straightforward. Let us make a simple change of variable $\boldsymbol{z} = \rho^{-1}\boldsymbol{\varphi}(\boldsymbol{u})$. The inverse transformation is also an affine transformation, and it is given by $\boldsymbol{u} = \boldsymbol{V}^{-1}(2\rho\boldsymbol{z} + \bar{\boldsymbol{p}}_2 - \bar{\boldsymbol{p}}_1)$, hence C can be written as

$$C = \left\{ \boldsymbol{V}^{-1}(2\rho\boldsymbol{z} + \bar{\boldsymbol{p}}_2 - \bar{\boldsymbol{p}}_1) \ : \ \|\boldsymbol{z}\| < 1 \right\} = 2\rho\boldsymbol{V}^{-1}B - \boldsymbol{V}^{-1}(\bar{\boldsymbol{p}}_1 - \bar{\boldsymbol{p}}_2),$$

where B is the unit circle. By using (3.3) and (3.5) we easily obtain σ_C in the case of straight trajectories:

$$\sigma_C(\boldsymbol{\eta}) = 2\rho \left\| \boldsymbol{V}^{-\mathsf{T}} \boldsymbol{\eta} \right\| - \left\langle \boldsymbol{V}^{-\mathsf{T}} \boldsymbol{\eta},\, \bar{\boldsymbol{p}}_1 - \bar{\boldsymbol{p}}_2 \right\rangle .$$

This strategy is not effective with other kinds of conflicting trajectory pairs. To solve (4.38) in general, I propose to use a barrier method. Since the function c defines C by (4.30), problem (4.38) is transformed into the following unconstrained optimization problem:

$$\min_{\boldsymbol{u} \in \mathbb{R}^2} \quad - \langle \boldsymbol{\eta},\, \boldsymbol{u} \rangle - \mu \ln \left(-c(\boldsymbol{u}) \right).$$

Interior-point methods are guaranteed to converge to the global extremizer of the optimization problem they are applied to if the constraint set is convex. Here, C is, in general, not convex. However, by inspecting all conflicting pairs and their collision set, I lay out two conjectures that can be used to guarantee that an interior-point method is suitable for solving (4.38). Let $Q_{\mathrm{II,IV}} = \{\boldsymbol{\eta} \,:\, \eta_1 \eta_2 < 0\}$ denote the union of the second and fourth quadrant of the $\mathbb{R}^2$ plane. We are interested in computing σ_C in $\mathrm{dom}(\sigma_C) \cap Q_{\mathrm{II,IV}}$, because the main objective is verifying condition (4.33). Since $\omega_1 > 0$ and $\omega_2 > 0$, then $\boldsymbol{\nu} \in Q_{\mathrm{II,IV}}$.

Conjecture 4.1. *Every local maximum of (4.38) is strict,* $\forall \boldsymbol{\eta} \in \mathrm{int}(\mathrm{dom}(\sigma_C)) \cap Q_{\mathrm{II,IV}}$.

This conjecture also implies the existence of $\nabla \sigma_C(\boldsymbol{\eta})$ on $\mathrm{int}(\mathrm{dom}(\sigma_C)) \cap Q_{\mathrm{II,IV}}$.

Conjecture 4.2. *Let* $\boldsymbol{\eta} \in \mathrm{dom}(\sigma_C) \cap Q_{\mathrm{II,IV}}$ *and* $\boldsymbol{\zeta} \in \mathbb{R}^2$ *a vector perpendicular to* $\boldsymbol{\eta}$*. Then,* C *is* $\boldsymbol{\zeta}$*-convex.*

Based on these two conjectures, I give the following result.

Proposition 4.3. *Provided that* C *is path-connected, for all* $\boldsymbol{\eta} \in \mathrm{int}(\mathrm{dom}(\sigma_C)) \cap Q_{\mathrm{II,IV}}$*, every local maximizer of (4.38) is also a global maximizer.*

Proof. The proof is in Appendix A. □

Given this result, although not proven, it is to be expected that if an interior-point algorithm converges, the solution is the value of the support function of the connected component that contains the starting point $\boldsymbol{u}^{(0)}$. The choice of $\boldsymbol{u}^{(0)}$ is crucial, as it must belong to the right connected component of C, if there is more than one. Suppose that such a suitable starting point is provided. Although interior-point methods are efficient algorithms, solving problem (4.38) yields the value of the support function only for the given value of $\boldsymbol{\eta}$. A naïve approach would be to run the optimization algorithm each time the support function is needed for a given value of $\boldsymbol{\eta}$; however, this is not practical, as it would need too much time, whereas vehicles need to compute it as fast as possible. To speed up the computation of $\sigma_C(\boldsymbol{\eta})$, two approximation methods are proposed. Both methods leverage the knowledge about support functions and their gradients and are based on some kind of interpolation between known values of σ_C. Furthermore, since support functions are positively homogeneous, the algorithm only needs to compute $\sigma_C(\boldsymbol{\eta})$ for $\|\boldsymbol{\eta}\| = 1$, i.e., $\boldsymbol{\eta} = \begin{bmatrix} \cos\theta & \sin\theta \end{bmatrix}^{\mathsf{T}}$, with $\theta \in [\pi/2, 0] \cup [\pi/2, \pi]$, so that $\boldsymbol{\eta} \in Q_{\mathrm{II,IV}}$.

In particular, suppose that $\sigma_C(\boldsymbol{\eta}_a)$, $\sigma_C(\boldsymbol{\eta}_b)$, $\boldsymbol{a} = \nabla \sigma_C(\boldsymbol{\eta}_a)$, and $\boldsymbol{b} = \nabla \sigma_C(\boldsymbol{\eta}_b)$ were computed, with $\boldsymbol{\eta}_a = \begin{bmatrix} \cos\theta_a & \sin\theta_a \end{bmatrix}^{\mathsf{T}}$ and $\boldsymbol{\eta}_b = \begin{bmatrix} \cos\theta_b & \sin\theta_b \end{bmatrix}^{\mathsf{T}}$, $\theta_a, \theta_b \in [\pi/2, 0] \cup [\pi/2, \pi]$, and we need to compute $\sigma_C(\boldsymbol{\eta})$, with $\theta_a < \theta < \theta_b$. $\boldsymbol{\eta}_a$ and $\boldsymbol{\eta}_b$ are the *endpoints* of the approximation. Then, given a tolerance $\epsilon > 0$, the approximation $\tilde{\sigma}_C$ must be good enough, that is, $|\tilde{\sigma}_C - \sigma(\boldsymbol{\eta})| < \epsilon$. If this is not the case, the approximation can be still used to find a

starting point that is close to $\arg\max_{\boldsymbol{u}\in C}\langle\boldsymbol{\eta}, \boldsymbol{u}\rangle$, in order to reduce the execution time of the interior-point method. The values computed by solving (4.38) are stored so they can be used as endpoints: this way, the $\sigma_C(\boldsymbol{\eta})$ is sampled and eventually the sampling becomes fine enough that the values obtained by approximation are within the given tolerance.

The computation algorithm separately considers the second and fourth quadrant of the $\boldsymbol{\eta}$-plane, hence it starts by computing the endpoints corresponding to the extreme values of $\mathrm{dom}(\sigma_C)$ with respect to the unit circles. This determines two separate intervals and four initial endpoints. These endpoints are computed exactly by using an interior-point method where the starting point is evaluated considering the different shapes of C. In particular, when C is bounded, the two initial points are computed by gradient descent starting from $\begin{bmatrix}1 & 0\end{bmatrix}^\mathsf{T}$ and $\begin{bmatrix}0 & 1\end{bmatrix}^\mathsf{T}$, as shown in Figure 4.15; when C is non-convex, it results into two separate starting points, corresponding to the minimizers of c; conversely, when C is convex, the starting points coincide. When the trajectories share the inbound lane, the starting point is determined by gradient descent starting from the point $\begin{bmatrix}0 & 0\end{bmatrix}^\mathsf{T}$; in the case of shared outbound lane, the gradient descent starts from $\begin{bmatrix}1 & 1\end{bmatrix}^\mathsf{T}$.

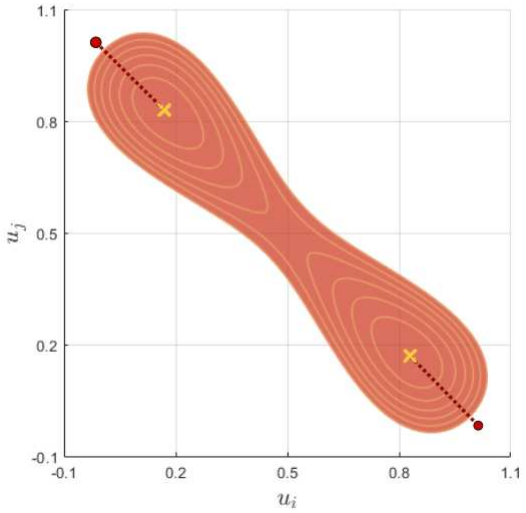


Figure 4.15: *Gradient descent for computation of the starting points.*

4.5.2.1 Approximation by bounding

The first proposed method leverages the convexity of σ_C. Since $\boldsymbol{\eta}_a$, $\boldsymbol{\eta}_b$, and $\boldsymbol{\eta}$ are not collinear, $\boldsymbol{\eta}$ cannot be expressed as $\lambda\boldsymbol{\eta}_a + (1-\lambda)\boldsymbol{\eta}_b$, for some $\lambda \in [0, 1]$. Consider the secant vectors $\boldsymbol{\eta}'_a$ and $\boldsymbol{\eta}'_b$ that are obtained by extending the directions of $\boldsymbol{\eta}_a$ and $\boldsymbol{\eta}_b$ on the line that is tangent to the unit circle at $\boldsymbol{\eta}$, as shown in Figure 4.16.

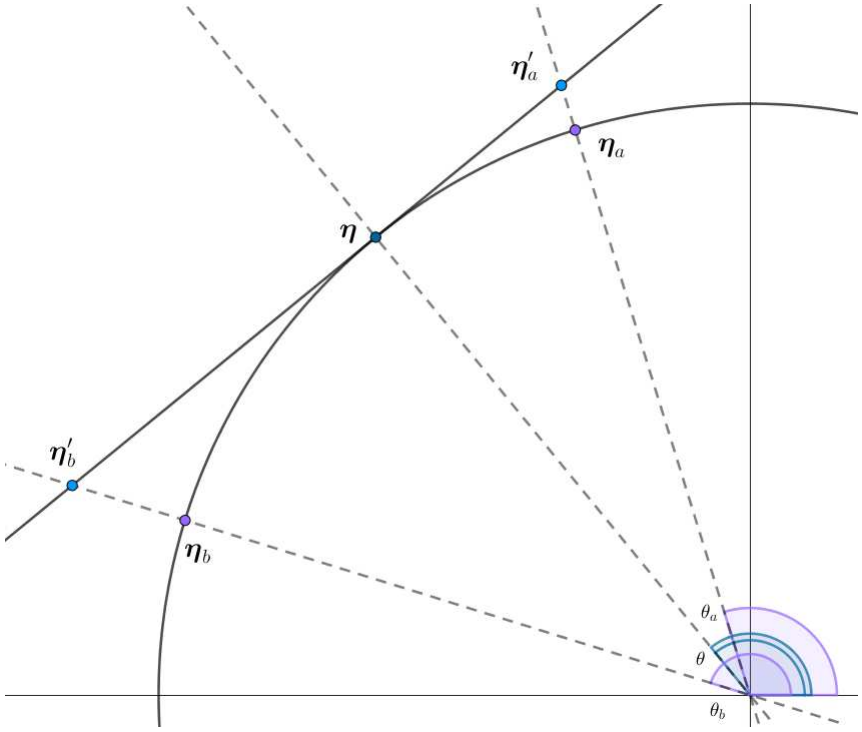


Figure 4.16: *Construction of* $\boldsymbol{\eta}'_a$ *and* $\boldsymbol{\eta}'_b$.

The expressions of $\boldsymbol{\eta}'_a$ and $\boldsymbol{\eta}'_b$ are given by

$$\boldsymbol{\eta}'_a = \frac{1}{\langle \boldsymbol{\eta},\, \boldsymbol{\eta}_a \rangle} \boldsymbol{\eta}_a, \quad \boldsymbol{\eta}'_b = \frac{1}{\langle \boldsymbol{\eta},\, \boldsymbol{\eta}_b \rangle} \boldsymbol{\eta}_b.$$

Note that both $\langle \boldsymbol{\eta},\, \boldsymbol{\eta}_a \rangle > 0$ and $\langle \boldsymbol{\eta},\, \boldsymbol{\eta}_b \rangle > 0$ because we are operating inside one quadrant of the plane. Because of the positive homogeneity of σ_C, it holds

$$\sigma_C(\boldsymbol{\eta}'_a) = \frac{1}{\langle \boldsymbol{\eta},\, \boldsymbol{\eta}_a \rangle} \sigma_C(\boldsymbol{\eta}_a), \quad \sigma_C(\boldsymbol{\eta}'_b) = \frac{1}{\langle \boldsymbol{\eta},\, \boldsymbol{\eta}_b \rangle} \sigma_C(\boldsymbol{\eta}_b).$$

$\boldsymbol{\eta}'_a$, $\boldsymbol{\eta}'_b$, and $\boldsymbol{\eta}$ are collinear, therefore we can write

$$\boldsymbol{\eta} = \lambda \boldsymbol{\eta}'_a + (1 - \lambda) \boldsymbol{\eta}'_b,$$

with

$$\lambda = \frac{\|\boldsymbol{\eta}'_b - \boldsymbol{\eta}\|}{\|\boldsymbol{\eta}'_b - \boldsymbol{\eta}'_a\|}.$$

With this setup, the goal is to find an upper bound $\tilde{\sigma}_C^{(up)}$ and a lower bound $\tilde{\sigma}_C^{(low)}$ of $\sigma_C(\boldsymbol{\eta})$. Because of the convexity of σ_C and $\mathrm{dom}(\sigma_C)$, $\forall \boldsymbol{\eta}_a, \boldsymbol{\eta}_b \in \mathrm{dom}(\sigma_C)$ it holds

$$\sigma_C(\boldsymbol{\eta}) = \sigma_C\left(\lambda \boldsymbol{\eta}'_a + (1 - \lambda) \boldsymbol{\eta}'_b\right) \leq \lambda \sigma_C(\boldsymbol{\eta}'_a) + (1 - \lambda) \sigma_C(\boldsymbol{\eta}'_b) = \tilde{\sigma}_C^{(up)} \tag{4.39}$$

and

$$\begin{cases} \sigma_C(\boldsymbol{\eta}) \geq \sigma_C(\boldsymbol{\eta}'_a) + \langle \nabla \sigma_C(\boldsymbol{\eta}'_a),\, \boldsymbol{\eta} - \boldsymbol{\eta}'_a \rangle\,, & \text{(4.40a)} \\ \sigma_C(\boldsymbol{\eta}) \geq \sigma_C(\boldsymbol{\eta}'_b) + \langle \nabla \sigma_C(\boldsymbol{\eta}'_b),\, \boldsymbol{\eta} - \boldsymbol{\eta}'_b \rangle\,. & \text{(4.40b)} \end{cases}$$

Let us use λ to create a convex combination of these two inequalities. By multiplying (4.40a) by λ and (4.40b) by $1-\lambda$ and by adding the results, we obtain

$$\sigma_C(\boldsymbol{\eta}) \geq \lambda\sigma_C(\boldsymbol{\eta}'_a) + (1-\lambda)\sigma_C(\boldsymbol{\eta}'_b) + \lambda\left\langle \nabla\sigma_C(\boldsymbol{\eta}'_a) - \nabla\sigma_C(\boldsymbol{\eta}'_b),\, \boldsymbol{\eta} - \boldsymbol{\eta}'_a \right\rangle = \tilde{\sigma}_C^{(low)} \tag{4.41}$$

Then, the approximation is given by averaging the upper bound in (4.39) and the lower bound in (4.41):

$$\tilde{\sigma}_C = \frac{\tilde{\sigma}_C^{(up)} + \tilde{\sigma}_C^{(low)}}{2} = \lambda\sigma_C(\boldsymbol{\eta}'_a) + (1-\lambda)\sigma_C(\boldsymbol{\eta}'_b) + \frac{\lambda}{2}\left\langle \boldsymbol{a}-\boldsymbol{b},\, \boldsymbol{\eta} - \boldsymbol{\eta}'_a \right\rangle. \tag{4.42}$$

The maximum error produced by this approximation is

$$\sup_{\boldsymbol{\eta}} |\tilde{\sigma}_C - \sigma(\boldsymbol{\eta})| = \frac{\tilde{\sigma}_C^{(up)} - \tilde{\sigma}_C^{(low)}}{2},$$

therefore, if this value is smaller than the given tolerance ϵ, $\tilde{\sigma}_C$ can be taken as an approximation of $\sigma(\boldsymbol{\eta})$.

Conversely, if the obtained approximation is not good enough, $\boldsymbol{a}$ and $\boldsymbol{b}$ are used to heuristically construct a new starting point for the interior-point method, given by

$$\boldsymbol{u}^{(0)} = \lambda\boldsymbol{a} + (1-\lambda)\boldsymbol{b}.$$

4.5.2.2 Approximation by Bézier curves

The boundary ∂C is a planar curve and it is known that $\boldsymbol{a}, \boldsymbol{b} \in \partial C$. The idea behind this method is to use a quadratic Bézier curve $\boldsymbol{q}(\lambda)$ to approximate ∂C: then, its support function can be computed in closed form by (3.6).

The quadratic Bézier curve is constructed so that its endpoints correspond to $\boldsymbol{a}$ and $\boldsymbol{b}$ and its tangent vector is also tangent to ∂C in both $\boldsymbol{a}$ and $\boldsymbol{b}$. To do so, I take the intersection of the tangent lines to ∂C in $\boldsymbol{a}$ and $\boldsymbol{b}$ to be the control handle of the Bézier curve, as Figure 4.17 shows.

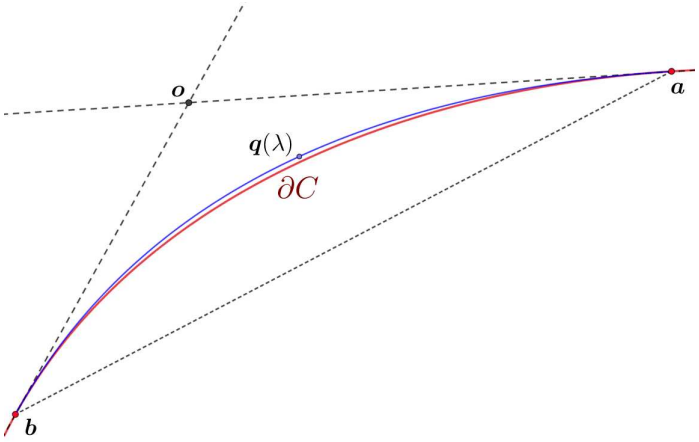


Figure 4.17: *Approximation of ∂C via a quadratic Bézier curve.*

Therefore, the point $\boldsymbol{o}$ is given by

$$\begin{cases} \langle \boldsymbol{\eta}_a, \boldsymbol{o} \rangle - \sigma_C(\boldsymbol{\eta}_a) = 0, \\ \langle \boldsymbol{\eta}_b, \boldsymbol{o} \rangle - \sigma_C(\boldsymbol{\eta}_b) = 0, \end{cases}$$

hence, since $\boldsymbol{\eta}_a$ and $\boldsymbol{\eta}_b$ are not parallel, we have

$$\boldsymbol{o} = \begin{bmatrix} \boldsymbol{\eta}_a & \boldsymbol{\eta}_b \end{bmatrix}^{-1} \begin{bmatrix} \sigma_C(\boldsymbol{\eta}_a) \\ \sigma_C(\boldsymbol{\eta}_b) \end{bmatrix}.$$

Since $\boldsymbol{\eta}_a \neq \boldsymbol{\eta}_b$, $\boldsymbol{a}$, $\boldsymbol{b}$, and $\boldsymbol{o}$ are not collinear, thus they can be used to construct a Bézier curve and to compute the approximation of $\sigma_C(\boldsymbol{\eta})$ by (3.6):

$$\tilde{\sigma}_C = \frac{\langle \boldsymbol{\eta}, \boldsymbol{a} \rangle \langle \boldsymbol{\eta}, \boldsymbol{b} \rangle - \langle \boldsymbol{\eta}, \boldsymbol{o} \rangle^2}{\langle \boldsymbol{\eta}, \boldsymbol{a} - 2\boldsymbol{o} + \boldsymbol{b} \rangle}, \tag{4.43}$$

and the tangency point $\boldsymbol{q}(\lambda^*)$ is obtained for $\lambda = \lambda^*$ given by (3.7). Although not proven, it was experimentally determined that (4.43) produces a slightly better approximation than (4.42). Ultimately, when the approximation is not good enough, the new starting point is given by

$$\boldsymbol{u}^{(0)} = \lambda^* \boldsymbol{a} + (1 - \lambda^*) \boldsymbol{b}.$$

4.6 Distributed scheduling

Given the results of Section 4.3, I propose a version of (4.15) which uses the energy associated with the control input as the cost functional. Considering a generic vehicle i, its initial arrival state $\boldsymbol{\lambda}_{i,0}$, and its agreed arrival state $\boldsymbol{\lambda}_i$, there may be multiple kinematic laws that can be used to transfer the arrival state value from $\boldsymbol{\lambda}_{i,0}$ to $\boldsymbol{\lambda}_i$. I assume that the vehicle chooses the kinematic law with the least associated cost. Given this, I consider the cost to depend on $\boldsymbol{\lambda}_i$ only, and not on γ_i. The cost of vehicle i as a function of $\boldsymbol{\lambda}_i$ is:

$$f_i(\boldsymbol{\lambda}_i) = \min_{\gamma_i \in \Gamma_i} F_i \left[\gamma_i(\boldsymbol{\lambda}_i, \cdot) \right],$$

where γ_i was expressed as a function of both $\boldsymbol{\lambda}_i$ and t. For example, let us assume that the energy associated to the acceleration is the chosen cost functional. The expression of each kinematic law γ_i depends on $\boldsymbol{\lambda}_i$ within the control zone and so does the energy $f_i(\boldsymbol{\lambda}_i)$, as can be seen in (4.22). The agent chooses γ_i so that the resulting $f(\boldsymbol{\lambda}_i)$ is minimal with that value of $\boldsymbol{\lambda}_i$. Furthermore, I use (4.11) as the collision condition for simplicity, but the results apply to any overapproximation of the occupancy functions.

Now, we can rewrite the functional optimization problem (4.15) as a normal optimization problem where $\boldsymbol{\lambda}_i$ is the decision variable:

$$\begin{aligned} \text{minimize} \quad & f_i(\boldsymbol{\lambda}_i) \\ \text{subject to} \quad & \left\| \boldsymbol{p}_i(\gamma_i(t)) - \boldsymbol{p}_j(\gamma_j(t)) \right\| < \rho_i + \rho_j \quad \forall t \in \mathbb{R},\ i \neq j \\ \text{and} \quad & \boldsymbol{\lambda}_i \in \Lambda_i. \end{aligned} \tag{4.44}$$

Also, I present the full distributed algorithm for problem (4.44), which does not need a central infrastructure agent.

The algorithm is based on a flooding strategy, where at each iteration k, each AV shares data about itself and about other vehicles it knows of with its neighbors. It then uses the received information to carry out collision avoidance, while minimizing its own cost function. The value of $\boldsymbol{\lambda}_i$ at iteration k is denoted with $\boldsymbol{\lambda}_i^{(k)}$, whereas the collision set and the configuration trajectory of vehicles i and j are denoted by $C_{i,j}$ and $\Gamma_{i,j}$, respectively.

For the generic vehicle i, the algorithm is described by the following sequence of steps.

Step 0: The vehicle starts with the initial value $\boldsymbol{\lambda}_i^{(0)}$.

Step 1: The vehicle broadcasts $\boldsymbol{\lambda}_i^{(k)}$, along with $\boldsymbol{p}_i$, ρ_i and χ_i to all connected vehicles of $\mathcal{N}_i$. This means that it receives $\boldsymbol{\lambda}_j^{(k)}$ from all $j \in \mathcal{D}_i(1)$, as well as $\boldsymbol{\lambda}_j^{(k-1)}$, $\forall j \in \mathcal{D}_i(2)$ and so on. In general, vehicle i is provided with $\boldsymbol{\lambda}_j^{(k-d+1)}$ for all $j \in \mathcal{D}_i(d)$, for $d = 1, ..., k$. The set of vehicles known by vehicle i at iteration k is denoted with $\mathcal{K}_i^{(k)} = \{j \in \mathcal{V} \,|\, \delta(i,j) = 1, ..., k\}$. For each $j \in \mathcal{K}_i^{(k)}$, vehicle i estimates $\boldsymbol{\lambda}_j$ to be $\tilde{\boldsymbol{\lambda}}_j^{(k)} = \boldsymbol{\lambda}_j^{(k-d+1)}$; the components of $\tilde{\boldsymbol{\lambda}}_j^{(k)}$ are $\tilde{\tau}_j^{(k)}$ and $\tilde{\omega}_j^{(k)}$.

Step 2: Firstly, a feasible value of $\boldsymbol{\lambda}_i$ is sought. A value of $\boldsymbol{\lambda}_i$ is considered feasible if does not produce collisions with any known vehicle with higher priority, i.e., $\Gamma_{i,j} \cap C_{i,j} = \emptyset$, $\forall j \in \mathcal{K}_i^{(k)}$, $\chi_j > \chi_i$, and $\boldsymbol{\lambda}_i \in \Lambda_i$. The output of this step is denoted by $\bar{\boldsymbol{\lambda}}_i^{(k+1)}$. If $\boldsymbol{\lambda}_i^{(k)}$ is feasible, then $\bar{\boldsymbol{\lambda}}_i^{(k+1)} = \boldsymbol{\lambda}_i^{(k)}$; otherwise, the output of this step is computed by using the results exposed in Section 4.4. Specifically, for a set of possible values of $\omega_i \in \left[\omega_i^-, \omega_i^+\right]$, τ_i is chosen so that $\Gamma_{i,j}$ supports $C_{i,j}$, as follows:

$$\begin{aligned} \bar{\boldsymbol{\lambda}}_i^{(k+1)} &= \arg\min_{\boldsymbol{\lambda}_i} f_i(\boldsymbol{\lambda}_i) \\ \text{s.t.} \quad & \tau_i = \tau_j - \sigma_C(-\boldsymbol{\nu}_{j,i}) \ \vee \ \tau_i = \tau_j + \sigma_C(\boldsymbol{\nu}_{j,i}) \\ \text{and} \quad & \bar{\boldsymbol{\lambda}}_i^{(k+1)} \in \Lambda_i \wedge \tilde{\Gamma}_{i,j}^{(k)} \cap C_{i,j} = \emptyset, \quad \forall j \in \mathcal{K}_i^{(k)}, \end{aligned}$$

where $\boldsymbol{\nu}_{j,i} = \begin{bmatrix} \omega_j^{-1} & -\omega_i^{-1} \end{bmatrix}^\mathsf{T}$ and $\tilde{\Gamma}_{i,j}^{(k)}$ is $\Gamma_{i,j}$ obtained with the estimate $\tilde{\boldsymbol{\lambda}}_j^{(k)}$. Theorem 4.1 guarantees that there exists at least one solution to this problem; if multiple solutions exist, the vehicle indifferently chooses one of them. The pseudocode that implements this step is described in Algorithm 3.

Step 3: Once the feasible value $\bar{\boldsymbol{\lambda}}_i^{(k+1)}$ is determined by **Step 2**, the vehicle can safely proceed to further minimize its cost function $f_i(\boldsymbol{\lambda}_i)$; gradient descent can be employed to perform this step. Given a predetermined sufficiently small number $\epsilon > 0$, the descent function is defined as $\text{desc}(\boldsymbol{\lambda}) = \boldsymbol{\lambda} - \epsilon \nabla f_i(\boldsymbol{\lambda})$ and I use a subscript to denote repeated application, e.g., $\text{desc}_2(\boldsymbol{\lambda}) = \text{desc}(\text{desc}(\boldsymbol{\lambda}))$. The next value $\boldsymbol{\lambda}_i^{(k+1)}$ is computed by applying gradient descent for a predefined number of times, or until it yields an unfeasible solution. Therefore $\boldsymbol{\lambda}_i^{(k+1)} = \text{desc}_N(\bar{\boldsymbol{\lambda}}_i^{(k+1)})$, where N is equal to a predefined maximum number of successive applications of the gradient descent or such that $\text{desc}_{N+1}(\bar{\boldsymbol{\lambda}}_i^{(k+1)})$ is unfeasible. Until the **Stop criterion** is not met, the vehicle goes back to **Step 1**.

Stop criterion: The algorithm stops if the vehicle has reached the CZ.

This sequence of steps is implemented by the pseudocode described in Algorithm 1. Now, I prove that this algorithm converges to a feasible solution.

Theorem 4.2. *If $\mathcal{G}$ is connected, Algorithm 1 produces a feasible solution to problem (4.44) in a finite number of iterations.*

Proof. The proof is by induction. Without loss of generality, I assume that the vehicles are sorted by priority, i.e., $\chi_i > \chi_{i+1}$, $\forall i = 1, ..., n-1$.

At iteration $k = 0$, vehicle 1 has already reached the final value of its decision vector $\boldsymbol{\lambda}_1$, since it is the vehicle with maximum priority.

Then, I show that when the first $m-1$ vehicles have reached their final decision values, at iteration K_m, the m-th vehicle can determine a feasible value of $\boldsymbol{\lambda}_m$ in a finite number of iterations. For $i = 1, ..., m$, we have that $\boldsymbol{\lambda}_i^{(k+1)} = \boldsymbol{\lambda}_i^{(k)}$, $\forall k \geq K_m$, since these values are final. The graph is connected, therefore after $k_m = \max_{i \in \mathcal{V}} \delta(m, i)$ iterations, we have $\tilde{\boldsymbol{\lambda}}_i^{(k_m)} = \boldsymbol{\lambda}_i^{(k_m)}$, $\forall i \in \mathcal{V}$, $i \neq m$; therefore, **Step 2** and **Step 3** produce a feasible value of $\boldsymbol{\lambda}_m^{(k_m)}$.

□

Hereafter the pseudocode of the described algorithm is reported. In particular, Algorithm 2 performs the collision detection part; it returns `true` when a collision between vehicle i and any other known vehicle is detected and `false` otherwise. Algorithm 3 implements collision avoidance by taking the current value of $\boldsymbol{\lambda}_i$ and returning a new feasible value. Algorithm 1 is the full algorithm that comprises collision avoidance and optimization.

Algorithm 1 Full algorithm

Step 0:
Define $k \leftarrow 0$, $\boldsymbol{\lambda}_i^{(0)} \leftarrow \boldsymbol{\lambda}_{i,0}$

while the vehicle is in the MZ **do**
  Step 1:
  Broadcast all available data to each $j \in \mathcal{N}_i$
  Receive data from each $j \in \mathcal{N}_i$

  Step 2:
  $k \leftarrow k + 1$
  $\boldsymbol{\lambda}_i^{(k)} \leftarrow \text{AVOID}(\boldsymbol{\lambda}_i^{(k-1)})$

  Step 3:
  Assign a predefined value to N and ϵ
  while $N > 0$ **do**
    $\bar{\boldsymbol{\lambda}}_i \leftarrow \boldsymbol{\lambda}_i^{(k)} - \epsilon \nabla f_i(\boldsymbol{\lambda}_i^{(k)})$
    if $\neg$DETECT$(\bar{\boldsymbol{\lambda}}_i)$ **then**
      $\boldsymbol{\lambda}_i^{(k)} \leftarrow \bar{\boldsymbol{\lambda}}_i$
      $N \leftarrow N - 1$
    else
      $N \leftarrow 0$
    end if
  end while
end while

Algorithm 2 DETECT function (Collision detection)

Input: $\boldsymbol{\lambda}_i = \begin{bmatrix}\tau_i & \omega_i\end{bmatrix}^\mathsf{T}$
Output: **true** or **false**
for $j \in \mathcal{K}_i^{(k)}$ **do**
  $\boldsymbol{\nu}_{j,i} \leftarrow \begin{bmatrix}\omega_j^{-1} & -\omega_i^{-1}\end{bmatrix}^\mathsf{T}$.
  $\sigma_C^+ \leftarrow \sigma_C(\boldsymbol{\nu}_{j,i})$ and $\sigma_C^- \leftarrow \sigma_C(-\boldsymbol{\nu}_{j,i})$.
  if $-\sigma_C^- < \tau_i - \tau_j < \sigma_C^+$ **then**
    return true
  end if
end for
return false

Algorithm 3 AVOID Function (Collision avoidance)

Input: $\boldsymbol{\lambda}_i$
Output: $\boldsymbol{\lambda}_i$
if $\neg$ DETECT$(\boldsymbol{\lambda}_i)$ **then**
  return
end if
$F \leftarrow +\infty$.
for $j \in \mathcal{K}_i^{(k)}$ **do**
  for $\omega_i \in \omega_i^-, ..., \omega_i^+$ **do**
    $\boldsymbol{\nu}_{j,i} \leftarrow \begin{bmatrix}\omega_j^{-1} & -\omega_i^{-1}\end{bmatrix}^\mathsf{T}$.
    $\tau_i \leftarrow \tau_j - \sigma_C(-\boldsymbol{\nu}_{j,i})$ or $\tau_i \leftarrow \tau_j + \sigma_C(\boldsymbol{\nu}_{j,i})$
    if $\begin{bmatrix}\tau_i & \omega_i\end{bmatrix}^\mathsf{T} \in \Lambda_i \ \wedge \ \neg$ DETECT$(\tau_i, \omega_i) \ \wedge \ f_i(\tau_i, \omega_i) < F$ **then**
      $\boldsymbol{\lambda}_i \leftarrow \begin{bmatrix}\tau_i & \omega_i\end{bmatrix}^\mathsf{T}$
      $F \leftarrow f_i(\tau_i, \omega_i)$
    end if
  end for
end for

4.7 Simulations and numerical results

In this section, I present the numerical results obtained by testing the proposed algorithm through a simulation in MATLAB. The case study is an intersection of perpendicular roads ($\alpha = 90°$), with a lane width of 5 meters. 30 seconds of continuous traffic were simulated, with a total of $n = 44$ vehicles, which makes an average of 0.37 vehicles per second per lane. The interarrival time at the SZ is generated using an exponential distribution while ensuring a minimum safety distance of 3 meters between the vehicles. The bounding radii range from 1.5 to 2.1 meters. Both the SZ and the CZ are 30 meter long and connections between vehicles are established as soon as their distance is less than 60 meters; the communication happens at a frequency of 10 Hz, which means that each iteration must take 0.1 seconds at most. All vehicles use the minimum energy control and the employed cost function is proportional to (4.22):

$$f_i(\boldsymbol{\lambda}_i) = \frac{3\omega_{0,i}^2\left(\tau_i - \tau_{0,i}\right)^2}{\tau_i^3} + \frac{3\omega_{0,i}\left(\tau_i - \tau_{0,i}\right)\left(\omega_i - \omega_{0,i}\right)}{\tau_i^2} + \frac{\left(\omega_i - \omega_{0,i}\right)^2}{\tau_i}.$$

As for the constraints, I considered a minimum speed of $v^- = 10$ km/h and a maximum speed of $v^+ = 50$ km/h, whereas the maximum acceleration/deceleration value is $a^+ = -a^- = 1.5$ m/s^2. The `fmincon` function of MATLAB's Optimization Toolbox was used to compute the support function σ_C by (4.38). The presented results are obtained by a personal computer equipped with an Intel Core i7-9750H CPU, 16 GB of RAM and a NVIDIA GeForce GTX 1650 graphics card.

In the simulation, all vehicles crossed the intersection without rear-end or lateral collisions. The average delay caused by the intersection was 1.02 seconds and the histogram showing the distribution of the delays is shown in figure 4.19.

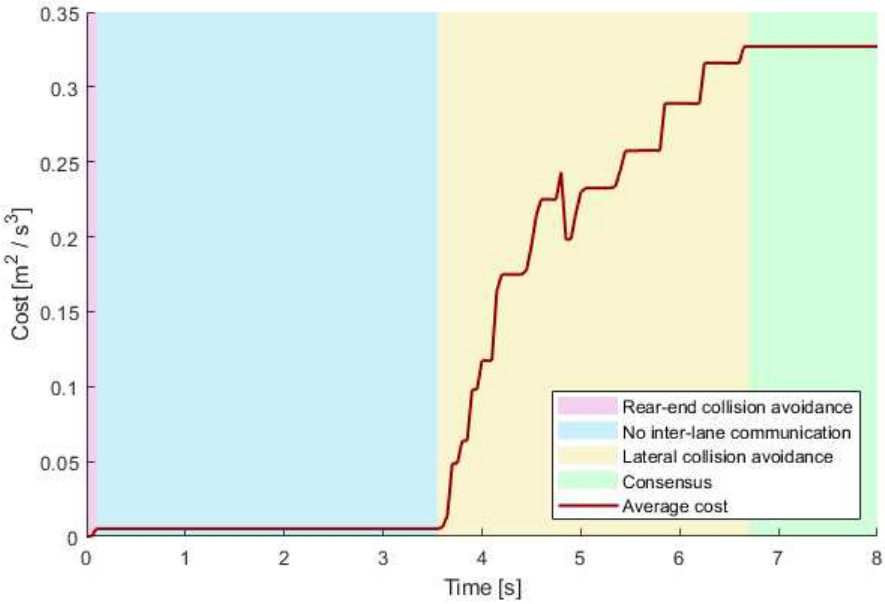


Figure 4.18: *Average cost over time.*

Figure 4.18, instead, shows the average cost over time. We can distinguish three phases.

1. In the very first iterations, vehicles from different lanes are too far apart to communicate, therefore only rear-end collision avoidance happens between vehicles of the same lane (pink and blue regions of the plot).

2. As soon as the leading vehicles come near enough, the communication graph gets connected and the lateral collision avoidance is performed.

3. After a finite amount of time, 3 seconds in this case, the vehicles reach consensus on the crossing schedule.

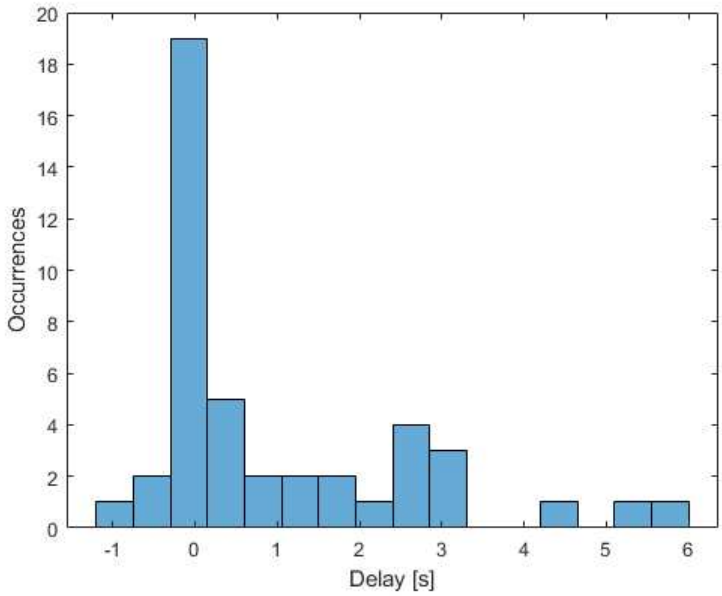


Figure 4.19: *Distribution of delays caused by the intersection.*

As for performances, the maximum duration of a single iteration was 44 milliseconds, which is small enough for the 10 Hz communication frequency.

The video output of the simulation can be visualized on YouTube at `https://youtu.be/Y6wzARyBVog`.

Chapter 5

Leaderless control of multi-agent systems

The aim of this section is to study a consensus protocol to be applied by a leaderless network of autonomous agents that have to reach a common velocity while forming a uniformly spaced string. The main objective is to optimize the protocol parameters to maximize the convergence speed by avoiding oscillations. A secondary objective is to study the effect of constant delay on the dynamics and the stability of the system, given the proposed control strategy. I propose and prove the conditions that the consensus parameters have to satisfy to guarantee the asymptotic stability of the multi-agent system dynamics. Then, I show that the consensus protocol parameters can be optimized in order to maximize the convergence speed and avoid oscillations if the network topology is described by a class of connected digraphs. For what concerns delays, a necessary and sufficient condition that guarantees consensus achievement under constant communication time delay is proved, and an explicit expression on how large the delay can be to guarantee consensus is reached.

5.1 A second-order consensus problem

In this context, first-order consensus algorithms are not suitable due to the governing dynamics of physical systems, such as vehicles, being dictated by Newton's laws of motion, implying a second-order dynamic system. The foundation of this work lies in existing second-order consensus protocols proposed, for instance, in [83], [101], [121], and it considers a protocol to be applied by a leaderless network of autonomous agents that have to reach a common velocity while forming a uniformly spaced string. More precisely, the value of the final common velocity (reference velocity) is decided by the agents through the consensus protocol, starting from an initial desired value for each agent. Furthermore, the agents communicate in a communication network described by a directed graph (digraph) having a directed spanning tree.

5.1.1 Consensus problem statement

The communication topology of the group of agents is described by a directed graph (digraph) that consists of a set of n vertices $\mathcal{V} = [\![1, n]\!]$ connected by a set of edges $\mathcal{E} \subset \mathcal{V} \times \mathcal{V}$. The dynamics of agent i, for $i \in [\![1, n]\!]$, is described by a second-order system where x_i denotes the position of the agent, v_i its velocity and u_i the control input:

$$\begin{bmatrix}\dot{x}_i\\ \dot{v}_i\end{bmatrix} = \begin{bmatrix}0 & 1\\ 0 & 0\end{bmatrix}\begin{bmatrix}x_i\\ v_i\end{bmatrix} + \begin{bmatrix}0\\ 1\end{bmatrix} u_i.$$

The positions, velocities, and inputs of the multi-agent system are grouped in the n-dimensional vectors $\boldsymbol{x}$, $\boldsymbol{v}$, and $\boldsymbol{u}$, respectively. Thus, the multi-agent system dynamics can be written as

$$\begin{bmatrix}\dot{\boldsymbol{x}}\\ \dot{\boldsymbol{v}}\end{bmatrix} = \begin{bmatrix}\boldsymbol{O}_n & \boldsymbol{I}_n\\ \boldsymbol{O}_n & \boldsymbol{O}_n\end{bmatrix}\begin{bmatrix}\boldsymbol{x}\\ \boldsymbol{v}\end{bmatrix} + \begin{bmatrix}\boldsymbol{0}_n\\ \boldsymbol{u}\end{bmatrix}.$$

The multi-agent system control problem is to attain the following behavior of the agent dynamics:

1. Each agent must reach and steadily keep a common reference velocity $\bar{v}$.
2. All the agents must be spaced with uniform interspace gap $\bar{d}$.

Denoting by $\boldsymbol{d}(t) = \boldsymbol{H}\boldsymbol{x}(t) \in \mathbb{R}^{n-1}$ the inter-agent distance vector, the objective of the multi-agent system control problem can be formally denoted as follows:

$$\lim_{t\to\infty} \boldsymbol{d}(t) = \bar{\boldsymbol{d}}, \quad \lim_{t\to\infty} \boldsymbol{v}(t) = \bar{\boldsymbol{v}}, \tag{5.1}$$

where $\bar{\boldsymbol{d}} = \boldsymbol{1}_{n-1}\bar{d}$ and $\bar{\boldsymbol{v}} = \boldsymbol{1}_n\bar{v}$. I assume the common reference velocity that each agent has to reach is unknown to the agents, hence they must estimate it to have a reference velocity to follow. More precisely, each agent i starts from an initial estimate $y_i(0)$ of the reference velocity; this estimate is updated using the values communicated by the other agents, using the following first-order consensus protocol:

$$\dot{y}_i = -\eta \sum_{j\in\mathcal{N}(i)} (y_i - y_j), \tag{5.2}$$

with $\eta \in \mathbb{R}_+$. This equation can be put in vector form, resulting in

$$\dot{\boldsymbol{y}} = -\eta \boldsymbol{L}\boldsymbol{y}. \tag{5.3}$$

5.1.2 Control law synthesis via Lyapunov's method

In addition to its usefulness in stability analysis, Lyapunov's direct method can be used to design control laws. This time, stability in its strictest sense is not of interest: in fact, the movement of a group of cars at constant speed can certainly not be called stable, since their position coordinates increase indefinitely over time. Therefore, a new system that encodes the goals of the multi-agent system control problem is defined, such that its stability implies (5.1).

Before introducing this new system, I introduce the difference matrix and provide a basic result that is used throughout the chapter. The *difference matrix* is the matrix $\boldsymbol{H} \in \mathbb{R}^{(n-1)\times n}$ defined as

$$\boldsymbol{H} = \begin{bmatrix} 1 & -1 & & & \\ & 1 & -1 & & \\ & & \ddots & \ddots & \\ & & & 1 & -1 \end{bmatrix}.$$

It is called difference matrix because the product of its i-th row with a vector equals the difference between the i-th and $(i+1)$-th components of the vector. $\boldsymbol{H}$ has full row-rank, therefore its (right) Moore-Penrose inverse can be computed as $\boldsymbol{H}^+ = \boldsymbol{H}^\mathsf{T}(\boldsymbol{H}\boldsymbol{H}^\mathsf{T})^{-1}$. $\boldsymbol{H}^+$ is also called the (right) pseudo-inverse of $\boldsymbol{H}$ because $\boldsymbol{H}\boldsymbol{H}^+ = \boldsymbol{H}\boldsymbol{H}^\mathsf{T}(\boldsymbol{H}\boldsymbol{H}^\mathsf{T})^{-1} = \boldsymbol{I}_n$. Moreover, it is easy to check $\mathcal{K}(\boldsymbol{H}) = \mathcal{S}(\mathbf{1}_n)$. Now, I provide a basic result.

Lemma 5.1. *Let $\boldsymbol{T} \in \mathbb{R}^{n\times n}$ and $\boldsymbol{U} \in \mathbb{R}^{m\times n}$, with $m < n$. If $\mathcal{K}(\boldsymbol{U}) \subseteq \mathcal{S}(\boldsymbol{T})$, then the following equality holds:*

$$\boldsymbol{T} = \boldsymbol{T}\boldsymbol{U}^+\boldsymbol{U}. \tag{5.4}$$

Proof. The proof is in Appendix A. □

In this case of study, we have that $\mathcal{K}(\boldsymbol{H}) = \mathcal{S}(\mathbf{1}_n) \subseteq \mathcal{K}(\boldsymbol{L})$; therefore, Lemma 5.1 can be applied to $\boldsymbol{L}$ and $\boldsymbol{H}$:

$$\boldsymbol{L} = \boldsymbol{L}\boldsymbol{H}^+\boldsymbol{H}. \tag{5.5}$$

In other words, regardless of the graph topology, $\boldsymbol{L}$ can be decomposed as the product of matrix $\boldsymbol{L}\boldsymbol{H}^+$ and the difference matrix $\boldsymbol{H}$.

We can use $\boldsymbol{H}$ to define the variables of the new system as follows:

$$\boldsymbol{p} = \boldsymbol{H}\boldsymbol{x} - \bar{\boldsymbol{d}}, \quad \boldsymbol{q} = \boldsymbol{v} - \bar{\boldsymbol{v}}. \tag{5.6}$$

Note that this transformation is not a change of coordinates: in fact, it reduces the dimensionality of the system, from $2n$ to $2n-1$; therefore it is not even invertible. Noting that $\boldsymbol{H}\bar{\boldsymbol{v}} = \mathbf{0}$, the dynamics of the new variables (5.6) are given by

$$\begin{aligned} \dot{\boldsymbol{p}} &= \boldsymbol{H}\dot{\boldsymbol{x}} = \boldsymbol{H}\boldsymbol{v} = \boldsymbol{H}(\boldsymbol{q} + \bar{\boldsymbol{v}}) = \boldsymbol{H}\boldsymbol{q}, \\ \dot{\boldsymbol{q}} &= \dot{\boldsymbol{v}} = \boldsymbol{u}. \end{aligned}$$

The control input $\boldsymbol{u}$ must be chosen such that the origin of the system is made an asymptotically stable equilibrium point. This would imply (5.1), as

$$q_i = 0 \Rightarrow x_i - x_{i+1} - \bar{d} \quad \text{and} \quad r_i = 0 \Rightarrow v_i = \bar{v}.$$

A standard Lyapunov function is used in the design of a control law with the goal of stabilizing the origin:

$$V(\boldsymbol{p}, \boldsymbol{q}) = \frac{1}{2}\|\boldsymbol{p}\|^2 + \frac{1}{2}\|\boldsymbol{q}\|^2,$$

which is obviously positive-definite in the origin. The computation if its time derivative is immediate; with abuse of notation, I write:

$$\dot{V}(\boldsymbol{p},\boldsymbol{q}) = \frac{\partial V}{\partial \boldsymbol{p}}\dot{\boldsymbol{p}} + \frac{\partial V}{\partial \boldsymbol{q}}\dot{\boldsymbol{q}} = \boldsymbol{p}^\mathsf{T}\dot{\boldsymbol{p}} + \boldsymbol{q}^\mathsf{T}\dot{\boldsymbol{q}} = \boldsymbol{p}\boldsymbol{H}\boldsymbol{q} + \boldsymbol{q}^\mathsf{T}\boldsymbol{u} = \boldsymbol{q}^\mathsf{T}(\boldsymbol{H}^\mathsf{T}\boldsymbol{p} + \boldsymbol{u}).$$

It is apparent that one cannot choose $\boldsymbol{u}$ so that $\dot{V}$ is negative-definite in the origin, but at least it can be made negative-semidefinite, and then one can prove the stability with other methods. A desirable choice of $\boldsymbol{u}$ would be

$$\boldsymbol{u} = -\boldsymbol{H}^\mathsf{T}\boldsymbol{p} + \boldsymbol{K}\boldsymbol{q}, \tag{5.7}$$

with $\boldsymbol{K}$ being a negative definite matrix. A possible choice for $\boldsymbol{K}$ was proposed from [122] and [123]:

$$\boldsymbol{K} = -(\gamma\boldsymbol{L} + \kappa\boldsymbol{I}_n),$$

with γ and κ being positive scalars. Unfortunately, $-\boldsymbol{H}^\mathsf{T}\boldsymbol{p}$ is not always a feasible control input, in the sense that agents might not be able to compute it. In fact, considering its expression,

$$\boldsymbol{H}^\mathsf{T}\boldsymbol{p} = \boldsymbol{H}^\mathsf{T}(\boldsymbol{H}\boldsymbol{x} - \bar{\boldsymbol{d}}) = \boldsymbol{H}^\mathsf{T}\boldsymbol{H}\boldsymbol{x} - \boldsymbol{H}^\mathsf{T}\bar{\boldsymbol{d}},$$

the $\boldsymbol{H}^\mathsf{T}\boldsymbol{H}\boldsymbol{x}$ part can be computed if and only if each vehicle can communicate with its predecessor and its successor, except for the first and last vehicles that must have access to their successor and predecessor only, respectively. This is apparent by considering the components of the $\boldsymbol{H}^\mathsf{T}\boldsymbol{H}\boldsymbol{x}$ vector. Firstly, the product $\boldsymbol{H}^\mathsf{T}\boldsymbol{H}$ is:

$$\boldsymbol{H}^\mathsf{T}\boldsymbol{H}\boldsymbol{x} = \begin{bmatrix} 1 & & & \\ -1 & 1 & & \\ & -1 & \ddots & \\ & & \ddots & 1 \\ & & & -1 \end{bmatrix} \begin{bmatrix} 1 & -1 & & & \\ & 1 & -1 & & \\ & & \ddots & \ddots & \\ & & & 1 & -1 \end{bmatrix} = \begin{bmatrix} 1 & -1 & & & \\ -1 & 2 & -1 & & \\ & \ddots & \ddots & \ddots & \\ & & -1 & 2 & -1 \\ & & & -1 & 1 \end{bmatrix}.$$

Then, we have

$$\boldsymbol{H}^\mathsf{T}\boldsymbol{H}\boldsymbol{x} = \begin{bmatrix} 1 & -1 & & & \\ -1 & 2 & -1 & & \\ & \ddots & \ddots & \ddots & \\ & & -1 & 2 & -1 \\ & & & -1 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ \vdots \\ x_i \\ x_n \end{bmatrix} = \begin{bmatrix} x_1 - x_2 \\ \vdots \\ -x_{i-1} + 2x_i - x_{i+1} \\ \vdots \\ -x_{n-1} + x_n \end{bmatrix}.$$

However, this result provides useful insights into the structure of a possibly feasible control law. I consider a different control law that is feasible and similar to (5.7):

$$\boldsymbol{u} = -(\boldsymbol{L}\boldsymbol{x} - \boldsymbol{h}) - (\gamma\boldsymbol{L} + \kappa\boldsymbol{I}_n)\boldsymbol{q},$$

with $\boldsymbol{h}$ being a constant vector whose expression is to be determined. Intuitively, $\boldsymbol{u}$ should depend on $\boldsymbol{p}$, because the control law must be able to react adequately when $\boldsymbol{p} \neq \boldsymbol{0}_n$. Therefore, I exploit (5.5) and choose of $\boldsymbol{h} = \boldsymbol{L}\boldsymbol{H}^+\bar{\boldsymbol{d}}$:

$$\boldsymbol{Lx} - \boldsymbol{h} = \boldsymbol{LH}^{+}\boldsymbol{Hx} - \boldsymbol{LH}^{+}\bar{\boldsymbol{d}} = \boldsymbol{LH}^{+}(\boldsymbol{Hx} - \bar{\boldsymbol{d}}) = \boldsymbol{LH}^{+}\boldsymbol{p}.$$

By noting that $\boldsymbol{L}\bar{\boldsymbol{v}} = \boldsymbol{0}_n$, the control law has now the following expression:

$$\boldsymbol{u} = -\boldsymbol{LH}^{+}\boldsymbol{p} - (\gamma\boldsymbol{L} + \kappa\boldsymbol{I}_n)\boldsymbol{q} = -\boldsymbol{Lx} + \boldsymbol{LH}^{+}\bar{\boldsymbol{d}} - \gamma\boldsymbol{Lv} - \kappa(\boldsymbol{v} - \bar{\boldsymbol{v}}). \tag{5.8}$$

Finally, I clarify how $\boldsymbol{LH}^{+}\bar{\boldsymbol{d}}$ is computed by each agent.

Proposition 5.1. *The expression of the i-th component of vector* $\boldsymbol{h} = \boldsymbol{LH}^{+}\bar{\boldsymbol{d}}$ *is given by*

$$h_i = d \sum_{j \in \mathcal{N}(i)} (i - j).$$

Proof. The proof is in Appendix A. □

5.1.3 Consensus protocol

The proposed consensus protocol is based on (5.8), but assumes each vehicle i uses y_i as an estimate of $\bar{v}$. The proposed control law is

$$\boldsymbol{u} = -\boldsymbol{Lx} + \boldsymbol{LH}^{+}\bar{\boldsymbol{d}} - \gamma\boldsymbol{L}(\boldsymbol{v} - \boldsymbol{y}) - \kappa(\boldsymbol{v} - \boldsymbol{y}). \tag{5.9}$$

The actual control law for agent i is obtained by breaking down (5.9) into individual components:

$$u_i = -\sum_{j \in \mathcal{N}(i)} \big((x_i - x_j) - \bar{d}(i - j) - \gamma(v_i - v_j) + \gamma(y_i - y_j)\big) - \kappa(v_i - y_i). \tag{5.10}$$

In other words, by using (5.10), each agent can communicate its own velocity and the relative distance with its neighbors. Such an assumption is very common in the related literature [85], [103], [104], [121]. Assuming that each agent has to reach a common reference velocity $\bar{v}$ and the same inter-space distance $\bar{d}$, the rationale of algorithm (5.10) is the following: by the first terms each agent communicates the actual distance from its neighbor and the objective inter-space to be imposed between two nearby agents; analogously by the second term, the agents communicate the actual difference between the velocities of its neighbors and the reference velocity. Parameters γ and κ determine the weights given to the relative and absolute velocity feedbacks, respectively, with respect to the distance (relative) feedback. Therefore, γ and κ can be referred to as *realtive feedback* and *absolute feedback*, respectively. In other words, with a higher value of γ, the control puts more weight on the global error. Instead, with a higher value of κ each agent becomes more "selfish", giving its absolute velocity error more importance.

Protocol (5.2), instead, makes y_i, for $i = 1, ..., n$ converge to a common value, which is the agreed velocity of the group and depends on the communication topology and on the initial values $y_i(0)$. This avoids the need for a leader and is independent of the initial velocities $v_i(0)$.

The complete dynamics of the system are given by combining (5.9) and (5.3):

$$\begin{bmatrix} \dot{\boldsymbol{x}} \\ \dot{\boldsymbol{v}} \\ \dot{\boldsymbol{y}} \end{bmatrix} = \underbrace{\begin{bmatrix} \boldsymbol{O}_n & \boldsymbol{I}_n & \boldsymbol{O}_n \\ -\boldsymbol{L} & -(\gamma\boldsymbol{L} + \kappa\boldsymbol{I}_n) & \gamma\boldsymbol{L} + \kappa\boldsymbol{I}_n \\ \boldsymbol{O}_n & \boldsymbol{O}_n & -\eta\boldsymbol{L} \end{bmatrix}}_{\boldsymbol{A}} \begin{bmatrix} \boldsymbol{x} \\ \boldsymbol{v} \\ \boldsymbol{y} \end{bmatrix} + \begin{bmatrix} \boldsymbol{0}_n \\ \boldsymbol{L}\boldsymbol{H}^{+}\bar{\boldsymbol{d}} \\ \boldsymbol{0}_n \end{bmatrix}. \tag{5.11}$$

Matrix $\boldsymbol{A}$ can be partitioned as follows:

$$\boldsymbol{A}_{1,1} = \begin{bmatrix} \boldsymbol{O}_n & \boldsymbol{I}_n \\ -\boldsymbol{L} & -(\gamma\boldsymbol{L} + \kappa\boldsymbol{I}_n) \end{bmatrix}, \quad \boldsymbol{A}_{1,2} = \begin{bmatrix} \boldsymbol{O}_n \\ \gamma\boldsymbol{L} + \kappa\boldsymbol{I}_n \end{bmatrix},$$

$$\boldsymbol{A}_{2,1} = \begin{bmatrix} \boldsymbol{O}_n & \boldsymbol{O}_n \end{bmatrix}, \quad \boldsymbol{A}_{2,2} = -\eta\boldsymbol{L}.$$

Because of the block structure of matrix $\boldsymbol{A}$, its spectrum is the union of the spectra of matrices $\boldsymbol{A}_{1,1}$ and $\boldsymbol{A}_{2,2}$. Let μ_i, with $i = 0, ..., n-1$, denote the eigenvalues of the Laplacian matrix $\boldsymbol{L}$. As it is shown in [124], the characteristic polynomial of $\boldsymbol{A}_{1,1}$ is the following:

$$p_{\boldsymbol{A}_{1,1}}(\lambda) = \prod_{i=0}^{n-1} \left(\lambda^2 + (\kappa + \gamma\mu_i)\lambda + \mu_i\right),$$

hence, the eigenvalues of $\boldsymbol{A}_{1,1}$ are

$$\lambda_{i,\pm} = \frac{-(\kappa + \gamma\mu_i) \pm \sqrt{(\kappa + \gamma\mu_i)^2 - 4\mu_i}}{2}. \tag{5.12}$$

Moreover, the characteristic polynomial of $\boldsymbol{A}_{2,2}$ is

$$p_{\boldsymbol{A}_{2,2}}(\lambda) = \prod_{i=0}^{n-1} (\lambda + \eta\mu_i).$$

It is clear that all eigenvalues of $\boldsymbol{A}_{2,2}$, except for the null one, have negative real parts if and only if the graph is connected. The dynamics of $\boldsymbol{y}$ are independent of the dynamics of the rest of the multi-agent system. Under the assumption of connectedness, by (3.15), we obtain the convergence value of this protocol:

$$\lim_{t\to\infty} \boldsymbol{y}(t) = \bar{v}\boldsymbol{1}_n, \tag{5.13}$$

with $\bar{v} = \boldsymbol{\xi}^{\mathsf{T}}\boldsymbol{y}(0)$, where $\boldsymbol{\xi}$ is the left eigenvector of $-\eta\boldsymbol{L}$ associated to its null eigenvalue, chosen so that $\boldsymbol{\xi}^{\mathsf{T}}\boldsymbol{1}_n = 1$.

5.2 Convergence

For a given communication topology, hence considering fixed values of μ_i, as (5.12) shows, the values of γ and κ have a direct impact on the eigenvalues of the system.

Firstly, note that each eigenvalue μ_i of $\boldsymbol{L}$ has two associated eigenvalues of $\boldsymbol{A}_{1,1}$, namely $\lambda_{i,+}$ and $\lambda_{i,-}$. The two eigenvalues $\lambda_{0,+}$ and $\lambda_{0,-}$ associated with $\mu_0 = 0$ are

$$\lambda_{0,+} = 0, \quad \lambda_{0,-} = -\kappa.$$

To prove the convergence of the system so that (5.1) is reached, a necessary condition is for all other eigenvalues to have negative real parts. However, this is not sufficient, because the asymptotic behavior of the system cannot be deduced from its eigenvalues only.

To solve this problem, I perform a change of variables on the lines of (5.6), by defining the transformations

$$\begin{aligned} \boldsymbol{p} &= \boldsymbol{H}\boldsymbol{x} - \bar{\boldsymbol{d}}, \\ \boldsymbol{q} &= \boldsymbol{v} - \boldsymbol{y}, \\ \boldsymbol{r} &= \boldsymbol{H}\boldsymbol{y} \end{aligned} \tag{5.14}$$

The derivatives of the new variables are

$$\begin{aligned} \dot{\boldsymbol{p}} &= \boldsymbol{H}\dot{\boldsymbol{x}} = \boldsymbol{H}\boldsymbol{v} = \boldsymbol{H}(\boldsymbol{q}+\boldsymbol{y}) = \boldsymbol{H}\boldsymbol{q} + \boldsymbol{r}, \\ \dot{\boldsymbol{q}} &= \dot{\boldsymbol{v}} - \dot{\boldsymbol{y}} = -\boldsymbol{L}\boldsymbol{x} + \boldsymbol{L}\boldsymbol{H}^{+}\bar{\boldsymbol{d}} - \gamma\boldsymbol{L}(\boldsymbol{v}-\boldsymbol{y}) - \kappa(\boldsymbol{v}-\boldsymbol{y}) - \eta\boldsymbol{L}\boldsymbol{y} = \\ &= -\boldsymbol{L}\boldsymbol{H}^{+}\boldsymbol{p} - (\gamma\boldsymbol{L} + \kappa\boldsymbol{I}_n)\boldsymbol{q} + \eta\boldsymbol{L}\boldsymbol{H}^{+}\boldsymbol{r}, \\ \dot{\boldsymbol{r}} &= \boldsymbol{H}\dot{\boldsymbol{y}} = -\eta\boldsymbol{H}\boldsymbol{L}\boldsymbol{y} = -\eta\boldsymbol{H}\boldsymbol{L}\boldsymbol{H}^{+}\boldsymbol{H}\boldsymbol{y} = -\eta\boldsymbol{H}\boldsymbol{L}\boldsymbol{H}^{+}\boldsymbol{r}. \end{aligned} \tag{5.15}$$

By putting these three equations in a compact form, we obtain the dynamics of system (5.11) under transformations (5.14):

$$\begin{bmatrix} \dot{\boldsymbol{p}} \\ \dot{\boldsymbol{q}} \\ \dot{\boldsymbol{r}} \end{bmatrix} = \underbrace{\begin{bmatrix} \boldsymbol{O}_{n-1} & \boldsymbol{H} & \boldsymbol{I}_{n-1} \\ -\boldsymbol{L}\boldsymbol{H}^{+} & -(\gamma\boldsymbol{L} + \kappa\boldsymbol{I}_n) & \eta\boldsymbol{L}\boldsymbol{H}^{+} \\ \boldsymbol{O}_{n-1} & \boldsymbol{O}_{n-1,n} & -\eta\boldsymbol{H}\boldsymbol{L}\boldsymbol{H}^{+} \end{bmatrix}}_{\boldsymbol{F}} \begin{bmatrix} \boldsymbol{p} \\ \boldsymbol{q} \\ \boldsymbol{r} \end{bmatrix}. \tag{5.16}$$

In this state space, control law (5.9) is written as follows:

$$\boldsymbol{u} = -\boldsymbol{L}\boldsymbol{H}^{+}\boldsymbol{p} - (\gamma\boldsymbol{L} + \kappa\boldsymbol{I}_n)\boldsymbol{q}. \tag{5.17}$$

Lemma 5.2. *Equations (5.1) and (5.13) hold if and only if system (5.16) is asymptotically stable.*

Proof. (Only if) Suppose equations (5.1) and (5.13) are verified. Then, trivially,

$$\bar{\boldsymbol{d}} = \lim_{t\to\infty} \boldsymbol{d}(t) = \lim_{t\to\infty} \boldsymbol{H}\boldsymbol{x}(t) = \lim_{t\to\infty} \boldsymbol{p}(t) + \bar{\boldsymbol{d}}$$

and $\lim_{t\to\infty} \boldsymbol{p}(t) = \boldsymbol{0}_n$ follows. Then,

$$\bar{\boldsymbol{v}} = \lim_{t\to\infty} \boldsymbol{v}(t) = \lim_{t\to\infty} \boldsymbol{y}(t),$$

therefore $\lim_{t\to\infty} \boldsymbol{q}(t) = \boldsymbol{0}_n$ and $\lim_{t\to\infty} \boldsymbol{r}(t) = \boldsymbol{0}_n$ follow.

(If) Suppose system (5.16) is asymptotically stable. Then, $\lim_{t\to\infty} \boldsymbol{r}(t) = \lim_{t\to\infty} \boldsymbol{H}\boldsymbol{y}(t) = \boldsymbol{0}_n$. $\mathcal{K}(\boldsymbol{H}) = \mathcal{S}(\boldsymbol{1}_n)$ implies $\lim_{t\to\infty} \boldsymbol{y}(t) = \boldsymbol{1}_n\bar{v}$, for some $\bar{v}$. From this and $\boldsymbol{0}_n = \lim_{t\to\infty} \boldsymbol{p}(t) = \lim_{t\to\infty} \boldsymbol{q}(t)$, (5.1) immediately follows.

□

Asymptotic stability of (5.16), is implied by the eigenvalues of $\boldsymbol{F}$ having strictly negative real parts. Let us partition $\boldsymbol{F}$ into 4 blocks:

$$\boldsymbol{F}_{1,1} = \begin{bmatrix} \boldsymbol{O}_{n-1} & \boldsymbol{H} \\ -\boldsymbol{L}\boldsymbol{H}^{+} & -(\gamma\boldsymbol{L} + \kappa\boldsymbol{I}_n) \end{bmatrix}, \quad \boldsymbol{F}_{1,2} = \begin{bmatrix} \boldsymbol{I}_{n-1} \\ \eta\boldsymbol{L}\boldsymbol{H}^{+} \end{bmatrix},$$

$$\boldsymbol{F}_{2,1} = \begin{bmatrix} \boldsymbol{O}_n & \boldsymbol{O}_{n,n-1} \end{bmatrix}, \quad \boldsymbol{F}_{2,2} = -\eta\boldsymbol{H}\boldsymbol{L}\boldsymbol{H}^{+}.$$

Again, because of the block structure, the spectrum of $\boldsymbol{F}$ is the union of the spectra of $\boldsymbol{F}_{1,1}$ and $\boldsymbol{F}_{2,2}$. Now, I prove it coincides with the spectrum of $\boldsymbol{A}$, up to the two null eigenvalues.

Lemma 5.3. *The eigenvalues of $\boldsymbol{F}$ are the solutions to*

$$(\lambda + \kappa)\prod_{i=1}^{n-1}(\lambda + \eta\mu_i)\left(\lambda^2 + (\kappa + \gamma\mu_i)\lambda + \mu_i\right) = 0.$$

Proof. Let us start with the determination of the characteristic polynomial of $\boldsymbol{F}_{2,2}$. The Jordan canonical form of $\boldsymbol{L}$ is expressed by highlighting the null eigenvalue and the corresponding eigenvectors:

$$\boldsymbol{L} = \boldsymbol{P}\boldsymbol{J}\boldsymbol{Q} = \begin{bmatrix} \mathbf{1}_n & \boldsymbol{P}' \end{bmatrix} \begin{bmatrix} 0 & \\ & \boldsymbol{J}' \end{bmatrix} \begin{bmatrix} \boldsymbol{\xi}^* \\ \boldsymbol{Q}' \end{bmatrix} = \boldsymbol{P}'\boldsymbol{J}'\boldsymbol{Q}'.$$

Furthermore, the product of $\boldsymbol{P}$ and $\boldsymbol{Q}$ is the identity matrix and can be written as

$$\boldsymbol{I}_n = \boldsymbol{P}\boldsymbol{Q} = \begin{bmatrix} \mathbf{1}_n & \boldsymbol{P}' \end{bmatrix} \begin{bmatrix} \boldsymbol{\xi}^* \\ \boldsymbol{Q}' \end{bmatrix} = \mathbf{1}_n\boldsymbol{\xi}^* + \boldsymbol{P}'\boldsymbol{Q}'.$$

The characteristic polynomial of $\boldsymbol{F}_{2,2}$ is

$$\begin{aligned} p_{\boldsymbol{F}_{2,2}}(\lambda) &= |\lambda\boldsymbol{I}_{n-1} - \boldsymbol{F}_{2,2}| = \left|\lambda\boldsymbol{H}\boldsymbol{I}_n\boldsymbol{H}^{+} + \eta\boldsymbol{H}\boldsymbol{L}\boldsymbol{H}^{+}\right| = \\ &= \left|\lambda\boldsymbol{H}(\mathbf{1}_n\boldsymbol{\xi}^* + \boldsymbol{P}'\boldsymbol{Q}')\boldsymbol{H}^{+} + \eta\boldsymbol{H}\boldsymbol{P}'\boldsymbol{J}'\boldsymbol{Q}'\boldsymbol{H}^{+}\right| = \left|\boldsymbol{H}\boldsymbol{P}'(\lambda\boldsymbol{I}_{n-1} + \eta\boldsymbol{J}')\boldsymbol{Q}'\boldsymbol{H}^{+}\right|. \end{aligned}$$

The eigenvalues of $\boldsymbol{F}_{2,2}$ are the solutions to $p_{\boldsymbol{F}_{2,2}}(\lambda) = 0$, which are also the eigenvalues of $\lambda\boldsymbol{I}_{n-1} + \eta\boldsymbol{J}'$, given by:

$$\prod_{i=1}^{n-1}(\lambda + \eta\mu_i) = 0. \tag{5.18}$$

The characteristic polynomial of $\boldsymbol{F}_{1,1}$ is

$$p_{\boldsymbol{F}_{1,1}}(\lambda) = |\lambda\boldsymbol{I}_{2n-1} - \boldsymbol{F}_{1,1}| = \begin{vmatrix} \lambda\boldsymbol{I}_{n-1} & -\boldsymbol{H} \\ \boldsymbol{L}\boldsymbol{H}^{+} & (\lambda + \kappa)\boldsymbol{I}_n + \gamma\boldsymbol{L} \end{vmatrix}.$$

Since the first block of this matrix is invertible, the determinant becomes

$$p_{\boldsymbol{F}_{1,1}}(\lambda) = |\lambda\boldsymbol{I}_{n-1}|\left|(\lambda + \kappa)\boldsymbol{I}_n + \gamma\boldsymbol{L} + \boldsymbol{L}\boldsymbol{H}^{+}\lambda^{-1}\boldsymbol{H}\right| = \lambda^{-1}\left|\lambda(\lambda + \kappa)\boldsymbol{I}_n + (1 + \gamma\lambda)\boldsymbol{L}\right|$$

As shown in [124], we have that

$$|\lambda(\lambda+\kappa)\boldsymbol{I}_n + (1+\gamma\lambda)\boldsymbol{L}| = \prod_{i=0}^{n-1} \left(\lambda(\lambda+\kappa) + (1+\gamma\lambda)\mu_i\right),$$

therefore, the eigenvalues of $\boldsymbol{F}_{1,1}$ are given by

$$\lambda^{-1}\prod_{i=0}^{n-1} \left(\lambda(\lambda+\kappa) + (1+\gamma\lambda)\mu_i\right) = \lambda(\lambda+\kappa)\prod_{i=1}^{n-1} \left((\lambda+\kappa) + (1+\gamma\lambda)\mu_i\right) = 0. \tag{5.19}$$

The thesis is obtained by combining (5.18) and (5.19).

□

The characterization of the eigenvalues of $\boldsymbol{F}$ given by this lemma is now used to prove the conditions that γ and κ must satisfy to ensure that system (5.16) is asymptotically stable, implying protocol (5.9) successfully achieve (5.1). Before introducing the result, I recall that $\alpha_i = \Re(\mu_i)$ and $\beta_i = \Im(\mu_i)$.

Theorem 5.1. *System (5.16) is asymptotically stable if $\mathcal{G}$ has a directed spanning tree and*

$$\alpha_i(\alpha_i^2+\beta_i^2)\gamma^2 + (2\alpha_i^2+\beta_i^2)\gamma\kappa + \alpha_i\kappa^2 - \beta_i^2 > 0, \quad \forall i = 1, ..., n-1. \tag{5.20}$$

Proof. If $\mathcal{G}$ has a directed spanning tree, then we know that $\alpha_i > 0$, for all $i = 1, ..., n-1$. The eigenvalue λ_i given by (5.18) has negative real part, since $\Re(\lambda_i) = \Re(-\eta\mu_i) = -\eta\alpha_i < 0$, for all $i = 1, ..., n-1$. Now, let us consider the eigenvalues given by (5.19), namely

$$\lambda_i^2 + (\kappa + \gamma\mu_i)\lambda + \mu_i = 0.$$

Using the Routh-Hurwitz criterion for second-order polynomials with complex coefficients, we have that matrix $\boldsymbol{F}_{1,1}$ is Hurwitz-stable if and only if

$$\Re(\kappa + \gamma\mu_i) > 0, \quad \forall i = 1, ..., n-1 \tag{5.21}$$

and

$$\Re(\kappa + \gamma\mu_i)\,\Re((\kappa+\gamma\mu_i)\mu_i) - \Im(\mu_i)^2 > 0, \quad \forall i = 1, ..., n-1. \tag{5.22}$$

Condition (5.21) is always verified, since $\gamma > 0$, $\kappa > 0$, and $\alpha_i > 0$, for all $i = 1, ..., n-1$. If we write condition (5.22) with $\Re(\mu_i) = \alpha_i$ and $\Im(\mu_i) = \beta_i$, we obtain

$$(\kappa + \gamma\alpha_i)\left((\kappa+\gamma\alpha_i)\alpha_i + \gamma\beta_i^2\right) - \beta_i^2 > 0\ \forall i = 1, ..., n-1.$$

After some calculations, (5.20) follows.

□

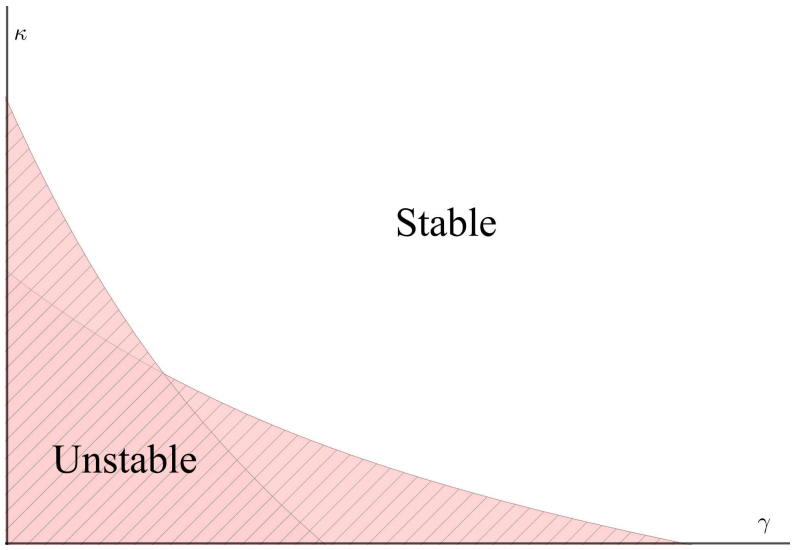


Figure 5.1: *Two critical hyperbolae determining the stability region.*

Let us consider the (γ, κ) plane: each inequality of (5.20) defines a region of stability in the plane, limited by a hyperbola. More precisely, in order to obtain stability, (γ, κ) must lie beyond a set of *critical hyperbolae*, where the i-th hyperbola depends only on μ_i. Figure 5.1 shows the stable and unstable regions of the plane that such inequalities produce.

Theorem 5.1 generalizes the stability conditions presented in the related literature [86], [101] and [102] that consider either $\gamma = 0$ or $\kappa = 0$. Indeed, if $\gamma = 0$ (no relative feedback on velocity control), (5.20) becomes

$$\kappa > \frac{|\beta_i|}{\sqrt{\alpha_i}},$$

as found by [86]. Moreover, if $\kappa = 0$ (no absolute feedback on velocity control), we get

$$\gamma > \sqrt{\frac{1}{\alpha_i}\frac{\beta_i^2}{\alpha_i^2 + \beta_i^2}},$$

aligned with [101] and [102].

5.3 Eigenvalue allocation

In this section the eigenvalues of matrix $\boldsymbol{F}$ are optimized by choosing parameters γ, κ, and η in order to maximize the convergence speed and avoid large oscillations. The hypothesis I make is to consider a digraph $\mathcal{G}$ with real eigenvalues. This applies to all symmetric digraphs and digraphs with symmetric strongly connected components communicating like Example 3.1. In this case, $\mu_i \in \mathbb{R}$, for all $i = 0, ..., n-1$. Without loss of generality, the eigenvalues $\{\mu_i\}_{i \in [\![0, n-1]\!]}$ are in non-decreasing order.

The objectives of the eigenvalue allocation problem are:

1. To avoid large oscillations and speed up the convergence by selecting a real dominant eigenvalue and maximizing its absolute value;

2. Allocating the non-dominant eigenvalues as far away as possible from the imaginary axis.

Let $\Lambda(\gamma,\kappa)$ denote the spectrum of $\boldsymbol{F}$, that is, the set of its eigenvalues, obtained with the given values of γ and κ. The eigenvalues are also functions of γ and κ, therefore we can write $\lambda(\gamma,\kappa)\in\Lambda(\gamma,\kappa)$. I define the following function:

$$f_1(\gamma,\kappa)=\min\left\{-\Re\left(\lambda(\gamma,\kappa)\right)\ :\ \lambda(\gamma,\kappa)\in\Lambda(\gamma,\kappa)\right\}.$$

Moreover, let $\bar{\Lambda}(\gamma,\kappa)$ denote the set of eigenvalues with minimum absolute real part for given values of γ and κ

$$\bar{\Lambda}(\gamma,\kappa)=\{\lambda(\gamma,\kappa)\in\Lambda(\gamma,\kappa)\ :\ -\Re\left(\lambda(\gamma,\kappa)\right)=f_1(\gamma,\kappa)\}$$

and let $\Lambda'(\gamma,\kappa)=\Lambda(\gamma,\kappa)\setminus\bar{\Lambda}(\gamma,\kappa)$ be the set of the remaining eigenvalues.
Now, a second function is defined as follows:

$$f_2(\gamma,\kappa)=\min\left\{-\Re\left(\lambda(\gamma,\kappa)\right)\ :\ \lambda(\gamma,\kappa)\in\Lambda(\gamma,\kappa)\right\}.$$

Problems 1 and 2 can be formally defined by the following optimization problems P1 and P2, respectively:

P1): $\max_{\gamma,\kappa\in\mathbb{R}_+} f_1(\gamma,\kappa)$, s.t. $\Im\left(\lambda(\gamma,\kappa)\right)=0$;

P2): $\max_{\gamma,\kappa\in\mathbb{R}_+} f_2(\gamma,\kappa)$.

The following theorem provides a solution to problems P1 and P2 in closed form.

Theorem 5.2. *Consider a set of agents that communicate in a network topology described by a symmetric digraph $\mathcal{G}$ that has a directed spanning tree. Then the eigenvalues of the controlled system (5.16) that solve P1 and P2 are obtained by the following values of γ, κ, and η:*

$$\bar{\gamma}=\frac{2\sqrt{\mu_1}}{\mu_1+\mu_2},\quad \bar{\kappa}=\frac{2\mu_2\sqrt{\mu_1}}{\mu_1+\mu_2},\tag{5.23}$$

and $\bar{\eta}>\frac{1}{\sqrt{\mu_1}}$.

Proof. Firstly, the eigenvalues of $\boldsymbol{F}_{2,2}$, provided by (5.18) depend on η, but not on γ and κ. Therefore, they can be allocated after optimizing γ and κ. Given this fact, I initially do not consider them in the computation of f_1 and f_2.

Let us consider the eigenvalues of the $\boldsymbol{F}_{1,1}$ block, whose expression is given by (5.12). Firstly, note that for each pair $(\lambda_{i,+},\lambda_{i,-})$, their real part is maximized in absolute value when they are real and coincident ($\kappa+\gamma\mu_i-2\sqrt{\mu_i}=0$), therefore

$$\max_{\gamma,\kappa\in\mathbb{R}}\left\{-\Re\left(\lambda_{i,+}\right),-\Re\left(\lambda_{i,-}\right)\right\}=\sqrt{\mu_i}.$$

Next, to solve P1, the (γ,κ)-plane, for $\gamma,\kappa\in\mathbb{R}_+$, is divided into four regions (A, B, C, and D) by the following straight line, as shown in Figure 5.2:

$$\mu_1\gamma+\kappa-2\sqrt{\mu_1}=0.\tag{5.24}$$

Then, I determine or bound the value of the objective function f_1 in each region.

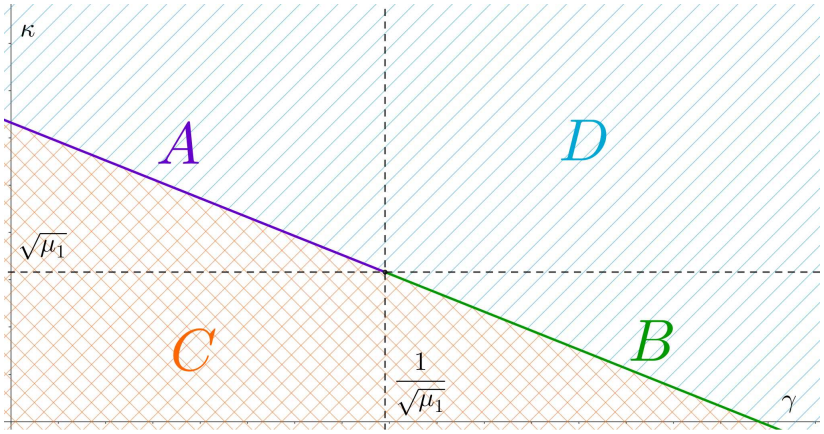


Figure 5.2: *The (γ, κ)-plane divided into regions A, B, C, and D.*

Region A is the segment defined by (5.24) and

$$0 < \gamma \leq \frac{1}{\sqrt{\mu_1}} \iff \sqrt{\mu_1} \leq \kappa < 2\sqrt{\mu_1}. \tag{5.25}$$

In this region, $\lambda_{1,+} = \lambda_{1,-} = -\sqrt{\mu_1}$, whereas, by substituting (5.24) into (5.12), we get

$$\lambda_{i,\pm} = -\sqrt{\mu_1} - \frac{\gamma(\mu_1 - \mu_i)}{2} \pm \frac{\sqrt{\gamma^2(\mu_i - \mu_1)^2 + 4(\mu_i - \mu_1)(\gamma\sqrt{\mu_1} - 1)}}{2},$$

for $i = 2, ..., n-1$. Imposing (5.25), it holds $-\Re(\lambda_{i,+}) = -\Re(\lambda_{i,-}) \geq \sqrt{\mu_1}$, $-\Re(\lambda_0) \geq \sqrt{\mu_1}$. Therefore, in region A the value of the objective function is $f_1(\gamma, \kappa) = \sqrt{\mu_1}$.

Region B is the segment defined by (5.24) and

$$\frac{1}{\sqrt{\mu_1}} < \gamma < \frac{2}{\sqrt{\mu_1}} \iff 0 \leq \kappa < \sqrt{\mu_1}.$$

It is immediate that $-\Re(\lambda_0) = \kappa < \sqrt{\mu_1}$, hence in this region $f_1(\gamma, \kappa) < \sqrt{\mu_1}$.

Region C is defined by $\mu_1\gamma + \kappa - 2\sqrt{\mu_1} < 0$. In this case, $\lambda_{1,\pm}$ are complex conjugates and it holds

$$-\Re(\lambda_{1,\pm}) = \frac{\mu_1\gamma + \kappa}{2} < \sqrt{\mu_1}.$$

As a consequence, $f_1(\gamma, \kappa) < \sqrt{\mu_1}$ in this region.

Region D is defined by $\mu_1\gamma + \kappa - 2\sqrt{\mu_1} > 0$. In this case, $\lambda_{i,+}$ and $\lambda_{i,-}$ are both real and it holds

$$-\lambda_{1,+} = \frac{\kappa + \gamma\mu_1 - \sqrt{(\kappa + \gamma\mu_1)^2 - 4\mu_1}}{2} < \sqrt{\mu_1}.$$

I conclude that P1 is solved when (γ, κ) is in region A defined by (5.24) and (5.25), and $\max_{\gamma,\kappa \in \mathbb{R}_+} f_1(\gamma, \kappa) = \sqrt{\mu_1}$.

To solve P2, I note that the eigenvalues $\lambda_{2,\pm}$ associated with μ_2 are vertically aligned with $\lambda_0 = -\kappa$ when

$$\Re\left(\frac{-(\kappa+\gamma\mu_2)\pm\sqrt{(\kappa+\gamma\mu_2)^2-4\mu_2}}{2}\right)=-\kappa,$$

which holds when

$$\kappa=\gamma\mu_2. \tag{5.26}$$

By combining (5.26) and (5.26), we obtain (5.23). Now, I prove that $\bar{\gamma}$ and $\bar{\kappa}$ solve both P1 and P2. If (5.26) holds, there are 3 possible cases to consider: $\kappa=\bar{\kappa}$, $\kappa<\bar{\kappa}$, and $\kappa>\bar{\kappa}$.

Let us consider case 1: as just shown, $-\Re(\lambda_{2,\pm})=-\kappa=-\bar{\kappa}$. The other eigenvalues are

$$\lambda_{i\pm}=\frac{\bar{\kappa}}{2}\left(-\left(1+\frac{\mu_i}{\mu_2}\right)\pm\sqrt{\left(1+\frac{\mu_i}{\mu_2}\right)^2-\frac{\mu_i(\mu_1+\mu_2)^2}{\mu_1\mu_2^2}}\right), \tag{5.27}$$

for $i=3,...,n-1$. Let us verify that $-\Re(\lambda_{i,\pm})>\bar{\kappa}$. If $\lambda_{i,\pm}$ are complex conjugates or real coincident, we have

$$-\Re(\lambda_{i,\pm})=\frac{\bar{\kappa}}{2}\left(1+\frac{\mu_i}{\mu_2}\right)=\frac{\bar{\kappa}}{2}\left(2+\frac{\mu_i-\mu_2}{\mu_2}\right)>\bar{\kappa},$$

because $\mu_i-\mu_2>0$. If $\lambda_{i,\pm}$ are real distinct, we only have to check that $-\Re(\lambda_{i,+})>\bar{\kappa}$, since $-\Re(\lambda_{i,-})>\bar{\kappa}$ is trivially verified. Since the inequality

$$(\mu_i-\mu_2)-\sqrt{(\mu_i-\mu_2)^2-\frac{\mu_i}{\mu_1}(\mu_2-\mu_1)^2}>0$$

holds, we have that (5.27) implies $-\Re(\lambda_{i,+})>\bar{\kappa}$. Thus, in case 1 we have $f_2(\gamma,\kappa)=\bar{\kappa}$.

In case 2, since $\lambda_0=\kappa<\bar{\kappa}$, we have $f_2(\gamma,\kappa)<\bar{\kappa}$.

Lastly, in case 3 it is verified that $\lambda_{2,\pm}$ are complex conjugates. If we compute the derivative of $-\Re(\lambda_{2,\pm})$ with respect to κ, we obtain

$$\frac{\partial\left(-\Re(\lambda_{2,\pm})\right)}{\partial\kappa}=\frac{1}{2}\left(1+\mu_2\frac{\partial\gamma}{\partial\kappa}\right)=\frac{1}{2}\left(1-\frac{\mu_2}{\mu_1}\right)<0.$$

Since the derivative is negative for all $\kappa<\bar{\kappa}$, we conclude that $-\Re(\lambda_{2,\pm})<\bar{\kappa}$, hence also $f_2(\gamma,\kappa)<\bar{\kappa}$.

The last thing to prove is that $(\bar{\gamma},\bar{\kappa})$ are in region A and solve P1. (5.25) is satisfied when $0<\sqrt{\mu_1}\gamma\leq 1$. When $\gamma=\bar{\gamma}$, we have

$$\sqrt{\mu_1}\gamma=\frac{2\mu_1}{\mu_1+\mu_2}=\frac{2}{1+\frac{\mu_2}{\mu_1}}.$$

Obviously, $\bar{\gamma}>0$. Since $\mu_2>\mu 1$, $\mu_2/\mu_1>1$, hence $\sqrt{\mu_1}\gamma<1$ is verified.

Finally, I place the eigenvalues of $\boldsymbol{F}_{2,2}$ so that they are farther away from the imaginary axis than the dominant eigenvalues $\lambda_{i,\pm}=-\sqrt{\mu_1}$. This is simple, as it is sufficient to impose $-\bar{\eta}\mu_i>-\sqrt{\mu_1}$ for all $i=1,...,n-1$, i.e.,

$$\bar{\eta}>\frac{1}{\sqrt{\mu_1}}.$$

□

The reason why f_1 attains a maximum when $\lambda_{1,+} = \lambda_{1,-} = -\sqrt{\mu_1}$ is that these eigenvalues of $\boldsymbol{F}$ are associated with the smallest non-null eigenvalue μ_1 of $\boldsymbol{L}$, therefore they are more critical than the others, meaning that they are more subject to getting closer to the imaginary axis when γ and κ vary. This is also the rationale behind the specific subdivision of the plane into 4 regions determined by (5.24).

5.4 Communication delays

The practical application of a consensus protocol needs information flow among the agents, which may be accompanied by a communication time delay. This section aims to study consensus protocol (5.10) in the presence of communication delay. In particular, a necessary and sufficient condition that guarantees the consensus achievement under constant communication time delay is proved and an explicit expression on how large the delay can be to guarantee consensus is obtained; then, a modified control law to ensure stability with an arbitrary constant time delay is proposed.

I assume that each agent compares its position and speed values with the respective values of the other agents with a delay τ. The problem is determining the maximum admissible value of τ, that is, the critical value τ_c so that if $\tau > \tau_c$, then the system is unstable. To account for this delay in the model, each term of (5.15) that is multiplied by $\boldsymbol{L}$ is assumed delayed by τ seconds:

$$\begin{cases} \dot{\boldsymbol{p}}(t) = \boldsymbol{H}\boldsymbol{q}(t) + \boldsymbol{r}(t), \\ \dot{\boldsymbol{q}}(t) = -\boldsymbol{L}\boldsymbol{H}^{+}\boldsymbol{p}(t-\tau) - \gamma\boldsymbol{L}\boldsymbol{q}(t-\tau) - \kappa\boldsymbol{q}(t) + \eta\boldsymbol{L}\boldsymbol{H}^{+}\boldsymbol{r}(t-\tau), \\ \dot{\boldsymbol{r}}(t) = -\eta\boldsymbol{H}\boldsymbol{L}\boldsymbol{H}^{+}\boldsymbol{r}(t-\tau), \end{cases}$$

and using the Laplace transform we obtain

$$\begin{cases} s\tilde{\boldsymbol{p}}(s) = \boldsymbol{H}\tilde{\boldsymbol{q}}(s) + \tilde{\boldsymbol{r}}(s), \\ s\tilde{\boldsymbol{q}}(s) = -\boldsymbol{L}\boldsymbol{H}^{+}e^{-\tau s}\tilde{\boldsymbol{p}}(s) - \gamma\boldsymbol{L}e^{-\tau s}\tilde{\boldsymbol{q}}(s) - \kappa\tilde{\boldsymbol{q}}(s) + \eta\boldsymbol{L}\boldsymbol{H}^{+}e^{-\tau s}\tilde{\boldsymbol{r}}(s), \\ s\tilde{\boldsymbol{r}}(s) = -\eta\boldsymbol{H}\boldsymbol{L}\boldsymbol{H}^{+}e^{-\tau s}\tilde{\boldsymbol{r}}(s). \end{cases} \tag{5.28}$$

Let us introduce the delayed Laplacian matrix $\boldsymbol{L}_\tau(s) = \boldsymbol{L}e^{-\tau s}$ and its eigenvalues are $\mu_i e^{-\tau s}$, for $i = 0, ..., n-1$. This matrix encodes both communication topology and communication delay. (5.28) can be represented compactly by

$$\underbrace{\begin{bmatrix} s\boldsymbol{I}_{n-1} & -\boldsymbol{H} & \boldsymbol{I}_{n-1} \\ \boldsymbol{L}_\tau(s)\boldsymbol{H}^{+} & \gamma\boldsymbol{L}_\tau(s) + (s+\kappa)\boldsymbol{I}_n & -\eta\boldsymbol{L}_\tau(s)\boldsymbol{H}^{+} \\ \boldsymbol{O}_{n-1} & \boldsymbol{O}_{n-1,n} & s\boldsymbol{I}_{n-1} + \eta\boldsymbol{H}\boldsymbol{L}_\tau(s)\boldsymbol{H}^{+} \end{bmatrix}}_{\boldsymbol{\Omega}(s)} \begin{bmatrix} \tilde{\boldsymbol{p}}(s) \\ \tilde{\boldsymbol{q}}(s) \\ \tilde{\boldsymbol{r}}(s) \end{bmatrix} = \begin{bmatrix} \boldsymbol{0}_{n-1} \\ \boldsymbol{0}_n \\ \boldsymbol{0}_{n-1} \end{bmatrix} \tag{5.29}$$

This equation resembles (5.16) and can be obtained from it by taking a Laplace transform and replacing $\boldsymbol{L}$ with $\boldsymbol{L}_\tau(s)$. The poles of the system are the values of s that make $\boldsymbol{\Omega}(s)$ singular, i.e., the solutions to $|\boldsymbol{\Omega}(s)| = 0$.

Lemma 5.4. *The poles of $\boldsymbol{\Omega}(s)$ are the solutions to*

$$(s+\kappa)\prod_{i=1}^{n-1}\left(s+\eta\mu_i e^{-\tau s}\right)\left(s^2+\left(\kappa+\gamma\mu_i e^{-\tau s}\right)s+\mu_i e^{-\tau s}\right)=0. \tag{5.30}$$

The proof is omitted, since it is identical to the proof of Lemma 5.3, except for the fact that λ is replaced by s and μ_i is replaced by $\mu_i e^{-\tau s}$.

Now, the following theorem determines the maximum value that the delay can assume to guarantee the dynamics of the multi-agent system (5.29).

Theorem 5.3. *System (5.29) is asymptotically stable if $\mathcal{G}$ has a directed spanning tree, (5.20) holds and*

$$0\leq\tau<\tau_c=\min\left(\bar{\tau}_\eta,\bar{\tau}_{\gamma\kappa}\right),$$

with

$$\bar{\tau}_\eta=\min_{i\in[\![0,n-1]\!]}\frac{1}{\eta\left|\mu_i\right|}\arctan\left(-\frac{\alpha_i}{\beta_i}\right)$$

$$\bar{\tau}_{\gamma,\kappa}=\min_{i\in[\![0,n-1]\!]}\frac{1}{\bar{\omega}_i}\arctan\left(\frac{\left(\alpha_i-\beta_i\gamma\bar{\omega}_i\right)\kappa-\left(\beta_i+\alpha_i\gamma\bar{\omega}_i\right)\bar{\omega}_i}{\left(\beta_i+\alpha_i\gamma\bar{\omega}_i\right)\kappa+\left(\alpha_i-\beta_i\gamma\bar{\omega}_i\right)\bar{\omega}_i}\right),$$

and

$$\bar{\omega}_i=\sqrt{\frac{\left(\gamma^2\left|\mu_i\right|^2-\kappa^2\right)+\sqrt{\left(\gamma^2\left|\mu_i\right|^2-\kappa^2\right)^2+4\left|\mu_i\right|^4}}{2}}.$$

Proof. Let us consider the factor of (5.30) given by

$$s+\eta\mu_i e^{-\tau s}=0, \tag{5.31}$$

for some i. This is a pseudo-polynomial of order 1 and to determine whether its all roots have negative real part, I first find the conditions under which it admits a pair of imaginary roots $s=\pm\jmath\omega$, with $\omega>0$. Let us consider only one of the two roots (the same applies to the other one); by setting $s=\jmath\omega$, in (5.31), we obtain

$$\jmath\omega+\eta\mu_i e^{-\jmath\omega\tau}=0. \tag{5.32}$$

Let us evaluate $\mu_i e^{-\jmath\omega\tau}$:

$$\begin{aligned}\mu_i e^{-\jmath\omega\tau}&=\left(\alpha_i+\jmath\beta_i\right)\left(\cos\left(\omega\tau\right)-\jmath\sin\left(\omega\tau\right)\right)=\\&=\left(\alpha_i\cos\left(\omega\tau\right)+\beta_i\sin\left(\omega\tau\right)\right)+\jmath\left(\beta_i\cos\left(\omega\tau\right)-\alpha_i\sin\left(\omega\tau\right)\right).\end{aligned}$$

With this, (5.32) becomes

$$\jmath\omega+\eta\left(\alpha_i\cos\left(\omega\tau\right)+\beta_i\sin\left(\omega\tau\right)\right)+\jmath\eta\left(\beta_i\cos\left(\omega\tau\right)-\alpha_i\sin\left(\omega\tau\right)\right)=0.$$

This equation can be split into two equations by considering the real and imaginary parts of each member:

$$\begin{cases} \eta\left(\alpha_i \cos\left(\omega\tau\right) + \beta_i \sin\left(\omega\tau\right)\right) = 0, \\ \eta\left(\beta_i \cos\left(\omega\tau\right) - \alpha_i \sin\left(\omega\tau\right)\right) = -\omega. \end{cases} \tag{5.33}$$

Squaring these two equations and adding them up yields

$$\omega^2 = \eta^2(\alpha_i^2 + \beta_i^2),$$

whose positive solution is $\omega = \eta\left|\mu_i\right|$. The smallest positive value of τ that satisfy the first equation of (5.33) is

$$\bar{\tau} = \frac{1}{\eta\left|\mu_i\right|} \arctan\left(-\frac{\alpha_i}{\beta_i}\right).$$

Since the delay-free case ($\tau = 0$) was proven in Theorem 5.1, by continuity [125], the stability of (5.31) is guaranteed for all values of $\tau < \bar{\tau}$. This condition must hold for all $i = 1, ..., n-1$, that is,

$$\tau < \tau_\eta = \min_{i \in [\![0, n-1]\!]} \frac{1}{\eta\left|\mu_i\right|} \arctan\left(-\frac{\alpha_i}{\beta_i}\right). \tag{5.34}$$

Now, let us take the other kind of factors of (5.30), namely,

$$s^2 + \left(\kappa + \gamma\mu_i e^{-\tau s}\right) s + \mu_i e^{-\tau s} = 0. \tag{5.35}$$

This time, we have a pseudo-polynomial of order 2. I repeat the same procedure, by letting $s = \jmath\omega$ and obtaining

$$-\omega^2 + \jmath\kappa\omega + \gamma\mu_i e^{-j\omega\tau}\jmath\omega + \mu_i e^{-\jmath\omega\tau} = 0,$$

that, when separating the real and imaginary parts, becomes:

$$\begin{cases} \left(\alpha_i - \beta_i\gamma\omega\right)\cos\left(\omega\tau\right) + \left(\beta_i + \alpha_i\gamma\omega\right)\sin\left(\omega\tau\right) = \omega^2, \\ \left(\beta_i + \alpha_i\gamma\omega\right)\cos\left(\omega\tau\right) - \left(\alpha_i - \beta_i\gamma\omega\right)\sin\left(\omega\tau\right) = -\kappa\omega. \end{cases} \tag{5.36}$$

Again, the sum of the squares of the two equations produces

$$\omega^4 + \left(\kappa^2 - \gamma^2\left|\mu_i\right|^2\right)\omega^2 - \left|\mu_i\right|^2 = 0,$$

whose only acceptable solution is denoted by $\bar{\omega}_i$:

$$\bar{\omega}_i = \sqrt{\frac{\left(\gamma^2\left|\mu_i\right|^2 - \kappa^2\right) + \sqrt{\left(\gamma^2\left|\mu_i\right|^2 - \kappa^2\right)^2 + 4\left|\mu_i\right|^4}}{2}}.$$

Solving system (5.36) for $\cos\left(\bar{\omega}_i\tau\right)$ and $\sin\left(\bar{\omega}_i\tau\right)$ yields

$$\tan\left(\bar{\omega}_i\tau\right) = \frac{\left(\alpha_i - \beta_i\gamma\bar{\omega}_i\right)\kappa - \left(\beta_i + \alpha_i\gamma\bar{\omega}_i\right)\bar{\omega}_i}{\left(\beta_i + \alpha_i\gamma\bar{\omega}_i\right)\kappa + \left(\alpha_i - \beta_i\gamma\bar{\omega}_i\right)\bar{\omega}_i}.$$

The smallest value of τ that satisfies this equation is

$$\bar{\tau} = \frac{1}{\bar{\omega}_i} \arctan\left(\frac{(\alpha_i - \beta_i\gamma\bar{\omega}_i)\,\kappa - (\beta_i + \alpha_i\gamma\bar{\omega}_i)\,\bar{\omega}_i}{(\beta_i + \alpha_i\gamma\bar{\omega}_i)\,\kappa + (\alpha_i - \beta_i\gamma\bar{\omega}_i)\,\bar{\omega}_i}\right)$$

and, by continuity [125], the stability of (5.35) is guaranteed for all values of $\tau < \bar{\tau}$. This condition must hold for all $i = 1, ..., n-1$, that is,

$$\tau < \bar{\tau}_{\gamma,\kappa} = \min_{i\in[\![0,n-1]\!]} \frac{1}{\bar{\omega}_i} \arctan\left(\frac{(\alpha_i - \beta_i\gamma\bar{\omega}_i)\,\kappa - (\beta_i + \alpha_i\gamma\bar{\omega}_i)\,\bar{\omega}_i}{(\beta_i + \alpha_i\gamma\bar{\omega}_i)\,\kappa + (\alpha_i - \beta_i\gamma\bar{\omega}_i)\,\bar{\omega}_i}\right). \tag{5.37}$$

Combining (5.34) and (5.37) completes the proof.

□

5.5 Simulations and numerical results

In this section, I provide numerical results and comparisons to show the performance of the proposed consensus protocols. Two scenarios are tested: a delay-free system and a system where communication delays are present.

5.5.1 Delay-free scenario

I assume that the communication topology is described by a graph $\mathcal{G}_C$, shown in Figure 5.3, that is characteristic of a group of agents that have to move in a chain and do not include a leader.

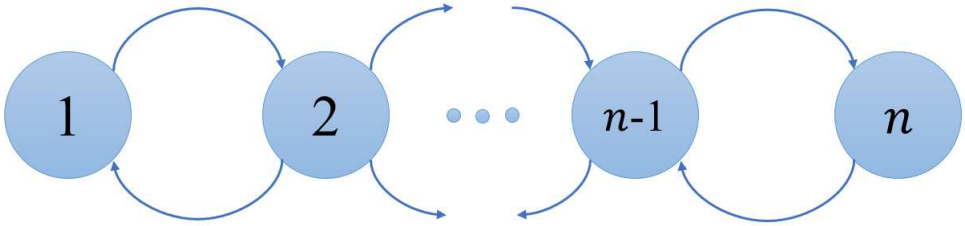


Figure 5.3: *Chain communication topology corresponding to the digraph $\mathcal{G}_C$.*

By Theorem 4 of [126], the real eigenvalues of the corresponding Laplacian matrix are the following:

$$\mu_i = 2\left(1 - \cos\left(\frac{i\pi}{n}\right)\right), \quad \text{for } i = 0, ..., n-1.$$

The tested scenario consists of $n = 8$ agents, e.g., robots or AGVs in industrial environments. The agents queue up to each other with the initial conditions reported in Table 5.1 that shows the initial inter-agent distance vector $\boldsymbol{d}(0)$ (in meters), the initial velocities $\boldsymbol{v}(0)$ (in m/s), the initial reference velocities $\boldsymbol{y}(0)$ (in m/s, selected higher than the initial velocities), the fixed distance $\bar{d}$ (in meters) and the values of the parameters $\gamma = \bar{\gamma}$ and $\kappa = \bar{\kappa}$ and chosen according to (5.23), with $\eta = 10/\sqrt{\mu_1}$.

The simulation outputs are reported in Figure 5.4 and 5.5, showing the position and velocity trends over time, respectively.

Parameters	UM	Values
$\bar{\gamma}$	unit	1.0574
$\bar{\kappa}$	unit	0.6194
$\bar{\eta}$	unit	25.62
$\bar{d}$	m	4
$\boldsymbol{d}(0)$	m	$[4.26, 1.44, 5.40, 2.73, 2.11, 2.85, 2.32]^\mathsf{T}$
$\boldsymbol{v}(0)$	m/s	$[0.47, 0.10, 1.58, 0.70, 1.35, 1.74, 1.25, 1.34]^\mathsf{T}$
$\boldsymbol{y}(0)$	m/s	$[1.94, 1.62, 1.99, 2.35, 2.37, 1.77, 1.71, 2.07]^\mathsf{T}$

Table 5.1: *Simulation values used in the delay-free scenario.*

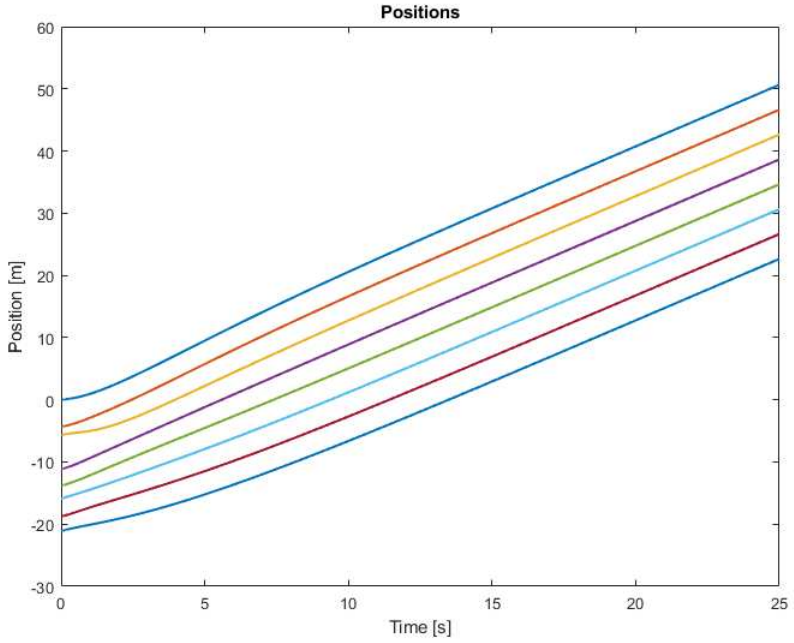


Figure 5.4: *Positions over time for the network topology $\mathcal{G}_C$.*

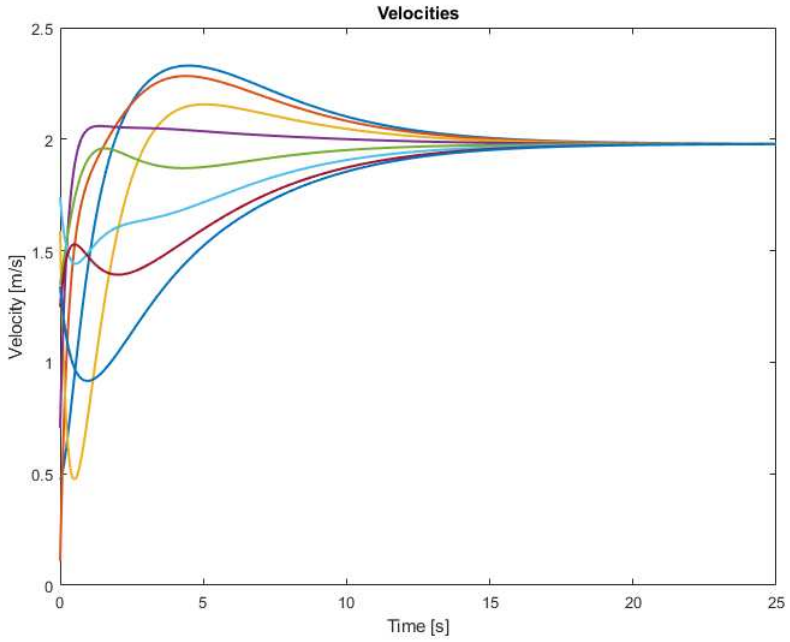


Figure 5.5: *Velocities over time for the network topology $\mathcal{G}_C$.*

Now, I consider the same protocol applied to a different communication topology. The new setup is shown in Figure 5.6, where two connections of $\mathcal{G}_C$ are dropped, obtaining a new graph, $\mathcal{G}_{C'}$.

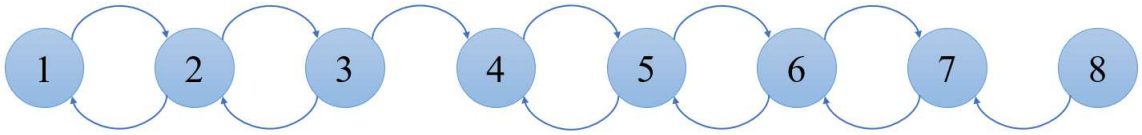


Figure 5.6: *Representation of $\mathcal{G}_{C'}$.*

It is apparent that has a directed spanning tree and by analyzing the structure of its Laplacian matrix, its eigenvalues are real. The results of the simulation are reported in Figure 5.7 and 5.8, showing how the agents reach the convergence values of positions and velocities, respectively.

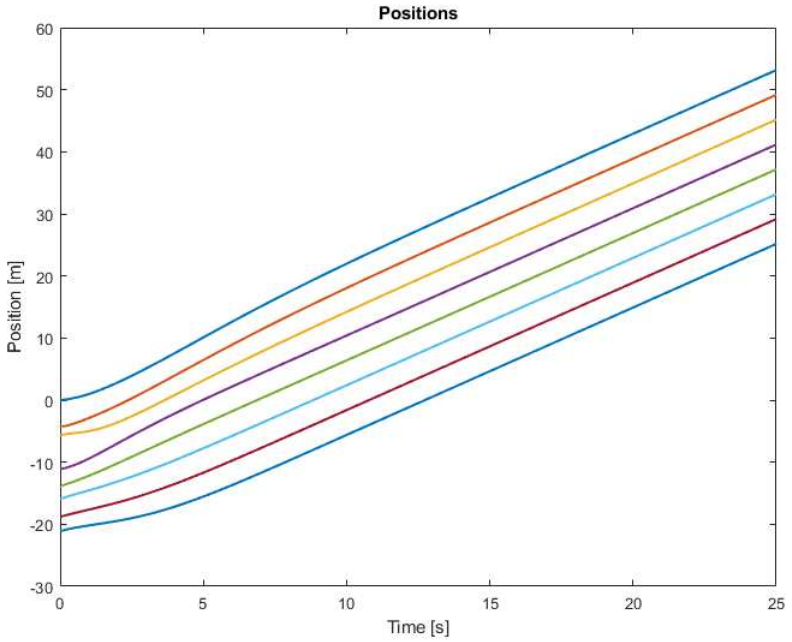


Figure 5.7: *Positions over time for the network topology* $\mathcal{G}_{C'}$.

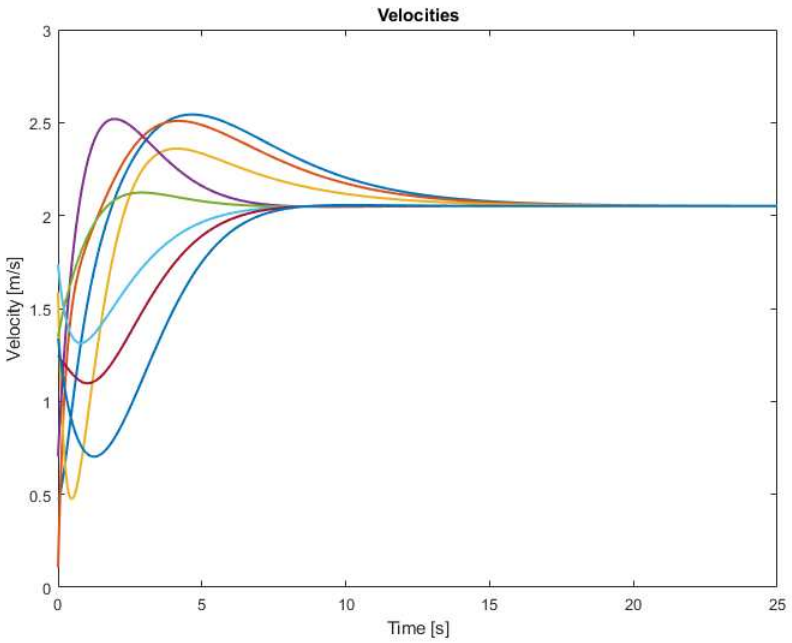


Figure 5.8: *Velocities over time for the network topology* $\mathcal{G}_{C'}$.

Now, I compare the proposed protocol with similar methods presented in the related literature. In particular, I consider the protocols proposed in [101] and [102] that are applied to undirected graph topologies characterized by Laplacian matrices with real non-negative eigenvalues:

$$u_i = -\sum_{i=1}^{n}(x_i - x_j) - \gamma_1 \sum_{i=1}^{n}(v_i - v_j). \tag{5.38}$$

Note that by choosing the parameter values according to (5.23), the dominant eigenvalues of the controlled system (5.11) determine an exponential decay of type $te^{-\sqrt{\mu_1}t}$, so the convergence speed is said to be $\sqrt{\mu_1}$. The maximum convergence speed of (5.38) is

$$\sqrt{\frac{\mu_1\mu_{n-1}}{2\mu_{n-1}\mu_1}} = \sqrt{\mu_1}\sqrt{\frac{\mu_{n-1}}{\mu_{n-1} + (\mu_{n-1} - \mu_1)}} < \sqrt{\mu_1}.$$

It is apparent that by introducing the new parameter κ, the proposed protocol can reach a greater convergence speed than protocol (5.38). Let us compare the two algorithms by some numerical results. I test some scenarios where agents communicate by the network topology $\mathcal{G}$ shown in Figure 5.3. The initial conditions of protocol (5.10) are selected randomly using uniform distributions with the same mean and variance of the data that Table 5.1 reports. Moreover, the values of $\bar{d}$, γ, κ, and η are shown in Table 5.1. In addition, for the comparison algorithm (5.10) is adapted to algorithm (5.38) by imposing $\bar{d} = 0$ and $\boldsymbol{v}(0) = \boldsymbol{y}(0)$. Furthermore, I set $\gamma_1 = 3.6609$ in (5.38). The comparison is performed by defining the following scalar function:

$$V(t) = \left\| \boldsymbol{v}(t) - n^{-1}\mathbf{1}_n\mathbf{1}_n^{\mathsf{T}}\boldsymbol{v}(0) \right\|. \tag{5.39}$$

$n^{-1}\mathbf{1}_n^{\mathsf{T}}\boldsymbol{v}(0)$ is the average of the initial values of the speeds and, because of the symmetry of $\mathcal{G}$, i.e., $\boldsymbol{L} = \boldsymbol{L}^{\mathsf{T}}$, it is also the convergence value of $\boldsymbol{v}(t)$. I use as $t_{0.5\%}$ as performance index, defined by

$$V(t) \geq 0.005V(0) \quad \forall t \geq t_{0.5\%}.$$

I ran 1000 cases for each of the two algorithms and averaged out the results. By applying protocol (5.10), I obtained

$$t_{0.5\%} = 27.05\,[\text{s}],$$

whereas, by applying (5.38) the resulting performance index is

$$t_{0.5\%} = 29.82\,[\text{s}].$$

The proposed protocol with respect to the similar protocol (5.38) offers a better convergence speed and the possibility of reaching a consensus value of the velocities, independently of the initial values of the velocities.

5.5.2 Delayed communication scenario

I consider a system composed of 8 agents with the communication topology shown in Figure 5.6. The initial conditions and the parameter values used for this simulation are listed in Table 5.2. The critical value of the delay is $\tau_c = 0.3727$ and the simulation used $\tau = \tau_c/2 = 0.1864$.

The simulation outputs for the delayed system is reported in Figure 5.9.

Parameters	UM	Values
$\bar{\gamma}$	unit	1.0461
$\bar{\kappa}$	unit	0.2072
$\bar{\eta}$	unit	1.6649
$\bar{d}$	m	4
τ	s	0.1864
τ_c	s	0.3727
$\boldsymbol{d}(0)$	m	$[1.54, 4.49, 3.32, 3.20, 1.58, 4.41, 3.49]^\mathsf{T}$
$\boldsymbol{v}(0)$	m/s	$[1.27, 0.51, 1.59, 0.70, 1.35, 1.74, 1.25, 1.34]^\mathsf{T}$
$\boldsymbol{y}(0)$	m/s	$[1.85, 2.01, 1.90, 1.58, 1.74, 1.62, 1.68, 1.74]^\mathsf{T}$

Table 5.2: *Simulation values used in delayed communication scenario.*

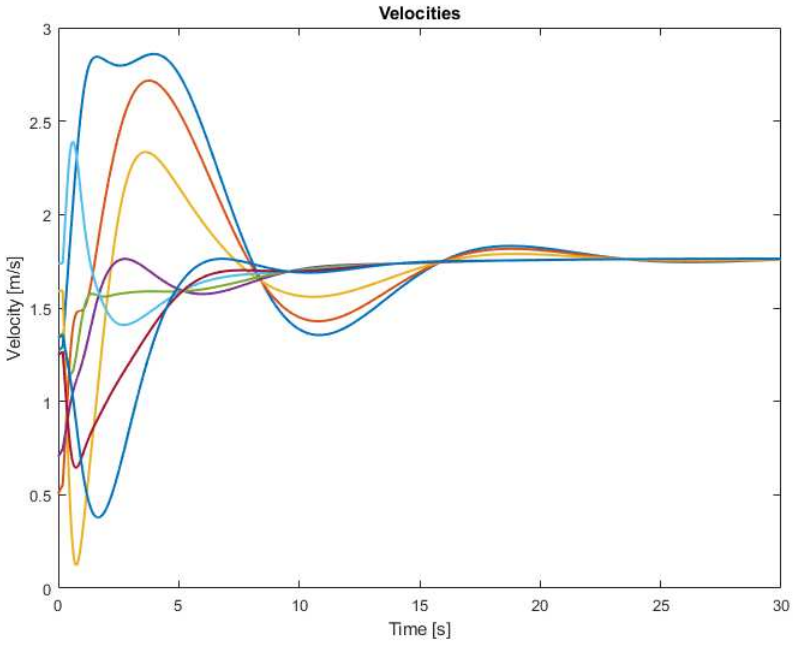


Figure 5.9: *Velocities over time for the network topology $\mathcal{G}_C$ with delay.*

In order to compare the convergence speed in the absence and in the presence of time delays, another simulation campaign is carried out. In particular, (5.39) is used again to evaluate the performance. The value of the $t_{0.5\%}$ index that was obtained after 1000 simulations with random initial conditions is

$$t_{0.5\%} = 40.11\,[\text{s}].$$

The simulation results show that the introduction of the delay slows down the convergence in the two communication topologies.

Lastly, I confirm the correctness of the computed critical delay value by running a simulation with $\tau = \tau_c$, obtaining an undamped oscillatory speed profile, depicted in Figure 5.10.

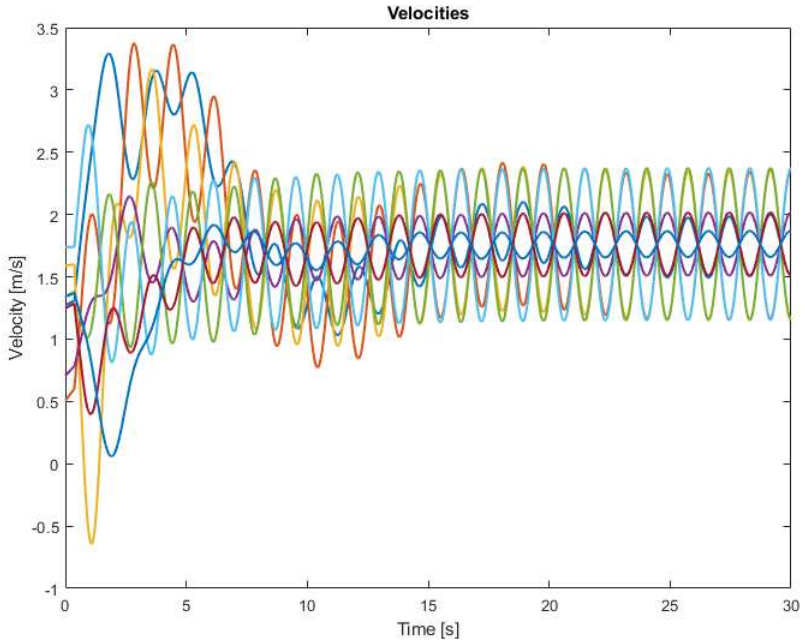


Figure 5.10: *Velocities over time for the network topology $\mathcal{G}_C$ with critical delay.*

Chapter 6

Conclusions

In this study, a novel framework for modeling autonomous intersections has been introduced, accompanied by new results in the realm of multi-agent systems. The proposed theoretical framework not only enhances the understanding of this complex problem but also establishes a robust mathematical foundation for addressing it. I believe the formulation of the Intersection Crossing Problem emerges as a promising perspective within the autonomous intersection management scenario.

The novelty of the proposed approach leaves ample room for future developments. Although self-consistent, the proposed theory is general enough to allow integration with existing state-of-the-art approaches, some of which dive deeper into practical issues that have not been fully considered as of today. Real-world scenarios are often far from theory: including nonidealities in the model and studying their impact on the given results is crucial for developing a framework that works in practice. The sensitivity to disturbances, modeling errors, and uncertainties is a broad topic that requires extensive analysis and grounded mathematical considerations, suggesting possibilities for future research in this domain. Furthermore, despite including safety margins in the vehicle occupancy model, the available approximations within the realm of collision detection could be further refined. In this work, the focus was directed towards a specific type of intersection termed α-intersection. Given the generality of the theory, I expect it could be easily extended to encompass a broader category of intersection types, potentially encompassing all variations: different shapes, angles, varying numbers of roads, and configurations featuring more than one lane per road. Since the proposed distributed algorithm relies on specific properties of the communication graph, such as connectedness, it is appropriate to incorporate the theory of dynamic graphs into the proposed model. This aims to define the requirements for the communication topology and protocols, especially when analyzing the propagation time of exchanged information. As for theory, there remain numerous aspects requiring further assessment. Certain concepts and results currently rest on conjectures that should be transformed into theorems via rigorous proofs. Future endeavors should be dedicated to addressing these fundamental issues, enriching the theoretical foundation of the proposed model. In Section 4.6, the ICP was converted from a functional optimization problem to a regular one. It would be interesting to solve the ICP as a functional optimization challenge, seeking a closed-form solution within a simplified scenario; perhaps with only two vehicles crossing the intersection. The proposed collision avoidance system and the distributed algorithm are based on the concept of a priority associated with each vehicle. Although it is very common in literature, it is a simplistic model. An intriguing avenue for future exploration involves incorporating more sophisticated methods,

such as game theory, into the collision avoidance and optimization framework. The application of game-theoretic approaches could enhance the overall effectiveness and adaptability of the collision avoidance system within the context of autonomous intersection management. From a practical standpoint, refining the computation strategy for the support function of the collision set could significantly reduce its computation time, making it more viable for real-time applications. Finally, to ensure the broad applicability of the proposed results, rigorous validation across an extensive range of scenarios must be undertaken. This validation step is essential for ensuring that the proposed model is not only theoretically sound but also practically reliable across a spectrum of real-world scenarios.

Within the context of multi-agent systems, this work considers a leaderless consensus protocol that a MAS can apply in order to reach a common velocity while forming a uniformly spaced string. The advantage of the method is that a leader is not required and by the optimized protocol parameters the fastest rate of convergence avoiding oscillations is guaranteed. However, the optimization of control parameter values depends on the communication graph topology: if the topology changes, the parameters must be updated by calculating the eigenvalues of the Laplacian matrix. Future research directions will focus on the assessment of the protocol in the presence of constraints on agent velocities and accelerations. Moreover, investigations about the impact on the stability and convergence of the delays of communication will be analyzed. To this aim, suitable conditions will be sought to guarantee correct behavior and good performance of the protocol.

Bibliography

[1] United Nations: Department of Economic and Social Affairs: Population Division, *World urbanization prospects 2018*. Population studies, New York, NY: United Nations, June 2020.

[2] Statista, "Number of cars sold worldwide from 2010 to 2022, with a 2023 forecast." `https://www.statista.com/statistics/200002/international-car-sales-since-1990/`, 2023. [Accessed: 05-02-2024].

[3] C. B. Rafter, B. Anvari, and S. Box, "Traffic responsive intersection control algorithm using gps data," in *2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC)*, pp. 1–6, 2017.

[4] D. Schrank, L. Albert, B. Eisele, and T. Lomax, "2021 urban mobility report," 2021.

[5] R. Azimi, G. Bhatia, R. R. Rajkumar, and P. Mudalige, "Stip: Spatio-temporal intersection protocols for autonomous vehicles," in *2014 ACM/IEEE International Conference on Cyber-Physical Systems (ICCPS)*, pp. 1–12, 2014.

[6] N. A. Stanton and P. M. Salmon, "Human error taxonomies applied to driving: A generic driver error taxonomy and its implications for intelligent transport systems," *Safety Science*, vol. 47, no. 2, pp. 227–237, 2009.

[7] A. A. Hassan and H. A. Rakha, "A fully-distributed heuristic algorithm for control of autonomous vehicle movements at isolated intersections," *International Journal of Transportation Science and Technology*, vol. 3, no. 4, pp. 297–309, 2014.

[8] J. Kolodko and L. Vlacic, "Cooperative autonomous driving at the intelligent control systems laboratory," *IEEE Intelligent Systems*, vol. 18, pp. 8–11, jul 2003.

[9] A. Festag, "Cooperative intelligent transport systems standards in europe," *IEEE Communications Magazine*, vol. 52, no. 12, pp. 166–172, 2014.

[10] Q. Chen, D. Jiang, and L. Delgrossi, "Ieee 1609.4 dsrc multi-channel operations and its implications on vehicle safety communications," in *2009 IEEE Vehicular Networking Conference (VNC)*, pp. 1–8, 2009.

[11] Y. Cao, W. Yu, W. Ren, and G. Chen, "An overview of recent progress in the study of distributed multi-agent coordination," *IEEE Transactions on Industrial Informatics*, vol. 9, no. 1, pp. 427–438, 2013.

[12] A. Gholamhosseinian and J. Seitz, "A comprehensive survey on cooperative intersection management for heterogeneous connected vehicles," *IEEE Access*, vol. 10, pp. 7937–7972, 2022.

[13] E. Namazi, J. Li, and C. Lu, "Intelligent intersection management systems considering autonomous vehicles: A systematic literature review," *IEEE Access*, vol. 7, pp. 91946–91965, 2019.

[14] M. Khayatian, M. Mehrabian, E. Andert, R. Dedinsky, S. Choudhary, Y. Lou, and A. Shrivastava, "A survey on intersection management of connected autonomous vehicles," *ACM Transactions on Cyber-Physical Systems*, vol. 4, pp. 1–27, 08 2020.

[15] L. Chen and C. Englund, "Cooperative intersection management: A survey," *IEEE Transactions on Intelligent Transportation Systems*, vol. 17, no. 2, pp. 570–586, 2016.

[16] K. Dresner and P. Stone, "A multiagent approach to autonomous intersection management.," *J. Artif. Intell. Res. (JAIR)*, vol. 31, pp. 591–656, 01 2008.

[17] D. Fajardo, T.-C. Au, S. T. Waller, P. Stone, and D. Yang, "Automated intersection control: Performance of future innovation versus current traffic signal control," *Transportation Research Record: Journal of the Transportation Research Board*, vol. 2259, p. 223–232, Jan. 2011.

[18] N. Shahidi, T.-C. Au, and P. Stone, "Batch reservations in autonomous intersection management (extended abstract)," pp. 1225–1226, 01 2011.

[19] X. Wei, G. Tan, and N. Ding, "Batch-light: An adaptive intelligent intersection control policy for autonomous vehicles," in *2014 IEEE International Conference on Progress in Informatics and Computing*, pp. 98–103, 2014.

[20] J. H. Dahlberg and V. Tuul, "Intelligent traffic intersection management using motion planning for autonomous vehicles," 2017.

[21] M. Vasirani and S. Ossowski, "A computational market for distributed control of urban road traffic systems," *IEEE Transactions on Intelligent Transportation Systems*, vol. 12, no. 2, pp. 313–321, 2011.

[22] M. Vasirani and S. Ossowski, "A market-inspired approach for intersection management in urban road traffic networks," 2014.

[23] P. Wang, S. Fang, L. Zhang, and J. Wang, "A vehicle collision detection algorithm at t-shaped intersections based on location-based service," pp. 308–317, 10 2015.

[24] K. Zhang, A. Yang, H. Su, A. de La Fortelle, K. Miao, and Y. Yao, "Service-oriented cooperation models and mechanisms for heterogeneous driverless vehicles at continuous static critical sections," *IEEE Transactions on Intelligent Transportation Systems*, vol. 18, no. 7, pp. 1867–1881, 2017.

[25] A. I. Morales Medina, F. Creemers, E. Lefeber, and N. van de Wouw, "Optimal access management for cooperative intersection control," *IEEE Transactions on Intelligent Transportation Systems*, vol. 21, no. 5, pp. 2114–2127, 2020.

[26] Y. Feng, C. Yu, and H. X. Liu, "Spatiotemporal intersection control in a connected and automated vehicle environment," *Transportation Research Part C: Emerging Technologies*, vol. 89, p. 364–383, Apr. 2018.

[27] Q. Lu and K.-D. Kim, "Intelligent intersection management of autonomous traffic using discrete-time occupancies trajectory," *Journal of Traffic and Logistics Engineering*, 2016.

[28] Q. Lu and K.-D. Kim, "A genetic algorithm approach for expedited crossing of emergency vehicles in connected and autonomous intersection traffic," *Journal of Advanced Transportation*, vol. 2017, p. 1–14, 2017.

[29] Q. Lu and K.-D. Kim, "A mixed integer programming approach for autonomous and connected intersection crossing traffic control," in *2018 IEEE 88th Vehicular Technology Conference (VTC-Fall)*, pp. 1–6, 2018.

[30] Y. Chang and P. Edara, "Arebic: Autonomous reservation-based intersection control for emergency evacuation," in *2017 IEEE Intelligent Vehicles Symposium (IV)*, IEEE, June 2017.

[31] B. Xu, X. J. Ban, Y. Bian, J. Wang, and K. Li, "V2i based cooperation between traffic signal and approaching automated vehicles," in *2017 IEEE Intelligent Vehicles Symposium (IV)*, pp. 1658–1664, 2017.

[32] X. Meng and C. G. Cassandras, "Optimal control of autonomous vehicles for non-stop signalized intersection crossing," in *2018 IEEE Conference on Decision and Control (CDC)*, pp. 6988–6993, 2018.

[33] M. Khayatian, Y. Lou, M. Mehrabian, and A. Shirvastava, "Crossroads+: A time-aware approach for intersection management of connected autonomous vehicles," *ACM Transactions on Cyber-Physical Systems*, vol. 4, p. 1–28, Nov. 2019.

[34] A. Katriniok, S. Kojchev, E. Lefeber, and H. Nijmeijer, "Distributed scenario model predictive control for driver aided intersection crossing," in *2018 European Control Conference (ECC)*, pp. 1746–1752, 2018.

[35] A. Katriniok, P. Kleibaum, C. Ress, and L. Eckstein, "Automation of road vehicles using v2x: An application to intersection automation," in *SAE Technical Paper Series*, ANNUAL, SAE International, Mar. 2017.

[36] B. Liu, Q. Shi, Z. Song, and A. El Kamel, "Trajectory planning for autonomous intersection management of connected vehicles," *Simulation Modelling Practice and Theory*, vol. 90, p. 16–30, Jan. 2019.

[37] R. Azimi, G. Bhatia, R. Rajkumar, and P. Mudalige, "Ballroom intersection protocol: Synchronous autonomous driving at intersections," in *2015 IEEE 21st International Conference on Embedded and Real-Time Computing Systems and Applications*, pp. 167–175, 2015.

[38] H. Kowshik, D. Caveney, and P. R. Kumar, "Provable systemwide safety in intelligent intersections," *IEEE Transactions on Vehicular Technology*, vol. 60, no. 3, pp. 804–818, 2011.

[39] Y. Wang, E. Wenjuan, D. Tian, G. Lu, G. Yu, and Y. Wang, "Vehicle collision warning system and collision detection algorithm based on vehicle infrastructure integration," in *7th Advanced Forum on Transportation of China (AFTC 2011)*, pp. 216–220, 2011.

[40] W. Wu, J. Zhang, A. Luo, and J. Cao, "Distributed mutual exclusion algorithms for intersection traffic control," *IEEE Transactions on Parallel and Distributed Systems*, vol. 26, no. 1, pp. 65–74, 2015.

[41] J. Khoury and J. Khoury, "Passive, decentralized, and fully autonomous intersection access control," in *17th International IEEE Conference on Intelligent Transportation Systems (ITSC)*, pp. 3028–3033, 2014.

[42] A. Bazzi, A. Zanella, and B. M. Masini, "A distributed virtual traffic light algorithm exploiting short range v2v communications," *Ad Hoc Networks*, vol. 49, pp. 42–57, 2016.

[43] Y. Jiang, M. Zanon, R. Hult, and B. Houska, "Distributed algorithm for optimal vehicle coordination at traffic intersections," *IFAC-PapersOnLine*, vol. 50, p. 11577–11582, July 2017.

[44] G. Rodrigues de Campos, P. Falcone, R. Hult, H. Wymeersch, and J. Sjöberg, "Traffic coordination at road intersections: Autonomous decision-making algorithms using model-based heuristics," *IEEE Intelligent Transportation Systems Magazine*, vol. 9, no. 1, pp. 8–21, 2017.

[45] L. Tu and C.-M. Huang, "Forwards: A map-free intersection collision-warning system for all road patterns," *IEEE Transactions on Vehicular Technology*, vol. 59, no. 7, pp. 3233–3248, 2010.

[46] J.-B. Tomas-Gabarron, E. Egea-Lopez, and J. Garcia-Haro, "Vehicular trajectory optimization for cooperative collision avoidance at high speeds," *IEEE Transactions on Intelligent Transportation Systems*, vol. 14, no. 4, pp. 1930–1941, 2013.

[47] G. Lu, L. Li, Y. Wang, R. Zhang, Z. Bao, and H. Chen, "A rule based control algorithm of connected vehicles in uncontrolled intersection," in *17th International IEEE Conference on Intelligent Transportation Systems (ITSC)*, pp. 115–120, 2014.

[48] Y. Rahmati and A. Talebpour, "Towards a collaborative connected, automated driving environment: A game theory based decision framework for unprotected left turn maneuvers," in *2017 IEEE Intelligent Vehicles Symposium (IV)*, pp. 1316–1321, 2017.

[49] N. Murgovski, G. R. de Campos, and J. Sjöberg, "Convex modeling of conflict resolution at traffic intersections," in *2015 54th IEEE Conference on Decision and Control (CDC)*, pp. 4708–4713, 2015.

[50] A. A. Malikopoulos and L. Zhao, "A closed-form analytical solution for optimal coordination of connected and automated vehicles," in *2019 American Control Conference (ACC)*, pp. 3599–3604, 2019.

[51] S. Joerer, M. Segata, B. Bloessl, R. Lo Cigno, C. Sommer, and F. Dressler, "A vehicular networking perspective on estimating vehicle collision probability at intersections," *IEEE Transactions on Vehicular Technology*, vol. 63, no. 4, pp. 1802–1812, 2014.

[52] G. R. de Campos, A. H. Runarsson, F. Granum, P. Falcone, and K. Alenljung, "Collision avoidance at intersections: A probabilistic threat-assessment and decision-making system for safety interventions," in *17th International IEEE Conference on Intelligent Transportation Systems (ITSC)*, pp. 649–654, 2014.

[53] A. Mirheli, M. Tajalli, L. Hajibabai, and A. Hajbabaie, "A consensus-based distributed trajectory control in a signal-free intersection," *Transportation Research Part C: Emerging Technologies*, vol. 100, p. 161–176, Mar. 2019.

[54] S. Azimi, G. Bhatia, R. Rajkumar, and P. Mudalige, "Reliable intersection protocols using vehicular networks," in *2013 ACM/IEEE International Conference on Cyber-Physical Systems (ICCPS)*, pp. 1–10, 2013.

[55] L. Makarem and D. Gillet, "Fluent coordination of autonomous vehicles at intersections," in *2012 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pp. 2557–2562, 2012.

[56] A. Katriniok, P. Sopasakis, M. Schuurmans, and P. Patrinos, "Nonlinear model predictive control for distributed motion planning in road intersections using panoc," in *2019 IEEE 58th Conference on Decision and Control (CDC)*, pp. 5272–5278, 2019.

[57] M. Hafner, D. Cunningham, L. Caminiti, and D. Del Vecchio, "Automated vehicle-to-vehicle collision avoidance at intersections," vol. 7, 01 2011.

[58] M. R. Hafner, D. Cunningham, L. Caminiti, and D. Del Vecchio, "Cooperative collision avoidance at intersections: Algorithms and experiments," *IEEE Transactions on Intelligent Transportation Systems*, vol. 14, no. 3, pp. 1162–1175, 2013.

[59] A. Colombo and D. Del Vecchio, "Efficient algorithms for collision avoidance at intersections," *Proceedings of the 15th ACM international conference on Hybrid Systems: Computation and Control (HSCC '12). ACM, New York, NY, USA*, pp. 145–154, 2012.

[60] J. Yu and L. Petnga, "Space-based collision avoidance framework for autonomous vehicles," *Procedia Computer Science*, vol. 140, pp. 37–45, 2018. Cyber Physical Systems and Deep Learning Chicago, Illinois November 5-7, 2018.

[61] J. Wang, X. Xue, R. Chai, and N. Cao, "A temporal-spatial collision warning method at non-signalized intersection," *Procedia Engineering*, vol. 137, p. 827–835, 2016.

[62] F. Belkhouche, "Collaboration and optimal conflict resolution at an unsignalized intersection," *IEEE Transactions on Intelligent Transportation Systems*, vol. 20, no. 6, pp. 2301–2312, 2019.

[63] F. Belkhouche, "Control of autonomous vehicles at an unsignalized intersection," in *2017 American Control Conference (ACC)*, pp. 1340–1345, 2017.

[64] S. Leroy, J.-P. Laumond, and T. Siméon, "Multiple path coordination for mobile robots: A geometric algorithm.," vol. 2, pp. 1118–1123, 01 1999.

[65] J. Gregoire, "Priority-based coordination of mobile robots," 2014.

[66] X. Qian, J. Gregoire, A. de La Fortelle, and F. Moutarde, "Decentralized model predictive control for smooth coordination of automated vehicles at intersection," in *2015 European Control Conference (ECC)*, pp. 3452–3458, 2015.

[67] M. Kneissl, A. Molin, H. Esen, and S. Hirche, "Any-time feasible coordination for multivehicle systems," *IEEE Transactions on Control Systems Technology*, vol. 30, no. 5, pp. 1807–1820, 2022.

[68] Y. Mo, M. Wang, T. Zhang, and Q. Zhang, "Autonomous cooperative vehicle coordination at road intersections," in *2018 IEEE International Conference on Communication Systems (ICCS)*, pp. 192–197, 2018.

[69] E. Steinmetz, R. Hult, Z. Zou, R. Emardson, F. Brännström, P. Falcone, and H. Wymeersch, "Collision-aware communication for intersection management of automated vehicles," *IEEE Access*, vol. 6, pp. 77359–77371, 2018.

[70] A. A. Malikopoulos, C. G. Cassandras, and Y. J. Zhang, "A decentralized energy-optimal control framework for connected automated vehicles at signal-free intersections," *Automatica*, vol. 93, pp. 244–256, 2018.

[71] R. Olfati-Saber and R. Murray, "Consensus problems in networks of agents with switching topology and time-delays," *IEEE Transactions on Automatic Control*, vol. 49, no. 9, pp. 1520–1533, 2004.

[72] F. Xiao and L. Wang, "Asynchronous consensus in continuous-time multi-agent systems with switching topology and time-varying delays," *IEEE Transactions on Automatic Control*, vol. 53, no. 8, pp. 1804–1816, 2008.

[73] F. Xiao, T. Chen, and H. Gao, "Synchronous hybrid event- and time-driven consensus in multiagent networks with time delays," *IEEE Transactions on Cybernetics*, vol. 46, no. 5, pp. 1165–1174, 2016.

[74] Y. G. Sun and L. Wang, "Consensus of multi-agent systems in directed networks with nonuniform time-varying delays," *IEEE Transactions on Automatic Control*, vol. 54, no. 7, pp. 1607–1613, 2009.

[75] P. Lin and Y. Jia, "Multi-agent consensus with diverse time-delays and jointly-connected topologies," *Automatica*, vol. 47, no. 4, pp. 848–856, 2011.

[76] X. Yu, F. Yang, C. Zou, and L. Ou, "Stabilization parametric region of distributed pid controllers for general first-order multi-agent systems with time delay," *IEEE/CAA Journal of Automatica Sinica*, vol. 7, no. 6, pp. 1555–1564, 2020.

[77] Z. Meng, W. Ren, Y. Cao, and Z. You, "Leaderless and leader-following consensus with communication and input delays under a directed network topology," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 41, no. 1, pp. 75–88, 2011.

[78] C. Huang, G. Zhai, and G. Xu, "Necessary and sufficient conditions for consensus in third order multi-agent systems," *IEEE/CAA Journal of Automatica Sinica*, vol. 5, no. 6, pp. 1044–1053, 2018.

[79] Y. G. Sun and L. Wang, "Consensus problems in networks of agents with double-integrator dynamics and time-varying delays," *International Journal of Control*, vol. 82, p. 1937–1945, Aug. 2009.

[80] P. Lin and Y. Jia, "Consensus of a class of second-order multi-agent systems with time-delay and jointly-connected topologies," *IEEE Transactions on Automatic Control*, vol. 55, no. 3, pp. 778–784, 2010.

[81] K. Liu, Z. Ji, and W. Ren, "Necessary and sufficient conditions for consensus of second-order multiagent systems under directed topologies without global gain dependency," *IEEE Transactions on Cybernetics*, vol. 47, no. 8, pp. 2089–2098, 2017.

[82] W. Ren, R. W. Beard, and E. M. Atkins, "Information consensus in multivehicle cooperative control," *IEEE Control Systems Magazine*, vol. 27, no. 2, pp. 71–82, 2007.

[83] W. Ren, "On consensus algorithms for double-integrator dynamics," *IEEE Transactions on Automatic Control*, vol. 53, no. 6, pp. 1503–1509, 2008.

[84] W. Yu, G. Chen, and M. Cao, "Some necessary and sufficient conditions for second-order consensus in multi-agent dynamical systems," *Automatica*, vol. 46, no. 6, pp. 1089–1095, 2010.

[85] J. Qin, W. X. Zheng, and H. Gao, "Consensus of multiple second-order vehicles with a time-varying reference signal under directed topology," *Automatica*, vol. 47, pp. 1983–1991, 09 2011.

[86] S. Santini, A. Salvi, A. S. Valente, A. Pescapè, M. Segata, and R. L. Cigno, "Platooning maneuvers in vehicular networks: A distributed and consensus-based approach," *IEEE Transactions on Intelligent Vehicles*, vol. 4, no. 1, pp. 59–72, 2019.

[87] M. Lv, W. Yu, J. Cao, and S. Baldi, "Consensus in high-power multiagent systems with mixed unknown control directions via hybrid nussbaum-based control," *IEEE Transactions on Cybernetics*, vol. 52, no. 6, pp. 5184–5196, 2022.

[88] M. Lv, W. Yu, J. Cao, and S. Baldi, "A separation-based methodology to consensus tracking of switched high-order nonlinear multiagent systems," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 33, no. 10, pp. 5467–5479, 2022.

[89] Y. Liu, D. Yao, H. Li, and R. Lu, "Distributed cooperative compound tracking control for a platoon of vehicles with adaptive nn," *IEEE Transactions on Cybernetics*, vol. 52, no. 7, pp. 7039–7048, 2022.

[90] Z. Meng, Z. Zhao, and Z. Lin, "On global leader-following consensus of identical linear dynamic systems subject to actuator saturation," *Systems & Control Letters*, vol. 62, no. 2, pp. 132–142, 2013.

[91] J. Fu, G. Wen, W. Yu, T. Huang, and X. Yu, "Consensus of second-order multiagent systems with both velocity and input constraints," *IEEE Transactions on Industrial Electronics*, vol. 66, no. 10, pp. 7946–7955, 2019.

[92] M. Jafarian and C. De Persis, "Formation control using binary information," *Automatica*, vol. 53, pp. 125–135, 2015.

[93] Q. Wei, X. Wang, X. Zhong, and N. Wu, "Consensus control of leader-following multi-agent systems in directed topology with heterogeneous disturbances," *IEEE/CAA Journal of Automatica Sinica*, vol. 8, no. 2, pp. 423–431, 2021.

[94] D. Yao, H. Li, R. Lu, and Y. Shi, "Distributed sliding-mode tracking control of second-order nonlinear multiagent systems: An event-triggered approach," *IEEE Transactions on Cybernetics*, vol. 50, no. 9, pp. 3892–3902, 2020.

[95] R. Palmieri, "Leaderless consensus: The state of the art," in *2016 IEEE International Parallel and Distributed Processing Symposium Workshops (IPDPSW)*, IEEE, May 2016.

[96] H. Rezaee and F. Abdollahi, "Adaptive leaderless consensus control of strict-feedback nonlinear multiagent systems with unknown control directions," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 51, no. 10, pp. 6435–6444, 2021.

[97] Y. Kim and M. Mesbahi, "On maximizing the second smallest eigenvalue of a state-dependent graph laplacian," *IEEE Transactions on Automatic Control*, vol. 51, no. 1, pp. 116–120, 2006.

[98] J. Sheng and Z. Ding, "Optimal consensus control of linear multi-agent systems with communication time delay," *IET Control Theory & Applications*, vol. 7, p. 1899–1905, Oct. 2013.

[99] W. Yang, X. Wang, and H. Shi, "Optimal second order consensus protocol with time delay," in *2011 50th IEEE Conference on Decision and Control and European Control Conference*, pp. 3470–3475, 2011.

[100] Z. Jiandong, "Maximum consensus speed of second-order multi-agent network systems under a kind of dynamic consensus protocols," in *Proceedings of the 29th Chinese Control Conference*, pp. 796–800, 2010.

[101] Z. Yu and L. Xie, "Convergence rate of distributed consensus for second-order multi-agent systems," in *IEEE ICCA 2010*, pp. 2284–2289, 2010.

[102] W. Xing, Y. Zhao, and H. R. Karimi, "Convergence analysis on multi-auv systems with leader-follower architecture," *IEEE Access*, vol. 5, pp. 853–868, 2017.

[103] A. Eichler and H. Werner, "Closed-form solution for optimal convergence speed of multi-agent systems with discrete-time double-integrator dynamics for fixed weight ratios," *Systems & Control Letters*, vol. 71, p. 7–13, 09 2014.

[104] A. Eichler and H. Werner, "Optimal convergence speed with constrained damping of double-integrator multi-agent systems with undirected topology," in *2015 54th IEEE Conference on Decision and Control (CDC)*, pp. 2211–2216, 2015.

[105] C. Bergenhem, S. Shladover, E. Coelingh, C. Englund, and S. Tsugawa, "Overview of platooning systems," 01 2012.

[106] B. Bamieh, F. Paganini, and M. Dahleh, "Distributed control of spatially invariant systems," *IEEE Transactions on Automatic Control*, vol. 47, no. 7, pp. 1091–1107, 2002.

[107] D. Swaroop and J. Hedrick, "String stability of interconnected systems," *IEEE Transactions on Automatic Control*, vol. 41, no. 3, pp. 349–357, 1996.

[108] M. Bashiri, H. Jafarzadeh, and C. H. Fleming, "Paim: Platoon-based autonomous intersection management," in *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*, pp. 374–380, 2018.

[109] R. W. Timmerman and M. A. A. Boon, "Platoon forming algorithms for intelligent street intersections," *Transportmetrica A: Transport Science*, vol. 17, pp. 278 – 307, 2019.

[110] R. T. Rockafellar, *Convex Analysis*. Princeton University Press, 2009.

[111] D. P. Bertsekas, *Convex Optimization Theory*. Athena Scientific, 2009.

[112] E. K. P. Chong and S. H. Żak, *An Introduction to Optimization, Second Edition*. John Wiley & Sons, Inc, 2001.

[113] N. Andreasson, A. Evgrafov, M. Patriksson, E. Gustavsson, and M. Onnheim, *Introduction to continuous optimization*. Lund, Sweden: Studentlitteratur, Oct. 2013.

[114] A. S. Nemirovski and M. J. Todd, "Interior-point methods for optimization," *Acta Numerica*, vol. 17, p. 191–234, Apr. 2008.

[115] M. H. Wright, "Interior methods for constrained optimization," *Acta Numerica*, vol. 1, p. 341–407, Jan. 1992.

[116] L. Li and Z.-P. Liu, "A connected network-regularized logistic regression model for feature selection," *Applied Intelligence*, vol. 52, 01 2022.

[117] INCOSE, "Systems engineering and system definitions." `https://www.incose.org/docs/default-source/default-document-library/final_-se-definition.pdf`, January 2019. [Accessed: 06-01-2024].

[118] S. Rinaldi, *Teoria dei sistemi*. CittàStudi, 1992.

[119] W. Ren and R. W. Beard, *Distributed consensus in multi-vehicle cooperative control – Theory and applications*. Springer, 2008.

[120] J. Ziegler and C. Stiller, "Fast collision checking for intelligent vehicle motion planning," in *2010 IEEE Intelligent Vehicles Symposium*, pp. 518–522, 2010.

[121] W. Ren and E. Atkins, "Distributed multi-vehicle coordinated control via local information exchange," *International Journal of Robust and Nonlinear Control*, vol. 17, 07 2007.

[122] W. Ren and E. Atkins, "Second-order consensus protocols in multiple vehicle systems with local interactions," *Collection of Technical Papers - AIAA Guidance, Navigation, and Control Conference*, vol. 5, 08 2005.

[123] E. Tegling and H. Sandberg, "On the coherence of large-scale networks with distributed pi and pd control," *IEEE Control Systems Letters*, vol. 1, no. 1, pp. 170–175, 2017.

[124] W. Ren, "Consensus based formation control strategies for multi-vehicle systems," in *2006 American Control Conference*, pp. 6 pp.–, 2006.

[125] R. Sipahi, S.-i. Niculescu, C. T. Abdallah, W. Michiels, and K. Gu, "Stability and stabilization of systems with time delay," *IEEE Control Systems Magazine*, vol. 31, no. 1, pp. 38–65, 2011.

[126] W.-C. Yueh, "Eigenvalues of several tridiagonal matrices," *Applied Mathematics E-Notes [electronic only]*, vol. 5, pp. 66–74, 2005.

[127] M. Hazewinkel, *Encyclopaedia of Mathematics: Supplement, Volume 1*. Kluwer Academic Pub, 1997.

Appendixes

Appendix A

Proofs

A.1 Chapter 3

Proof of Lemma 3.1

The intersection of the half-spaces $H_<(\boldsymbol{\eta}, a)$ and $H_<(-\boldsymbol{\eta}, b)$, with $a, b \in \bar{\mathbb{R}}$, is the strip $S(\boldsymbol{\eta}, a, b) = H_<(\boldsymbol{\eta}, a) \cap H_<(-\boldsymbol{\eta}, b)$:

$$\begin{aligned} S(\boldsymbol{\eta}, a, b) &= \{\boldsymbol{x} \in \mathbb{R}^n \ : \ \langle \boldsymbol{\eta}, \boldsymbol{x} \rangle < a\} \cap \{\boldsymbol{x} \in \mathbb{R}^n \ : \ \langle -\boldsymbol{\eta}, \boldsymbol{x} \rangle < b\} = \\ &= \{\boldsymbol{x} \in \mathbb{R}^n \ : \ -b < \langle \boldsymbol{\eta}, \boldsymbol{x} \rangle < a\}. \end{aligned}$$

For a given $d \in \mathbb{R}$, the intersection between the hyperplane $H(\boldsymbol{\eta}, d)$ and the strip $S(\boldsymbol{\eta}, a, b)$ is

$$\begin{aligned} H(\boldsymbol{\eta}, d) \cap S(\boldsymbol{\eta}, a, b) &= \{\boldsymbol{x} \in \mathbb{R}^n \ : \ -b < \langle \boldsymbol{\eta}, \boldsymbol{x} \rangle < a \wedge \langle \boldsymbol{\eta}, \boldsymbol{x} \rangle = d\} = \\ &= \begin{cases} H(\boldsymbol{\eta}, d) & \text{if} \quad -b < d < a, \\ \emptyset & \text{if} \quad d > a \vee d < -b. \end{cases} \end{aligned}$$

By (3.1), we have that $C \subset H_<(\boldsymbol{\eta}, \sigma_C(\boldsymbol{\eta}))$ and $C \subset H_<(-\boldsymbol{\eta}, \sigma_C(-\boldsymbol{\eta}))$, hence $C \subset S(\boldsymbol{\eta}, \sigma_C(\boldsymbol{\eta}), \sigma_C(-\boldsymbol{\eta}))$. From here, it follows that

$$d \leq -\sigma_C(-\boldsymbol{\eta}) \ \vee \ d \geq \sigma_C(\boldsymbol{\eta}) \ \Rightarrow \ H(\boldsymbol{\eta}, d) \cap S(\boldsymbol{\eta}, \sigma_C(\boldsymbol{\eta}), \sigma_C(-\boldsymbol{\eta}) = \emptyset,$$

thus the thesis.

Proof of Proposition 3.1

By the definition of $\sigma_{\boldsymbol{q}}$, we have

$$\sigma_{\boldsymbol{q}}(\boldsymbol{\eta}) = \sup_{\lambda \in [0,1]} \langle \boldsymbol{\eta}, \boldsymbol{q}(\lambda) \rangle .$$

First, note that the derivative of $\boldsymbol{q}$ is

$$\boldsymbol{q}'(\lambda) = -2(1-\lambda)\boldsymbol{a} + 2(1-2\lambda)\boldsymbol{o} + 2\lambda\boldsymbol{b}.$$

To find the extremum of $\langle \boldsymbol{\eta},\, \boldsymbol{q}(\lambda)\rangle$ inside the $[0,1]$ interval, let us first compute its derivative:

$$\frac{\mathrm{d}}{\mathrm{d}\lambda}\langle \boldsymbol{\eta},\, \boldsymbol{q}(\lambda)\rangle = \langle \boldsymbol{\eta},\, \boldsymbol{q}'(\lambda)\rangle = -2(1-\lambda)\langle \boldsymbol{\eta},\, \boldsymbol{a}\rangle + 2(1-2\lambda)\langle \boldsymbol{\eta},\, \boldsymbol{o}\rangle + 2\lambda\langle \boldsymbol{\eta},\, \boldsymbol{b}\rangle .$$

λ^* is the value of λ in which this derivative vanishes:

$$-2(1-\lambda^*)\langle \boldsymbol{\eta},\, \boldsymbol{a}\rangle + 2(1-2\lambda^*)\langle \boldsymbol{\eta},\, \boldsymbol{o}\rangle + 2\lambda^*\langle \boldsymbol{\eta},\, \boldsymbol{b}\rangle = 0.$$

By simple calculations, we obtain

$$\lambda^* = \frac{\langle \boldsymbol{\eta},\, \boldsymbol{a}-\boldsymbol{o}\rangle}{\langle \boldsymbol{\eta},\, \boldsymbol{a}-2\boldsymbol{o}+\boldsymbol{b}\rangle}$$

and

$$1-\lambda^* = \frac{\langle \boldsymbol{\eta},\, \boldsymbol{b}-\boldsymbol{o}\rangle}{\langle \boldsymbol{\eta},\, \boldsymbol{a}-2\boldsymbol{o}+\boldsymbol{b}\rangle}.$$

It must be verified that

$$\langle \boldsymbol{\eta},\, \boldsymbol{a}-2\boldsymbol{o}+\boldsymbol{b}\rangle \neq 0, \tag{A.1}$$

otherwise, λ^* would not be defined.
Imposing $0 \leq \lambda^* \leq 1$ yields

$$0 \leq \langle \boldsymbol{\eta},\, \boldsymbol{a}-\boldsymbol{o}\rangle \leq \langle \boldsymbol{\eta},\, \boldsymbol{a}-2\boldsymbol{o}+\boldsymbol{b}\rangle . \tag{A.2}$$

and, then, the following set of inequalities

$$\begin{cases} \langle \boldsymbol{\eta},\, \boldsymbol{a}-\boldsymbol{o}\rangle \geq 0, \\ \langle \boldsymbol{\eta},\, \boldsymbol{b}-\boldsymbol{o}\rangle \leq 0, \end{cases}$$

which is equivalent to the intersection of the closed half-spaces $H_{\geq}(\boldsymbol{a}-\boldsymbol{o}, 0)$ and $H_{\leq}(\boldsymbol{b}-\boldsymbol{o}, 0)$.

Under the hypothesis of non-collinearity of the control points, we can verify (A.1): to see this, suppose that $\langle \boldsymbol{\eta},\, \boldsymbol{a}-2\boldsymbol{o}+\boldsymbol{b}\rangle = 0$. Then, by (A.2), it implies $\langle \boldsymbol{\eta},\, \boldsymbol{a}-\boldsymbol{o}\rangle = 0$ and $\langle \boldsymbol{\eta},\, \boldsymbol{b}-\boldsymbol{o}\rangle = 0$, which means that $\boldsymbol{a}$, $\boldsymbol{o}$, and $\boldsymbol{b}$ are collinear, leading to a contradiction.

Now, it is easy to verify that $\langle \boldsymbol{\eta},\, \boldsymbol{q}(t)\rangle$ attains a maximum at λ^* by evaluating its second derivative:

$$\frac{\mathrm{d}^2}{\mathrm{d}\lambda^2}\langle \boldsymbol{\eta},\, \boldsymbol{q}(\lambda)\rangle = 2\langle \boldsymbol{\eta},\, \boldsymbol{a}\rangle - 4\langle \boldsymbol{\eta},\, \boldsymbol{o}\rangle + 2\langle \boldsymbol{\eta},\, \boldsymbol{b}\rangle = 2\langle \boldsymbol{\eta},\, \boldsymbol{a}-2\boldsymbol{o}+\boldsymbol{b}\rangle ,$$

which, by (A.1) and (A.2), is strictly greater than 0.

Before directly evaluating $\langle \boldsymbol{\eta},\, \boldsymbol{q}(\lambda^*)\rangle$, let us first report the expression of $\boldsymbol{q}(\lambda^*)$:

$$\boldsymbol{q}(\lambda^*) = \left(\frac{\langle \boldsymbol{\eta},\, \boldsymbol{b}-\boldsymbol{o}\rangle}{\langle \boldsymbol{\eta},\, \boldsymbol{a}-2\boldsymbol{o}+\boldsymbol{b}\rangle}\right)^2 \boldsymbol{a} + \frac{2\langle \boldsymbol{\eta},\, \boldsymbol{a}-\boldsymbol{o}\rangle\langle \boldsymbol{\eta},\, \boldsymbol{b}-\boldsymbol{o}\rangle}{\langle \boldsymbol{\eta},\, \boldsymbol{a}-2\boldsymbol{o}+\boldsymbol{b}\rangle^2}\boldsymbol{o} + \left(\frac{\langle \boldsymbol{\eta},\, \boldsymbol{a}-\boldsymbol{o}\rangle}{\langle \boldsymbol{\eta},\, \boldsymbol{a}-2\boldsymbol{o}+\boldsymbol{b}\rangle}\right)^2 \boldsymbol{b} =$$
$$= \frac{\langle \boldsymbol{\eta},\, \boldsymbol{b}-\boldsymbol{o}\rangle^2 \boldsymbol{a} + 2\langle \boldsymbol{\eta},\, \boldsymbol{a}-\boldsymbol{o}\rangle\langle \boldsymbol{\eta},\, \boldsymbol{b}-\boldsymbol{o}\rangle \boldsymbol{o} + \langle \boldsymbol{\eta},\, \boldsymbol{a}-\boldsymbol{o}\rangle^2 \boldsymbol{b}}{\langle \boldsymbol{\eta},\, \boldsymbol{a}-2\boldsymbol{o}+\boldsymbol{b}\rangle^2}.$$

Then, after some calculations, we obtain

$$\sigma_{\boldsymbol{q}}(\boldsymbol{\eta}) = \langle \boldsymbol{\eta}, \boldsymbol{q}(\lambda^*) \rangle = \frac{\langle \boldsymbol{\eta}, \boldsymbol{a} \rangle \langle \boldsymbol{\eta}, \boldsymbol{b} \rangle - \langle \boldsymbol{\eta}, \boldsymbol{o} \rangle^2}{\langle \boldsymbol{\eta}, \boldsymbol{a} - 2\boldsymbol{o} + \boldsymbol{b} \rangle}.$$

By its expression, we can see that $\sigma_{\boldsymbol{q}}$ is differentiable in every point of its domain, thus its gradient $\nabla \sigma_{\boldsymbol{q}}(\boldsymbol{\eta})$ equals $\arg\max_{\lambda \in [0,1]} \langle \boldsymbol{\eta}, \boldsymbol{q}(\lambda) \rangle = \boldsymbol{q}(\lambda^*)$.

A.2 Chapter 4

Proof of Proposition 4.2

To prove the first statement, it is enough to show that $\boldsymbol{w}'$ is the only direction of recession of $\mathrm{conv}(C)$.

If $\boldsymbol{p}_1 \downarrow \boldsymbol{p}_2$, by definition it exists $\bar{u}_1$ and $\bar{u}_2$ such that $\boldsymbol{p}_1(\mathbb{R}_{\geq \bar{u}_1}) = \boldsymbol{p}_2(\mathbb{R}_{\geq \bar{u}_2})$. Given that every trajectory of P_α is a straight line when its abscissa is greater than 1, $\bar{u}_1$ and $\bar{u}_2$ can be chosen so that both $\boldsymbol{p}_1(u_1)$ and $\boldsymbol{p}_2(u_2)$ are straight lines for $u_1 \geq \bar{u}_1$ and $u_2 \geq \bar{u}_2$. Let $\bar{\boldsymbol{p}} = \boldsymbol{p}_1(\bar{u}_1) = \boldsymbol{p}_2(\bar{u}_2)$. Then, we can write

$$\begin{cases} \boldsymbol{p}_1(u_1) = \bar{\boldsymbol{p}} + \boldsymbol{v}_1(u_1 - \bar{u}_1), \quad \forall u_1 \geq \bar{u}_1 & \text{(A.3a)} \\ \boldsymbol{p}_2(u_2) = \bar{\boldsymbol{p}} + \boldsymbol{v}_2(u_2 - \bar{u}_2), \quad \forall u_2 \geq \bar{u}_2. & \text{(A.3b)} \end{cases}$$

Since $\boldsymbol{p}_1$ and $\boldsymbol{p}_2$ share the same image within the intervals of interests, they are reparametrizations of the same curve, i.e., there exist $\bar{\lambda} \in \mathbb{R}$ and two non-decreasing $\beta_1 : \mathbb{R}_{\geq \bar{\lambda}} \to \mathbb{R}_{\geq \bar{u}_1}$, $\beta_2 : \mathbb{R}_{\geq \bar{\lambda}} \to \mathbb{R}_{\geq \bar{u}_2}$, such that $\boldsymbol{p}_1(\beta_1(\lambda)) = \boldsymbol{p}_2(\beta_2(\lambda))$, $\forall \lambda > \bar{\lambda}$. To see this, let us equal (A.3a) and (A.3b):

$$\bar{\boldsymbol{p}} + \boldsymbol{v}_1(u_1 - \bar{u}_1) = \bar{\boldsymbol{p}} + \boldsymbol{v}_2(u_2 - \bar{u}_2). \tag{A.4}$$

It is easy to show that (A.4) can hold only if $\boldsymbol{v}_1$ and $\boldsymbol{v}_2$ are parallel, hence we can write $\boldsymbol{v}_1 = \hat{\boldsymbol{v}} w_1$ and $\boldsymbol{v}_2 = \hat{\boldsymbol{v}} w_2$. Then, (A.4) becomes

$$\hat{\boldsymbol{v}} \left(w_1(u_1 - \bar{u}_1) - w_2(u_2 - \bar{u}_2) \right) = 0.$$

From here, we derive that $\boldsymbol{p}_1(u_1) = \boldsymbol{p}_2(u_2)$ if and only if

$$\begin{cases} u_1 = \bar{u}_1 + w_2 \lambda, \\ u_2 = \bar{u}_2 + w_1 \lambda, \end{cases} \qquad \lambda \geq 0. \tag{A.5}$$

In this case, we obtained $\beta_1(\lambda) = \bar{u}_1 + w_2\lambda$, $\beta_2(\lambda) = \bar{u}_2 + w_1\lambda$, and $\bar{\lambda} = 0$. Equations (A.5) can be written more compactly as

$$\boldsymbol{u} = \bar{\boldsymbol{u}} + \boldsymbol{w}' \lambda, \quad \lambda \geq 0.$$

Now, showing that $\boldsymbol{w}'$ is the only direction of $\mathrm{conv}(C)$ is straightforward: $c(\bar{\boldsymbol{u}} + \lambda \boldsymbol{w}') = -\rho^2 < 0$, $\forall \lambda \geq 0$, that is,

$$\bar{\boldsymbol{u}} + \lambda \boldsymbol{w}' \in C, \quad \forall \lambda \geq 0.$$

The proof of the second statement is analogous and shows that when $\boldsymbol{p}_1 \uparrow \boldsymbol{p}_2$, $-\boldsymbol{w}'$ is the only direction of recession of $\mathrm{conv}(C)$.

Proof of Proposition 4.3

The proof is by contradiction. Suppose that $\boldsymbol{u}^*$ and $\bar{\boldsymbol{u}}$ are a local and a global maximizer of (4.38), respectively. Because, by hypothesis, C is path-connected, then there exists a path from $\boldsymbol{u}^*$ to $\bar{\boldsymbol{u}}$, i.e., a continuous function $\boldsymbol{\phi} : [0,1] \to C$, such that $\boldsymbol{\phi}(0) = \boldsymbol{u}^*$ and $\boldsymbol{\phi}(1) = \bar{\boldsymbol{u}}$.

Let g denote the objective function of problem (4.38), that is, $g(\boldsymbol{u}) = \langle \boldsymbol{\eta}, \boldsymbol{u} \rangle$. Since $\boldsymbol{u}^*$ is a strict maximizer, by definition, it exists $\epsilon > 0$ such that

$$g^* = g(\boldsymbol{u}^*) > g(\boldsymbol{u}), \quad \forall \boldsymbol{u} \in C \setminus \{\boldsymbol{u}^*\} \ : \ \|\boldsymbol{u} - \boldsymbol{u}^*\| \leq \epsilon. \tag{A.6}$$

Let us choose ϵ small enough so that $\|\bar{\boldsymbol{u}} - \boldsymbol{u}\| < \epsilon$. By the continuity of $\boldsymbol{\phi}$, there exists $\lambda_0 \in [0,1]$ such that $\|\boldsymbol{\phi}(\lambda_0) - \boldsymbol{u}^*\| = \epsilon$. Let $g_0 = g(\boldsymbol{\phi}(\lambda_0)) < g^*$ and $\bar{g} = g(\bar{\boldsymbol{u}}) > g^*$. $g \circ \boldsymbol{\phi}$ is continuous because it is a composition of continuous functions, hence the intermediate value theorem applies: in the $[\lambda_0, 1]$ interval, $g \circ \boldsymbol{\phi}$ assumes all values between $g(\boldsymbol{\phi}(\lambda_0)) = g_0$ and $g(\boldsymbol{\phi}(1)) = \bar{g}$. In particular, since $g_0 < g^* < \bar{g}$, there must exist λ_1 such that $g(\boldsymbol{\phi}(\lambda_1)) = g^*$ and $\boldsymbol{\phi}_1 = \boldsymbol{\phi}(\lambda_1) \neq \boldsymbol{u}^*$, because $g(\boldsymbol{\phi}(\lambda))$ is decreasing whenever $\boldsymbol{\phi}(\lambda) = \boldsymbol{u}^*$. Now let $\boldsymbol{\zeta} = \boldsymbol{\phi}_1 - \boldsymbol{u}^*$. $\boldsymbol{\zeta}$ is perpendicular to $\boldsymbol{\eta}$, since

$$\langle \boldsymbol{\eta}, \boldsymbol{\zeta} \rangle = \langle \boldsymbol{\eta}, \boldsymbol{\phi}_1 - \boldsymbol{u}^* \rangle = \langle \boldsymbol{\eta}, \boldsymbol{\phi}_1 \rangle - \langle \boldsymbol{\eta}, \boldsymbol{u}^* \rangle = g^* - g^* = 0.$$

Consider the vector $\boldsymbol{u}_\epsilon = \boldsymbol{u}^* + \epsilon \frac{\boldsymbol{\zeta}}{\|\boldsymbol{\zeta}\|}$. The value of g at $\boldsymbol{u}_\epsilon$ is

$$g(\boldsymbol{u}_\epsilon) = \left\langle \boldsymbol{\eta}, \boldsymbol{u}^* + \epsilon \frac{\boldsymbol{\zeta}}{\|\boldsymbol{\zeta}\|} \right\rangle = \langle \boldsymbol{\eta}, \boldsymbol{u}^* \rangle = g(\boldsymbol{u}^*).$$

Because $\boldsymbol{u}^*$ is a local maximum and $\|\boldsymbol{u}_\epsilon - \boldsymbol{u}^*\| = \epsilon$, by (A.6) the only possibility is $\boldsymbol{u}_\epsilon \notin C$, contradicting the hypothesis of C being $\boldsymbol{\zeta}$-convex.

A.3 Chapter 5

Proof of Lemma 5.1

Let us rewrite (5.4) as

$$\boldsymbol{T}(\boldsymbol{I}_n - \boldsymbol{U}^+\boldsymbol{U}) = \boldsymbol{O}_n. \tag{A.7}$$

It is known that $\boldsymbol{U}^+\boldsymbol{U}$ is the projector onto $\mathcal{K}^\perp(\boldsymbol{U})$ along $\mathcal{K}(\boldsymbol{U})$ and $\boldsymbol{I}_n - \boldsymbol{U}^+\boldsymbol{U}$ is the projector onto $\mathcal{K}(\boldsymbol{U})$ along $\mathcal{K}^\perp(\boldsymbol{U})$. This implies that $\mathcal{S}(\boldsymbol{I}_n - \boldsymbol{U}^+\boldsymbol{U}) = \mathcal{K}(\boldsymbol{U})$. Therefore, for any vector $\boldsymbol{w} \in \mathbb{R}^n$, we have

$$(\boldsymbol{I}_n - \boldsymbol{U}^+\boldsymbol{U})\boldsymbol{w} \in \mathcal{K}(\boldsymbol{U}).$$

Since, by hypothesis, $\mathcal{K}(\boldsymbol{U}) \subseteq \mathcal{K}(\boldsymbol{T})$, it holds

$$\boldsymbol{T}(\boldsymbol{I}_n - \boldsymbol{U}^+\boldsymbol{U})\boldsymbol{w} = \boldsymbol{0}_n.$$

This equation holds $\forall \boldsymbol{w} \in \mathbb{R}^n$, hence (A.7) is proven.

Proof of Proposition 5.1

Let us report the expression of $\boldsymbol{h}$:

$$\boldsymbol{h} = -\boldsymbol{L}\boldsymbol{H}^{+}\bar{\boldsymbol{d}}.$$

Let $\boldsymbol{g} = \boldsymbol{H}^{+}\bar{\boldsymbol{d}} \in \mathbb{R}^n$, so that $\boldsymbol{h} = -\boldsymbol{L}\boldsymbol{g}$. Since $\boldsymbol{H}\boldsymbol{H}^{+} = \boldsymbol{I}_n$, we can write $\bar{\boldsymbol{d}} = \boldsymbol{H}\boldsymbol{H}^{+}\bar{\boldsymbol{d}} = \boldsymbol{H}\boldsymbol{g}$, or, by considering its i-th component,

$$\bar{d} = g_i - g_{i+1}.$$

This equations defines the arithmetic progression $g_{i+1} = g_i - \bar{d}$, whose general term is

$$g_i = g_1 + (i-1)\bar{d}.$$

Considering the i-th component of $\boldsymbol{h} = -\boldsymbol{L}\boldsymbol{g}$, we obtain

$$h_i = -\sum_{j \in \mathcal{N}(i)} (g_i - g_j) = -\sum_{j \in \mathcal{N}(i)} \big(\big(g_1 + (i-1)\bar{d}\big) - \big(g_1 + (j-1)\bar{d}\big)\big) = \bar{d} \sum_{j \in \mathcal{N}(i)} (i-j).$$