



Politecnico di Bari

Repository Istituzionale dei Prodotti della Ricerca del Politecnico di Bari

Exploiting predictive capabilities in motion control for autonomous agents operating in crowded environments

This is a PhD Thesis

Original Citation:

Exploiting predictive capabilities in motion control for autonomous agents operating in crowded environments / Gaetan, Elisa. - ELETTRONICO. - (2026).

Availability:

This version is available at <http://hdl.handle.net/11589/295760> since: 2026-01-17

Published version

DOI:

Publisher: Politecnico di Bari

Terms of use:

(Article begins on next page)



Italian National Ph.D. Program in Autonomous Systems

ACADEMIC DISCIPLINE: SYSTEMS AND CONTROL ENGINEERING (IINF-04/A)

Final Dissertation

Exploiting Predictive capabilities in motion control for autonomous agents operating in crowded environments

by

Elisa Gaetan

Administrative Headquarters:

Politecnico di Bari – Department of Electrical and Information Engineering

Hosting University:

Università degli Studi di Modena e Reggio Emilia – Department of Engineering “Enzo Ferrari”

Referees:

Prof. Adriano Fagiolini

Prof. Johannes Betz

Supervisors:

Prof. Laura Giarre

Prof. Paolo Falcone



PAOLO FALCONE
16.01.2026
16:23:15
GMT+01:00

Coordinator of Ph.D Program

Prof. Mariagrazia Dotoli

LIBERATORIA PER L'ARCHIVIAZIONE DELLA TESI DI DOTTORATO

Al Magnifico Rettore
del Politecnico di Bari

Il/la sottoscritto/a ELISA GAETAN nato/a a CASTELFRANCO VENETO (TV) il 25/08/1998
residente a CASTELFRANCO VENETO (TV) in via BRIGATA CESARE BATTISTI 22 e-mail elisa.gaetan@gmail.com
iscritto al 3° anno di Corso di Dottorato di Ricerca in AUTONOMOUS SYSTEMS DAUSY ciclo XXXVIII
ed essendo stato ammesso a sostenere l'esame finale con la prevista discussione della tesi dal titolo:
EXPLOITING PREDICTIVE CAPABILITIES IN MOTION CONTROL FOR AUTONOMOUS AGENTS OPERATING IN
CROWDED ENVIRONMENTS

DICHARA

- 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 AUTONOMOUS SYSTEMS DAUSY ciclo XXXVIII, 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 archiverà 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**/inviolate/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 CASTELFRANCO V.TO, 15/01/2026

Firma



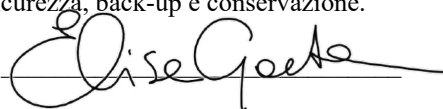
Il/La sottoscritto, con l'autoarchiviazione della propria tesi di dottorato nell'Archivio Istituzionale ad accesso aperto del Politecnico di Bari (POLIBA-IRIS), pur mantenendo su di essa tutti i diritti d'autore, morali ed economici, ai sensi della normativa vigente (Legge 633/1941 e ss.mm.ii.),

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 CASTELFRANCO V.TO, 15/01/2026

Firma





Elisa Gaetan

Exploiting Predictive capabilities in motion control for autonomous agents operating in crowded environments

Thesis submitted for the degree of Philosophiae Doctor

Italian National Ph.D. Program in Autonomous Systems
Università degli Studi di Modena e Reggio Emilia

Tutors

Prof. *Laura Giarre*

Prof. *Paolo Falcone*



2025



Ministero
dell'Università
e della Ricerca



Italiadomani
PIANO NAZIONALE
DI RIPRESA E RESILIENZA



Politecnico
di Bari



UNIMORE
UNIVERSITÀ DEGLI STUDI DI
MODENA E REGGIO EMILIA

The doctoral scholarship was funded by the European Union - Next Generation EU,
Mission 4 Component 1 CUP *D93C22000850005*.

Dissertation submitted for the degree of *Philosophiae Doctor*
Italian National Ph.D. Program in Autonomous Systems

Cycle:

38th

Administrative Headquarters:

Politecnico di Bari

Hosting University:

Università degli Studi di Modena e Reggio Emilia

Title:

Exploiting Predictive capabilities in motion control for autonomous agents operating in
crowded environments

Ph.D Candidate:

Elisa Gaetan, Università degli Studi di Modena e Reggio Emilia (Modena, Italy)

Tutors:

Prof. Laura Giarre, Università degli Studi di Modena e Reggio Emilia (Modena, Italy)

Prof. Paolo Falcone, Università degli Studi di Modena e Reggio Emilia (Modena, Italy)

Coordinator:

Prof. Mariagrazia Dotoli, Politecnico di Bari (Bari, Italy)

External Reviewers:

Prof. Adriano Fagiolini, Università degli Studi di Palermo (Palermo, Italy)

Prof. Johannes Betz, Technical University of Munich (Munich, Germany)

Last version:

January 14, 2026

*All rights reserved. No part of this publication may be reproduced or transmitted, in any
form or by any means, without permission.*

Abstract

The study of interacting entities—ranging from biological organisms to autonomous machines—offers a unifying framework for understanding the emergence of collective behavior in complex systems. Across disciplines such as physics, control theory, computer science, and the social sciences, a central challenge lies in capturing how local interactions among agents give rise to macroscopic patterns and decision dynamics. Deterministic models, while analytically tractable, often fail to account for the uncertainty and heterogeneity inherent in real-world environments. Stochastic approaches, particularly those based on Markovian frameworks, provide a principled alternative by representing agents as probabilistically transitioning between states under local interaction rules. This perspective enables the modeling of emergent phenomena such as consensus, competition, and polarization, while maintaining computational simplicity.

Building upon this foundation, this thesis presents two main contributions that bridge modeling and control of interacting decision-making agents. The first concerns the problem of modeling, inferring, and forecasting agents' decisions within dynamic environments. A Markov chain-based framework is developed to represent the evolution of agents' decisions in a network, allowing for probabilistic characterization of behavioral transitions. This model is analyzed in two domains: (i) social networks, where agents are interconnected or influenced by environmental factors, and (ii) traffic scenarios, where a decision-making model for human drivers is proposed based on scenario-specific rules and empirically grounded behavioral observations.

The second contribution focuses on the integration of these decision-making models within a control framework. The developed models are incorporated into closed-loop systems—particularly in traffic scenarios—where an autonomous agent acts as the actuator of a collective control strategy. In addition, an alternative controller is designed for pedestrian environments, emphasizing collision avoidance under stochastic predictions of human obstacle trajectories. These implementations demonstrate how probabilistic models of behavior can inform adaptive, decentralized control mechanisms in complex, interactive settings.

Ultimately, the investigation of interacting agents seeks to explain how simple local rules generate complex global behaviors, and how these processes can be shaped through decentralized, adaptive control strategies. By integrating stochastic modeling, decision theory, and behavioral insights, this research contributes to the design of autonomous and socially aware systems that operate safely, efficiently, and cooperatively within dynamic, human-centered environments.

Vola solo chi osa farlo.
Luis Sepúlveda

Contents

Acronyms	x
Preface	xi
List of Papers Written by the Author	xi
Introduction	
1 Introduction	2
1.1 Overview	2
1.2 Related Works	4
1.3 Contributions	7
1.4 Thesis organization	7
Opinion Dynamics	
2 Modeling Agents Interactions with Markov Chains	10
2.1 Introduction	10
2.2 Related Works	10
2.3 Contribution	11
3 Repulsive Models for Interacting Agents	12
3.1 Problem Definition	12
3.2 Repulsive Forces	12
3.2.1 Indirect repulsion	13
3.2.2 Direct repulsion	13
3.3 Marginalized Model	14
3.4 Comparison between direct and indirect repulsion models	15
3.5 Feasibility under similarity and marginalizability constraints	16
3.5.1 No marginalization constraints	17
3.5.2 $\mathbb{D}$ is marginalizable	17
3.5.3 $\mathbb{D}$ and $\mathbb{I}$ are marginalizable	18
3.6 Traffic Example	18
3.7 Discussion	19
4 Influencing Agents Decisions	21
4.1 Influence of Decisions in Stochastic Multi-Agent Systems	21
4.1.1 Mathematical Model	21
4.1.2 Controller	24
4.1.3 Simulation and Results	27
4.2 Feedback Control in Multi-Agent Markovian networks	29
4.2.1 Explicit solution for the constrained reference tracking problem	29
4.2.2 Results	32
4.3 Discussion	36
Mixed-Autonomy Traffic	
5 Mixed-Autonomy Traffic Scenarios: Predicting and Exploiting Drivers' Decisions	38

5.1	Introduction	38
5.2	Related Works	39
5.3	Contributions	39
6	Exploiting Markov Chains to forecast traffic scenarios	41
6.1	Modeling Traffic Scenarios with Markov Chains	41
6.1.1	Discrete-Time Markovian Model	41
6.1.2	Fitting the Markovian model for traffic scenarios	41
6.1.3	Learning the Transition Matrix	42
6.1.4	Simulation and Results	43
6.2	A Behavioral Model for Interacting Vehicles	47
6.2.1	Problem definition	47
6.2.2	Set of interacting vehicles	48
6.2.3	Decision-making model of the Autonomous Agent	50
6.2.4	Parametric Transition Matrix Design	50
6.2.5	Simulations and Results	54
6.3	Discussion	57
7	Scenario-Based Model Predictive Control for Vehicle interactions in Highway setting	58
7.1	Problem definition	58
7.2	Control architecture	59
7.2.1	Decision-making block	59
7.2.2	Scenario-tree model predictive control algorithm	60
7.3	Simulations and Results	61
7.4	Discussion	62
Social Robotics		
8	Interactions in Social Robotics: Motion Planners Design for Pedestrian Environments	65
8.1	Introduction	65
8.2	Related Works	67
8.3	Contributions	68
9	Game-Theory-based Motion Planner	69
9.1	Problem definition	70
9.2	Potential Games	71
9.3	Game between the autonomous agent and the human	72
9.4	Discussion	75
10	Control Barrier Function for Collision Avoidance in Uncertain Scenarios	76
10.1	Scenario of application	76
10.2	Robot and Pedestrian Dynamics	77
10.3	Chance-Constrained CBF	78
10.4	Application of CC-CBF: a wheelchair motion planner	81
10.5	Discussion	86
11	Hierarchical Motion Planner for Autonomous Agents based on Soft Control Barrier Functions	88
11.1	Microscopic Motion Planner: a Soft CBF Approach for Collision Avoidance	88
11.1.1	Comparison with hard CBF	93
11.1.2	Comparison with CC-CBF	93
11.1.3	Safety improvement	96
11.1.4	Dynamics of relaxation parameters	98
11.1.5	Including the attentiveness of pedestrians	99

11.1.6	Approximated Microscopic Planner	104
11.2	Macroscopic Motion Planner: how to move through a crowd	105
11.2.1	Macroscopic Features	106
11.2.2	Cells Score Function	107
11.2.3	Planning the Macroscopic path	109
11.3	Hierarchical double-layer Motion Planner	111
11.4	Simulations and Results	112
11.5	Discussion	115
Conclusions		
12	Conclusions	118
Appendices		120
A	Markov Chains	121
A.0.1	Continuous-Time Markov Chain (CTMC)	121
A.0.2	Discrete-Time Markov Chain (DTMC)	122
B	Interaction forces between agents	123
B.0.1	Attraction forces	123
B.0.2	Repulsion forces	123
B.0.3	Marginalized Model	123
C	Cell-Transmission Model	125
C.0.1	Cell Transmission Model	125
C.0.2	Modeling the input to the traffic scenario	125
D	Scenario-Tree MPC	126
E	Control Barrier Functions	127
F	Hierarchical Motion Planner Additional Simulation Results	128
G	Autonomous Wheelchair System Description	130

List of Figures

3.1	Indirect and Direct repulsion.	12
3.2	CTMCs for single agents and their network.	17
3.3	Agent decision probabilities and eigenvalues.	19
4.1	Evolution of decision-making processes for G_1 and G_2	28
4.2	Percentage variation of the price of different t-shirt colors.	28
4.3	Price trends of the t-shirts.	29
4.4	Decision-making process evolution of $\mathcal{I}_1$ and $\mathcal{I}_2$	33
4.5	Variation of interest rates.	34
4.6	Π_k as valid probability vector.	34
4.7	Tracking performances by EFC algorithm.	35
4.8	Π_k as valid probability vector with EFC.	35
6.1	Comparison between probability evolutions of the homogeneous and non-homogeneous DTMCs.	44
6.2	Comparison of CTM modal configurations under low and high arrival rates.	45
6.3	Evolution of the estimated DTMC configuration chain for high rate of arrivals.	46
6.4	Highway scenario.	48
6.5	Highway scenarios 1 and 2.	49
6.6	Highway scenario 3.	49
6.7	Highway scenario 4.	50
6.8	Highway scenario 5.	50
6.9	Inputs of the controlled vehicles.	55
6.10	Initial conditions of autonomous and human-driven vehicles.	56
6.11	Motion of autonomous vehicle.	56
6.12	Collision risk between autonomous and human-driven vehicles.	57
7.1	Two-lane highway scenario.	58
7.2	Control logics of Scenario-Based MPC.	59
7.3	Markovian behavioral and decision-making model block.	59
7.4	Speed evolution of autonomous and human-driven vehicles.	62
7.5	Lane trajectories of autonomous and human-driven vehicles.	62
7.6	Snapshots of simulation with Scenario-Based MPC.	63
9.1	Interaction between two players.	69
9.2	Comparison of trajectories changing the type of game between agents.	74
9.3	Distance to the nominal trajectory.	74
10.1	Scenario of application of social robot.	77
10.2	Comparison between CC-CBF and the conservative approach.	84
10.3	Snapshots of the trajectories obtained with CC-CBF and the conservative approach.	84
10.4	Distance between the robot and the obstacle, comparing CC-CBF with the conservative approach.	85
10.5	Inputs applied to the robot, comparing CC-CBF with the conservative approach.	85
10.6	Trajectories of the robot varying γ	86
10.7	Distance between the robot and the obstacle varying γ	86

11.1	Comparison between the trajectories obtained with hard and soft CBF. .	93
11.2	Snapshots of the simulation with hard CBF.	94
11.3	Snapshots of the simulation with soft CBF.	94
11.4	CC-CBF simulation.	95
11.5	Trajectory of the autonomous agent for varying probability thresholds. .	95
11.6	Evolution of the probabilistic constraint along the prediction horizon for varying probability thresholds.	96
11.7	Indices of unsafe trajectories at the first prediction instant for varying probability thresholds.	96
11.8	Snapshots of the simulation not including δ dynamics.	99
11.9	Snapshots of the simulation including δ dynamics.	100
11.10	Attentiveness coefficient heatmap.	101
11.11	Relative facing angle, distance, attentiveness coefficient, and weight on the slack variables.	102
11.12	Snapshots of the simulation.	102
11.13	Relative facing angle, distance, attentiveness coefficient, and weight on the slack variables for two obstacles scenario.	103
11.14	Snapshots of the simulation with two obstacles.	103
11.15	Snapshots of the simulation considering the two optimization problems cascade.	105
11.16	Discretized sensing area.	106
11.17	Density, kinetic energy and entropy layers.	107
11.18	Distribution of parameters for score function (11.35)	109
11.19	Distribution of parameters for score function (11.38)	109
11.20	Score and target functions of the macro path with bird-view.	110
11.21	Score and target functions of the macro path with sensor-view.	111
11.22	Hierarchical Controller block scheme.	111
11.23	Output of the macro planner considering bird-view approach.	113
11.24	Controlled trajectory of the robot considering bird-view approach.	113
11.25	Snapshots of simulation considering bird-view approach.	114
11.26	Robot's speed considering bird-view approach.	114
11.27	Control inputs considering bird-view approach.	114
11.28	Output of the macro planner considering the sensor-view approach.	115
11.29	Controlled trajectory of the robot considering the sensor-view approach. .	115
11.30	Snapshots of simulation considering the sensor-view approach.	115
11.31	Robot's speed considering the sensor-view approach.	116
11.32	Control inputs considering the sensor-view approach.	116
D.1	Scenario-Tree Model Predictive Control.	126
F.1	Macroscopic path evolution during the simulation considering the bird-view approach.	129
F.2	Macroscopic path evolution during the simulation considering the sensor- view approach.	129
G.1	Block scheme of the wheelchair.	130
G.2	Pictures of the autonomous wheelchair.	132

List of Tables

6.1	CTM configurations varying flows and arrival rate.	47
6.2	Vehicles' initial state.	55
7.1	Scenario-Based MPC simulation parameters.	61
9.1	Static game with 2 agents representing the game in Fig. 9.1.	70
11.1	Parameter estimation of score function $f_{C_i,k}$	108
11.2	Parameter estimation of score function $f_{C_i,k}^\theta$	108
F.1	Parameters employed in the simulations of the hierarchical controller. . .	128

Acronyms

- AV** *Autonomous Vehicle.* 39, 40, 118, 119
- AVs** *Autonomous Vehicles.* 38
- CAV** *Connected and Automated Vehicle.* 38
- CAVs** *Connected and Automated Vehicles.* 6, 38
- CBC** *Control Barrier Constraint.* 78, 79, 86
- CBF** *Control Barrier Function.* 7, 8, 76, 78, 83, 86, 87, 89, 103, 119
- CBFs** *Control Barrier Functions.* 76
- CC-CBF** *Chance-Constrained Control Barrier Function.* vii, 79, 81, 87, 88
- CTE** *Cross-Tracking Error.* 112
- CTM** *Cell-Transmission Model.* 41–47, 125
- CTMC** *Continuous-Time Markov Chain.* 5, 11, 15, 121, 122
- DTMC** *Discrete-Time Markov Chain.* 21–24, 26–29, 32, 41–47, 50, 54, 122
- EFC** *Explicit Feedback Control.* vii, 32, 33, 35
- LQR** *Linear Quadratic Regulator.* 6, 29, 30, 119
- MAS** *Multi-Agent Systems.* 5
- MC** *Markov Chain.* 5, 7, 8, 12, 36, 40, 53, 119, 121
- MDP** *Markov Decision Process.* 5, 6, 69
- MDPs** *Markov Decision Processes.* 11
- MLE** *Maximum-Likelihood Estimator.* 42, 43
- MPC** *Model Predictive Control.* 5, 6, 24, 29, 32, 33, 35, 36, 39, 60, 76, 118, 119, 126
- NN** *Neural Network.* 100, 105
- V2I** *Vehicle-to-Infrastructure.* 38
- V2V** *Vehicle-to-Vehicle.* 38

Preface

This thesis is submitted in partial fulfillment of the requirements for the degree of *Philosophiae Doctor* in Autonomous Systems (DAUSY).

List of Papers Written by the Author

International Journal Articles

- [J1] Heiker, C. - J., **Gaetan, E.**, Giarré, L., and Falcone, P., “Repulsive markovian models for opinion dynamics,” *Systems and Control Letters*, vol. 185, p. 105720, 2024.
- [J2] **Gaetan, E.**, Giarré, L., and Falcone, P., “Scenario-tree model predictive control for vehicle interactions in highway setting,” *IEEE Control Systems Letters*, vol. 8, pp. 1162–1167, 2024.
- [J3] Cimolato, C., **Gaetan, E.**, Pasotti, L., Schenato, L., and Bellato, M., “The essential role of ribosomal feedback in bacterial cells growth and metabolic load,” *submitted to IEEE Control Systems Magazine*, 2025.
- [J4] **Gaetan, E.**, Falcone, P., M.Bayen, A., and Giarré, L., “Soft control barrier functions for autonomous robots moving in human populated environments,” *submitted to IEEE Transactions on Systems, Man and Cybernetics*, 2025.

International Conference Proceedings

- [C1] **Gaetan, E.**, Giarré, L., Sacone, S., Falcone, P., and Heiker, C. - J., “Modelling traffic scenarios via markovian opinion dynamics,” in *2023 IEEE 26th International Conference on Intelligent Transportation Systems (ITSC)*, 2023, pp. 1072–1077.
- [C2] **Gaetan, E.**, Giarré, L., Sacone, S., and Falcone, P., “Control of decisions in stochastic multi-agent systems,” in *2024 European Control Conference (ECC)*, 2024, pp. 2356–2361.
- [C3] **Gaetan, E.**, Giarré, L., and Falcone, P., “Feedback control in multi-agent markovian networks,” in *2024 IEEE 20th International Conference on Automation Science and Engineering (CASE)*, 2024, pp. 1264–1269.
- [C4] **Gaetan, E.**, Giarré, L., and Falcone, P., “A behavioral model for interacting vehicles in highway scenarios,” in *2024 IEEE 27th International Conference on Intelligent Transportation Systems (ITSC)*, 2024, pp. 633–638.
- [C5] Bianchin, F., Lefringhausen, R., **Gaetan, E.**, Tesfazgi, S., and Hirche, S., “A set-theoretic robust control approach for linear quadratic games with unknown counterparts,” *accepted and to appear in 2025 IEEE Control and Decisions Conference (CDC)*.
- [C6] **Gaetan, E.**, Jeddi, M., Falcone, P., and Giarré, L., “Human-Attention-Aware Motion Planning for Social Robots,” *submitted to 2026 European Control Conference (ECC)*.

Oral Presentations, Posters and Workshops

- [P1] **Gaetan, E.**, Giarré, L., Sacone, S., Falcone, P., and Heiker, C.- J., “Modeling traffic scenarios via Markovian Opinion Dynamics,” in *Automatica.it*, Catania, Italy, 2023.
- [P2] **Gaetan, E.**, “Exploiting Opinion Dynamics Models for Traffic Predictions (Workshop),” in *2023 IEEE 26th International Conference on Intelligent Transportation Systems (ITSC)*, Bilbao, Spain, 2023.
- [P3] **Gaetan, E.**, Giarré, L., and Falcone, P., “Scenario-Tree Model Predictive Control fro vehicle interactions in Highway setting,” in *Automatica.it*, Bolzano, Italy, 2024.
- [P4] **Gaetan, E.**, Jeddi, M., Mattioli, A., Falcone, P., and Giarré, L., “Hierarchical Motion Control for Autonomous Wheelchairs moving in a Crowded Environment,” in *I-RIM Conference*, Rome, Italy, 2025.
- [P5] **Gaetan, E.**, Falcone, P., and Giarré, L., “Autonomous Robot Safety Moving in a Crowded Human Populated Environment,” in *Automatica.it*, Perugia, Italy, 2025.

Acknowledgments

This PhD journey has meant a great deal to me. It was not just a job or a period spent at the university — it was an opportunity to discover a new world, filled with wonderful people and experiences I could have never dreamed of. I am grateful for every moment: for the difficult ones, which taught me resilience, and for the joyful ones, which brought deep satisfaction and happiness.

First, I would like to thank my supervisors, Prof. Laura Giarre and Prof. Paolo Falcone, for their invaluable guidance, support, and encouragement throughout these three years. Their advice has shaped my professional growth, and their trust has allowed me to experience wonderful opportunities.

From the Università di Modena e Reggio Emilia, I would also like to thank my colleagues, and most importantly, my friends from the Autolab group and from Ideativa. Sharing both the office and many moments of free time with you has been one of the greatest joys of this journey. A special thanks to Dario, Niccolò, Matteo, Giovanni and Davide: everything we shared together has made these three years truly special.

I am also deeply grateful to my host research group at UC Berkeley. My sincere thanks to Prof. Alexandre M. Bayen for giving me the incredible opportunity to join his lab. I am thankful as well to all the wonderful people I met there, with whom I shared both stimulating discussions in the lab and memorable football matches memories. A special thanks to Arwa and Zhe, who always made me feel welcome.

I would also like to thank my co-author, Prof. Simona Sacone, who introduced me to the world of conferences for the first time. A special thanks goes to Valeria, with whom I shared so much over these years, from PhD bureaucracy to conference adventures.

My heartfelt thanks also go to my friends, who have always been by my side and supported me through everything. Thank you for the laughter, the jokes, and the calls, even when we were far apart. A special mention to Mariangela and Francesco, who taught me how to stand strong and face challenges with courage.

Finally, I would like to thank Konstantin, for his unconditional love and support, and my family, my parents and Anna, for making me feel supported every single day.

Thank you all!

Introduction

Chapter 1

Introduction

1.1 Overview

The world is characterized by millions of entities, both living and inanimate, that interact daily with one another — either with peers, entities of different species, or the surrounding environment. These entities are generally modeled as complex systems of various types, which experience interactions that can be social, technological, biological, or economic in nature. In such systems, the behavior of a single entity is influenced by that of the group, and vice versa. Capturing, modeling, and understanding how these interactions affect the overall system is therefore a central problem across multiple disciplines. Understanding individual and collective behaviors is not only relevant in engineering applications but also in the social sciences, where, for example, it provides insights into how people relate to one another and how connections can be leveraged to disseminate information. It is equally significant in biological research, where it aids in studying how animals cooperate in nature or preserve their species, and in medical studies, where it contributes to understanding phenomena such as bacterial aggregation or how drugs can interfere with these mechanisms. From an engineering perspective, studying interactions—particularly within groups—is essential for exploiting them to achieve coordinated behavior, as in opinion dynamics, or for optimizing large-scale systems, such as in traffic scenarios. In general, this is a fundamental challenge that spans disciplines including physics, computer science, economics, and systems theory. Agents adapt their decisions based on partial information received from the environment and their surroundings, leading them to change their goals or strategies. This is particularly evident in human and animal agents. Such adaptive dynamics can generate a rich variety of emergent behaviors, including cooperation, competition, consensus, and polarization, all of which profoundly influence the efficiency, stability, and resilience of the system as a whole.

However, modeling these dynamics is often challenging, as it cannot be assumed that all agents act according to the same rationale. Moreover, most of the relevant information about them—such as their objectives, traits, or level of understanding of the scenario—is typically unknown. Hence, it is necessary to balance descriptive realism with analytical tractability. Deterministic models, in this sense, may lack detail, as they account for only a single possible outcome and often rely on complete knowledge of the system’s parameters and states, which is difficult to obtain in large-scale environments or when dealing with living and reasoning agents. Stochastic frameworks therefore offer a compelling alternative, as they naturally incorporate uncertainty arising from sensor noise, incomplete understanding, or heterogeneity in how agents reason. Among these, a particularly appealing class of stochastic models is the Markovian framework, which is both linear and computationally tractable. It provides a natural and versatile way to represent decision-making processes, where states correspond to the agents’ decisions. By describing agents as probabilistically transitioning between states according to local interaction rules, such models capture essential features of collective decision-making and opinion evolution while maintaining analytical simplicity. This approach enables the study of how local behaviors and information exchange give rise to macroscopic patterns observable at the population level.

A natural question arises when considering Markovian models: how can the model be structured so that interactions among agents are properly represented? In this framework, interactions are typically captured through modifications of the transition rates governing the system dynamics. These modifications can follow a specific structure. For instance, processes of attraction and agreement tend to promote alignment among agents, whereas mechanisms of repulsion, disagreement, or competition can preserve diversity or lead to

fragmentation within the group. In this context, reaching consensus or disagreement is not always the ultimate goal. Rather, understanding these mechanisms is valuable both for explanatory purposes and for guiding or influencing system behavior. Many applications, in fact, require such models to be employed as tools for steering decisions—at least in a probabilistic sense, given the stochastic nature of the framework—toward certain states or configurations. Consequently, the relationship between modeling and control becomes a central theme in the analysis of interacting-agent systems. It is therefore important to clarify what *control* means in this context. It is neither reasonable nor ethical to assume that the decisions of individual agents can be deterministically controlled. The objective is instead to construct a model that represents, at least, the average behavior of the agents, and to identify a suitable control strategy that can probabilistically guide this representative agent toward desired states. This outcome is, of course, not guaranteed, as it depends on each agent’s individual rationale. Nonetheless, similar mechanisms are already applied in various real-world settings—for example, in marketing strategies designed to influence consumer behavior. The purpose of such models is analogous: to represent the decision processes of agents and, through a suitable control mechanism, determine an effective control action to influence their probabilistic evolution.

Moreover, this type of control strategy should not be understood as being limited to marketing applications. Indeed, the ability to characterize and influence decision-making processes is equally valuable when integrating autonomous agents into human-centered environments.

In this sense, the domain of transportation provides a concrete example of such collective dynamics. Road networks can be interpreted as large-scale systems composed of interacting decision-makers whose local choices determine global outcomes such as traffic flow, safety, and environmental impact. Modern mobility systems generate considerable social and economic benefits but also impose significant external costs, including congestion, emissions, and accidents. Traditional management strategies—based on static control measures or centralized regulation—often lack the flexibility required to address the growing complexity of road traffic. The emergence of connected and automated vehicles has transformed this landscape by introducing agents capable of autonomous decision-making, communication, and adaptation. Each vehicle thus becomes both a participant in and a regulator of collective traffic behavior, giving rise to new forms of coordination and interaction. Modeling such systems therefore, demands approaches that can accommodate heterogeneity, uncertainty, and feedback effects among agents.

These considerations extend beyond transportation and characterize a wide variety of multi-agent systems. Real-world environments typically involve diverse populations of agents that differ in objectives, capabilities, and available information. Capturing this diversity is essential for describing global system properties and for designing policies that remain effective under uncertainty. Stochastic, data-driven, and predictive modeling frameworks provide a promising direction for analyzing and controlling such systems, as they combine statistical inference with decision-making principles. The underlying challenge is to understand how local, decentralized interactions can be harnessed to produce globally coherent, efficient, and stable outcomes across domains ranging from mobility and communication networks to distributed computation and socio-economic systems.

A further example of multi-agent interaction arises in the context of human-machine and human-robot collaboration. As autonomous systems increasingly operate in shared human spaces, it becomes imperative to design control and motion algorithms that can both perceive and predict human behavior, in addition to responding promptly. In these settings, interaction extends beyond physical coordination to encompass social understanding and communication. Robots navigating among pedestrians or collaborating with humans must respect implicit social norms, maintain appropriate interpersonal distances, and communicate their intentions in a legible and socially acceptable manner. Achieving such behavior requires models that integrate motion planning, prediction, and social reasoning within a unified framework.

The mutual influence between human and machine actions introduces specific

challenges. Classical planning methods that treat other agents as passive or non-responsive entities fail to capture the reciprocity inherent in social interactions. More advanced frameworks model this reciprocity explicitly, often through game-theoretic or stochastic approaches that account for the interdependence of decisions. However, these formulations frequently encounter scalability and generalization issues, as the dimensionality of possible interactions grows rapidly and human behavior exhibits substantial variability across contexts. Despite these difficulties, studying coupled decision-making remains fundamental for enabling safe, efficient, and socially compatible autonomous systems.

At a broader level, the study of interacting agents addresses a fundamental scientific question: how do simple local rules give rise to complex collective behavior? Across natural and artificial systems, similar principles recur. Agents operate with limited and noisy information, adapt their strategies based on local feedback, and collectively generate large-scale structures and dynamics. These emergent behaviors may exhibit stability, oscillations, or abrupt transitions depending on how information propagates and how interactions are organized. Understanding such processes is critical for explaining self-organization and for designing systems that exploit, rather than resist, emergent dynamics. In engineered settings, this knowledge supports the development of distributed algorithms and control mechanisms that operate effectively under decentralization and uncertainty.

However, another challenge must be addressed. Currently, robots are solely responsible for ensuring movement safety. In scenarios where autonomous agents transport or guide humans, always requiring the robot to take longer paths to accommodate human movement may not be fair or efficient. Therefore, a central question arises: is it possible to design a controller that, based on predictions of the motion of surrounding agents, can strategically exploit their behavior to be more or less conservative, assuming a degree of cooperation from humans?

Research on autonomous and interacting agents thus occupies an interdisciplinary position, bridging mathematics, control theory, artificial intelligence, behavioral science, and the social sciences. Progress in this field depends on integrating methods from these areas while maintaining an appropriate balance between abstraction and empirical realism. Models must remain sufficiently simple to yield insight, yet sufficiently expressive to capture the essential characteristics of real-world interactions. Likewise, strategies for influencing or coordinating agents must respect the decentralized nature of decision-making while leveraging local information to achieve system-level objectives.

In summary, the investigation of autonomous agent interactions provides a unifying framework for understanding and shaping complex systems. From collective decision-making and opinion dynamics to mobility management and social robotics, the same fundamental questions persist: how do local interactions determine global outcomes, how can uncertainty and heterogeneity be incorporated into predictive frameworks, and how can these systems be guided toward efficient, stable, and socially desirable behaviors? Addressing these questions advances both theoretical understanding and practical capability, enabling the design of systems that combine autonomy, adaptability, and social awareness within an increasingly interconnected world.

1.2 Related Works

Research on multi-agent systems and collective behavior spans diverse domains, from social dynamics to intelligent transportation and human–robot interaction. A central theme across these fields is understanding how individual decision-making and interaction mechanisms give rise to emergent group behavior—and how such processes can be modeled, predicted, and controlled.

Early research in opinion dynamics provided foundational insights into how agents influence one another and how consensus or polarization emerges. Markovian frameworks have been particularly influential, enabling the probabilistic description of opinion evolution and interaction among agents [1], [2], [3].

In these works, the authors proposed to model opinion influence in social networks by exploiting a *Markov Chain* (MC), which, despite its simplicity and linearity, offers considerable flexibility. Along with the model formulation, they also introduced a framework to interpret and structure the interactions among agents, each governed by its own MC, in order to derive a collective network-level interaction model. Once this model was obtained, they demonstrated how it could be marginalized, that is, reduced in state dimension, thereby showing that the state of a single agent within the network can be inferred from the overall network configuration. Furthermore, they conducted a transient analysis of the MC, considering both unbiased and biased influence processes.

An alternative formulation aimed at modeling the evolution of opinions in a social network is presented in [4], where the interactions are represented as connections in a weighted graph, and the influence matrix encodes the strength of these connections. Moreover, in [5], the authors modeled the interactions among agents in a network as a weighted graph as well, but introduced the presence of clusters, where members within the same cluster cooperate, while different clusters exhibit antagonistic or opposing behaviors.

Beyond their descriptive role, stochastic models have also inspired control-oriented approaches. In [6], the authors proposed a framework based on a *Multi-Agent Systems* (MAS) with a leader–follower structure, and designed an algorithm for leader selection aimed at minimizing the convergence time to a unique consensus state. In [7], a similar MAS-based framework was introduced, where the complex system, or social network, is represented by a graph. The objective is to identify the subset of nodes to be controlled through suitable inputs, in order to steer the network toward consensus. In [8], the control of leader–follower networks was further extended to account for time-varying interactions. Although these approaches are highly interesting, they do not directly address the case of social networks that do not necessarily aim to reach consensus. Moreover, identifying a clear leader is not always possible. To illustrate this, consider a traffic scenario: one might assume that the leading vehicle is always the car in front. However, this is not necessarily true. For instance, if a following vehicle initiates an overtaking maneuver, its action can trigger a reaction in the car ahead. In that situation, the follower effectively becomes the leader. Additionally, game-theoretic and *Markov Decision Process* (MDP) [9] formulations have also been proposed to steer collective outcomes. In [10], opinion dynamics are modeled through mean-field games, where convergence is sought not toward a strict consensus, but toward an equilibrium state with the property that no agent can improve its outcome by unilaterally deviating from it.

More recently, opinion dynamics models have been employed directly for the probabilistic control and manipulation of decision processes. In [11], the authors proposed a mathematical framework to represent different forms of opinion manipulation, and investigated how the underlying network structure can amplify or attenuate the influence of an external input aimed at steering the agents’ opinions.

These ideas have been fruitfully applied to traffic systems, where vehicle interactions and driver decisions can be viewed as analogous to social influence processes. Markovian and stochastic models have been used to represent lane-selection behavior, vehicle arrivals, and driver cooperation.

In [12], [13] and [14], Markovian models are employed to predict the behavior of traffic agents. In the first work, a *Continuous-Time Markov Chain* (CTMC) is used to model and predict agents’ behavior at intersections, whereas the second focuses on the evolution of a highway environment, still relying on MC to forecast traffic congestion. These studies represent some of the few examples of the use of MC in the field of traffic optimization and control, where the prediction of system-level conditions through the modeling of individual decision processes has not yet been extensively explored.

Despite these developments, more traditional control strategies—such as speed regulation, lane-changing guidance, and *Model Predictive Control* (MPC)—have long been central to traffic management. In [15], a comprehensive review of lane-changing models established the foundations for understanding the factors influencing drivers’ lane-changing behavior, emphasizing the critical role of peer influence. The main distinction among

existing lane-changing models lies in their underlying strategy. They can be categorized as *driving-assistance models*, primarily focused on preventing collisions and supporting the driver during lane-change maneuvers, or as *driving-decision models*, where either an advisory signal is provided to the driver or a control action is executed directly, in the case of autonomous vehicles. In both cases, the decision process is strongly conditioned by the surrounding environment.

In [16], a mainstream feedback controller was designed based on variable speed limits along a highway. However, this controller does not incorporate predictions of drivers' behavior and assumes full compliance with the prescribed speed limits. In contrast, in [17], the authors proposed a robust MPC for lane keeping and obstacle avoidance, explicitly considering an uncertain driver model to predict vehicle trajectories. In [18], the focus is on the design of an MPC-based controller for a single vehicle traveling on a multi-lane road, with the goal of improving anticipatory behavior and achieving smoother driving. The vehicle is assumed to be connected to the environment, thus capable of receiving information about surrounding traffic. In [19], the concept of variable speed limits introduced in [16] is extended to include lane-changing capabilities, allowing the controller to generate upstream lane-change recommendations in the presence of bottlenecks. Finally, in [20], the optimal trajectory planning problem near a highway exit is addressed, where a decision manager determines the most suitable target lane.

However, their scalability is limited by infrastructure-based actuation. The advent of *Connected and Automated Vehicles* (CAVs) offers a more flexible platform for real-time, distributed control, enabling fine-grained coordination based on predictive or data-driven models. Related studies have also explored the integration of MDPs and reinforcement learning for lane-following and cooperative driving. In [21], [22], the authors proposed a controller based on MDP, that is then applied for lane following application, focusing on collision avoidance and not leaving the street at any point. However, the prediction of the human driver decision-making remains a key open challenge.

A parallel line of research addresses navigation and collision avoidance in crowded environments, where interactions with pedestrians or other dynamic obstacles introduce substantial uncertainty. Classical optimization-based methods, such as LQR and MPC, ensure safety through hard or barrier-based collision-avoidance constraints, whereas alternative frameworks rely on predictive models of human behavior. In [23], an iterative constrained LQR controller is proposed for autonomous driving, where the constraints are incorporated into the optimization problem as components of the cost function after linearization. Although this approach is computationally efficient and intuitive, embedding collision-avoidance constraints solely within the cost function compromises their hard-constraint nature. Another example is provided in [24], where an adaptive constrained iterative LQR is used for motion-planning problems of autonomous vehicles. Here, the constraints are again treated within the cost function after linearization, and the authors further highlight the challenge of parameter tuning across varying scenarios, proposing an adaptive weight-tuning mechanism. Notably, these controllers do not explicitly model the behavior of obstacles.

Alternative formulations, based on energy minimization and social-force principles, aim to capture human objectives and mutual influences. A seminal work in this direction is [25], where social interactions among pedestrians are modeled through social forces. While conceptually appealing, this approach requires knowledge of quantities that are often unavailable, such as each pedestrian's goal or desired speed. An extension is presented in [26], where the framework is generalized from deterministic to stochastic by assuming a probability distribution over pedestrian velocities, thus accounting for multiple hypotheses.

Other approaches employ learning-based and data-driven frameworks, which often rely on pre-recorded trajectories. In [27], an algorithm for socially aware robot navigation is proposed, where the next obstacle trajectory is estimated based on inferred personality traits derived from a pre-recorded dataset. However, such models are highly dependent on the specific training scenario and tend to generalize poorly to unseen environments

or behaviors. Game-theoretical methods have also been explored to model cooperative interactions and anticipate human responses. In [28], a motion planner for drones is presented, based on dynamic potential games and the computation of the generalized Nash equilibrium. Similarly, in [29], a motion planner for autonomous vehicles operating in dynamic environments with moving obstacles—such as pedestrians or other vehicles—is developed, relying on a stochastic model that captures the possible choices of the obstacles. Despite their theoretical appeal, these approaches often depend on strong assumptions about human intent. In particular, they typically require prior knowledge of the objective functions guiding each agent and assume that all participants in the interaction pursue a Nash equilibrium. This assumption does not always hold in practice, especially in scenarios involving selfish, unaware, or unpredictable agents.

More recent probabilistic *Control Barrier Function* (CBF) formulations incorporate uncertainty directly into the safety guarantees, thereby bridging the gap between deterministic optimization and stochastic reasoning. Examples can be found in [30], [31], [32], which implement Chance-Constrained CBFs—stochastic extensions of standard CBFs—where the constraint enforces a desired probability of safety rather than a strict distance-based condition. While these methods enable the evaluation of multiple possible obstacle trajectories, they suffer from a critical limitation: their performance is highly sensitive to parameter tuning, which significantly affects robustness and practical deployment.

Across these domains, a unifying research frontier lies in developing stochastic, interaction-aware models that capture the coupling between agents’ decisions and their environment, while supporting predictive and control capabilities. Markovian and game-theoretical approaches provide a shared mathematical language for studying such systems—whether they involve opinions in social networks, vehicles in traffic flows, or humans in crowded spaces. The convergence of these perspectives points toward a generalized framework for multi-agent decision-making under uncertainty, where control, prediction, and interaction modeling are jointly integrated.

1.3 Contributions

This thesis presents two main contributions that are closely interconnected. The first concerns the problem of modeling, inferring, and forecasting agents’ decisions. In particular, the first part of this dissertation focuses on developing a simple yet effective model, based on a MC, to represent the evolution of agents’ decisions within a network. This framework is analyzed in two contexts: social networks, where agents are interconnected with each other or influenced by the environment, and traffic scenarios, where a decision-making model for human drivers is proposed based on scenario-specific rules and common behavioral observations.

The second contribution addresses how these decision-making models can be integrated within a control framework. Specifically, the models developed in the first part are incorporated into a control loop—particularly in traffic scenarios—where an autonomous agent acts as the actuator of the control strategy. Furthermore, an alternative controller is designed for a pedestrian environment, focusing on collision avoidance under stochastic predictions of human obstacle trajectories.

1.4 Thesis organization

This dissertation is organized into three main parts: *Opinion Dynamics*, which analyzes modeling and control schemes in the field of opinion dynamics, focusing on the evolution of opinions in social networks; *Mixed-Autonomy Traffic*, which extends similar concepts to traffic scenarios for forecasting and controlling vehicle behaviors; and *Social Robotics*, which addresses the problem of controlling an autonomous agent in pedestrian environments. Considering this structure, the content of each chapter is summarized as follows:

- Chapter 2 introduces the *Opinion Dynamics* section and reviews related works;
- Chapter 3 presents a framework for describing interactions in a multi-agent network based on attraction and repulsion forces;
- Chapter 4 develops a framework that models interactions between groups of agents and their surrounding environment, proposing external strategies for steering agents' decisions;
- Chapter 5 introduces the *Mixed-Autonomy Traffic* section and reviews related works;
- Chapter 6 presents both macroscopic and microscopic frameworks based on MC for forecasting traffic scenarios;
- Chapter 7 proposes a control framework that leverages the microscopic forecasting model to steer drivers' decisions in highway scenarios;
- Chapter 8 introduces the *Social Robotics* section and discusses relevant literature;
- Chapter 9 analyzes the motion planning problem for autonomous agents from a game-theoretical perspective;
- Chapter 10 introduces the first version of a CBF-based controller for uncertain pedestrian scenarios;
- Chapter 11 presents a hierarchical motion planner enabling autonomous agents to navigate crowded environments;
- Chapter 12 concludes the dissertation, summarizing the main findings and insights.

Opinion Dynamics

Chapter 2

Modeling Agents Interactions with Markov Chains

Overview

In many domains, ranging from market economies to social networks, politics, and vehicular traffic, individuals or groups are continuously required to make decisions. While the decision-making process of an individual can be realistically assumed to depend on personal objectives and status, it is also influenced by interactions with the surrounding environment and other agents. Within this context, models for forecasting decisions based on such interactions are of particular interest.

In the following two chapters, the focus will first be on modeling interaction forces between agents, leading to different formulations that can be analyzed to highlight their distinguishing features. In the second part, interaction models between agents and the environment will be developed, with an emphasis on their potential to influence individuals' opinions, thereby reflecting simplified marketing strategies. Finally, an explicit feedback law for this model will be derived.

2.1 Introduction

Collective decision-making is a fundamental process that arises in many domains, including markets, politics, transportation systems, and smart infrastructures. In such contexts, groups of agents interact, communicate, and influence one another, often leading to complex patterns of opinion formation and decision evolution. These dynamics are difficult to capture, particularly in large decentralized systems where uncertainty and incomplete information play a crucial role. Deterministic models, while valuable in theory, often require levels of detail that are not practically attainable, motivating the need for stochastic abstractions.

Markovian models provide a natural framework for describing decision processes in which agents switch between alternative states according to probabilistic rules. Such models can capture essential interaction mechanisms—such as consensus, disagreement, and influence—while remaining analytically tractable. They are especially useful when the aim is to study how localized interactions scale into global collective behavior.

A central challenge in this area is to understand how different forms of interaction shape the evolution of opinions. Agreement and attraction mechanisms may drive agents toward alignment, while disagreement and repulsion mechanisms can prevent consensus or create polarization. Beyond modeling, another key challenge lies in influencing collective dynamics: in many applications, it is necessary not only to observe how opinions evolve but also to design strategies that guide the system toward desired behaviors.

This part of the dissertation addresses these challenges by focusing on the stochastic modeling of opinion evolution and on the control of collective decision processes. The goal is to develop frameworks that balance descriptive power with analytical tractability, enabling both the understanding of natural interaction patterns and the design of interventions that influence outcomes.

2.2 Related Works

Research on opinion dynamics has developed along multiple directions, with an emphasis both on understanding the natural evolution of opinions and on shaping collective outcomes. A first line of work has concentrated on modeling how agents influence one

another. Markovian frameworks have been widely adopted in this regard, as they allow the evolution of decisions to be described through probabilistic transitions between states. These models have been used to explain how opinions spread in social networks [2], [3], how persistent minorities or stubborn agents affect consensus formation, and how the structure of a network influences the emergence of social power. Analytical reductions of these models have also been derived, providing tools for studying large-scale systems in a tractable way [1], [4].

Applications of these ideas extend beyond social networks. In transportation, for instance, *Continuous-Time Markov Chain* (CTMC) has been employed to describe how vehicles arrive at and depart from lanes, or how drivers make stochastic decisions at intersections [12], [13], [14]. This latter one also has enriched the modeling framework by explicitly incorporating attraction and repulsion mechanisms between agents, highlighting how agreement and disagreement can be viewed as forces that alter transition rates between states. Such models provide insight into the micro-level processes that underpin complex collective behavior.

Alongside descriptive studies, a growing body of research has addressed how to intervene in these systems. Strategies for influencing collective behavior have been explored through controllability analyses and leader selection, where certain agents are identified as leverage points for steering the network [6], [7], [8]. Alternative perspectives have emerged from game theory, where the interaction of rational agents is formalized as a strategic game [10], and from consensus theory, where the focus is on driving distributed systems toward coordinated outcomes [5]. *Markov Decision Processes* (MDPs) have also been employed to model stochastic decision-making under external control [9], offering a well-established framework for optimizing policies. More recently, opinion dynamics models themselves have been used not only for description but also as a means of control, with works demonstrating how Markovian formulations can be exploited to manipulate decision probabilities and design targeted interventions [11], [33].

The present work is situated within this landscape, bridging descriptive and control-oriented perspectives. It draws on the tradition of Markovian modeling of opinion dynamics while also aligning with recent efforts that view these systems not merely as phenomena to be observed but as processes that can be shaped through systematic interventions.

2.3 Contribution

As stated in the above introduction, the formulation of a stochastic model of opinion dynamics is intended to support the analysis and influence of opinions in single- and multi-agent networks. Within this framework, Markovian models provide a simple yet powerful linear representation that can be effectively exploited. Building on this foundation, the contributions of the next two chapters are summarized as follows:

- The formulation of a *direct repulsion force* as an interaction mechanism between agents, and its comparison with the *indirect repulsion force* through the property of *similarity* - Chapter 3;
- The development of a framework for influencing opinions through agent interactions - Chapter 4, Section 4.1;
- The derivation of an explicit feedback law solution, obtained by applying the above influence framework - Chapter 4, Section 4.2.

Chapter 3

Repulsive Models for Interacting Agents

In this Chapter, the problem of modeling the decision-making process within a network of agents is analyzed using a *Markov Chain* (MC), specifically in continuous time. Consequently, each agent is referred to as a *Markovian agent*; additional details are provided in Appendix A. The interactions among agents are interpreted as social forces, which can be either attractive or repulsive, as discussed in [14].

Focusing on the case of repulsive forces, two different modeling approaches are proposed and compared, with the objective of highlighting their similarities and potential advantages.

Relevant preliminaries from previous works are summarized in Appendix B.

The contents of this chapter are mainly derived from the work presented in [33].

3.1 Problem Definition

Before introducing the proposed framework, a simple illustrative example is presented to clarify the underlying concept of repulsion. In this example, two interpretations of repulsion are considered: *direct* and *indirect*. While the concept of indirect repulsion has already been introduced in [14], the notion of direct repulsion is here formulated and compared against it.

Example 3.1.1

Consider two drivers, a and b , shown in Fig. 3.1, who must choose between two highway lanes, L_1 and L_2 . Indirect repulsion causes driver a to choose L_2 with a higher frequency since b is not in L_2 (Fig. 3.1a). Conversely, direct repulsion leads a to select L_1 with a lower frequency when b is present in L_1 (Fig. 3.1b).

3.2 Repulsive Forces

In the following, the indirect repulsion force introduced in [14] is recalled, and an alternative direct formulation for modeling disagreement between stochastic agents is proposed.

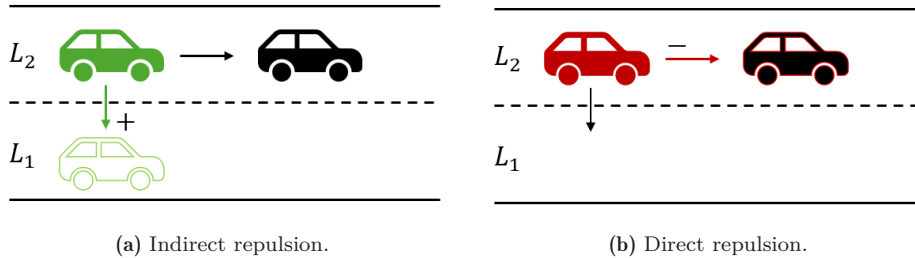


Figure 3.1: Drivers a and b decide between lanes L_1 and L_2 . Indirect repulsion increases the rate of different decisions, while direct repulsion decreases the rate of identical decisions.

3.2.1 Indirect repulsion

The indirect repulsion force is expressed as

$${}^+\xi_j^n(\mathcal{R}_\ell, \gamma) = \gamma \sum_k \Gamma_{\mathcal{R}_\ell}^{\mathcal{A}}[n, k](1 - \mathbf{1}_j^k) \quad (3.1)$$

and describes how the n :th agent, a member of the group $\mathcal{A}$, *increases* its transition rate to s_j . The increase is directly dependent on which agents, indexed by k , in the group $\mathcal{R}_\ell$ are in states *different from* s_j , an event denoted by the negated indicator function $(1 - \mathbf{1}_j^k)$.

The sets $\mathcal{A}$ and $\mathcal{R}_\ell$ are disjoint subsets of $\mathcal{N}$ that interact according to the directed graph $G_{\mathcal{R}_\ell}^{\mathcal{A}} = (\mathcal{N}, \Gamma_{\mathcal{R}_\ell}^{\mathcal{A}})$, where $\Gamma_{\mathcal{R}_\ell}^{\mathcal{A}}$ is an $N \times N$ row-normalized, non-negative real adjacency matrix representing the agent-to-agent influence strength. A group $\mathcal{A}$ can be repulsed by several disjoint groups $\mathcal{R}_\ell \in \mathcal{R}$, with each conflict described by a unique interaction graph. Moreover, each group can simultaneously be repulsed by other groups (assuming the role of $\mathcal{A}$ in one conflict) and repulse other groups (assuming the role of $\mathcal{R}_\ell$ in another).

The parameter $\gamma \geq 0$ determines the magnitude of the repulsion force in proportion to the weighted average decisions of $\mathcal{R}_\ell$. It may depend on both time-independent traits, such as the agent index n and the pair $(\mathcal{A}, \mathcal{R}_\ell)$, as well as on the time-dependent network state $s^\mathbb{X}$. However, state-independence is required for analytical marginalization to be feasible. This property is referred to as *marginalizability*.

To incorporate indirect repulsion into the network model described by (A.3), an $M^Z \times M^Z$ network transition rate matrix R^+ is defined, following the same indexing convention as Q . If a network transition from the a :th to the b :th state in $\mathcal{S}_\mathbb{X}$, denoted $s_a^\mathbb{X} \rightarrow s_b^\mathbb{X}$ with $a \neq b$, is caused by the n :th agent's transition $s_i \rightarrow s_j$, then

$$R^+[a, b] = \sum_{\mathcal{R}_\ell \in \mathcal{R}} {}^+\xi_j^n(\mathcal{R}_\ell, \gamma | s_a^\mathbb{X}, s_b^\mathbb{X}).$$

While the indicator functions in (3.1) are random variables, they can be evaluated deterministically given the network states $s_a^\mathbb{X}$ and $s_b^\mathbb{X}$, such that $R^+[a, b]$ becomes a non-negative real scalar. Each diagonal element in R^+ is the negated sum of the off-diagonal elements in the corresponding row.

A network modeled with indirect repulsion is denoted by $\mathbb{I}$, and its decision probabilities $\Pi_\mathbb{I}(t)$ are obtained by solving

$$\dot{\Pi}_\mathbb{I}(t) = (Q + R^+)^T \Pi_\mathbb{I}(t), \quad (3.2)$$

where $\dot{\Pi}_\mathbb{I}(t)$ denotes the first time derivative of $\Pi_\mathbb{I}(t)$, starting from an initial condition $\Pi_\mathbb{I}(0)$.

3.2.2 Direct repulsion

As an alternative formulation, the *direct* repulsion force is defined as

$${}^-\xi_j^n(\mathcal{R}_\ell, \gamma) = \gamma \sum_k \Gamma_{\mathcal{R}_\ell}^{\mathcal{A}}[n, k] \mathbf{1}_j^k. \quad (3.3)$$

This force, denoted by "-" in contrast to (3.1) (denoted "+"), directly *reduces* the n :th agent's transition rate to s_j , depending on which agents, indexed by k , are in s_j . To prevent negative transition rates, the parameter γ must be appropriately constrained, as discussed next.

Assuming state-independent influence parameters γ , consider the case in which the n :th agent experiences maximal repulsion, i.e., every group $\mathcal{R}_\ell$ has all its agents in s_j such that $\mathbf{1}_j^k = 1$ for all k and $\mathcal{R}_\ell$. Since each $\Gamma_{\mathcal{R}_\ell}^{\mathcal{A}}$ is row-normalized, the total rate reduction is

$$\sum_{\mathcal{R}_\ell \in \mathcal{R}} {}^-\xi_j^n(\mathcal{R}_\ell, \gamma) = \sum_{\mathcal{R}_\ell \in \mathcal{R}} \gamma.$$

If this value is less than or equal to the minimum isolated transition rate $q_{\min}^n = \min_{i \neq j} Q_n[i, j]$, none of the n :th agent's transition rates become negative. Assuming that γ depends on n and $(\mathcal{A}, \mathcal{R}_\ell)$, the following condition must hold:

$$\sum_{\mathcal{R}_\ell \in \mathcal{R}} \gamma \leq q_{\min}^n, \quad \forall n. \quad (3.4)$$

Analogously to indirect repulsion, an $M^Z \times M^Z$ transition rate matrix R^- is defined to describe direct repulsion in the network model (A.3). A network transition from the a :th to the b :th state in $\mathcal{S}_\mathbb{X}$, with $a \neq b$, caused by the n :th agent's transition $s_i \rightarrow s_j$ obtains the rate reduction

$$R^-[a, b] = \sum_{\mathcal{R}_\ell \in \mathcal{R}} -\xi_j^n(\mathcal{R}_\ell, \gamma | s_a^\mathbb{X}, s_b^\mathbb{X}).$$

Each diagonal element in R^- equals the negative sum of the off-diagonal elements in the corresponding row. A network described by direct repulsion is denoted by $\mathbb{D}$, and its decision probabilities $\mathbf{\Pi}_\mathbb{D}(t)$ are obtained by solving

$$\dot{\mathbf{\Pi}}_\mathbb{D}(t) = (Q - R^-)^T \mathbf{\Pi}_\mathbb{D}(t), \quad (3.5)$$

where $\dot{\mathbf{\Pi}}_\mathbb{D}(t)$ denotes the first time derivative of $\mathbf{\Pi}_\mathbb{D}(t)$, starting from an initial condition $\mathbf{\Pi}_\mathbb{D}(0)$.

Remark 3.2.1

Unlike indirect repulsion, direct repulsion must be constrained to avoid negative transition rates, thus requiring knowledge of $q_{\min}^n$. Moreover, the reduction in transition rates may undesirably affect the transient phase of the network decision probabilities. These limitations should be considered when selecting the appropriate repulsion model.

3.3 Marginalized Model

Through *marginalization*, the probability distribution of a single variable can be obtained by summing over a joint probability distribution. In this way, MZ individual agent decision probabilities can be derived from an M^Z -state network model. However, [14] demonstrated that the marginalization of $\mathbb{I}$ can be expressed analytically, without explicitly using the full network model, which results in a significantly more computationally efficient formulation.

Under the same conditions, it is possible to derive an analytical marginalization of $\mathbb{D}$, as stated below.

Theorem 3.3.1

Each row in the marginalized model describes the probability that the n :th agent is in state s_j as

$$\dot{\pi}_j^n = \sum_{i=1}^M Q^n[i, j] \pi_i^n + \pi_j^n \sum_{\mathcal{R}_\ell \in \mathcal{R}} \gamma - \sum_{\mathcal{R}_\ell \in \mathcal{R}} \gamma \sum_k \Gamma_{\mathcal{R}_\ell}^{\mathcal{A}}[n, k] \pi_j^k. \quad (3.6)$$

□

For the complete proof of this theorem, please refer to Appendix B.

Equation (3.6) is used to construct the marginalized model

$$\dot{\mathbf{\Pi}}_m(t) = (Q_m - R_m)^T \mathbf{\Pi}_m(t), \quad (3.7)$$

where Q_m and R_m are $MZ \times MZ$ matrices. These matrices are constructed from the terms in (B.8) and (B.12).

3.4 Comparison between direct and indirect repulsion models

To determine whether a network $\mathbb{D}$ defined with direct repulsion and a network $\mathbb{I}$ formulated with indirect repulsion can be used to similarly model a decision process, two criteria for *similarity* between general networks are defined:

1. the difference in the networks' expected decision-state holding times is minimized with respect to some criterion (e.g., (3.9)), and
2. the networks have identical stationary decision probabilities. Next, this property is formalized to derive a method that finds $\mathbb{D}$ similar to $\mathbb{I}$.

The first criterion for similarity requires that the time $\mathbb{D}$ spends in each decision state should be as close as possible to that of $\mathbb{I}$. As described in [34], the state holding time V_i of a state s_i in a CTMC with rate matrix Q is exponentially distributed with parameter $-Q[i, i]$. It follows that the expected holding time is $\mathbb{E}[V_i] = -1/Q[i, i]$.

For a state s_i^x in the networks $\mathbb{I}$ and $\mathbb{D}$, the cumulative distribution functions of the state holding times are

$$P[V_i^+ \leq t] = 1 - e^{(Q[i, i] + R^+[i, i])t}, \quad (3.8a)$$

$$P[V_i^- \leq t] = 1 - e^{(Q[i, i] - R^-[i, i])t}, \quad (3.8b)$$

respectively. Thus, $\mathbb{I}$ decreases the expected state holding time, whereas $\mathbb{D}$ increases it. To ensure that the decision-state holding times of both networks remain as close as possible, the increase in $\mathbb{E}[V_i^-]$ should be minimized, which is equivalent to minimizing the diagonal elements of R^- . For this purpose, the following objective function is introduced:

$$f(r, H) = r^T A^T H A r, \quad (3.9)$$

where r is a column vector such that r_i corresponds to the i th positive element of R^- , enumerated row-wise and left-to-right. The matrix A is a $(0, 1)$ -matrix such that $A r = -\text{diag}(R^-)$, and H is a tuning matrix that emphasizes the minimization of a subset of diagonal elements.

The second criterion for similarity between $\mathbb{D}$ and $\mathbb{I}$ requires that they reach identical stationary state probabilities. While the transition-rate matrix of an ergodic CTMC maps to a unique stationary distribution, multiple matrices can yield the same stationary state. Therefore, the off-diagonal elements of R^- must be found such that the stationary-state equations of (3.2) and (3.5), expressed as

$$(Q + R^+)^T \bar{\Pi} = 0, \quad (3.10a)$$

$$(Q - R^-)^T \bar{\Pi} = 0, \quad (3.10b)$$

hold for given Q , R^+ , and the shared stationary distribution $\bar{\Pi}$. Since Q cancels out when comparing the two, (3.10) reduces to

$$(R^-)^T \bar{\Pi} = -(R^+)^T \bar{\Pi}. \quad (3.11)$$

The linear combination $(R^-)^T \bar{\Pi}$ can be rearranged into a product between a known matrix $M_{\bar{\Pi}}$, constructed from the elements of $\bar{\Pi}$, and the vector r containing the unknown off-diagonal elements of R^- . The rate reductions in r must be positive but cannot exceed the corresponding transition rates in Q . Hence, the following conditions must hold:

$$M_{\bar{\Pi}} r = -(R^+)^T \bar{\Pi}, \quad (3.12a)$$

$$0 < r < q, \quad (3.12b)$$

where q is a vector containing the corresponding elements of Q , indexed in the same order as r .

All solutions to (3.12a) are given by

$$r = -M_{\bar{\Pi}}^{\dagger}(R^+)^T \bar{\Pi} + (I - M_{\bar{\Pi}}^{\dagger} M_{\bar{\Pi}})w, \quad (3.13)$$

where $\dagger$ denotes the *Moore–Penrose inverse* and w is an arbitrary vector (see, e.g., [35]). Solutions exist if and only if $M_{\bar{\Pi}}^{\dagger} M_{\bar{\Pi}}^{\dagger} = -(R^+)^T \bar{\Pi}$. The solution is unique if $M_{\bar{\Pi}}$ has full column rank, implying $I - M_{\bar{\Pi}}^{\dagger} M_{\bar{\Pi}} = 0$. Although this condition can be used to verify the existence of any solution r , the additional constraint $r < q$ must also be satisfied, which can be directly enforced in numerical optimization.

3.5 Feasibility under similarity and marginalizability constraints

Minimizing the objective function (3.9) under the constraints (3.12) defines the constrained optimization problem

$$\underset{r}{\text{minimize}} \quad f(r, H), \quad (3.14a)$$

$$\text{subject to} \quad M_{\bar{\Pi}} r = -(R^+)^T \bar{\Pi}, \quad (3.14b)$$

$$0 < r < q, \quad (3.14c)$$

which, for any number of agents and decisions, determines the off-diagonal elements of R^- such that $\mathbb{D}$ is *similar* to $\mathbb{I}$. In the following, the feasibility of (3.14) is analyzed with respect to the network structure and additional constraints required for obtaining marginalizable networks.

To show that similarity is not generally feasible for marginalizable networks, this section returns to Example 3.1.1 in the introduction and investigates (3.14) in three cases:

1. No additional marginalization constraints are imposed,
2. $\mathbb{D}$ is marginalizable, and
3. $\mathbb{D}$ and $\mathbb{I}$ are both marginalizable.

In Example 3.1.1, $Z = 2$ drivers, a and b , choose between $M = 2$ lanes at rates

$$Q_1 = \begin{bmatrix} -q_{12}^a & q_{12}^a \\ q_{21}^a & -q_{21}^a \end{bmatrix}, \quad Q_2 = \begin{bmatrix} -q_{12}^b & q_{12}^b \\ q_{21}^b & -q_{21}^b \end{bmatrix}, \quad (3.15)$$

respectively, so that $\mathbb{X}$ has a state space $\mathcal{S}_{\mathbb{X}} = \{s_1^1, s_2^1\} \times \{s_1^2, s_2^2\}$ according to (A.2). Its isolated transition rate matrix is

$$Q = \begin{bmatrix} -q_{12}^a - q_{12}^b & q_{12}^b & q_{12}^a & 0 \\ q_{21}^b & -q_{12}^a - q_{21}^b & 0 & q_{12}^a \\ q_{21}^a & 0 & -q_{21}^a - q_{12}^b & q_{12}^b \\ 0 & q_{21}^a & q_{21}^b & -q_{21}^a - q_{21}^b \end{bmatrix}. \quad (3.16)$$

Assuming mutual repulsion, the structure of the network transition rate matrices for indirect and direct repulsion are

$$R^+ = \begin{bmatrix} -(r_{12}^{a+} + r_{12}^{b+}) & r_{12}^{b+} & r_{12}^{a+} & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & r_{21}^{a+} & r_{21}^{b+} & -(r_{21}^{a+} + r_{21}^{b+}) \end{bmatrix}, \quad (3.17a)$$

$$R^- = \begin{bmatrix} 0 & 0 & 0 & 0 \\ r_{21}^{b-} & -(r_{12}^{a-} + r_{21}^{b-}) & 0 & r_{12}^{a-} \\ r_{21}^{a-} & 0 & -(r_{21}^{a-} + r_{12}^{b-}) & r_{12}^{b-} \\ 0 & 0 & 0 & 0 \end{bmatrix}. \quad (3.17b)$$

Here, $r_{ij}^{a/b+}$ and $r_{ij}^{a/b-}$ denote the rate increase and decrease for each agent's transition $s_i \rightarrow s_j$ as a result of indirect and direct repulsion, respectively. To formulate the constraints in (3.14b) and (3.14c), the following matrices and vectors are constructed:

$$M_{\bar{\Pi}} = \begin{bmatrix} \bar{\pi}_2 & 0 & \bar{\pi}_3 & 0 \\ -\bar{\pi}_2 & -\bar{\pi}_2 & 0 & 0 \\ 0 & 0 & -\bar{\pi}_3 & -\bar{\pi}_3 \\ 0 & \bar{\pi}_2 & 0 & \bar{\pi}_3 \end{bmatrix}, \quad (3.18a)$$

$$q = [q_{21}^b \quad q_{12}^a \quad q_{21}^a \quad q_{12}^b]^T, \quad (3.18b)$$

$$r = [r_{21}^{b-} \quad r_{12}^{a-} \quad r_{21}^{a-} \quad r_{12}^{b-}]^T, \quad (3.18c)$$

$$\bar{\Pi} = [\bar{\pi}_1 \quad \bar{\pi}_2 \quad \bar{\pi}_3 \quad \bar{\pi}_4]^T. \quad (3.18d)$$

Figure 3.2 illustrates the CTMC representations of a , b , and $\mathbb{X}$. For demonstration purposes, $\mathbb{X}$ includes both the indirect and direct repulsion rates in its isolated transition rates, thereby depicting $\mathbb{I}$ and $\mathbb{D}$ simultaneously. In this special case, indirect and direct repulsion never affect the same network transition, which is also reflected in the non-overlapping sparsity patterns of (3.17a) and (3.17b).

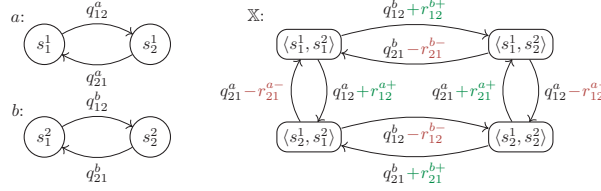


Figure 3.2: CTMCs a and b and their network CTMC $\mathbb{X}$.

3.5.1 No marginalization constraints

Evaluating the constraints (3.14b) and (3.14c) for Example network 3.1.1 yields

$$0 < r_{21}^{b-} = \frac{\bar{\pi}_1 r_{12}^{b+} - \bar{\pi}_4 r_{21}^{b+} + \bar{\pi}_3 x}{\bar{\pi}_2} < q_{21}^b, \quad (3.19a)$$

$$0 < r_{12}^{a-} = \frac{\bar{\pi}_4 r_{21}^{a+} + \bar{\pi}_4 r_{21}^{b+} - \bar{\pi}_3 x}{\bar{\pi}_2} < q_{12}^a, \quad (3.19b)$$

$$0 < r_{21}^{a-} = \frac{\bar{\pi}_1 r_{12}^{a+} + \bar{\pi}_4 r_{21}^{b+} - \bar{\pi}_3 x}{\bar{\pi}_3} < q_{21}^a, \quad (3.19c)$$

$$0 < r_{12}^{b-} = x < q_{12}^b. \quad (3.19d)$$

Hence, the rate x constitutes a free parameter on an interval $(x_{\min}, x_{\max})$ such that (3.19) is satisfied, indicating that similarity can be achieved without marginalization of the networks.

3.5.2 $\mathbb{D}$ is marginalizable

Following the reasoning in Section 3.4, choosing state-independent parameters when constructing R^- imposes the additional equality constraints $r_{21}^{a-} = r_{12}^{a-}$ and $r_{21}^{b-} = r_{12}^{b-}$, thereby halving the number of unknowns in (3.14). The lower bound of (3.19) then

reduces to

$$0 < r_{21}^{b-} = \frac{\bar{\pi}_1 r_{12}^{b+}}{\bar{\pi}_2} + \frac{\bar{\pi}_4 r_{21}^{a+}}{\bar{\pi}_2} + \frac{\bar{\pi}_1 r_{12}^{a+}}{\bar{\pi}_2 - \bar{\pi}_3} - \frac{\bar{\pi}_4 r_{21}^{a+}}{\bar{\pi}_2 - \bar{\pi}_3}, \quad (3.20a)$$

$$0 < r_{21}^{a-} = \frac{\bar{\pi}_4 r_{21}^{a+}}{\bar{\pi}_2 - \bar{\pi}_3} - \frac{\bar{\pi}_1 r_{12}^{a+}}{\bar{\pi}_2 - \bar{\pi}_3}, \quad (3.20b)$$

$$0 = \frac{\bar{\pi}_1 r_{12}^{a+}}{\bar{\pi}_3} - \frac{\bar{\pi}_1 r_{12}^{b+}}{\bar{\pi}_2} - \frac{\bar{\pi}_4 r_{21}^{a+}}{\bar{\pi}_2} + \frac{\bar{\pi}_4 r_{21}^{b+}}{\bar{\pi}_3}. \quad (3.20c)$$

The satisfaction of (3.20c) depends solely on R^+ and $\bar{\Pi}$ and cannot be adjusted through the choice of R^- . This counterexample therefore demonstrates that similarity between $\mathbb{I}$ and $\mathbb{D}$ is not necessarily determined by $\mathbb{D}$ when $\mathbb{D}$ is marginalizable.

3.5.3 $\mathbb{D}$ and $\mathbb{I}$ are marginalizable

Marginalizability of $\mathbb{I}$ requires that $r_{12}^{a+} = r_{21}^{a+}$ and $r_{12}^{b+} = r_{21}^{b+}$ hold in addition to (3.20). By expressing r_{12}^{b+} as a function of r_{12}^{a+} in (3.20c) and substituting it into (3.20a), together with (3.20b), two sets of conditions can be derived:

$$\frac{\bar{\pi}_2}{\bar{\pi}_3} > \frac{\bar{\pi}_4}{\bar{\pi}_1} > 1 \quad \text{or} \quad \frac{\bar{\pi}_2}{\bar{\pi}_3} < \frac{\bar{\pi}_4}{\bar{\pi}_1} < 1, \quad (3.21)$$

which ensure the existence and positivity of r .

Subsequently, the influence of the conditions in (3.21) on the positivity of r_{12}^{a+} and r_{12}^{b+} must be verified. From (3.20c), $r_{12}^{a+} > 0$ is assumed by construction. Therefore, $r_{12}^{b+} > 0$ if and only if

$$\frac{\bar{\pi}_3 \bar{\pi}_1 - \bar{\pi}_4 \bar{\pi}_2}{\bar{\pi}_2 \bar{\pi}_1 - \bar{\pi}_4 \bar{\pi}_3} > 0, \quad (3.22)$$

which yields two possible sets of inequalities:

$$\left\{ \begin{array}{l} \frac{\bar{\pi}_3}{\bar{\pi}_2} > \frac{\bar{\pi}_4}{\bar{\pi}_1} \\ \frac{\bar{\pi}_2}{\bar{\pi}_3} > \frac{\bar{\pi}_4}{\bar{\pi}_1} \end{array} \right\} \quad \text{or} \quad \left\{ \begin{array}{l} \frac{\bar{\pi}_3}{\bar{\pi}_2} < \frac{\bar{\pi}_4}{\bar{\pi}_1} \\ \frac{\bar{\pi}_2}{\bar{\pi}_3} < \frac{\bar{\pi}_4}{\bar{\pi}_1} \end{array} \right\}. \quad (3.23)$$

Finally, merging the conditions for existence and positivity in (3.21) with those in (3.23) reveals that no consistent solution exists, as each case leads to contradictory inequalities. Consequently, it is impossible to identify any combination of $\bar{\Pi}$ entries that satisfies all constraints simultaneously. This provides a counterexample for $Z = M = 2$, illustrating that similarity according to (3.14) is not generally feasible when both $\mathbb{D}$ and $\mathbb{I}$ are marginalizable.

3.6 Traffic Example

In Example 3.1.1, the similarity objective function (3.14a) cannot be minimized because the diagonal elements of (3.17b) become independent of the free rate r_{12}^{b-} due to (3.19). However, in the following large-scale example, it is shown that (3.14) can be solved when $\mathbb{I}$ is marginalizable, and that minimizing (3.14a) can lead to a shorter probability transient in $\mathbb{D}$.

Example 3.6.1

Consider $N = 8$ driver agents $c, d, \dots, j$ choosing between $M = 2$ lanes, with rates $Q_1[1, 2] = 1$, $Q_2[1, 2] = 2$, $\dots$, $Q_8[1, 2] = 8$, and $Q_{1\dots 8}[2, 1] = 5$, yielding a 256×256 Q matrix according to (A.6). Groups are disregarded, assuming fully connected repulsion among all agents, such that $\Gamma_{\mathcal{R}_\ell}^A$ in (3.1) and (3.3) always describes pairwise repulsion. For $\mathbb{I}$, we select state-independent parameters $0.14 \leq \gamma \leq 0.71$, ensuring that (3.4) holds for all agents and that $\mathbb{I}$ is marginalizable.

The objective is to find a $\mathbb{D}$ similar to $\mathbb{I}$ by solving (3.14) for the vector r containing the 2032 unknown rates in R^- . We assume that R^- results from state-dependent γ parameters, making $\mathbb{D}$ non-marginalizable. The matrix H in (3.14a) is set to the identity, leading to an unweighted minimization of all diagonal elements in R^- .

To demonstrate that these diagonal elements can indeed be minimized, two networks, $\mathbb{D}_1$ and $\mathbb{D}_2$, are compared. For $\mathbb{D}_1$, a lower bound of $r \geq 0.4$ is imposed, which is unnecessarily restrictive. For $\mathbb{D}_2$, the bound is relaxed to $r \geq 0.001$, which is closer to the minimal positivity constraint.

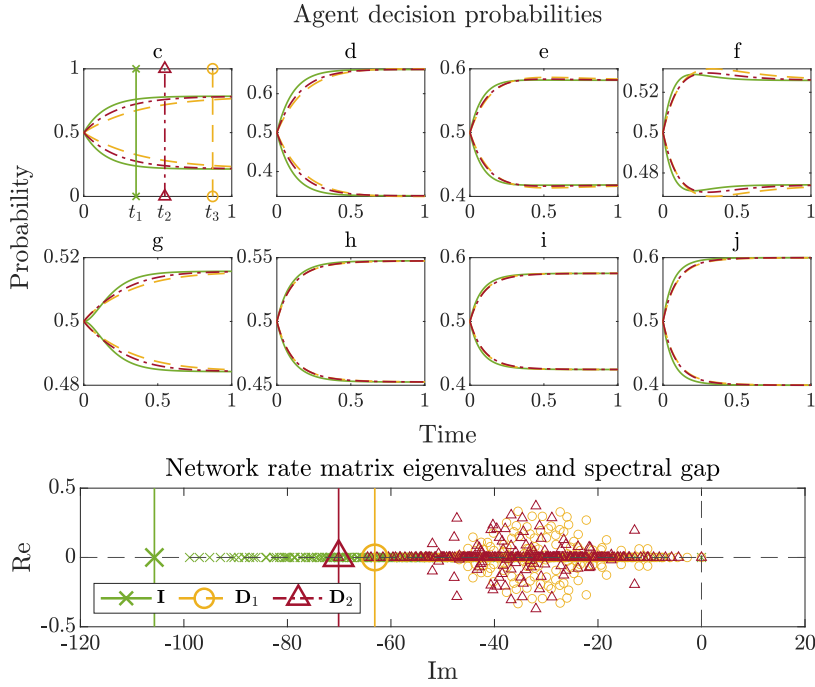


Figure 3.3: Agent decision probabilities and eigenvalues.

The networks $\mathbb{I}$, $\mathbb{D}_1$, and $\mathbb{D}_2$ are simulated to obtain their probability trajectories over the time interval $t \in [0, 1]$. By post-summing the network decision probabilities, the individual decision probabilities of each agent $c, \dots, j$ are extracted and compared in Fig. 3.3. In the first eight plots, it can be observed that $\mathbb{I}$, $\mathbb{D}_1$, and $\mathbb{D}_2$ yield identical stationary decision probabilities for all agents, confirming that the second similarity criterion is satisfied.

For agent c , the figure also reports the network *rise time*, defined as the time required to reach 95% of the stationary probability. Notably, $\mathbb{I}$ exhibits the fastest rise time, $t_1 \approx 0.35$, followed by $\mathbb{D}_2$ with $t_2 \approx 0.55$, and $\mathbb{D}_1$ with $t_3 \approx 0.87$. The eigenvalue plot further highlights that, in this case, a larger spectral gap—achieved by minimizing the difference in expected state holding time—corresponds to a shorter transient response.

Remark 3.6.1

The example above illustrates a case where a network $\mathbb{D}$ similar to a marginalizable $\mathbb{I}$ can be found. However, when marginalization constraints are also enforced on $\mathbb{D}$, no feasible solution exists.

3.7 Discussion

The *direct repulsion force* is introduced as an alternative to the *indirect repulsion force* for modeling group-wise disagreement in decision processes among Markovian agents in a network. To assess whether both methods can equivalently describe a decision

process, *similarity* is defined as a property between two network models, requiring equality between their stationary decision probabilities and a minimal difference (with respect to the cost (3.14a)) between their expected decision-state holding times. It is shown that the state-space dimension of a network defined with direct repulsion can also be significantly reduced through *marginalization*. Furthermore, for two networks $\mathbb{D}$ and $\mathbb{I}$ defined with direct and indirect repulsion, respectively, an example is provided in which a $\mathbb{D}$ similar to a marginalizable $\mathbb{I}$ can be obtained. However, a counterexample is also derived, demonstrating that the existence of a $\mathbb{D}$ similar to $\mathbb{I}$ cannot, in general, be guaranteed when $\mathbb{D}$ and/or $\mathbb{I}$ are marginalizable. Moreover, direct repulsion requires additional information and may increase the time required for the decision process to reach stationarity. Hence, indirect repulsion is recommended as a suitable model for disagreement in decision processes among Markovian agents.

Chapter 4

Influencing Agents Decisions

In this Chapter, models of interactions between agents and their surrounding environment are analyzed. In the first part (Section 4.1), a Markovian framework is introduced to describe the interactions between groups of individuals and the environment. An illustrative application in marketing is provided, showing how such a simple model can capture mechanisms that realistically occur in everyday life.

In the second part (Section 4.2), the same framework is further investigated with the goal of deriving a closed-form solution. This is achieved through dynamic programming, by first proceeding backward in time and subsequently forward to obtain the explicit solution.

The contents of this chapter are mainly derived from the works presented in [36], [37].

4.1 Influence of Decisions in Stochastic Multi-Agent Systems

This section presents a mathematical framework for modeling interactions between groups of agents and their environment. The following example illustrates the objectives and intended applications of this framework.

Example 4.1.1

A population of consumers is partitioned into two groups, G_1 and G_2 , based, for instance, on common attitudes or habits, with no interaction occurring within each group. The individuals of G_1 and G_2 are asked to make a decision and purchase a new t-shirt to replace their own. A correlation between the colors of the old and new t-shirts is assumed to hold with a known probability. Furthermore, the probability distribution of individuals in each group owning t-shirts of different colors, such as red, yellow, or blue, is known.

A t-shirt firm aims to increase the sales of blue t-shirts and investigates sales, sponsorship, and advertisement policies to persuade consumers in G_1 and G_2 . However, the groups are affected differently by the price, and the individuals in each group are influenced by the decisions made by those in the other group.

The objective is to determine the price of the t-shirts in order to control the probability that members of G_1 and G_2 purchase a new blue t-shirt.

In the remainder of this work, a modeling and control framework is developed that can be applied in scenarios where, similarly to Example 4.1.1, the behavior (alternatively, decisions or opinions) of a set of stochastic and interacting agents is influenced by the evolution of an externally controllable process, e.g., the surrounding environment.

4.1.1 Mathematical Model

Similarly to what has been done in the previous chapter 3, a discrete-time Markov Chain (DTMC) is employed to describe the decision-making process of a stochastic agent r . The DTMC is described by

$$\boldsymbol{\Pi}_{k+1}^r = (Q^r)^T \boldsymbol{\Pi}_k^r, \quad (4.1)$$

with initial condition $\boldsymbol{\Pi}_0^r$. Here, $\boldsymbol{\Pi}_k^r \in \mathbb{R}^{m \times 1}$ denotes the state probability vector at time instant k , such that $(\boldsymbol{\Pi}_k^r)^T \mathbf{1} = 1$ (where $\mathbf{1}$ is a column vector of ones), and m corresponds to the number of discrete states that the agent r can occupy. The transition probability matrix Q^r is row-stochastic ($Q^r \mathbf{1} = \mathbf{1}$) and has entries in the interval $[0, 1]$. A set of $Z \geq 1$ Markovian agents evolving according to (4.1) can be represented as a Markovian network [2]. The network evolution is given by

$$\boldsymbol{\Pi}_{k+1}^{\mathbf{x}} = (Q_k^{\mathbf{x}})^T \boldsymbol{\Pi}_k^{\mathbf{x}}, \quad (4.2)$$

where $\mathbf{\Pi}_k^{\mathbb{X}}$ denotes the probability vector of the network. To distinguish from the single-agent case, the states of the network are referred to as *configurations*. Specifically, a single network configuration corresponds to the combination of the states of all the agents. Assuming that each agent can occupy m states and denoting with $\mathcal{S}^r$ the state set of agent r , the set of configurations $\mathcal{S}^{\mathbb{X}}$ of the network has dimension $p = m^n$ and is given by

$$\mathcal{S}^{\mathbb{X}} = \mathcal{S}^1 \times \cdots \times \mathcal{S}^r \times \cdots \times \mathcal{S}^Z, \quad (4.3)$$

where $\times$ denotes the Cartesian product. If the agents are independent, then the transition matrix $Q \in \mathbb{R}^{p \times p}$ is simply given by

$$Q_k^{\mathbb{X}} = Q_k^1 \otimes \cdots \otimes Q_k^r \otimes \cdots \otimes Q_k^Z, \quad (4.4)$$

with $\otimes$ the Kronecker product. Conversely, if agents are not independent, their interactions influence one another, and this effect is captured by the structure of the transition matrix Q . In the literature (see [2], [14]), such interactions are modeled by modifying the transition matrix, leading to

$$\mathbf{\Pi}_{k+1}^{\mathbb{X}} = (\tilde{Q}_k^{\mathbb{X}})^T \mathbf{\Pi}_k = (Q_k^{\mathbb{X}} + \Omega(\bar{\alpha}))^T \mathbf{\Pi}_k^{\mathbb{X}}. \quad (4.5)$$

To simplify the notation, the shorthand $\mathbf{\Pi}_k^{\mathbb{X}} = \mathbf{\Pi}_k$, $Q_k^{\mathbb{X}} = Q_k$ is adopted in the following.

Here, $\Omega(\bar{\alpha})$ models the interactions through the parameter vector $\bar{\alpha}$. While the focus of [2] is on adjusting $\bar{\alpha}$ based on the network structure, the objective of the present work is to control $\mathbf{\Pi}_k$ with respect to a desired probability vector $\mathbf{\Pi}^{des}$.

To this end, the Markov model of one (4.1) or more (6.1.1) target agents is integrated with a model describing the evolution of a controllable agent (the *controller agent*), assumed to be independent of the targets as in [6]. In this setting, the controller agent is interpreted as the *environment* in which the target agents operate.

The model $\mathcal{W}$ describing the coupled evolution of target agents' opinions and the environment is

$$\mathbf{\Pi}_{k+1} = Q^T \mathbf{\Pi}_k + \boldsymbol{\mu}(k, \mathbf{x}_k), \quad (4.6a)$$

$$\mathbf{x}_{k+1} = f(\mathbf{x}_k, \mathbf{u}_k), \quad (4.6b)$$

where (4.6a) captures the evolution of the target agents, and (4.6b) models the dynamics of the environment. Specifically, $\mathbf{\Pi}_k$ represents the probability evolution of the target agents' states under their intrinsic behavior given by Q . The term $\boldsymbol{\mu}(k, \mathbf{x}_k)$ accounts for the influence exerted by the environment, while $\mathbf{x}_{k+1} = f(\mathbf{x}_k, \mathbf{u}_k)$ describes the evolution of the environment under the control input $\mathbf{u}_k$.

In this formulation, $\boldsymbol{\mu}(k, \mathbf{x}_k)$ can be interpreted as a control input directly acting on the DTMC. To maintain linearity, two simplifying assumptions are introduced:

1. The influence $\boldsymbol{\mu}(k, \mathbf{x}_k)$ exerted by the environment on the target agents is linear with respect to $\mathbf{x}_k$, i.e., $\boldsymbol{\mu}(k, \mathbf{x}_k) = H\mathbf{x}_k$, where H models the mapping from environmental states to agents' states.
2. The evolution of the environment follows a linear discrete-time system: $\mathbf{x}_{k+1} = A\mathbf{x}_k + B\mathbf{u}_k$.

Under these assumptions, the system $\mathcal{W}$ becomes

$$\underbrace{\begin{bmatrix} \mathbf{\Pi}_{k+1} \\ \mathbf{x}_{k+1} \end{bmatrix}}_{x_{a,k+1}} = \underbrace{\begin{bmatrix} Q_k^T & H \\ \mathbf{0}_{w \times p} & A \end{bmatrix}}_{\tilde{A}_k} \underbrace{\begin{bmatrix} \mathbf{\Pi}_k \\ \mathbf{x}_k \end{bmatrix}}_{x_{a,k}} + \underbrace{\begin{bmatrix} \mathbf{0}_{p \times l} \\ B \end{bmatrix}}_{\tilde{B}} u_k \quad (4.7)$$

$$y_{a,k} = \underbrace{\begin{bmatrix} I_{m \times m} & \mathbf{0}_{m \times w} \end{bmatrix}}_{\tilde{C}} x_{a,k} + \underbrace{\begin{bmatrix} \mathbf{0}_{m \times l} \end{bmatrix}}_{\tilde{D}} u_k$$

where $x_{a,k}$ is the augmented state, $y_{a,k}$ is the augmented output, and $\tilde{A}, \tilde{B}, \tilde{C}$ and $\tilde{D}$ are the augmented matrices

Before proceeding with the design of the controller, a preliminary study of the reachability properties of this system is undertaken.

4.1.1.1 Reachability of DTMC

In control theory [38], a discrete linear system is said to be *reachable* if the rank of the reachability matrix is full.

In (4.7), the dynamics of the DTMC are influenced by the state of the environment, denoted as $\mathbf{x}_k$, which can be interpreted as the control input of the DTMC itself. Assume that the evolution of the environment's state is fully reachable, i.e.,

$$\text{rank}(\mathcal{R}_E) = \text{rank}([B, AB, \dots, A^{m-1}B]) = m. \quad (4.8)$$

By construction, the environment has the same order as the DTMC. Therefore, for any initial state $\mathbf{x}_0$, it is feasible to reach any desired state of the environment. Consequently, there are no restrictions on the control input applied to the DTMC, and the primary concern becomes demonstrating the reachability of the DTMC portion of the model $\mathcal{W}$.

Consider the reachability matrix of the DTMC subsystem given by $\mathbf{\Pi}_{k+1} = Q^T \mathbf{\Pi}_k + H \mathbf{x}_k$ in (4.7), where $\mathbf{\Pi}_k, \mathbf{x}_k \in \mathbb{R}^{m \times 1}$ and $H \in \mathbb{R}^{m \times m}$ is non-singular:

$$\mathcal{R}_{MC} = [H, Q^T H, (Q^T)^2 H, \dots, (Q^T)^{m-1} H]. \quad (4.9)$$

If $\text{rank}(\mathcal{R}_{MC}) = m$, then the system is reachable. From (4.9), reachability is ensured (since H is non-singular), and the image of $\mathcal{R}_{MC}$ is

$$\mathcal{Im}(\mathcal{R}_{MC}) = \{\mathbf{e}_1, \mathbf{e}_2, \dots, \mathbf{e}_m\}, \quad (4.10)$$

where $\mathbf{e}_i$ is the i^{th} vector of the orthonormal basis.

Given a final state $\mathbf{\Pi}_k$ and initial condition $\mathbf{\Pi}_0$, it is of interest to investigate whether $\mathbf{\Pi}_k$ can be achieved through a feasible input sequence.

Remark 4.1.1

Henceforth, the forced evolution component $H \mathbf{x}_k$ of the DTMC dynamics will be abbreviated as $\boldsymbol{\mu}_k$, under the assumption that the environment is fully reachable and H is non-singular.

The reachability problem is solvable *iff*

$$\mathbf{\Pi}_k - (Q^T)^k \mathbf{\Pi}_0 = \mathcal{R}_{MC}^k \mathcal{U}_k, \quad (4.11)$$

where $\mathcal{U}_k = [\boldsymbol{\mu}_{k-1}, \dots, \boldsymbol{\mu}_0]^T$ and $\mathcal{R}_{MC}^k$ is the reachability matrix at step k . Eq. (4.11) admits a solution *iff*

$$\mathbf{\Pi}_k - (Q^T)^k \mathbf{\Pi}_0 \in \mathcal{Im}(\mathcal{R}_{MC}^k). \quad (4.12)$$

The vector $\mathbf{\Pi}_k - (Q^T)^k \mathbf{\Pi}_0$ can be expressed as a linear combination of the orthonormal basis vectors. Thus, from (4.10), condition (4.12) is satisfied.

It is important to note, however, that in this context, knowledge of system controllability alone is insufficient. The control input $\boldsymbol{\mu}_k$ in (4.6a) must preserve the structural properties of a Markov Chain. Therefore, the constraint

$$\boldsymbol{\mu}_k^T \mathbf{1} = 0, \quad \forall k, \quad (4.13)$$

must be enforced on the input sequence (see Section 4.1.2.1 for derivation). Additionally, the target state $\mathbf{\Pi}_k$ must satisfy the probability vector constraint $\mathbf{\Pi}_k^T \mathbf{1} = 1$.

The following theorem formalizes the existence of a feasible input sequence.

Theorem 4.1.1

Consider the linear system

$$\mathbf{\Pi}_{k+1} = Q^T \mathbf{\Pi}_k + \boldsymbol{\mu}_k, \quad (4.14)$$

where $\mathbf{\Pi}_k$ is a probability vector and Q is row-stochastic. If this system is reachable for initial condition $\mathbf{\Pi}_0$ and final state $\mathbf{\Pi}_k$, then the input sequence $\mathcal{U}_k = [\boldsymbol{\mu}_{k-1}, \dots, \boldsymbol{\mu}_0]^T$ satisfies

$$\mathcal{U}_k^T \mathbf{1} = \mathbf{0}. \quad (4.15)$$

□

Proof. From (4.11), $\mathbf{\Pi}_k$ and $\mathbf{\Pi}_0$ are probability vectors:

$$\mathbf{\Pi}_k^T \mathbf{1} = 1, \quad \mathbf{\Pi}_0^T \mathbf{1} = 1. \quad (4.16)$$

Therefore,

$$\begin{aligned} (\mathbf{\Pi}_k)^T \mathbf{1} &= ((Q^T)^k \mathbf{\Pi}_0 + \mathcal{R}_k \mathcal{U}_k)^T \mathbf{1} \\ &= \underbrace{((Q^T)^k \mathbf{\Pi}_0)^T \mathbf{1}}_a + \underbrace{(\mathcal{R}_k \mathcal{U}_k)^T \mathbf{1}}_b. \end{aligned} \quad (4.17)$$

For term a :

$$\mathbf{\Pi}_0^T ((Q^T)^k)^T \mathbf{1} = \mathbf{\Pi}_0^T \mathbf{1} = 1, \quad (4.18)$$

since Q is row-stochastic.

For term b , $(\mathcal{R}_k \mathcal{U}_k)^T \mathbf{1} = \mathcal{U}_k^T \mathcal{R}_k^T \mathbf{1}$, and

$$\mathcal{R}_k^T \mathbf{1} = \begin{bmatrix} I \\ Q \\ Q^2 \\ \vdots \\ Q^{k-1} \end{bmatrix} \mathbf{1} = \begin{bmatrix} \mathbf{1} \\ \mathbf{1} \\ \mathbf{1} \\ \vdots \\ \mathbf{1} \end{bmatrix}. \quad (4.19)$$

Combining (4.18) and (4.19), it follows that

$$1 = 1 + \mathcal{U}_k^T \mathbf{1} \implies \mathcal{U}_k^T \mathbf{1} = 0. \quad (4.20)$$

□

Hence, the DTMC is reachable with a feasible input sequence, assuming full reachability of the environment. It is important to remark that this assumption may not hold in all scenarios, potentially affecting overall system reachability.

In the next section, the derivation of the control policy u for (4.7) is presented, with the specific objective of controlling the evolution of the DTMC component.

4.1.2 Controller

In this section, a MPC problem for reference tracking is formulated for model (4.7), assuming the state is fully observable.

Following standard MPC theory [39], a quadratic cost function is considered:

$$J = \mathbf{x}_{a,N}^T S \mathbf{x}_{a,N} + \sum_{i=0}^{N-1} \left(\tilde{\mathbf{x}}_{a,k+i}^T Q_{err} \tilde{\mathbf{x}}_{a,k+i} + \mathbf{u}_{k+i}^T R \mathbf{u}_{k+i} \right), \quad (4.21)$$

where

- $\tilde{\mathbf{x}}_{a,k+i} = \mathbf{x}_{a,k+i} - \mathbf{x}_{a,k+i}^{ref}$ is the tracking error, with Q_{err} the corresponding weighting matrix;
- $\mathbf{u}_{k+i}$ is the control input, with R the corresponding weighting matrix;

- $\mathbf{x}_{a,N}$ is the terminal state, with S the corresponding weighting matrix;
- N denotes the finite control horizon.

It is important to note that the state vector $\mathbf{x}_a = [\mathbf{\Pi}, \mathbf{x}]^T$ contains the probability vector $\mathbf{\Pi}$, which must preserve a proper probabilistic structure. Therefore, when formulating the MPC minimization problem, constraints on the environment state must be incorporated:

$$F_{eq} \mathbf{x}_k = \mathbf{f}_{eq}, \quad (4.22a)$$

$$F_{ineq} \mathbf{x}_k \leq \mathbf{f}_{ineq}. \quad (4.22b)$$

Hence, the optimization problem considered is

Optimization Problem 4.1.1

$$\begin{aligned} \mathcal{U}_{k:k+N}^* = \operatorname{argmin}_{\mathbf{u}_k \in \mathcal{U}, \forall k} & \mathbf{x}_{a,N}^T S \mathbf{x}_{a,N} + \sum_{i=0}^{N-1} \left(\tilde{\mathbf{x}}_{a,k+i}^T Q_{err} \tilde{\mathbf{x}}_{a,k+i} + \mathbf{u}_{k+i}^T R \mathbf{u}_{k+i} \right) \\ \text{s.t. } & \begin{cases} \mathbf{x}_{a,k+1} = f(\mathbf{x}_{a,k}, \mathbf{u}_k), \\ F_{eq} \mathbf{x}_k = \mathbf{f}_{eq}, \\ F_{ineq} \mathbf{x}_k \leq \mathbf{f}_{ineq}. \end{cases} \quad \forall k = 1, \dots, N \end{aligned} \quad (4.23)$$

In the following subsections, the derivation of the constraint matrices $F_{eq}, \mathbf{f}_{eq}, F_{ineq}, \mathbf{f}_{ineq}$ is presented, along with the specification of the cost function matrices (7.1). The structure of the condensed MPC matrices is then made explicit.

4.1.2.1 Equality constraints

The equality constraint matrices corresponding to (4.22a) are derived as follows.

Given that the first submodel is a DTMC, the following properties must be satisfied:

1. $\sum_i \pi_{i,k} = 1 \forall k$, i.e., in matrix form, $\mathbf{\Pi}_k^T \mathbf{1} = 1$, ensuring that the probability vector remains normalized;
2. $Q\mathbf{1} = \mathbf{1}$ due to the row-stochastic nature of the transition matrix.

By applying these properties to the DTMC portion of (4.7), one obtains:

$$\begin{aligned} (\mathbf{\Pi}_{k+1})^T &= (Q^T \mathbf{\Pi}_k)^T + (H \mathbf{x}_k)^T \\ (\mathbf{\Pi}_{k+1})^T \mathbf{1} &= (Q^T \mathbf{\Pi}_k)^T \mathbf{1} + (H \mathbf{x}_k)^T \mathbf{1} \\ 1 &= \mathbf{\Pi}_k^T Q \mathbf{1} + (H \mathbf{x}_k)^T \mathbf{1} \\ 1 &= \mathbf{\Pi}_k^T \mathbf{1} + (H \mathbf{x}_k)^T \mathbf{1} \\ 1 &= 1 + (H \mathbf{x}_k)^T \mathbf{1}. \end{aligned} \quad (4.24)$$

Hence, the equality constraint for the environment's state is

$$(H \mathbf{x}_k)^T \mathbf{1} = 0, \quad \forall k, \quad (4.25)$$

which can equivalently be written as

$$\mathbf{1}^T H \mathbf{x}_k = 0, \quad (4.26)$$

so that

$$F_{eq} = \mathbf{1}^T H, \quad \mathbf{f}_{eq} = 0. \quad (4.27)$$

4.1.2.2 Inequality constraints

For the inequality constraint, the mathematical meaning of $H\mathbf{x}_k$ is considered: it represents the variation applied to the state probabilities. Therefore, it is reasonable to assume

$$[H\mathbf{x}_k]_i \in [-1, 1], \quad \forall i, \quad (4.28)$$

corresponding to the maximum allowable change in probability for any DTMC state. These boundaries can be integrated into the control problem as inequality constraints, as in (4.22b). Specifically, the following two sets of inequalities are imposed:

$$\begin{cases} H\mathbf{x}_k \geq -1 \Rightarrow -H\mathbf{x}_k \leq 1, \\ H\mathbf{x}_k \leq 1. \end{cases} \quad (4.29)$$

In matrix form, this yields

$$F_{ineq} = \begin{bmatrix} -H \\ H \end{bmatrix}, \quad \mathbf{f}_{ineq} = \mathbf{1}. \quad (4.30)$$

4.1.2.3 Cost function matrices

The weighting matrices in the cost function (7.1) are defined as

$$Q_{err} = \beta, \quad R = I, \quad S = Q_{err}, \quad (4.31)$$

where β is a tunable diagonal matrix. In this work, it is set as

$$\beta = \text{diag}([1, 1, 1, 0, 0, 0]), \quad (4.32)$$

indicating that the focus is solely on minimizing the reference tracking error of the DTMC state.

4.1.2.4 Condensed matrices

The MPC optimization can be expressed in a compact form using condensed matrices. Considering a finite control horizon $N = 10$, the condensed state and input vectors are defined as

$$\begin{aligned} \mathcal{X}_{c,k} &= [\mathbf{x}_{a,k+1}^T \quad \dots \quad \mathbf{x}_{a,k+N}^T]^T, \\ \mathcal{U}_{c,k} &= [\mathbf{u}_k^T \quad \dots \quad \mathbf{u}_{k+N-1}^T]^T, \end{aligned} \quad (4.33)$$

with the condensed system matrices

$$\begin{aligned} \mathcal{A}_c &= [\tilde{A} \quad \tilde{A}^2 \quad \dots \quad \tilde{A}^N]^T, \\ \mathcal{B}_c &= \begin{bmatrix} \tilde{B} & 0 & \dots & 0 \\ \tilde{A}\tilde{B} & \tilde{B} & \ddots & 0 \\ \vdots & \ddots & \ddots & 0 \\ \tilde{A}^{N-1}\tilde{B} & \dots & \tilde{A}\tilde{B} & \tilde{B} \end{bmatrix}. \end{aligned} \quad (4.34)$$

Thus, the system can be written in condensed form as

$$\mathcal{X}_{c,k} = \mathcal{A}_c \mathbf{x}_k + \mathcal{B}_c \mathcal{U}_{c,k}. \quad (4.35)$$

Similarly, the constraints (4.22a) and (4.22b) are expressed in condensed form as

$$\mathcal{F}_{c,eq} = I_{N \times N} \otimes F_{eq}, \quad \mathbf{f}_{c,eq} = I_{N \times 1} \otimes \mathbf{f}_{eq}, \quad (4.36)$$

$$\mathcal{F}_{c,ineq} = I_{N \times N} \otimes F_{ineq}, \quad \mathbf{f}_{c,ineq} = I_{N \times 1} \otimes \mathbf{f}_{ineq}, \quad (4.37)$$

and the cost function matrices are

$$Q_{err} = \text{diag}([Q_{err}, \dots, Q_{err}, S]), \quad \mathcal{R} = \text{diag}([R, \dots, R]). \quad (4.38)$$

For further details on the derivation and use of condensed matrices in MPC, see [40].

4.1.3 Simulation and Results

In this section, the model (4.7) is simulated to assess the performance of the proposed controller.

For the simulations, the scenario introduced in Example 4.1.1 is considered. In particular, there are 2 target agents corresponding to groups G_1 and G_2 , which are embedded within an environment representing the t-shirt prices. The company can increase or decrease these prices as part of its control strategy.

The target agents are called to choose among red, yellow, or blue t-shirts. Hence, the DTMC associated with each agent has $m = 3$ states representing the different colors. The objective of the company is to steer the agents' decisions toward purchasing red t-shirts.

The entries of the DTMC state probability vector Π_k are defined as the probabilities of purchasing a red, yellow, or blue t-shirt, respectively. The entries of the environment's state x_k correspond to the percentage price variation (relative to the initial cost) for red, yellow, and blue t-shirts. By construction, the percentage price variations are constrained to the interval $[-1, 1]$.

As an illustrative example, the matrices for the linear system in (4.7) are defined as follows. To enable a fair comparison between G_1 and G_2 , the two groups are assumed to have the same transition matrix: $Q^{(1)} = Q^{(2)} = Q$. The matrices Q and H are set as

$$Q = \begin{bmatrix} 0.3 & 0.5 & 0.2 \\ 0.1 & 0.2 & 0.7 \\ 0.4 & 0.5 & 0.1 \end{bmatrix}, \quad H = - \begin{bmatrix} 0.7 & 0.2 & 0.1 \\ 0.4 & 0.6 & 0 \\ 0.1 & 0.1 & 0.8 \end{bmatrix}, \quad (4.39)$$

where H characterizes the influence of price variations on the probability of selecting a particular color. Specifically, it assigns higher weight to the variation of the price of a specific color relative to its corresponding purchase probability.

In real-world scenarios, the entries of Q and H are typically unknown and should be estimated from data. The matrix Q can be estimated using a frequentist approach [41], while H can be inferred through surveys or experimental data capturing the effect of price changes on purchasing behavior.

The matrices A and B governing the evolution of the environment are set as

$$A = I, \quad B = I, \quad (4.40)$$

which implies that t-shirt prices remain constant over time unless actively modified by the company. Furthermore, altering the price of a specific color does not affect the other colors.

The reference and initial probability vectors are set as

$$\begin{aligned} \Pi_{ref} &= [0.3, 0.2, 0.5], \\ \Pi_0 &= [0.69, 0.12, 0.19], \end{aligned} \quad (4.41)$$

representing a shift in overall preference from red toward blue t-shirts. No reference is assigned for x , since the goal is to control only the DTMC component. If needed, one could define x_{ref} arbitrarily and assign zero weights to the corresponding tracking error in the cost function (7.1).

To illustrate different behaviors among agents, the members of G_1 are assumed to be strongly influenced by the environment, whereas the members of G_2 are considered unaffected by price variations.

4.1.3.1 Controller performances

All simulations were conducted in *Matlab R2023a* for a total of 30 steps, each corresponding to a day. Nevertheless, the forthcoming figures display only the initial 10 steps, since no substantial changes were observed beyond this point in the simulations.

The evolution of the probability vectors for G_1 (red line) and G_2 (blue line) is shown in Fig. 4.1.

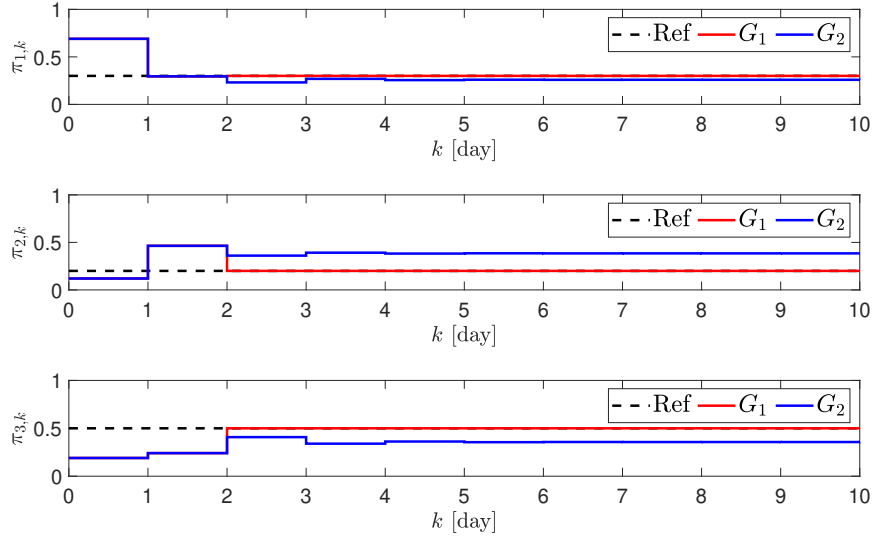


Figure 4.1: Evolution of the decision-making process of the target agents G_1 and G_2 .

Here, G_1 is affected by the environment, whereas the decision-making process of G_2 evolves naturally according to the DTMC described by the transition matrix Q in (4.39) and initial condition Π_0 in (4.41). In contrast, G_1 , although characterized by the same Q and Π_0 , is influenced by the environment through the control input.

Before confirming the effectiveness of the controller, it is verified that the controlled state $\Pi(k)$ maintains the properties of a probability vector, i.e., $\sum_i \pi_i(k) = 1 \forall k$. This ensures that the DTMC component of the controlled state remains valid throughout the simulation.

Next, the dynamics of the environment are analyzed to understand how the desired influence on G_1 is achieved. Assuming an initial t-shirt price of 30 euros for all colors, Fig. 4.2 shows the percentage variation in the price of each t-shirt (R = red, Y = yellow, B = blue). A zero value indicates that the price remains unchanged. Using these variations, the actual t-shirt prices are computed and illustrated in Fig. 4.3.

Finally, considering the initial objective of promoting blue t-shirt purchases and the influence matrix H (4.39), the simulation produces the intuitive result that the company should reduce the price of blue t-shirts to achieve the desired shift in purchasing behavior.

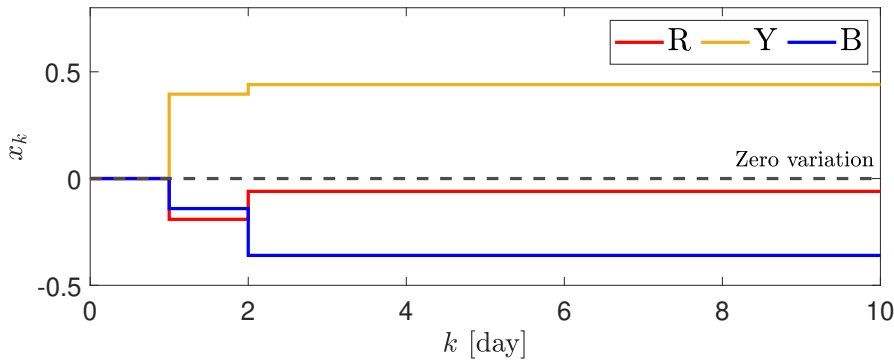


Figure 4.2: Percentage variation of the price of different t-shirt colors.

To summarize, it has proposed a control framework for an opinion dynamics model, specifically for a DTMC. Unlike common approaches in the literature, it assumes that

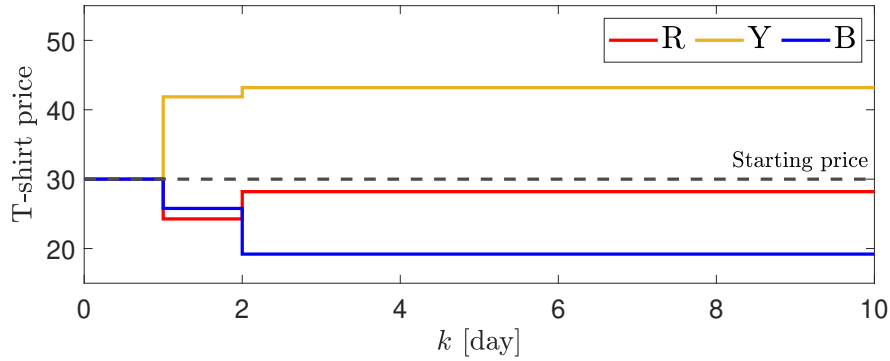


Figure 4.3: Price trends of the t-shirts.

the control input is external and acts on a controller agent, which could represent the environment. This controller, in turn, influences the behavior of the target agents within the network.

It first demonstrated that the system is controllable under suitable assumptions and then designed an MPC to steer the target agents toward the desired state. To illustrate the framework, it applied it to a small-scale, everyday-life example, showing the outcomes of the controller and highlighting potential applications. Although this work represents an initial step in controlling stochastic agents, it provides valuable insights into the capabilities and potential of this control scheme.

4.2 Feedback Control in Multi-Agent Markovian networks

Until now, the impact of the environment or the controller agent on the evolution of the DTMC has been represented as a (control) input denoted by the product $H\mathbf{x}_k$, where $\mathbf{x}_k$ is the state of the controller agent, manipulated by $\mathbf{u}_k$. In the previous Section 4.1, $\mathbf{u}_k$ was derived as the optimal control sequence from a Linear Quadratic optimization problem, corresponding to optimization problem (4.23). Nevertheless, it is now of interest to rewrite the control input as an affine transformation of the reference error and to derive an explicit feedback control (EFC) solution.

4.2.1 Explicit solution for the constrained reference tracking problem

The objective is to demonstrate that it is possible to revert from the formulation presented in (4.7) to a feedback control scheme of the form

$$\begin{cases} \mathbf{x}_{a,k+1} = \tilde{A}\mathbf{x}_{a,k} + \tilde{B}\mathbf{u}_k^*, \\ \mathbf{u}_k^* = F_k\tilde{\mathbf{x}}_{a,k} + \mathbf{v}_k, \end{cases} \quad (4.42)$$

where $\mathbf{u}_k^*$ denotes the optimal control sequence obtained as a solution of the *Linear Quadratic Regulator* (LQR) problem described in Section 4.1.2.

In general, one cannot definitively claim that the optimal control sequence $\mathbf{u}_k^*$, resulting from a constrained LQR optimization problem with a finite horizon, is linearly dependent on the state (or, in this case, on the reference tracking error). Nevertheless, in [42] and [43], algorithms were successfully derived to address a constrained LQR problem by establishing an affine relationship between the control and the state. In other words, it was shown that the optimal control sequence $\mathbf{u}_k^*$ for a LQR problem, potentially subject to linear constraints, can be expressed as the affine transformation

$$\mathbf{u}_k^*(x_k) = F_k\mathbf{x}_k + \mathbf{v}_k, \quad (4.43)$$

where F_k is the feedback matrix.

The algorithms proposed in [42] and [43] are, however, not directly applicable to the present case. This is due to the fact that the cost function considered in (7.1) is quadratic with respect to the tracking error. Consequently, the problem formulation and its solution must be adapted in order to fit the specific LQR structure adopted here, such that the control input is rewritten as

$$\mathbf{u}_k^*(\tilde{\mathbf{x}}_k) = F_k \tilde{\mathbf{x}}_k + \mathbf{v}_k. \quad (4.44)$$

Here, the explicit solution to the LQR reference tracking problem is derived, and its general formulation is presented (i.e., without specifying the system matrices). As recalled in the introduction, this formulation is more suitable for the setting under consideration.

The LQR problem for reference tracking is given by

$$\min_{\mathbf{u}_0, \dots, \mathbf{u}_{T-1}} \quad cost_T(\tilde{\mathbf{x}}_T) + \sum_{k=0}^{T-1} cost_k(\tilde{\mathbf{x}}_k, \mathbf{u}_k), \quad (4.45)$$

where $cost_k$ denotes the stage cost at time k , and $cost_T$ denotes the terminal cost. The optimization problem (4.45) is subject to the following constraints:

$$dynamics_k(\mathbf{x}_{k+1}, \mathbf{x}_k, \mathbf{u}_k) = 0, \quad \forall k \in \{0, \dots, T-1\}, \quad (4.46a)$$

$$\mathbf{x}_0 = \mathbf{x}(0), \quad (\text{initial condition}) \quad (4.46b)$$

$$constraint_k(\mathbf{x}_k, \mathbf{u}_k) = 0, \quad \forall k \in \{0, \dots, T-1\}, \quad (4.46c)$$

$$constraint_T(\mathbf{x}_T) = 0, \quad (\text{terminal constraint}). \quad (4.46d)$$

The linear dynamics of the system are generally written as

$$dynamics_k(\mathbf{x}_{k+1}, \mathbf{x}_k, \mathbf{u}_k) = \mathbf{x}_{k+1} - (A_k \mathbf{x}_k + B_k \mathbf{u}_k + \mathbf{c}_k). \quad (4.47)$$

The terminal cost function $cost_T(\tilde{\mathbf{x}}_T)$ is expressed as

$$cost_T(\tilde{\mathbf{x}}_T) = \frac{1}{2} \begin{bmatrix} 0 \\ \tilde{\mathbf{x}}_T \end{bmatrix}^T \begin{bmatrix} 0 & q_{\tilde{\mathbf{x}}_T}^T \\ q_{\tilde{\mathbf{x}}_T} & Q_{\tilde{\mathbf{x}}_T \tilde{\mathbf{x}}_T} \end{bmatrix} \begin{bmatrix} 0 \\ \tilde{\mathbf{x}}_T \end{bmatrix}. \quad (4.48)$$

In the considered study case, it is assumed that $\mathbf{u}_k^{ref} = \mathbf{0}_{l \times 1}$ for all k , so that $\tilde{\mathbf{u}}_k = \mathbf{u}_k$. Accordingly, the stage cost function simplifies to

$$cost_k(\tilde{\mathbf{x}}_k, \mathbf{u}_k) = \frac{1}{2} \begin{bmatrix} 0 \\ \tilde{\mathbf{x}}_k \\ \mathbf{u}_k \end{bmatrix}^T \begin{bmatrix} 0 & q_{\tilde{\mathbf{x}}_k}^T & q_{\mathbf{u}_k}^T \\ q_{\tilde{\mathbf{x}}_k} & Q_{\tilde{\mathbf{x}}_k \tilde{\mathbf{x}}_k} & Q_{\mathbf{u}_k \tilde{\mathbf{x}}_k}^T \\ q_{\mathbf{u}_k} & Q_{\mathbf{u}_k \tilde{\mathbf{x}}_k} & Q_{\mathbf{u}_k \mathbf{u}_k} \end{bmatrix} \begin{bmatrix} 0 \\ \tilde{\mathbf{x}}_k \\ \mathbf{u}_k \end{bmatrix}. \quad (4.49)$$

The constraint functions must remain state-dependent (and not error-dependent), and therefore their structure is identical to the original formulation:

$$constraint_k(\mathbf{x}_k, \mathbf{u}_k) = G_{\mathbf{x}_k} \mathbf{x}_k + G_{\mathbf{u}_k} \mathbf{u}_k + \mathbf{g}_k, \quad (4.50a)$$

$$constraint_T(\mathbf{x}_T) = G_{\mathbf{x}_T} \mathbf{x}_T + \mathbf{g}_T. \quad (4.50b)$$

To compute the constrained control policies, the dynamic programming approach is adopted. Two auxiliary functions are defined:

- $cost_to_go_k$, which represents the minimum remaining cost from time k to the terminal horizon;
- $constraint_to_go_k$, which identifies the set of admissible states that ensure feasibility of $\mathbf{u}_k$.

In the modified setting, $cost_to_go_k$ depends on the tracking error, while $constraint_to_go_k$ remains state-dependent:

$$cost_to_go_k(\tilde{\mathbf{x}}) = \frac{1}{2} \begin{bmatrix} 1 \\ \tilde{\mathbf{x}} \end{bmatrix}^T \begin{bmatrix} 0 & \nu_{\tilde{\mathbf{x}}_k}^T \\ \nu_{\tilde{\mathbf{x}}_k} & V_{\tilde{\mathbf{x}}_k \tilde{\mathbf{x}}_k} \end{bmatrix} \begin{bmatrix} 1 \\ \tilde{\mathbf{x}} \end{bmatrix}, \quad (4.51a)$$

$$\text{constraint_to_go}_k(\mathbf{x}) = H_{\mathbf{x}_k} \mathbf{x} + \mathbf{h}_k. \quad (4.51b)$$

The initialization is performed by setting

$$\nu_{\tilde{\mathbf{x}}_T} = q_{\tilde{\mathbf{x}}_T}, \quad V_{\tilde{\mathbf{x}}\tilde{\mathbf{x}}_T} = Q_{\tilde{\mathbf{x}}\tilde{\mathbf{x}}_T},$$

for the cost-to-go function, and

$$H_{\mathbf{x}_T} = G_{\mathbf{x}_T}, \quad \mathbf{h}_T = \mathbf{g}_T,$$

for the constraint-to-go function.

The dynamic programming algorithm proceeds backwards from $T - 1$ to 0, by solving at each stage:

$$\mathbf{u}_k^*(\tilde{\mathbf{x}}_k) = \underset{\mathbf{u}_k}{\operatorname{argmin}} J(\tilde{\mathbf{x}}_k, \mathbf{u}_k), \quad (4.52)$$

where

$$J(\tilde{\mathbf{x}}_k, \mathbf{u}_k) = \text{cost}_k(\tilde{\mathbf{x}}_k, \mathbf{u}_k) + \text{cost_to_go}_{k+1}(\tilde{\mathbf{x}}_{k+1}), \quad (4.53)$$

subject to

$$0 = \text{dynamics}_k(\mathbf{x}_{k+1}, \mathbf{x}_k, \mathbf{u}_k), \quad (4.54a)$$

$$\mathbf{u}_k \in \underset{\mathbf{u}}{\operatorname{argmin}} \left\| \begin{bmatrix} \text{constraint}_k(\mathbf{x}_k, \mathbf{u}_k) \\ \text{constraint_to_go}_{k+1}(\mathbf{x}_{k+1}) \end{bmatrix} \right\|_2. \quad (4.54b)$$

To evaluate the summation in (4.53), the function (4.51a) must be expressed in terms of $\tilde{\mathbf{x}}_{k+1}$. By exploiting the definition of the tracking error and the system dynamics, one obtains

$$\tilde{\mathbf{x}}_{k+1} = \mathbf{x}_{k+1} - \mathbf{x}_{k+1}^{\text{ref}} = A_k \mathbf{x}_k + B_k \mathbf{u}_k + \mathbf{c}_k - \mathbf{x}_{k+1}^{\text{ref}}. \quad (4.55)$$

By adding and subtracting the term $A_k \mathbf{x}_k^{\text{ref}}$ to (4.55), this can be rewritten as

$$\tilde{\mathbf{x}}_{k+1} = A_k \mathbf{x}_k + B_k \mathbf{u}_k + \underbrace{(\mathbf{c}_k + A_k \mathbf{x}_k^{\text{ref}} - \mathbf{x}_{k+1}^{\text{ref}})}_{\gamma_k}. \quad (4.56)$$

Substituting (4.56) into cost_to_go_{k+1} and summing with the stage cost cost_k in (4.53) yields

$$J(\tilde{\mathbf{x}}_k, \mathbf{u}_k) = \frac{1}{2} \begin{bmatrix} 1 \\ \tilde{\mathbf{x}}_k \\ \mathbf{u}_k \end{bmatrix}^T \begin{bmatrix} m_k & m_{\tilde{\mathbf{x}}_k}^T & m_{\mathbf{u}_k}^T \\ m_{\tilde{\mathbf{x}}_k} & M_{\tilde{\mathbf{x}}\tilde{\mathbf{x}}_k} & M_{\mathbf{u}\tilde{\mathbf{x}}_k}^T \\ m_{\mathbf{u}_k} & M_{\mathbf{u}\tilde{\mathbf{x}}_k} & M_{\mathbf{u}\mathbf{u}_k} \end{bmatrix} \begin{bmatrix} 1 \\ \tilde{\mathbf{x}}_k \\ \mathbf{u}_k \end{bmatrix}, \quad (4.57)$$

with coefficients defined as

$$m_k = \gamma_k^T \nu_{\tilde{\mathbf{x}}_{k+1}} + \nu_{\tilde{\mathbf{x}}_{k+1}}^T \gamma_k + \gamma_k^T V_{\tilde{\mathbf{x}}\tilde{\mathbf{x}}_{k+1}} \gamma_k, \quad (4.58a)$$

$$m_{\tilde{\mathbf{x}}_k} = A_k^T \nu_{\tilde{\mathbf{x}}_{k+1}} + A_k^T V_{\tilde{\mathbf{x}}\tilde{\mathbf{x}}_{k+1}} \gamma_k + q_{\tilde{\mathbf{x}}_k}, \quad (4.58b)$$

$$m_{\mathbf{u}_k} = B_k^T \nu_{\tilde{\mathbf{x}}_{k+1}} + B_k^T V_{\tilde{\mathbf{x}}\tilde{\mathbf{x}}_{k+1}} \gamma_k + q_{\mathbf{u}_k}, \quad (4.58c)$$

$$M_{\tilde{\mathbf{x}}\tilde{\mathbf{x}}_k} = A_k^T V_{\tilde{\mathbf{x}}\tilde{\mathbf{x}}_{k+1}} A_k + Q_{\tilde{\mathbf{x}}\tilde{\mathbf{x}}_k}, \quad (4.58d)$$

$$M_{\mathbf{u}\tilde{\mathbf{x}}_k} = B_k^T V_{\tilde{\mathbf{x}}\tilde{\mathbf{x}}_{k+1}} A_k + Q_{\mathbf{u}\tilde{\mathbf{x}}_k}, \quad (4.58e)$$

$$M_{\mathbf{u}\mathbf{u}_k} = B_k^T V_{\tilde{\mathbf{x}}\tilde{\mathbf{x}}_{k+1}} B_k + Q_{\mathbf{u}\mathbf{u}_k}. \quad (4.58f)$$

Finally, by exploiting the constraint formulation (4.50) together with the dynamics (4.47), the feasibility condition (4.54b) can be rewritten as

$$\mathbf{u}_k \in \underset{\mathbf{u}_k}{\operatorname{argmin}} \|N_{\mathbf{x}_k} \mathbf{x}_k + N_{\mathbf{u}_k} \mathbf{u}_k + \mathbf{n}_k\|_2, \quad (4.59)$$

where

$$N_{\mathbf{x}_k} = \begin{bmatrix} G_{\mathbf{x}_k} \\ H_{\mathbf{x}_{k+1}} A_k \end{bmatrix}, \quad N_{\mathbf{u}_k} = \begin{bmatrix} G_{\mathbf{u}_k} \\ H_{\mathbf{x}_{k+1}} B_k \end{bmatrix}, \quad \mathbf{n}_k = \begin{bmatrix} \mathbf{g}_k \\ H_{\mathbf{x}_{k+1}} \mathbf{c}_k + \mathbf{h}_k \end{bmatrix}.$$

The optimal control input can be expressed as the sum of a constrained and a free component,

$$\mathbf{u}_k^* = P_{\mathbf{y}_k} \mathbf{y}_k^* + Z_{\mathbf{w}_k} \mathbf{w}_k^* = F_k \tilde{\mathbf{x}}_k + \mathbf{v}_k, \quad (4.60)$$

where $P_{\mathbf{y}_k}$ and $Z_{\mathbf{w}_k}$ denote the projection matrices associated with the equality constraints, and $\mathbf{y}_k^*$, $\mathbf{w}_k^*$ are the corresponding optimal decision variables. As the derivation of $\mathbf{y}_k^*$, $\mathbf{w}_k^*$, and the matrices $P_{\mathbf{y}_k}$, $Z_{\mathbf{w}_k}$ is identical to that in [42], [43], it is omitted here. Importantly, since $\mathbf{y}_k^*$ and $\mathbf{w}_k^*$ depend on the tracking error $\tilde{\mathbf{x}}_k$, the left-hand side of (4.60) can be rearranged to match the affine form on the right-hand side.

Consequently, the feedback control law in (4.44) is obtained by identifying F_k and $\mathbf{v}_k$ as

$$F_k = -P_{\mathbf{y}_k} (N_{\mathbf{u}_k} P_{\mathbf{y}_k})^\dagger N_{\mathbf{x}_k} - Z_{\mathbf{w}_k} (Z_{\mathbf{w}_k}^T M_{\mathbf{u}\mathbf{u}_k} Z_{\mathbf{w}_k})^{-1} Z_{\mathbf{w}_k}^T M_{\mathbf{u}\tilde{\mathbf{x}}_k}, \quad (4.61a)$$

$$\mathbf{v}_k = -P_{\mathbf{y}_k} (N_{\mathbf{u}_k} P_{\mathbf{y}_k})^\dagger (N_{\mathbf{x}_k} \mathbf{x}_k^{ref} + \mathbf{n}_k) - Z_{\mathbf{w}_k} (Z_{\mathbf{w}_k}^T M_{\mathbf{u}\mathbf{u}_k} Z_{\mathbf{w}_k})^{-1} Z_{\mathbf{w}_k}^T m_{\mathbf{u}_k}. \quad (4.61b)$$

Since the procedure is iterative, the matrices of the *cost_to_go* and *constraint_to_go* functions must be updated at each step. The update rules for $H_{\mathbf{x}_k}$, $\mathbf{h}_k$, $V_{\tilde{\mathbf{x}}_k}$, $\nu_{\tilde{\mathbf{x}}_k}$ follow directly from the original algorithm in [42], [43], and are therefore omitted for brevity.

4.2.2 Results

Here, the optimal control input sequence obtained with the MPC solution is compared to the EFC solution. The setup follows the scenario introduced in Example 4.1.1, but is adapted to highlight the differences between the two approaches.

Example 4.2.1

A broker assists two groups of clients, denoted by $\mathcal{I}_1$ and $\mathcal{I}_2$, in managing their investment portfolios. Periodically, the broker updates the portfolios by reallocating among the available financial products, here simplified to three options: $\mathcal{P}_1$, $\mathcal{P}_2$, and $\mathcal{P}_3$.

The broker receives instructions from the company to maximize profits while accounting for customers' risk and return profiles. In practice, this means that recommending low-risk investments to high-risk clients is acceptable, but the reverse is not. For simplicity, this restriction is omitted in the present example.

The company further requires the broker to steer clients' preferences toward a specific product $\mathcal{P}^{ref}$. To influence decisions, the broker has two levers:

- adjusting interest rates by incorporating shares with different levels of risk into the products;
- modifying the price of a product by increasing or decreasing the quantity of stocks.

The clients exhibit different behaviors: $\mathcal{I}_1$ is a fearless investor, naturally drawn to risky but potentially profitable products, whereas $\mathcal{I}_2$ is largely independent and uses the broker only for transaction execution.

This scenario motivates a mathematical formulation where the broker seeks to influence the probability distribution of clients' choices by controlling the interest rates.

To capture these dynamics, the decision-making processes of the investors are modeled as a DTMC with three states, where each state corresponds to a financial product. The evolution of the system is described by the dynamics of the interest rates $\mathbf{x}_k$, which are influenced by the broker's actions $\mathbf{u}_k$. Here, $\mathbf{u}_k$ represents adjustments to the packages, achieved by adding or removing stocks with different risk levels to modify the corresponding interest rates.

The intrinsic decision-making behavior of the investors is governed by a time-invariant transition probability matrix Q , assumed to be known. To correctly analyze how the broker's actions influence both investors, it is assumed that their inherent (baseline) behaviors are identical, i.e.,

$$Q_k^1 = Q_k^2 = Q \quad \forall k.$$

The heterogeneity of the investors is introduced instead through the influence matrices H , which encode how sensitive each investor is to variations in the interest rates. Specifically,

$$Q = \begin{bmatrix} 0.6 & 0.3 & 0.1 \\ 0.4 & 0.4 & 0.2 \\ 0.3 & 0.5 & 0.2 \end{bmatrix}, \quad H_1 = \begin{bmatrix} 0.5 & 0.25 & 0.25 \\ 0.3 & 0.5 & 0.2 \\ 0.2 & 0.2 & 0.6 \end{bmatrix},$$

where H_1 corresponds to the fearless investor $\mathcal{I}_1$. For the independent investor $\mathcal{I}_2$, the influence of the interest rate is null, and therefore $H_2 = \mathbf{0}_{3 \times 3}$.

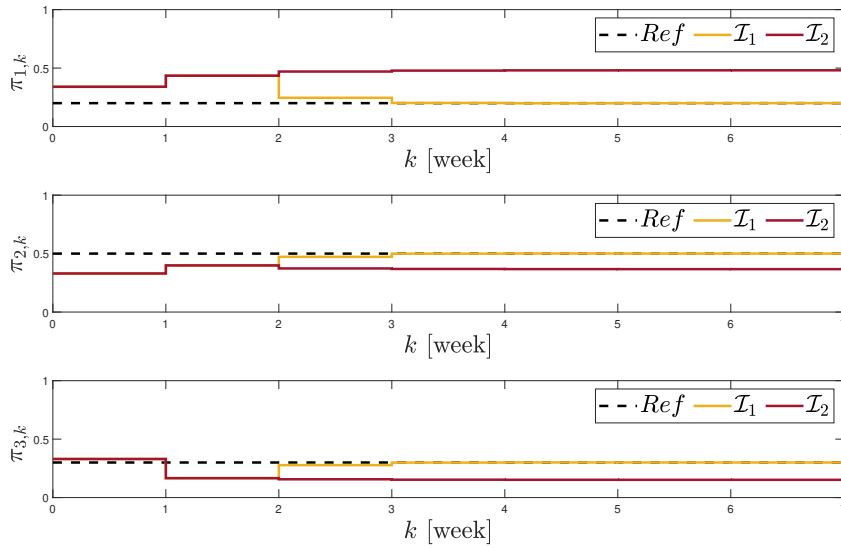


Figure 4.4: Decision-making process evolution of $\mathcal{I}_1$ (yellow line) and $\mathcal{I}_2$ (red line). The reference probability is shown as the black dashed line.

In the considered case study, the dynamics of the interest rate variations are simplified as a linear system:

$$\mathbf{x}_{k+1} = A\mathbf{x}_k + B\mathbf{u}_k.$$

For this example, assume to set $A = I_{3 \times 3}$, implying that rates remain unchanged in the absence of external actions. The input matrix B specifies how the broker's actions impact the interest rates. A straightforward choice is $B = I_{3 \times 3}$, meaning the broker can directly and independently act on each package by buying or selling shares with different levels of risk. Crucially, the state matrices in (4.7) must be chosen such that the system is fully controllable.

In the following, the evolution of the opinions of the fearless investor $\mathcal{I}_1$ is analyzed under the broker's actions. Trivially, since $H_2 = \mathbf{0}_{3 \times 3}$, the independent investor $\mathcal{I}_2$ cannot be influenced. The focus, therefore, is on deriving the control actions required to steer the decision-making process of $\mathcal{I}_1$.

4.2.2.1 MPC performances

Here, the results obtained with the MPC are presented. In the following section, these results will be compared with the performances achieved using the EFC.

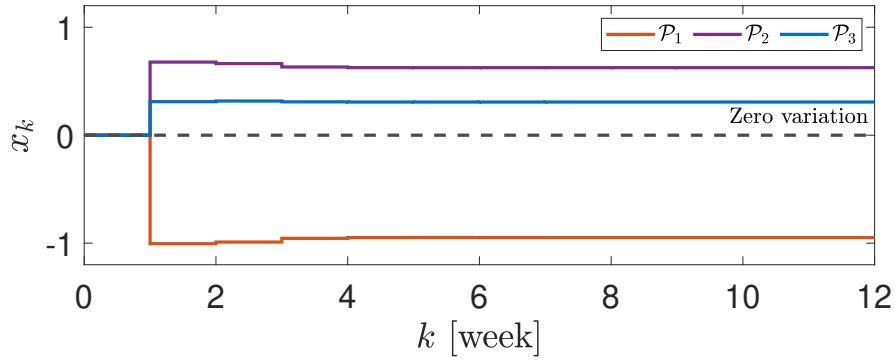


Figure 4.5: Variation of interest rates resulting from the broker's actions on the share composition for influencing the fearless investor $\mathcal{I}_1$. The rates associated with products $\mathcal{P}_1$, $\mathcal{P}_2$, and $\mathcal{P}_3$ are shown in red, purple, and blue, respectively.

All simulations were conducted in *Matlab R2023a* over a horizon of 12 steps, each corresponding to one week. For clarity, the figures display only the first 7 steps, as no significant changes were observed beyond this point.

The reference probability vector and the initial conditions for both Π and $\mathbf{x}$ were set as

$$\Pi^{ref} = [0.2, 0.5, 0.3], \quad \Pi_0 = [1/3, 1/3, 1/3], \quad \mathbf{x}_0 = [0, 0, 0].$$

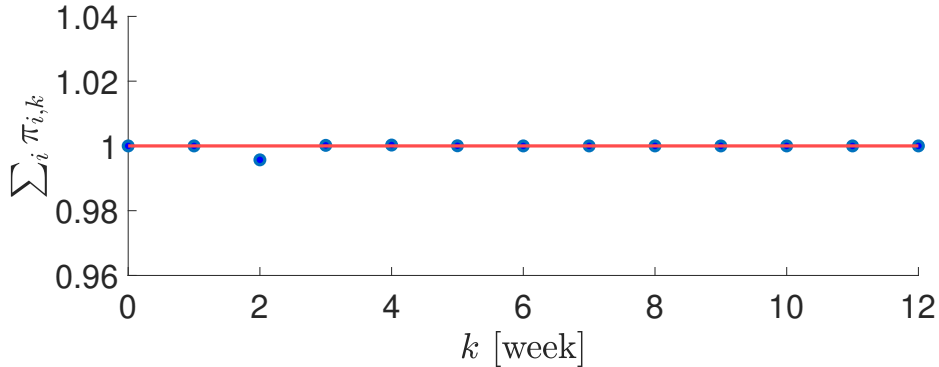


Figure 4.6: Verification that Π_k remains a valid probability vector under the MPC solution. The sum of its entries must equal 1.

The control objective is to shift the overall preference of $\mathcal{I}_1$ towards the share package $\mathcal{P}_2$. No explicit reference is assigned to the state x_k (although trivially $\mathbf{x}^{ref} = \mathbf{0}_{3 \times 1}$ could be set), since the unique objective is to steer the investor's probability vector. Accordingly, the weighting matrix is chosen as

$$Q_{err} = \text{diag}([100, 100, 100, 0, 0, 0]),$$

which penalizes tracking errors in the probability vector but not in $\mathbf{x}$. The input weighting matrix is set as

$$R = \text{diag}([0.1, 0.1, 0.1]).$$

The evolution of the probability vectors of both investors is reported in Fig. 4.4. As expected, $\mathcal{I}_1$ is influenced (and thus controlled) by the broker's actions on the share composition, while $\mathcal{I}_2$ remains unaffected. The zero tracking error is achieved, confirming successful control of the preference dynamics.

It is then instructive to analyze the controlled evolution of the interest rates $\mathbf{x}_k$. As illustrated in Fig. 4.5, the broker increases the interest rate of the corresponding package

in order to steer $\mathcal{I}_1$ towards $\mathcal{P}_2$. This adjustment reflects the risk-seeking profile of the investor, who is naturally attracted to stock packages with higher interest rates.

Finally, Fig. 4.6 confirms that Π_k is maintained as a valid probability vector throughout the simulation. A small deviation is observed at $k = 2$, where the sum of entries slightly differs from 1. However, the error is promptly corrected in subsequent steps, restoring the stochasticity of the vector.

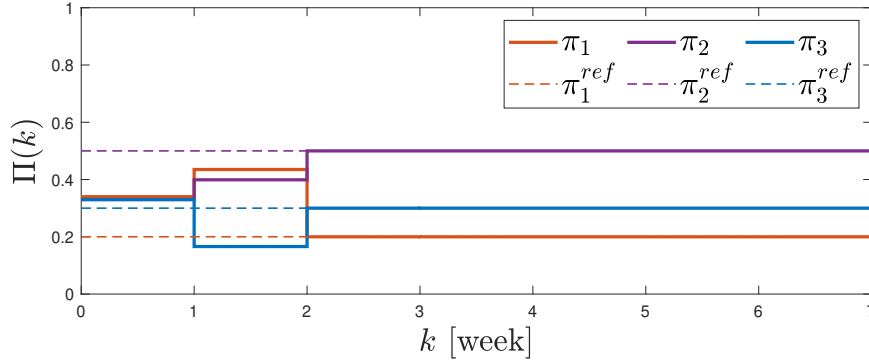


Figure 4.7: Tracking performances by EFC algorithm. The reference and actual probability values are indicated by a dashed and a continuous line respectively.

4.2.2.2 EFC performances

In this section, the results obtained using the EFC algorithm presented in Section 4.2.1 are reported. To ensure a fair comparison, the initial conditions Π_0 and $\mathbf{x}_0$, the cost matrices Q_{err} and R , and the reference Π^{ref} are the same as those used in Section 4.2.2.1. Additionally, the terminal cost matrix is set as $S = Q_{err}$.

The performance of the EFC controller is illustrated in Fig. 4.7, which shows zero tracking error. Notably, the feedback control policy generated by the algorithm achieves faster convergence compared to the MPC solution. This behavior can be explained by recalling that the EFC algorithm computes, at each time step, both a feedback matrix F_k and an affine vector $\mathbf{v}_k$, thereby providing more degrees of freedom than MPC, which can only influence the system through the control input vector $\mathbf{u}_k$.

Furthermore, the stochasticity of the probability vector is verified. As shown in Fig. 4.8, the vector Π remains a valid probability vector at each time step, i.e., the sum of all elements equals 1.

Overall, these results confirm that the modified EFC algorithm generates an effective and reliable control policy.

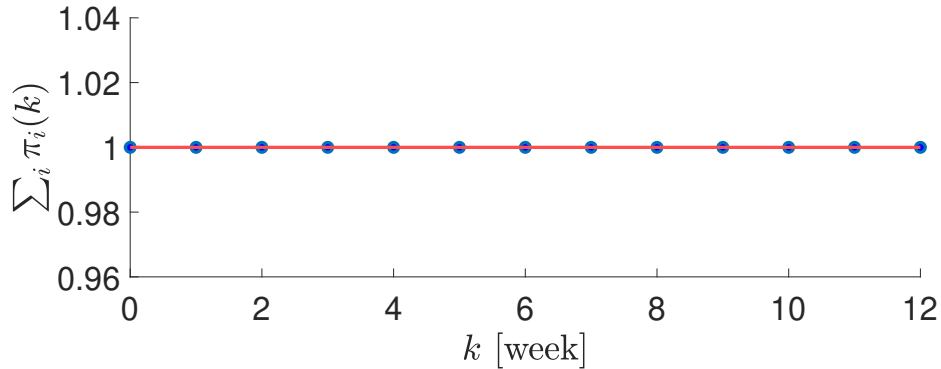


Figure 4.8: Verification that Π_k remains a valid probability vector throughout the simulation under the EFC policy. The sum of all entries must equal 1.

4.3 Discussion

In this chapter, a framework based on MC is presented for modeling the decision-making process of agents interacting with their surrounding environment. In particular, it is assumed that the environment influences this process, and it is demonstrated how a simple MPC scheme can be employed to represent everyday-life scenarios, such as marketing dynamics and social influence. Moreover, an explicit solution to this problem is derived, resulting in a feedback control law that characterizes the agent's optimal decision-making behavior.

In the next chapter, this type of framework is applied to traffic scenarios to model the decision-making process of human-driven vehicles.

Mixed-Autonomy Traffic

Chapter 5

Mixed-Autonomy Traffic Scenarios: Predicting and Exploiting Drivers' Decisions

Overview

Nowadays, emerging technologies are driving the transition toward a future of full autonomy, passing through an intermediate stage of mixed autonomy, where autonomous and human-driven vehicles coexist. This scenario necessarily demands the development of new control strategies for autonomous vehicles, as they must interact seamlessly with human drivers. Consequently, these technologies also require predictive tools capable of forecasting human driving decisions.

In the following two chapters, the problem of inferring human decisions is addressed in the specific context of highway driving, where individuals must comply with a defined set of traffic rules. The proposed framework is designed not only to capture such decision-making behavior but also to serve as a building block within broader control schemes for autonomous vehicle coordination.

5.1 Introduction

Road transportation plays a central role in the global economy but also represents one of the largest contributors to environmental pollution and societal costs [44]. The growing reliance on road mobility generates significant negative externalities, including emissions, noise, congestion, and safety risks, which in Europe alone were estimated at approximately 100 billion euros in 2020 (around 1% of the European Union's GDP) [45]. Addressing these challenges requires not only the design of sustainable infrastructures but also the deployment of advanced traffic management strategies capable of meeting increasing mobility demand efficiently.

Among existing traffic management solutions, *mainstream control* has attracted considerable attention, with strategies such as variable speed limits or lane-change recommendations being widely studied. Traditional implementations, however, rely heavily on variable message signs that provide collective advice at fixed locations, which limits their effectiveness. The emergence of *Connected and Automated Vehicles* (CAVs), equipped with *Vehicle-to-Vehicle* (V2V) and *Vehicle-to-Infrastructure* (V2I) communication, enables individualized and continuous control actions along the road network, thus opening new possibilities for traffic optimization.

The effectiveness of such strategies can be further enhanced by predictive capabilities, either model-based or data-driven. While traffic dynamics in high-density networks have been extensively studied, reliable models capturing the behavior of mixed traffic—including both human-driven vehicles and CAVs—remain limited. The heterogeneity of vehicles and the complexity of their interactions challenge traditional modeling techniques. In this context, learning-based approaches, supported by the availability of large-scale traffic datasets, provide a promising avenue for developing predictive tools tailored to control applications.

Parallel to advances in traffic-level modeling, the development of *Autonomous Vehicles* (AVs) has shifted attention toward safety and decision-making in individual vehicle motion. Beyond physical sensing and collision avoidance, effective autonomy requires anticipating interactions with surrounding vehicles and predicting other drivers' intentions. Human drivers naturally account for such social interactions when making decisions, yet this aspect

remains insufficiently represented in most current AV control algorithms. Consequently, modeling driver interactions and their influence on traffic evolution is a central challenge for developing cooperative and safe control systems in mixed traffic environments.

In this context, the present work addresses driver interactions in highway scenarios through a Markovian behavioral framework. This model is embedded into a broader control strategy for lane-change maneuvers, aiming to enhance both safety and efficiency. Since lane-changing exerts a major influence on both macroscopic and microscopic traffic performance, its effective coordination can reduce travel times, mitigate congestion, and improve safety. Particular attention is given to the role of controlled vehicles acting as *safety cars*, capable of influencing surrounding traffic behavior. To address the nonlinear, stochastic, and hybrid nature of the proposed behavioral model, the control framework adopts an approach inspired by scenario-based *Model Predictive Control* (MPC), which provides a computationally tractable alternative to classical stochastic MPC formulations.

5.2 Related Works

Traffic control has long been a central research topic, and a wide literature addresses strategies for managing road networks [46], [47]. Among them, mainstream control schemes focus on speed regulation and lane-changing recommendations [16], [19]. Despite promising results, their large-scale implementation has been constrained by the limitations of infrastructure-based actuation (e.g., variable message signs). CAV technology provides a more effective platform by enabling continuous, individualized, and network-wide control.

Predictive methods have also been explored to improve control efficiency. Traditional approaches employ macroscopic or microscopic traffic models, which have demonstrated effectiveness in several control frameworks. However, incorporating connected vehicle dynamics and heterogeneous interactions remains an open challenge. Data-driven approaches and learning-based models are increasingly investigated as alternatives to address these complexities [48].

In parallel, research on AV safety has extensively studied obstacle detection, environment perception, and collision avoidance, often leveraging sensors and control algorithms [49], [50], [51]. Nonetheless, these approaches frequently neglect the role of driver decision-making and interactions among road users. Some works have incorporated predictive techniques for traffic flow [52] or trajectory forecasting [53], though comprehensive modeling of social interactions remains limited. Studies such as [54] and [55] highlight the importance of accounting for behavioral interactions, particularly in merging or intersection scenarios.

Recently, deterministic and stochastic multi-agent models have attracted attention for capturing communication and interaction patterns across different domains, including social networks [33], smart grids [56], and traffic systems [57]. In the traffic context, interactions among vehicles strongly influence both individual performance (e.g., fuel efficiency, travel time) and global metrics (e.g., average speed, congestion, safety). Predicting driver cooperation levels has thus emerged as a promising direction to improve path planning and traffic control.

Finally, lane-changing control has been widely investigated, with many contributions relying on MPC-based frameworks [15], [17], [18], [20]. To address the computational complexity of nonlinear and stochastic systems, scenario-based MPC methods have been proposed [58], [59], [60], [61]. Related works have also explored the integration of Markov decision processes and reinforcement learning for lane-following and cooperative control [21], [22], though their predictive use for driver decision-making remains underexplored.

5.3 Contributions

As discussed in the introduction, emerging vehicle technologies demand new tools for prediction, control, and optimization of traffic, particularly in mixed-autonomy scenarios. In this section of the thesis, two predictive tools are developed—one macroscopic and one

microscopic—both based on the *Markov Chain* (MC) framework. The microscopic tool is then integrated into a control framework for managing the maneuvers of AVs in highway scenarios.

To summarize, the contributions of this chapter are:

- The formulation of a macroscopic tool for predicting the conditions of highway sections, based on the MC framework - Chapter 6, Section 6.1;
- The formulation of a microscopic tool for predicting the individual decisions of human drivers in highway scenarios, also within the MC framework - Chapter 6, Section 6.2;
- The design of a controller for AVs leveraging the microscopic prediction tool - Chapter 7.

Chapter 6

Exploiting Markov Chains to forecast traffic scenarios

In this chapter, two different tools for forecasting traffic scenarios are presented. In the first part, Section 6.1, a macroscopic tool is described for predicting the conditions of the highway, specifically the vehicle density in each discretized cell.

The second part, described in Section 6.2, presents a microscopic forecasting tool for predicting the maneuvers of human-driven vehicles, based on the surrounding environment and the decisions of nearby vehicles.

The contents of this chapter are mainly derived from the works presented in [57], [62].

6.1 Modeling Traffic Scenarios with Markov Chains

In this first section, a modeling framework for forecasting highway conditions at the macroscopic level is introduced. The central idea is the development of an opinion dynamics-based approach to traffic modeling and control, with the objective of overcoming the limitations of purely model-driven methods as well as data-driven techniques.

First, the general formulation of a discrete-time Markovian model is presented as the preliminary. Then, this formulation is adapted and tailored to highway traffic scenarios, resulting in the proposed opinion dynamics model designed to capture and predict traffic evolution.

6.1.1 Discrete-Time Markovian Model

The decision process of a single agent r is once again described as a *Discrete-Time Markov Chain* (DTMC) over a set $\mathcal{S} = \{s_1, \dots, s_M\}$ of M decision states, as in Chapters 3,4. The state transition probabilities $\Pr(s_{k+1}^r = s_j \mid s_k^r = s_i) = p_{ij,k}^r$ at the discrete time step $k \in \mathbb{Z}$ are encoded in the non-negative, row-stochastic matrix $Q_k^r \in \mathbb{R}^{M \times M}$, where $Q_{i,j}^r(k) = p_{ij,k}^r$ for all $i, j \in \{1, \dots, M\}$. The probability distribution $\Pi_k^r = [\pi_{1,k}^r, \dots, \pi_{M,k}^r]^T$ over $\mathcal{S}$ describes the decision probabilities of agent r and evolves according to

$$\Pi_{k+1}^r = (Q_k^r)^T \Pi_k^r, \quad (6.1)$$

starting from an initial distribution Π_0^r .

A network $\mathbb{X}$ of Z discrete-time Markovian agents is then defined, where multiple agents may undergo state transitions simultaneously at each step k . Following [2], the network state is denoted by $\mathbb{X} = \langle s^1, \dots, s^Z \rangle$, and the corresponding state set is $\mathcal{S}^{\mathbb{X}} = \mathcal{S}^1 \times \dots \times \mathcal{S}^Z$. Since $\mathcal{S}^r$ represents the decision set of each agent r , the network admits M^Z states. The probability distribution of the entire network evolves as

$$\Pi_{k+1}^{\mathbb{X}} = (Q_k^{\mathbb{X}})^T \Pi_k^{\mathbb{X}}, \quad (6.2)$$

starting from the initial network distribution $\Pi_0^{\mathbb{X}}$.

6.1.2 Fitting the Markovian model for traffic scenarios

It now adapts the generic formulation of a DTMC from the previous section to a one-way road traffic scenario. In particular, it is interested in describing the same scenario as the CTM, namely a road stretch divided into cells crossed by vehicles. For further details on CTM, please refer to Appendix C.

The sets of agents $\mathcal{R}$ and states $\mathcal{S}$ for the network of Markovian agents are first defined. Considering each vehicle as an individual agent may seem intuitive. However,

due to the property of Markov Chains where dynamics can potentially be observed for an infinite time, defining vehicles as agents would restrict observations to a finite time interval, undermining this property. To preserve such a characteristic, it considers the cells in the CTM as agents of the DTMC:

$$\mathcal{C} \equiv \mathcal{R} \iff \{c_1, \dots, c_L\} \equiv \{r_1, \dots, r_Z\}.$$

Henceforth, the index r for a generic Markovian agent is replaced with c .

Next, the possible states that a single agent can assume are discussed. Three distinct states are defined:

- $s_1 = \text{"The number of vehicles increases"};$
- $s_2 = \text{"The number of vehicles decreases"};$
- $s_3 = \text{"The number of vehicles remains constant"}.$

Such states describe *modes* that can be useful also when dealing with lane change or lane merging decisions, especially in the context of a potential control loop. Thus, for the traffic application, the set of decision states is $\mathcal{S} = \{s_1, s_2, s_3\}$.

A relationship between such decision states and the states of the CTM can be established by rewriting (C.1a) as

$$n_{c,k+1} - n_{c,k} = d_{c,k} - d_{c+1,k}, \quad (6.3)$$

where the left-hand side represents the variation of vehicles in cell c over one time step.

Hence, the decision states in $\mathcal{S}$ correspond to the sign of such variation as follows:

- $n_{c,k+1} - n_{c,k} > 0 \iff s_1^c(k),$
- $n_{c,k+1} - n_{c,k} < 0 \iff s_2^c(k),$
- $n_{c,k+1} - n_{c,k} = 0 \iff s_3^c(k).$

Without loss of generality, a unitary step size $\Delta k = 1$ is chosen. Thus, the above discrete derivative of the number of vehicles in cell c corresponds to the state s_i .

Furthermore, it is important to note that the fundamental property of a Markov chain, known as the *memoryless property* [34], holds for the proposed model. This property states that, given the current state s_i , the next state depends only on s_i and not on the past history.

Indeed, according to (6.3), the next state can also be expressed as $s_i^c(k) = d_{c,k} - d_{c+1,k}$. In turn, the inflow $d_{c,k}$ and outflow $d_{c+1,k}$ of the cell c are evaluated following (C.1b). Thus, the numbers of vehicles $n_{c-1,k}$ and $n_{c,k}$ depend on the variation expressed by the previous state $s_i^c(k-1)$ and not on the entire past history. Therefore, the proposed model respects the memoryless property.

6.1.3 Learning the Transition Matrix

Next, the problem of inferring from data the network transition matrices $Q_k^{\mathbb{X}}$ is addressed. This task involves finding the *Maximum-Likelihood Estimator* (MLE) $\hat{Q}_k^{\mathbb{X}}$ [63]. In the literature, the frequentist and Bayesian approaches are commonly considered for this purpose [41]. Since no prior information about the probability distribution of the transition matrix is available, the frequentist approach is adopted.

As described in [64], the MLE $\hat{p}_{ij,k} = [\hat{Q}_k^{\mathbb{X}}]_{(i,j)}$ for the transition probability $p_{ij,k} = [Q_k^{\mathbb{X}}]_{(i,j)}$ in a non-homogeneous system is given by:

$$\hat{p}_{ij,k} = \frac{V_{ij}(k, k+1)}{V_{i+}(k, k+1)}, \quad (6.4)$$

where $V_{ij}(k, k+1)$ represents the number of transitions from configuration X_i to X_j in the time interval $[k, k+1]$, and $V_{i+}(k, k+1)$ is the number of chains that experience a transition starting from configuration X_i in the same time interval.

The result in (6.4) can be adapted to a homogeneous system, as explained in [65]-[66]. In this case, the MLE for p_{ij} becomes:

$$\hat{p}_{ij} = \frac{v_{ij}}{v_{i+}}, \quad (6.5)$$

where v_{ij} is the number of observations from configuration X_i to X_j , and $v_{i+} = \sum_{j=1}^M v_{ij}$ represents the total number of observations starting from configuration X_i .

Although building either (6.4) or (6.5) upon experimental data is of practical relevance, in this work state trajectories from the CTM are used. Using the CTM trajectories instead of experimental data allows it to be assessed whether the DTMC can reproduce the behavior emerging from a well-known and widely accepted model like the CTM. Thus, CTM is used both for learning the model and for validation.

While the MLE in the homogeneous case can be easily estimated by counting the occurrences of state transitions in a single state trajectory of the CTM, the estimator for the non-homogeneous case requires multiple trajectories to gather information about the time variability of the state transition rates.

6.1.3.1 Issue in the transition matrix learning process and its possible solution

The issue of learning non-row-stochastic matrices $Q_k^{\mathbf{x}}$ is a potential challenge in both homogeneous and non-homogeneous systems. This can arise because the predictions from the deterministic CTM may not cover all possible network configurations at each time step, leading to missing transitions in the inferred matrices. In the CTM framework, while the state of the first cell can vary from the initial steps due to the stochastic input $n_{\text{gate},k}$, the states of the remaining $Z - 1$ cells are deterministic and remain constant over the initial executions. Consequently, it is not guaranteed that all possible network configurations are reached at least once in the CTM executions, resulting in zero elements in certain rows of the matrix $Q_k^{\mathbf{x}}$.

Inspired by [67], where a stochastic version of CTM is developed, a stochastic term, namely a noise, is added to (C.1):

$$n_{c,k+1} = n_{c,k} + d_{c,k}\Delta k - d_{c+1,k}\Delta k + \eta_k, \quad (6.6a)$$

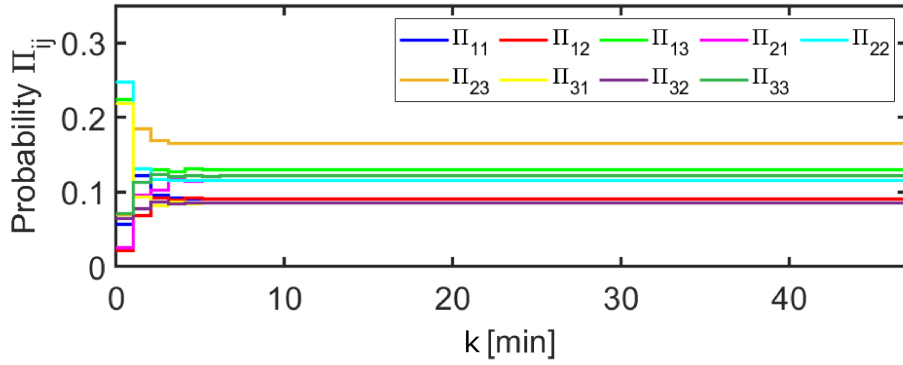
$$d_{c,k} = \min \left\{ \frac{n_{c-1,k}}{\Delta k}, D_{c,k}, \frac{N_{c,k} - n_{c,k}}{\Delta k} \right\}. \quad (6.6b)$$

The noise term η_k describes additional vehicles entering the system and is constrained to be an integer to preserve the feasibility of the model. To ensure that the vehicle count remains non-negative ($n_{c,k+1} \geq 0$), η_k must also be non-negative. Incorporating a disturbance term can help resolve the issue, but it may not be sufficient on its own. Simulations indicate that a few initial time steps must be discarded to accurately infer the transition matrices. This corresponds to the *burn-in* (or *warm-up*) phase of Markov chains ([68]). Owing to the memoryless property, discarding these initial steps does not negatively affect the Markov chain.

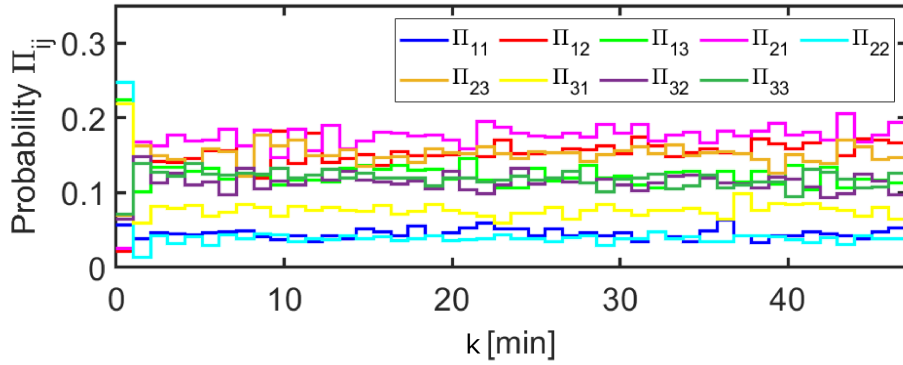
In conclusion, adding a disturbance term to the CTM equations can partially address the issue of non-row-stochastic matrices. However, further considerations, such as discarding initial time steps, may be necessary to ensure the validity of the inferred transition matrices.

6.1.4 Simulation and Results

In this section, the CTM and DTMC models are compared, and the obtained results are highlighted. It is argued that the non-homogeneous DTMC model is to be preferred over the homogeneous one. This claim is based on the superior accuracy of the non-homogeneous model in capturing the dynamics of the CTM, as shown next through the simulation results of a small-scale example.



(a) Homogeneous system.



(b) Non-homogeneous system.

Figure 6.1: Comparison between probability evolutions of the homogeneous and non-homogeneous DTMCs with $Z = 2$ and $M = 3$. Considering the configuration with the highest probability at each instant k as the actual configuration, the homogeneous DTMC can only predict which of the M^Z configurations is most probable in a single execution of the CTM. In contrast, the non-homogeneous DTMC can predict not only the most probable configurations at each instant k , but also the transitions between configurations.

6.1.4.1 Comparing homogeneous and non-homogeneous DTMCs

In this example, a CTM with 2 cells is considered to derive the transition matrices of DTMCs with $Z = 2$ and $M = 3$. Figure 6.1a illustrates the probability vector evolution $\Pi_{\mathcal{X},k}$ of the homogeneous DTMC. Due to the small dimension of the Markov chain, all network configurations are assumed to be visited at least once during the simulation, ensuring that the corresponding Markov chain is ergodic ([34]). Consequently, $\Pi_{\mathcal{X},k}$ converges to a unique, stationary state $\bar{\Pi}_{\mathcal{X}}$.

This result is compared with Fig. 6.1b, which shows the probability evolution of a non-homogeneous DTMC to establish which of the two models provides a more accurate description.

If the configuration with the highest probability at each instant k is considered as the occurred configuration, it is observed that while the homogeneous DTMC can only predict the most likely network configuration in a single CTM trajectory, the non-homogeneous DTMC can predict the configurations at each instant k as well as the transitions between them. Therefore, the non-homogeneous DTMC should be employed when predicting the evolution of configurations over time.

However, employing a non-homogeneous DTMC introduces a limitation: multiple CTM trajectories are necessary to infer the transition matrices of a single non-homogeneous DTMC. Consequently, the configurations predicted by a non-homogeneous DTMC should be compared against multiple CTM state trajectories.

For subsequent comparisons, the DTMC results are evaluated against the *modal*

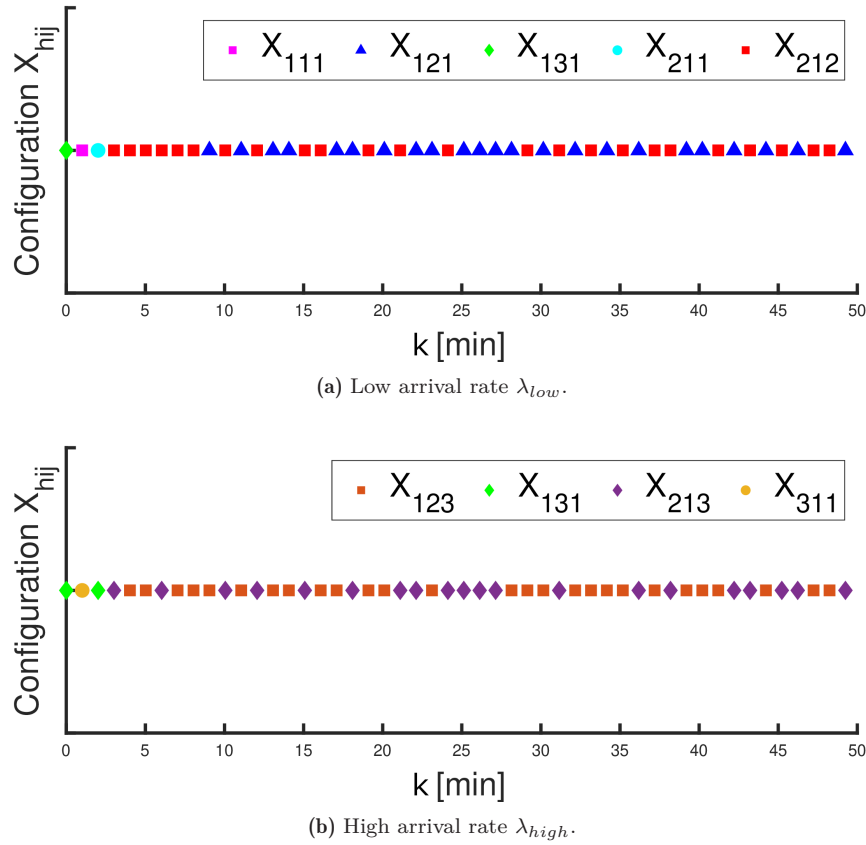


Figure 6.2: Comparison of CTM modal configurations under low and high arrival rates. Distinct symbols (varying in shape and color) represent different configurations X_{hij} exhibited by the CTM over time.

configurations of the CTM. Nonetheless, predicting the average behavior of one-way roads remains of considerable interest.

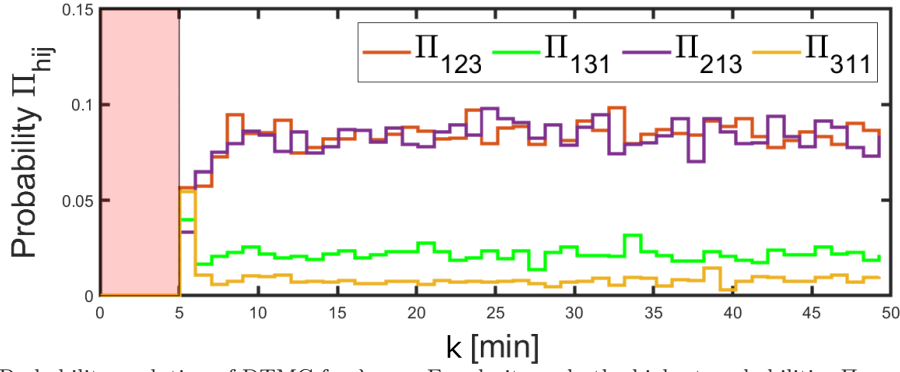
By integrating the DTMC framework into a predictive control algorithm, the model can anticipate the average traffic behavior of a road stretch. This predictive capability provides insights into future traffic conditions and supports planning of control actions. More generally, the DTMC framework allows a macroscopic description of traffic, emphasizing collective traffic flow behavior rather than individual vehicles. This macroscopic perspective captures overall trends and patterns in traffic dynamics, including variations in cell density, flow rates, and congestion levels.

6.1.4.2 Exploiting a small-scale network example to assess the agreement between CTM and DTMC predictions

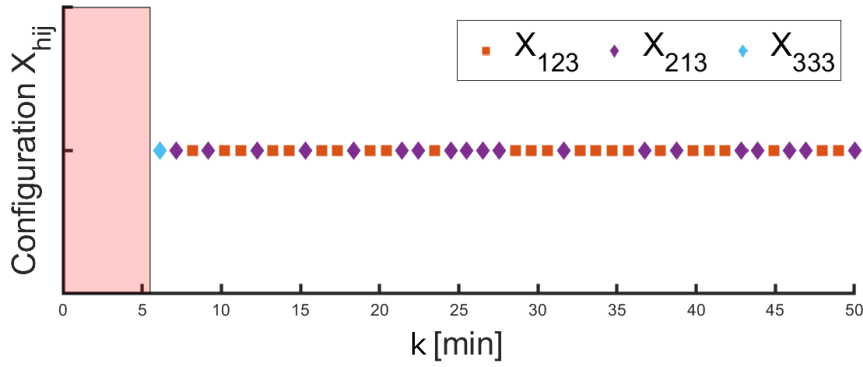
The consistency between CTM and DTMC predictions is analyzed across various traffic scenarios, emphasizing the influence of key parameters on the resulting configurations. The evolution of the vehicle count, described by (C.1), primarily depends on $D_{c,k}$ and $N_{c,k}$. Additionally, as discussed in Section C.0.2, the arrival rate λ impacts the system. By varying $(D_{c,k}, N_{c,k}, \lambda)$, multiple traffic scenarios can be generated.

Parameters are assumed to be time-invariant, reflecting a constant road structure. Two types of one-way roads are considered:

1. Uniform roads, where $N_c = N$ and $D_c = D$ for all $c \in \mathcal{C}$ (see Case 1).
2. Roads with a narrowing in the last cell, where $N_1 = \dots = N_{c_{L-1}} > N_{c_L}$ (see Case 2).



(a) Probability evolution of DTMC for λ_{high} . For clarity, only the highest probabilities Π_{123} and Π_{213} and the additional two configurations observed in Fig. 6.2b, Π_{131} and Π_{311} , are presented.



(b) DTMC configurations for λ_{high} derived from the probability evolution in Fig. 6.3a. Each configuration X_{hij} corresponds to the state with the highest probability Π_{hij} .

Figure 6.3: Evolution of the estimated DTMC configuration chain for λ_{high} , to compare with Fig. 6.2b. The highlighted red zone indicates unreliable predictions due to non-row-stochastic inferred transition matrices.

For the simulations, a small-scale network with $Z = 3$ agents and $M = 3$ states is considered, resulting in $M^Z = 27$ possible configurations. Using (6.6), state trajectories of the CTM are generated following the procedure in Section 6.1.2. These trajectories allow inference of the transition matrices $D_k^{\mathbb{X}}$ and subsequent simulation of the DTMC with random initial conditions. The DTMC results are then compared against the modal CTM predictions. In Case 1, the effect of λ is analyzed, whereas in Case 2, both Q_c and λ are varied.

- *Case 1:* Assuming $N_c = N = 50$ vehs and $D_c = D = 25$ vehs/min, with outflow $D_{out} = D$, and stochastic disturbance $\eta_k = \mathcal{U}(0 \text{ veh}, 2 \text{ vehs})$, two arrival rates are considered: $\lambda_{low} = 5$ vehs/min (Fig. 6.2a) and $\lambda_{high} = 50$ vehs/min (Fig. 6.2b). The modal CTM plots indicate that a low arrival rate alternates between X_{121} and X_{212} , while a high arrival rate alternates between X_{123} and X_{213} . This demonstrates that the arrival rate primarily affects the vehicle count in the last cell, particularly for uniform roads. High arrival rates result in a consistent density in the last cell.

Using the CTM trajectories, transition matrices are inferred, and the DTMC probability evolution is computed (Fig. 6.3). Fig. 6.3a shows $\Pi_{\mathcal{X}}$ according to (A.7). For clarity, only selected state probabilities Π_{hij} are presented. The corresponding configurations are determined as

$$X_{hij,k} = \arg \max_{\substack{\Pi_{hij,k} \in \Pi_k, \\ h \in \mathcal{S}^1, i \in \mathcal{S}^2, j \in \mathcal{S}^3}} \Pi_{hij,k}. \quad (6.7)$$

Initial configurations cannot be reliably inferred due to non-row-stochastic transition matrices, highlighted by the red zone in Fig. 6.3. Beyond this transient, the DTMC accurately reproduces the modal CTM behavior, confirming its ability to predict the average behavior of a uniform one-way road.

- *Case 2:* For $N_1 = N_2 = 50$ veh, $N_3 = 30$ veh, $D_1 = D_2 = 25$ veh/min, and $\eta_k = \mathcal{U}(0 \text{ veh}, 2 \text{ veh})$, four scenarios are considered by varying $D_3 = \{5, 25\}$ veh/min and $D_{out} = \{5, 25\}$ veh/min, with fixed $\lambda_{low} = 5$ veh/min and $\lambda_{high} = 50$ veh/min. Physically, D_3 and D_{out} correspond to the widths of the last cell entrance and exit, respectively. Table 6.1 summarizes the CTM configurations for each parameter combination. Unlike Case 1, the network behavior is predominantly determined by D_3 and D_{out} rather than λ , and the configurations remain consistent across low and high arrival rates.

Table 6.1: CTM configurations for Case 2 under different parameter combinations.

Network Parameters (D_3, D_{out}, λ)	Network Configurations	
	$\lambda_{low} = 5$ vehs/min	$\lambda_{high} = 50$ vehs/min
A. (5, 5)	X_{123}, X_{213}	X_{123}, X_{213}
B. (5, 25)	X_{333}	X_{333}
C. (25, 5)	X_{123}, X_{213}	X_{123}, X_{213}
D. (25, 25)	X_{121}, X_{212}	X_{121}, X_{212}

For combination A, minimal inflow and outflow result in a constant density in the last cell (s_3). Increasing the outflow (combination B) yields a constant density across all cells, as $D_{out} > D_3$ limits the transfer of vehicles to the last cell. In combination C, the limited outflow constrains vehicles leaving the network, maintaining a constant density in the third cell. Finally, combination D mirrors Case 1, with inflow and outflow insufficient to saturate the last cell ($D_3 = D_{out} < N_3$), preventing congestion.

Comparisons between modal CTM and DTMC predictions for all combinations demonstrate satisfactory agreement.

To sum up, the proposed framework models and analyzes traffic dynamics in a one-way road scenario through a DTMC. This approach captures variations in density across different cells along the road by representing the system dynamics as a sequence of discrete configurations and transitions between them. Each cell of the one-way road is modeled as an individual agent within the DTMC, while the vehicle density in each cell serves as the primary parameter of interest.

6.2 A Behavioral Model for Interacting Vehicles

In this Section, a mathematical framework is presented to predict drivers' intentions within a highway scenario by leveraging the social interactions among vehicles. Unlike the previous section, which provided a macroscopic description of highway evolution, this section employs a microscopic stochastic model to capture uncertainties in driver behavior while maintaining a tractable model structure.

The framework is designed to function as a predictive estimator of vehicle trajectories, thereby enhancing the performance of downstream control algorithms, which are discussed in the subsequent chapter. Once the model capturing vehicle interactions is established, it enables the development of advanced vehicle control strategies. For example, control algorithms can be deployed in one or multiple designated vehicles (e.g., a safety car) to influence traffic flow and guide it toward specific objectives.

6.2.1 Problem definition

Consider a highway scenario as depicted in Figure 6.4:

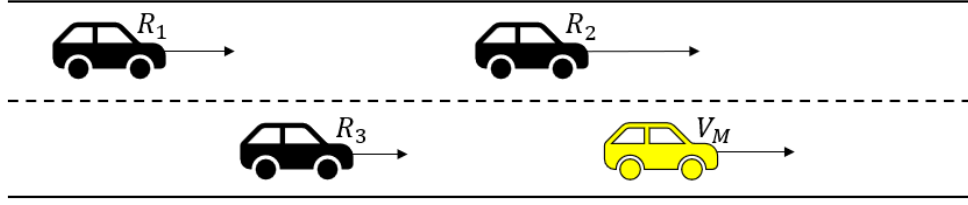


Figure 6.4: Highway scenario.

In this scenario, R_i denotes a human-driven vehicle, that will be modeled in the following as a i^{th} Markovian agent, while V_M represents the autonomous agent (namely, the controlled agent, otherwise called *ego* agent). In practical applications, such an ego agent may correspond to an autonomous vehicle implementing the control or a safety car influencing the behavior of surrounding vehicles.

The objective of this work is to model the interactions among the agents R_i , and to employ the ego agent V_M to control and manage the traffic flow, thereby inducing desired performance outcomes within the traffic system.

A simplified kinematic model is adopted for the dynamics of the ego vehicle, following [69], where the space coordinate y is replaced with the lane index. In a highway with N_ℓ lanes, let $y \in \mathbb{N}$ denote the lane index, where $y = 1$ and $y = N_\ell$ correspond to the rightmost and leftmost lanes, respectively. Moreover, lane changes are assumed to be instantaneous.

Accordingly, the discrete-time motion model is described as

$$\mathcal{M} : \begin{cases} x_{k+1} = x_k + v_k \Delta k, \\ y_{k+1} = y_k + \psi_k, \\ v_{k+1} = v_k + a_k \Delta k, \end{cases} \quad (6.8)$$

where $x_k \in \mathbb{R}$ denotes the longitudinal position, $y_k \in \mathbb{N}$ the lane index, and $v_k \in \mathbb{R}$ the velocity. The control inputs are $\psi_k \in \mathbb{Z}$, representing the lane-change command, and $a_k \in \mathbb{R}$, the longitudinal acceleration. Thus, the system state is given by $\mathbf{x}_k = [x_k, y_k, v_k]^T$, and the control input vector is $\mathbf{u}_k = [a_k, \psi_k]^T$.

The control actions $[a_k, \psi_k]$ are determined by each vehicle based on interactions with a subset of surrounding peers, according to the decision-making model that will be discussed in the following. It is therefore necessary to specify the set of surrounding vehicles with which the ego vehicle interacts.

6.2.2 Set of interacting vehicles

Consider an ego vehicle moving according to (6.8), whose decisions result from interactions with a set of surrounding vehicles. This set, defined with respect to the ego vehicle V_M , is denoted as $\mathbf{I}_M$. The interaction between two vehicles is evaluated on the basis of their relative position and velocity.

Definition 6.2.1 (Interacting Vehicles)

let two vehicles V_1 and V_2 be described by the state vector $[x, y, v]$ according to (6.8). Define the vehicle ahead as

$$V_f = \operatorname{argmax}_{V_i, i \in \{1,2\}} x_i,$$

and the vehicle behind as

$$V_b = \operatorname{argmin}_{V_i, i \in \{1,2\}} x_i.$$

The relative states of V_b with respect to V_f are

$$\begin{aligned}\tilde{x}_{bf} &= x_f - x_b, \\ \tilde{y}_{bf} &= y_f - y_b, \\ \tilde{v}_{bf} &= v_f - v_b,\end{aligned}\tag{6.9}$$

where $[x_f, y_f, v_f]$ and $[x_b, y_b, v_b]$ denote the position and velocity of the front and back vehicles, respectively. Two vehicles are said to be not interacting if

$$\begin{cases} \tilde{v}_{bf} \geq 0, \\ \text{At least two of the quantities in (6.9) satisfy} \\ \tilde{x}_{bf} \geq \Delta x, \quad \tilde{y}_{bf} \geq \Delta y, \quad \tilde{v}_{bf} \geq \Delta v, \end{cases}\tag{6.10}$$

where $[\Delta x, \Delta y, \Delta v]$ are predefined interaction thresholds.

Notably, the interaction is assumed to be mutual: if V_f is not interacting with V_b , then V_b is not interacting with V_f , and vice versa.

It is useful to provide a visual illustration of Definition 6.2.1. To this end, $\Delta y = 1$ is set in order to minimize the number of possible cases to be considered. Furthermore, it is assumed that the interaction threshold $[\Delta v, \Delta x, \Delta y]$ is given, with all thresholds strictly positive.

In the following, five distinct scenarios corresponding to the absence of interactions are presented.

- a) Assume $\tilde{v}_{ij} > \Delta v, \tilde{x}_{ij} > \Delta x, \tilde{y}_{ij} \leq \Delta y$. Then A_i and A_j are in the same lane or adjacent lanes. However, A_j is sufficiently far from A_i and faster than A_i .

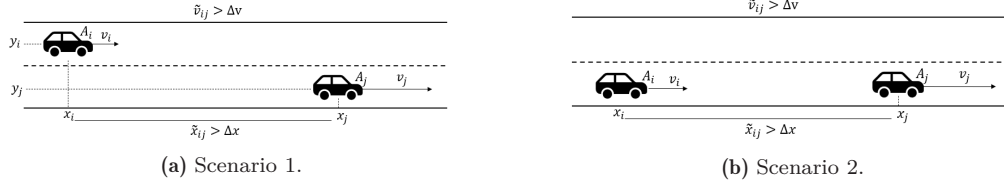


Figure 6.5: Possible scenarios with $\tilde{v}_{ij} > \Delta v, \tilde{x}_{ij} > \Delta x, \tilde{y}_{ij} \leq \Delta y, \Delta y = 1$.

- b) Assume $\tilde{v}_{ij} > \Delta v, 0 \leq \tilde{x}_{ij} \leq \Delta x, \tilde{y}_{ij} > \Delta y$. Then A_i and A_j are in not adjacent lanes and A_j is sufficiently faster than A_i .

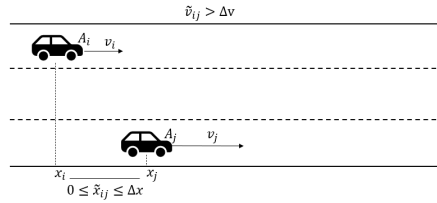


Figure 6.6: Scenario 3 with $\tilde{v}_{ij} > \Delta v, 0 \leq \tilde{x}_{ij} \leq \Delta x, \tilde{y}_{ij} > \Delta y$.

- c) Assume $0 \leq \tilde{v}_{ij} \leq \Delta v, \tilde{x}_{ij} > \Delta x, \tilde{y}_{ij} > \Delta y$. Then the speed of A_i and A_j is similar, but they are far enough distant.
- d) Assume $\tilde{v}_{ij} > \Delta v, \tilde{x}_{ij} > \Delta x, \tilde{y}_{ij} > \Delta y$. Then A_j is sufficiently far from A_i and faster than A_i .

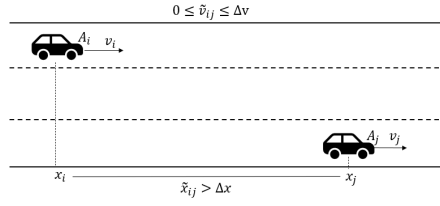


Figure 6.7: Scenario 4 with $0 \leq \tilde{v}_{ij} \leq \Delta v$, $\tilde{x}_{ij} > \Delta x$, $\tilde{y}_{ij} > \Delta y$.

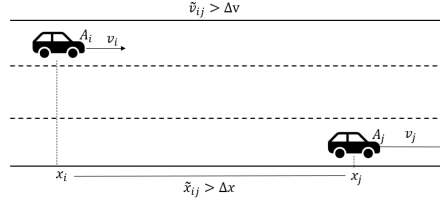


Figure 6.8: Scenario 5 with $\tilde{v}_{ij} > \Delta v$, $\tilde{x}_{ij} > \Delta x$, $\tilde{y}_{ij} > \Delta y$.

6.2.3 Decision-making model of the Autonomous Agent

The decision-making process $\mathcal{D}$ of a single vehicle V_M is modeled as a *Discrete-Time Markov Chain* (DTMC), which evolves according to the linear equation

$$\mathcal{D} : \Pi_{k+1}^r = (Q_k^r)^T \Pi_k^r, \quad (6.11)$$

with the initial condition $\Pi^r(0) = \Pi_0^r$. The state probability vector is $\Pi_k^r \in \mathbb{R}^{m \times 1}$, where m is the number of all possible states from the set $\mathcal{S}^r$. As a probability vector, the sum of all entries of Π_k^r must be equal to 1. The matrix Q_k^r is the transition probability matrix, with entries representing the transition probabilities from the state at time instant k to the state at $k+1$. By construction, Q_k^r is row-stochastic.

The states of the model $\mathcal{D}$ are defined as $\bar{u} = [a, \psi]$, which coincide with the inputs of the model $\mathcal{M}$ (6.8). Hence, the entries of the state vector Π^r of $\mathcal{D}$ represent the probabilities of applying input $\bar{u}$ in the model $\mathcal{M}$, meaning that the Markovian states are the *decisions* made by the vehicle V_M .

To limit the dimensionality of $\mathcal{D}$, a finite set of states is proposed. The set of possible acceleration inputs $\mathbf{A}$ and the set of lane-change commands $\mathbf{L}$ are given as

$$\mathbf{A} = \{-5, -3.5, -1.5, 0, 1.5, 2.5\} \text{m/s}^2, \quad (6.12a)$$

$$\mathbf{L} = \{-1, 0, +1\}. \quad (6.12b)$$

In (6.12b), $\psi = -1$ corresponds to moving to the right lane, $\psi = 0$ to maintaining the current lane, and $\psi = +1$ to moving to the left lane.

Consequently, the set of states $\mathcal{S}^r$ of $\mathcal{D}$ is defined as

$$\mathcal{S}^r = \mathbf{A} \times \mathbf{L} = \{s^M = (a, \psi) \mid a \in \mathbf{A}, \psi \in \mathbf{L}\}. \quad (6.13)$$

6.2.4 Parametric Transition Matrix Design

Here, the derivation of the transition matrix $Q^r \in \mathbb{R}^{m \times m}$ is described (where m denotes the cardinality of $\mathcal{S}^r$), based on the interactions among vehicles.

6.2.4.1 Physically unfeasible transitions

In the previous section, the set $\mathcal{S}^r$ is defined for $\mathcal{D}$, which corresponds to the input space of $\mathcal{M}$. Within $\mathcal{S}^r$ there exist states that $\mathcal{D}$ cannot reach due to physical limitations imposed by the scenario (e.g., number of lanes) or by the vehicle (e.g., feasible velocity

range). Consequently, transition probabilities toward such infeasible states must be set to zero.

Zero probability toward infeasible transitions is enforced by introducing scaling parameters that multiply the nominal transition probabilities. Denote by $\Pr(\cdot)$ the (unscaled) transition probability and by $\widehat{\Pr}(\cdot)$ its scaled version.

Example 6.2.1

Consider a highway with N_ℓ lanes. If vehicle V_M is in the rightmost lane, i.e., $y_M = 1$, then the lane-change command $\psi = -1$ is infeasible. Conversely, if $y_M = N_\ell$ (leftmost lane), then $\psi = +1$ is infeasible.

Accordingly, transitions toward $\psi = -1$ and $\psi = +1$ are modified as

$$\begin{aligned}\widehat{\Pr}(s_{k+1}^M = (a, -1) \mid s_k^M) &= \beta_{-1} \Pr(s_{k+1}^M = (a, -1) \mid s_k^M), \\ \widehat{\Pr}(s_{k+1}^M = (a, +1) \mid s_k^M) &= \beta_{+1} \Pr(s_{k+1}^M = (a, +1) \mid s_k^M),\end{aligned}\quad (6.14)$$

where $\beta_{-1} = (y_{M,k} - 1)$ and $\beta_{+1} = (N_\ell - y_{M,k})$ are scaling factors that account for lane-change feasibility.

Feasibility of acceleration/deceleration commands is treated analogously.

Example 6.2.2

If vehicle V_M is already traveling at the maximum allowed speed, positive accelerations $a \in \{1.5, 2.5\} \text{ m/s}^2$ are infeasible. Conversely, if the vehicle speed is near zero, further deceleration that would produce negative velocity is infeasible.

Using the vehicle dynamics in (6.8), the allowable acceleration and deceleration limits are computed as

$$a^{\max} = \frac{v^{\max} - v_{M,k}}{\Delta k}, \quad d^{\max} = \frac{v_{M,k}}{\Delta k}, \quad (6.15)$$

where $v^{\max}$ is the maximum permitted speed on the highway, $a^{\max}$ is the maximum admissible acceleration (to avoid exceeding $v^{\max}$), and $d^{\max}$ is the maximum admissible deceleration magnitude (to avoid negative speed).

These limits are translated into scaling factors for transition probabilities. Let $\mathbf{A}^+ = \{a \in \mathbf{A} \mid a > 0\}$ and $\mathbf{A}^- = \{a \in \mathbf{A} \mid a < 0\}$. The scaled transition probabilities are then

$$\begin{aligned}\widehat{\Pr}(s_{k+1}^M = (\mathbf{A}^+, \psi) \mid s_k^M) &= \beta_a \Pr(s_{k+1}^M = (\mathbf{A}^+, \psi) \mid s_k^M), \\ \widehat{\Pr}(s_{k+1}^M = (\mathbf{A}^-, \psi) \mid s_k^M) &= \beta_d \Pr(s_{k+1}^M = (\mathbf{A}^-, \psi) \mid s_k^M),\end{aligned}\quad (6.16)$$

with

$$\beta_a = 1 - I^{a_{M,k} > a^{\max}}, \quad \beta_d = 1 - I^{a_{M,k} < -d^{\max}},$$

where

$$I^{a_{M,k} > a^{\max}} = \begin{cases} 1 & \text{if } a_{M,k} > a^{\max}, \\ 0 & \text{otherwise,} \end{cases} \quad I^{a_{M,k} < -d^{\max}} = \begin{cases} 1 & \text{if } a_{M,k} < -d^{\max}, \\ 0 & \text{otherwise.} \end{cases} \quad (6.17)$$

Finally, the lane-change scaling factors $\{\beta_{-1}, \beta_{+1}\}$ are combined with the acceleration/deceleration scaling factors $\{\beta_a, \beta_d\}$ according to the state combination $s_k^M = (a_k, \psi_k)$. For example, the scaled transition probability toward the state $s_{k+1}^M = (2.5 \text{ m/s}^2, +1)$, independently of the previous state, is

$$\widehat{\Pr}(s_{k+1}^M \mid s_k^M) = \beta_a \beta_{+1} \Pr(s_{k+1}^M \mid s_k^M). \quad (6.18)$$

After assessing transition feasibility, the next step is to detail the modeling of lane-changing and acceleration/deceleration transition probabilities. The underlying objective is to design the transition matrix Q^r so that it reflects typical driver behavior given the surrounding scenario.

6.2.4.2 Modeling lane-changing moves transition probabilities

In the modeling of transition probabilities based on vehicle interactions, it is considered that a standard driver V_M typically pays more attention to vehicles ahead rather than behind. Consequently, the set of interacting vehicles $\mathbf{I}_M$ is partitioned into two subsets:

$$\begin{aligned} \mathbf{I}_M^+ &= \{V_i \in \mathbf{I}_M \mid \tilde{x}_{Mi} = x_i - x_M \geq 0\}, \\ \mathbf{I}_M^- &= \{V_i \in \mathbf{I}_M \mid \tilde{x}_{Mi} = x_i - x_M < 0\}, \end{aligned} \quad (6.19)$$

so that $\mathbf{I}_M^+ \cup \mathbf{I}_M^- = \mathbf{I}_M$ and $\mathbf{I}_M^+ \cap \mathbf{I}_M^- = \emptyset$. Each subset is associated with an *impact factor* modeling the influence of the interacting vehicles on V_M . Denote the impact factors as ϵ^+ and ϵ^- for $\mathbf{I}_M^+$ and $\mathbf{I}_M^-$, respectively.

To simplify the analysis, the following assumption is introduced:

Assumption 6.2.1

The study is limited to a highway with $N_\ell = 2$ lanes, and each driver aims to reach the maximum allowed speed v^{max} .

Notably, a possible extension for the above assumption could consider an attitude parameter to model driver behavior, ranging from cautious to aggressive.

The transition probabilities for the ego vehicle are derived by first determining the generally optimal state choice for V_M based on the surrounding environment, and then assigning a probability distribution centered on this ideal state. Dependence on the previous state is then included.

The determination of the optimal state is performed separately for lane-changing and acceleration/deceleration. The probability of performing a lane-change or maintaining the current lane is expressed as

$$\Pr(E) = p_\psi, \quad \Pr(\bar{E}) = 1 - p_\psi, \quad (6.20)$$

where E corresponds to *respecting the indication on the best lane ℓ^** . The best lane ℓ^* is obtained by evaluating lane conditions, such as lane density and average velocity, that standard drivers typically consider.

Based on Assumption 6.2.1 and the distribution in (6.20), the lane-change probabilities are

$$\begin{cases} \Pr(\psi = 0 \mid \ell^* = y_M) = p_\psi, \\ \Pr(\psi = \pm 1 \mid \ell^* = y_M) = 1 - p_\psi, \end{cases} \quad (6.21)$$

if the recommendation is to remain in the current lane, otherwise

$$\begin{cases} \Pr(\psi = \pm 1 \mid \ell^* \neq y_M) = p_\psi, \\ \Pr(\psi = 0 \mid \ell^* \neq y_M) = 1 - p_\psi. \end{cases} \quad (6.22)$$

The best lane ℓ^* is selected according to the cost function

$$\ell^* = \operatorname{argmin}_{\ell \in \{1,2\}} [\epsilon^+ J_\ell^+ + \epsilon^- J_\ell^-], \quad (6.23)$$

where J_ℓ^+ and J_ℓ^- are the cost functions associated with the sets of front and back interacting vehicles, weighted by their respective impact factors. Specifically, the cost function for front vehicles is

$$J_\ell^+ = \rho_d^+ d_\ell^+ - \rho_v^+ \hat{v}_\ell^+ - \rho_{dist}^+ dist_\ell^+, \quad (6.24)$$

where d_ℓ^+ is the normalized number of vehicles in $\mathbf{I}_M^+$ occupying lane ℓ , $\hat{v}_\ell^+$ is the normalized average velocity in lane ℓ , and $dist_\ell^+$ is the normalized minimum distance from V_M to the nearest vehicle in lane ℓ . The weights $\rho_d^+, \rho_v^+, \rho_{dist}^+$ adjust the relative influence of each factor. If no vehicles are present in lane ℓ , then $d_\ell^+ = 0$, $\hat{v}_\ell^+ = 1$, and $dist_\ell^+ = \Delta x / \Delta x^{safety}$.

Similarly, the cost function for back vehicles is

$$J_{\ell}^{-} = \rho_{dist}^{-} dist_{\ell}^{-} + \rho_{\ell}^{-} \tilde{v}_{\ell}^{-}, \quad (6.25)$$

where $dist_{\ell}^{-}$ is the normalized minimum distance from V_M to the closest vehicle behind in lane ℓ , and $\tilde{v}_{\ell}^{-}$ is the relative velocity to that vehicle. If no vehicles are present in lane ℓ , then $dist_{\ell}^{-} = \Delta x$. The evaluation of these interacting sets is repeated at every time step k .

Remark 6.2.1

The cost functions (6.24) and (6.25) are based on lane attributes, such as occupancy, average velocity, and relative distance, that drivers typically evaluate when choosing lanes. The relative importance of each attribute can vary with driver behavior, and the parameters in these cost functions should be estimated from real datasets to reflect diverse driver attitudes.

6.2.4.3 Modeling acceleration/deceleration transition probabilities

For modeling the acceleration/deceleration transition probabilities, a similar approach to the lane-changing case is employed. Specifically, a probability distribution is defined centered on the optimal acceleration/deceleration value. A Gaussian distribution $\mathcal{N}(a^*, \sigma)$ is used, with mean a^* corresponding to the best acceleration/deceleration and standard deviation σ .

The optimal acceleration a^* is determined by considering exclusively the influence of the nearest front vehicle $V_{i,\ell}$ in each lane $\ell = \{1, 2\}$:

$$V_{i,\ell} = \operatorname{argmin}_{V_j \in \mathcal{I}_M^+, y_j = \ell} \tilde{x}_{Mj}. \quad (6.26)$$

After identifying the front vehicle, the relative velocity $\tilde{v}_{Mi}$ is computed:

- If $\tilde{v}_{Mi} \geq 0$, the front vehicle is faster or has the same velocity as V_M , and the distance between vehicles is considered negligible. In this case, $a^* = \tilde{v}_{Mi}/\Delta k$.
- If $\tilde{v}_{Mi} < 0$, the front vehicle is slower and the relative distance must be accounted for to avoid collisions. Using the kinematic model (6.8), the optimal velocity v^* is first computed based on $\tilde{x}_{Mi}$, and then the optimal acceleration is obtained as $a^* = (v^* - v_M)/\Delta k$.

Once the probability distributions for lane-changing and acceleration/deceleration moves are defined, they are combined to construct the probability distribution over the state space $\mathcal{S}^r$ (6.13), denoted as

$$\Pr(s_i \in \mathcal{S}^r), \quad \forall i = 1, \dots, |\mathcal{S}^r|.$$

6.2.4.4 Consider the influence of the previous state

So far, a probability distribution for the states in $\mathcal{S}^r$ has been obtained that does not account for the influence of the previous state, as required for a MC. Therefore, the following considerations are introduced, which are intuitive for most drivers in highway traffic scenarios:

1. a lane-changing move is rarely followed by another lane-change in the next time instant, as such decisions are typically made for overtaking or moving to a faster lane; and
2. while transitions from high positive to low negative accelerations (corresponding to hard braking) are possible, the reverse is less likely due to the vehicle's physical limitations.

The influence of the previous lane-changing move is modeled as a scaling probability. For clarity, the state notation $s_1 = (a \in \mathbf{A}, \psi = 0)$ and $s_2 = (a \in \mathbf{A}, \psi = \pm 1)$ is used. If $\psi_{k-1} = \pm 1$, the scaling parameter $\eta_{\pm 1}$ is applied:

$$\begin{aligned}\widehat{\Pr}(s_k = s_1 | s_{k-1} = s_2) &= \Pr(s_k = s_1) \eta_{\pm 1}, \\ \widehat{\Pr}(s_k = s_2 | s_{k-1} = s_2) &= \Pr(s_k = s_2) (1 - \eta_{\pm 1}).\end{aligned}\quad (6.27)$$

If instead $\psi_{k-1} = 0$, the parameter η_0 is employed:

$$\begin{aligned}\widehat{\Pr}(s_k = s_1 | s_{k-1} = s_1) &= \Pr(s_k = s_1) \eta_0, \\ \widehat{\Pr}(s_k = s_2 | s_{k-1} = s_1) &= \Pr(s_k = s_2) (1 - \eta_0).\end{aligned}\quad (6.28)$$

The influence of the previous acceleration/deceleration move is modeled using a normal distribution $\mathcal{N}(a_{k-1}, \varsigma)$, centered at the previous acceleration a_{k-1} with standard deviation $\varsigma > \sigma$, to account for greater variability.

Finally, all contributions—lane-changing and acceleration move probabilities, dependence on the previous state, and scaling factors for unfeasible transitions—are combined to generate the transition matrix Q^r of the vehicle V_M , which evolves according to the surrounding environment.

As an example, the transition probability from $s_i^M = (0 \text{ m/s}^2, 0)$ to $s_j^M = (2.5 \text{ m/s}^2, +1)$ is given by

$$\widehat{\Pr}(s_j^M | s_i^M) = \beta_a \beta_{\pm 1} (1 - p_\psi) \mathcal{N}(a^*, \sigma) (1 - \eta_0) \mathcal{N}(0 \text{ m/s}^2, \varsigma) \Pr(s_j^M | s_i^M),$$

if $\ell^* = y_M$. Alternatively, if $\ell^* \neq y_M$,

$$\widehat{\Pr}(s_j^M | s_i^M) = \beta_a \beta_{\pm 1} p_\psi \mathcal{N}(a^*, \sigma) (1 - \eta_0) \mathcal{N}(0 \text{ m/s}^2, \varsigma) \Pr(s_j^M | s_i^M).$$

In both cases, $\mathcal{N}(a^*, \sigma)$ and $\mathcal{N}(0 \text{ m/s}^2, \varsigma)$ should be substituted with the probabilities of $a = 2.5 \text{ m/s}^2$, computed according to the respective normal distributions.

6.2.5 Simulations and Results

The decision-making model is applied to a single Markovian vehicle V_M within a highway scenario containing N_v other vehicles. The motion of all vehicles is described by (6.8). For V_M , the control input $\bar{u}_k$ is selected as the DTMC state with the highest probability:

$$\mathbf{u}_k = \operatorname{argmax}_{s \in \mathbf{S}^r} \Pi_k^M.$$

The objective is to evaluate how the decisions of V_M vary based on the motion of the surrounding vehicles, denoted as R_i , $i = \{1, \dots, N_v\}$, to distinguish them from V_M . The simulation considers a two-lane highway scenario with $N_v = 4$ manually controlled vehicles, along with the ego vehicle V_M .

6.2.5.1 Description of the highway scenario

To derive the set of interacting vehicles $\mathbf{I}_M$ for V_M at each time step, it is necessary to define the interaction thresholds $[\Delta x, \Delta y, \Delta v]$.

For the threshold on the x position, the safety distance Δx^{safety} is first computed assuming the worst-case scenario, i.e., when V_M is traveling at its maximum allowed velocity. The safety distance [70] is defined as

$$\Delta x^{\text{safety}} = D_1 + D_2,$$

where $D_1 = v \cdot t_r$ is the reaction distance, depending on the velocity v and reaction time t_r , and $D_2 = \frac{v^2}{2a_b}$ is the braking distance. Assuming a maximum reaction time $t_r \approx 1.5 \text{ s}$ [71], a maximum highway velocity $v = 130 \text{ km/h} \approx 36 \text{ m/s}$, and braking acceleration $a_b \approx 9.8 \text{ m/s}^2$, the safety distance results in $\Delta x^{\text{safety}} \approx 120 \text{ m}$.

Two vehicles are considered non-interacting if their relative distance significantly exceeds this safety distance. Therefore, the threshold is set to $\Delta x = 500 \text{ m} > 3 \cdot \Delta x^{safety}$.

For the lane distance, the threshold is set as $\Delta y = 2$, and for the velocity, $\Delta v = 20 \text{ km/h}$. These thresholds are chosen heuristically; in future work, they could be calibrated using real-world datasets.

All simulations are performed using *Matlab R2023a*, with a total simulation duration of $T = 60 \text{ s}$ and a time step of $\Delta k = 1 \text{ s}$.

6.2.5.2 Motion of the controlled vehicles

The motion of the vehicles R_i , $i = \{1, \dots, 4\}$ is manually controlled by modifying the inputs u_i applied to the kinematic model (6.8). Fig. 6.9 illustrates the inputs for each controlled vehicle.

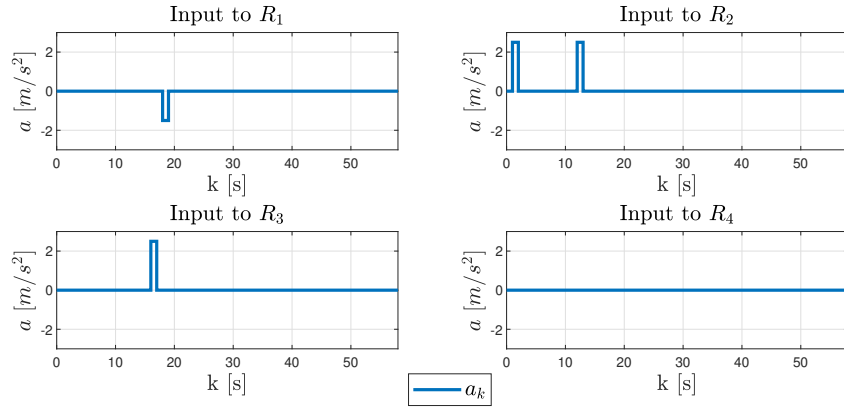


Figure 6.9: Inputs applied to the motion of the controlled vehicles R_i , $i = \{1, \dots, 4\}$. The blue line represents the acceleration input.

At this stage, no control strategy is applied to these inputs, as the objective is to evaluate whether the decision-making model for V_M can predict feasible behavior given the surrounding scenario. To simplify the simulation and prevent collisions among the controlled vehicles, it is assumed that none of the controlled vehicles performs a lane change during the simulation, i.e., $\psi_k = 0 \forall k = \{1, \dots, T\}$ and $\forall R_i$.

The initial states of both the controlled vehicles and the ego vehicle are depicted in Fig. 6.10, and resumed in Tab.6.2.

Table 6.2: Vehicles' initial state.

Vehicle	Vehicle's initial state		
	x	y	v
V_M	0 m	1 st lane	50 km/h
R_1	150 m	1 st lane	80 km/h
R_2	450 m	1 st lane	110 km/h
R_3	300 m	2 nd lane	120 km/h
R_4	600 m	2 nd lane	130 km/h

6.2.5.3 Decisions of the ego vehicles

Based on the motion of the controlled vehicles, the ego vehicle V_M adapts its decisions and, consequently, its kinematic state. The behavior of V_M is compared for a decision-making model that accounts exclusively for front vehicles ($\epsilon^+ = 1, \epsilon^- = 0$, Fig. 6.12(a)) and for a model that accounts for both front and back vehicles ($\epsilon^+ \neq 0, \epsilon^- \neq 0$, Fig. 6.12(b)).

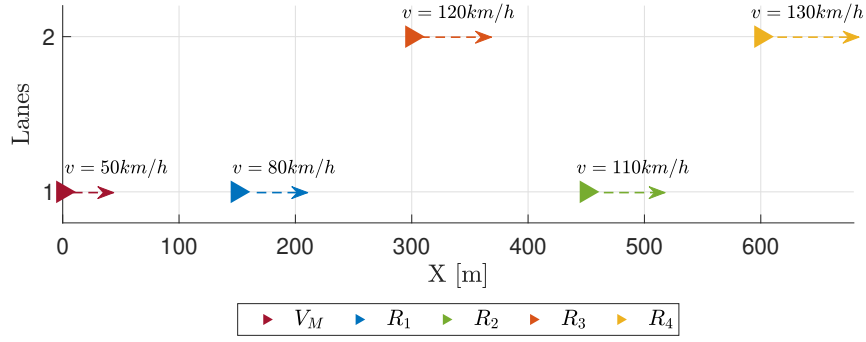


Figure 6.10: Initial conditions of V_M and R_i , $i = 1, \dots, 4$.

In both figures, the red zone indicates the collision risk area, with the dashed red line marking the collision between vehicles. The blue line represents the relative distance ($\tilde{x}_{iM}$, $i = \{1, \dots, 4\}$) between V_M and R_i , while the green zone highlights when the vehicles are in different lanes, preventing collisions even if their relative x distance is zero.

It is evident from Fig. 6.12(a) that, by considering only the front vehicles, V_M exhibits a high collision risk with R_1 . Indeed, V_M changes lanes without accounting for the presence of R_1 behind it, as shown by the blue line intersecting the collision risk zone around $k \approx 22$ s.

Conversely, by setting the impact factors to $\epsilon^+ = 0.7, \epsilon^- = 0.3$, Fig. 6.12(b) demonstrates that the collision risk between V_M and R_1 , as well as with the other controlled vehicles, is eliminated. The motion of V_M is displayed in Fig. 6.11. Notably, V_M cannot reach the maximum allowed speed due to the finite dimension of the acceleration set (6.12a).

For a more detailed analysis, snapshots of the simulation are provided in Fig. 10.3, highlighting the instants when V_M performs lane changes. Each snapshot is accompanied by a colormap representing the evolution of the probability vector Π , showing the probabilities of all possible states in $\mathcal{S}^r$. The x and y axes correspond to the order of entries in the sets $\mathcal{A}$ and $\mathcal{L}$, respectively. The Markovian state $s = (a, \psi)$ with the highest probability corresponds to the actual control input $\mathbf{u}$ applied to the kinematic model of V_M in the subsequent time step.

The modification of the impact factors results in a different behavior for V_M , which moves to the second lane to overtake the slower vehicle R_1 , and returns to the first lane once a safe distance is established from the other back vehicle. The values of the impact factors are chosen heuristically, reflecting that drivers generally consider front vehicles more than rear ones due to the ease of observing their movements.

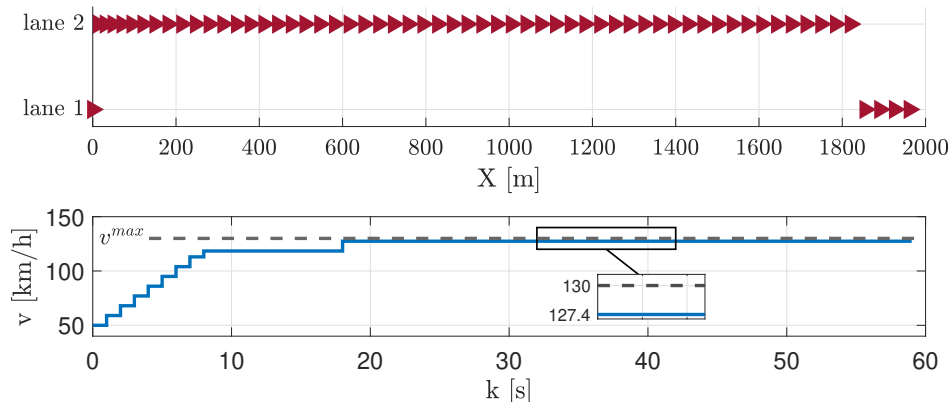


Figure 6.11: Motion of V_M . The top graph shows the spatial motion; the bottom graph the velocity over time.

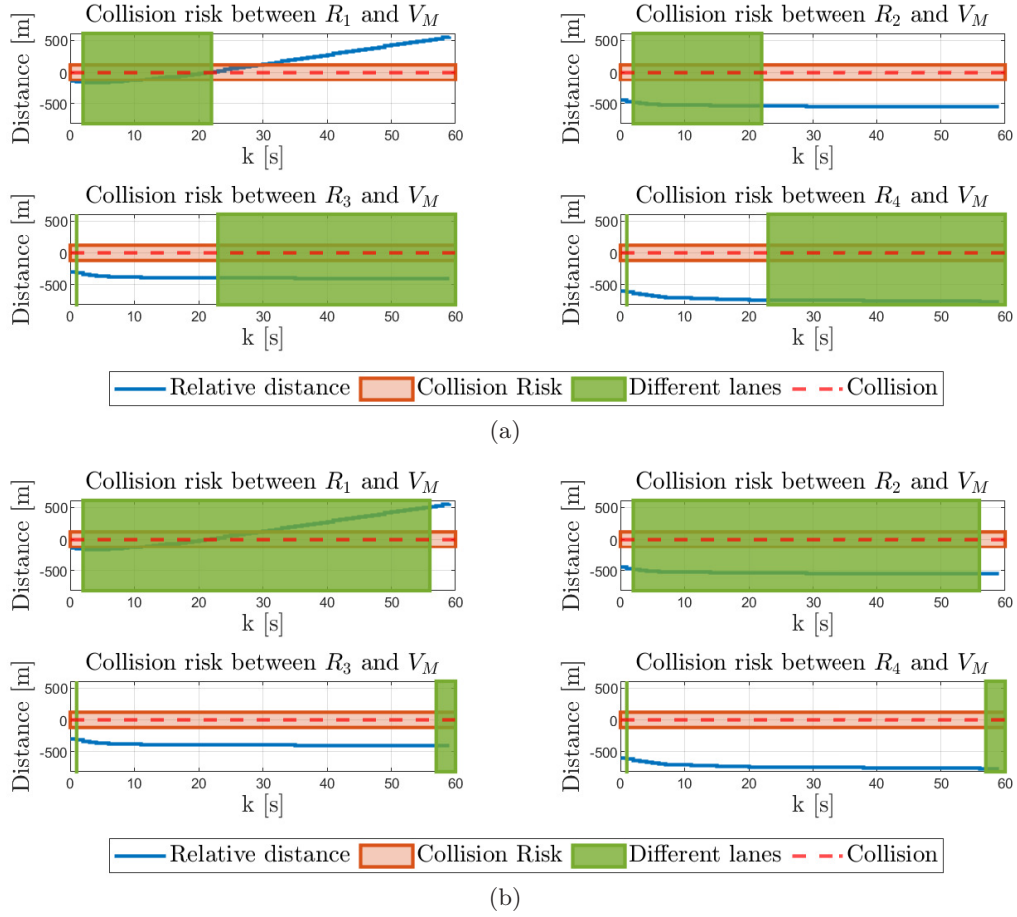


Figure 6.12: Collision risk between R_i and V_M when the decision-making model accounts exclusively for front vehicles (a), and for both front and back vehicles (b).

6.3 Discussion

In this chapter, two complementary approaches for forecasting highway traffic conditions have been presented. Section 6.1 introduced a macroscopic framework based on Markov Chain theory, demonstrating that, with appropriate tuning of the transition matrix, the model can capture the traffic dynamics emerging when the road is divided into sections. In contrast, Section 6.2 focused on a microscopic perspective, employing a Markov Chain model to learn the influences and interactions among individual vehicles. The macroscopic approach is particularly suitable for analyzing large-scale traffic trends and average flow behavior, whereas the microscopic approach is better suited to capturing local interactions and detailed dynamics at the vehicle level.

In the next chapter, the microscopic tool will be integrated into a control framework inspired by the concept of a safety car. In this setting, the controlled agent is tasked with influencing the decisions of another vehicle, leveraging the predictions generated by the proposed tool. This integration illustrates how the microscopic model can be employed not only for analysis but also as a basis for decision-making and control in interactive traffic scenarios.

Chapter 7

Scenario-Based Model Predictive Control for Vehicle interactions in Highway setting

In this chapter, a modeling and control design framework is presented for capturing and influencing drivers' decisions in highway scenarios by employing one or more vehicles as actuators. The proposed approach is based on a driver's decision-making model, which is embedded within a scenario-tree model predictive controller to compute acceleration and lane-change commands for a set of controlled vehicles. The effectiveness of the framework is demonstrated in a two-lane highway example involving two vehicles, one autonomous and one human-driven. Numerical simulation results show that the proposed method can efficiently influence the lane-change behavior of one vehicle by using the other as a control actuator.

The contents of this chapter are mainly derived from the work presented in [72].

7.1 Problem definition

The following illustrative example serves as motivation for the problem addressed in this work. Consider a two-lane highway scenario with two vehicles (Fig. 7.1). In this setup, the yellow vehicle A_C acts as a *control vehicle*, functioning as an *actuator*, and is tasked with influencing the velocity and/or lane position of the blue vehicle A_M . The concept is to utilize A_C in the role of a safety car, i.e., an agent capable of shaping the movements of A_M . In practice, A_C cannot anticipate the exact maneuvers of A_M in advance, but it can predict the most likely decisions of A_M using a decision-making model informed by past observations, and act to steer these toward a desired behavior.

The controller is locally implemented on A_C , and it is assumed that the (absolute or relative) position and velocity of A_M are available through onboard sensors or road infrastructure. In real-world applications, multiple control vehicles could be deployed to influence traffic flow. For example, in lane-merging scenarios, uncoordinated maneuvers of vehicles may cause queues to form along the highway. Equipping vehicles with the capability to influence the decisions of others by adjusting their velocities or lane positions could improve overall traffic performance, including reductions in travel time.

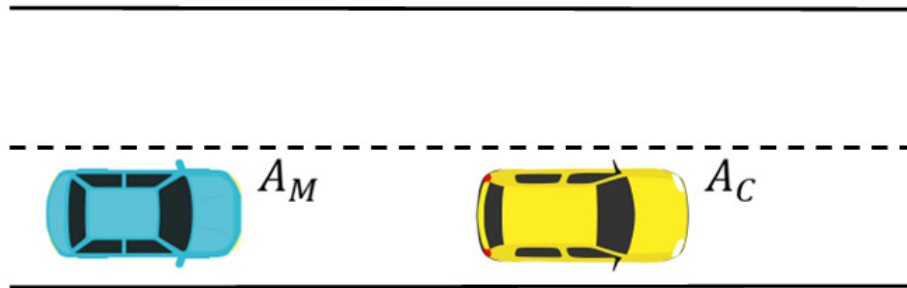


Figure 7.1: Two-lane highway scenario. A_C is the control agent, and A_M is the Markovian agent whose maneuvers are to be influenced.

Hence, the vehicle A_C is designated as the control vehicle, influencing the decisions of the other vehicles in a manner similar to an *autonomous safety car*. A control algorithm

is implemented on A_C , computing its control inputs so as to influence the vehicle A_M and guide its states (or decisions) toward a desired behavior. The decision-making processes within the network of agents are modeled as a Markov process, which yields a linear discrete-time model.

The dynamics of both vehicles are described by model $\mathcal{M}$ in Eq. (6.8), while the decision-making model follows the formulation introduced in Chapter 6, Section 6.2.

7.2 Control architecture

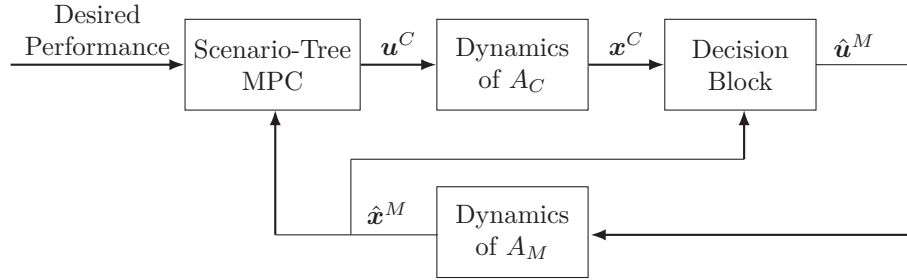


Figure 7.2: Control logics.

As stated in Section 7.1, the control vehicle A_C does not have prior knowledge of the decisions of A_M . However, A_C can predict A_M 's decisions by exploiting the Markovian behavioral model described in Section 6.2, and accordingly derive the best control action.

Fig. 7.2 illustrates the control logic, where the blocks *Kinematic motion model of control vehicle* and *Kinematic motion model of Markovian vehicle* implement the model (6.8) for A_C and A_M , respectively. The block *Markovian behavioral and decision-making model* captures the decision-making process of A_M , providing the input $\hat{u}^M$ to the model (6.8). Finally, the *Scenario-tree MPC* computes the optimal control action $\hat{u}^M$ for A_C , based on the current state of A_M and the control objectives.

The following subsections provide a detailed description of the *Markovian behavioral and decision-making model* and the *Scenario-tree MPC* blocks.

7.2.1 Decision-making block

The *Markovian behavioral and decision-making model* block represents the actual plant of the scheme. Specifically, this block captures the decision-making process of vehicle A_M , which depends on the current states x^C and $\hat{x}^M$, and outputs the predicted action $\hat{u}^M$.

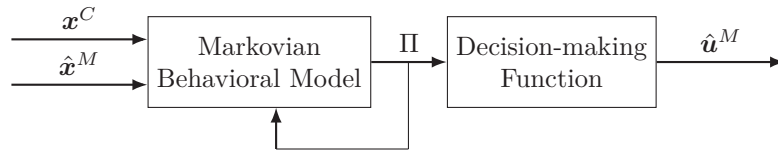


Figure 7.3: Markovian behavioral and decision-making model block.

As illustrated in Fig. 7.3, the *Markovian behavioral and decision-making model* consists of two main components: the behavioral model introduced in Section 6.2 and the *decision-making function*. The behavioral model constructs the transition matrix Q based on the current states of the vehicles and computes the next probability vector Π . The decision-making function, given the probability vector Π , selects one of the states whose probability exceeds a specified threshold p^* . This threshold can be set to zero, allowing A_M to choose among all admissible transitions, or to a positive value, restricting the selection to states with sufficiently high probabilities. The selected state corresponds to the action $\hat{u}^M$, which serves as the input to the kinematic motion model of A_M .

7.2.2 Scenario-tree model predictive control algorithm

The controller design in this approach is inspired by the principles of Model Predictive Control (MPC).

In MPC, the control action is obtained by solving a finite-horizon, possibly constrained, optimal control problem. This problem is formulated as an optimization task, where the plant's evolution is optimized according to a cost function over a finite prediction horizon. The procedure is repeated at each time step over a receding horizon, based on the current system state.

Several variants of MPC exist, addressing nonlinear and/or stochastic models and constraints. In stochastic MPC, a common approach is to construct a *scenario tree*, which describes the evolution of uncertainties or perturbations over the time horizon [58], [59], [60].

In the present setting, the decisions of A_M are modeled by the evolution of a Markov Chain, which yields a probability vector Π . A decision block then selects the input $\hat{\mathbf{u}}^M$ corresponding to the Markovian state in $\mathcal{S}$ with the highest probability. This structure naturally lends itself to a scenario-tree formulation, which can account for the evolution of the probability vector over the control horizon, as in [61].

Algorithm 7.1 Scenario-tree optimizer

```

1:  $T \leftarrow$  Simulation length
2:  $N_c \leftarrow$  Prediction horizon
3:  $\mathbf{x}^{ref} \leftarrow$  Reference
4: for  $k = 1, \dots, T$  do
5:   for iterations  $i = 1, \dots, N_c$  do
6:      $\mathcal{U}_i^C \leftarrow$  Compute all the feasible  $\mathbf{u}^C$  for  $A_C$ 
7:     for  $\mathbf{u}_j^C \in \mathcal{U}_i^C$  do
8:        $\mathbf{x}_j^C \leftarrow$  State of  $A_C$ , applying  $\mathbf{u}_j^C$  in (6.8)
9:       Derive the corresponding  $Q_j$  and  $\Pi_j$ .
10:       $\hat{\mathbf{u}}_j^M \leftarrow \operatorname{argmax}_{s \in \mathcal{S}} \Pi_j$ 
11:       $\hat{\mathbf{x}}_j^M \leftarrow$  State of  $A_M$ , applying  $\hat{\mathbf{u}}_j^M$  in (6.8)
12:       $\text{cost}_j^i \leftarrow \text{cost}_j^{i-1} + J(\mathbf{x}^{ref}, \hat{\mathbf{x}}_j^M)$ 
13:    end for
14:  end for
15:  Find the control input sequence that leads to the minimum cost and apply the first
    control action  $\mathbf{u}^C$ .
16: end for

```

The proposed algorithm is a scenario-tree optimizer (further details in Appendix D). A scenario tree is constructed by considering all admissible control inputs $\mathcal{U}_i^C$ for A_C (ensuring lane-keeping, collision avoidance, and nonnegative velocities), and deriving the corresponding transition matrix and state probability vector (Algorithm 7.1, lines 6–9). The action of the Markovian vehicle A_M is then assumed to correspond to the state with the highest probability (line 10). Once the estimated decision of A_M is obtained, the corresponding state $\hat{\mathbf{x}}^M$ is computed (line 11). Finally, the error with respect to the reference state $\mathbf{x}^{ref}$ is evaluated through a cost function $J(\mathbf{x}^{ref}, \hat{\mathbf{x}}^M)$, which is accumulated along the scenario tree (line 12). This cost function is defined as

$$J(\mathbf{x}^{ref}, \hat{\mathbf{x}}^M) = (\hat{\mathbf{x}}^M - \mathbf{x}^{ref})^T Q_{err} (\hat{\mathbf{x}}^M - \mathbf{x}^{ref}), \quad (7.1)$$

where Q_{err} is a diagonal weighting matrix, with entries corresponding to position, lane, and velocity errors.

Now the optimization problem can be formalized. Let $\mathbf{X}_\gamma = [\hat{\mathbf{x}}_1^M, \dots, \hat{\mathbf{x}}_{N_c}^M]$ denote the γ -th estimated scenario (corresponding to a tree branch). The optimal scenario $\mathbf{X}^*$

is given by

$$\mathbf{X}^* = \operatorname{argmin}_{\mathbf{X}_\gamma \in \mathcal{S}} \left\{ \sum_{i=1}^{N_c} J(\mathbf{X}_i^{ref}, \mathbf{X}_{\gamma,i}) \right\}, \quad (7.2)$$

where $\mathcal{S}$ is the set of all scenarios generated by the tree (with cardinality $|\mathcal{S}|$ equal to the number of leaves), and $\mathbf{X}^{ref}$ is the reference scenario.

After computing the optimal scenario, the corresponding optimal control sequence is the one leading to $\mathbf{X}^*$. Following the standard receding horizon principle, only the first input $\mathbf{u}^{C,*}$ is applied to A_C .

7.3 Simulations and Results

In this section, the results obtained from numerical simulations using the proposed control scheme in the traffic scenario illustrated in Fig. 7.1 are presented. The simulation duration is $T = 60$ s, with a time step of $\Delta k = 1$ s and a prediction horizon of $N_c = 3$. Each simulation requires an average execution time of 1 s.

The objective of the control is to guide A_M to achieve the maximum allowed velocity, set as $v_{\max} = 130$ km/h ≈ 36 m/s, while also controlling its lane position. The desired lane reference is set as $y^{ref} = 1$ until $k = 32$ s, and then switched to $y^{ref} = 2$, with the velocity reference $v^{ref} = v_{\max}$.

As no specific x -position reference is considered, the weighting matrix in the cost function (7.1) is chosen as $Q_{err} = \operatorname{diag}([0, \omega_\ell, \omega_v])$, where ω_ℓ and ω_v are the weights for lane and velocity errors, respectively. Considering the maximum possible errors during the simulation, namely 1 for lane error, since $N_{\text{lanes}} = 2$ and 36 m/s for the maximum velocity, and prioritizing lane tracking, it sets $\omega_\ell = 100$ and $\omega_v = 1$.

All other simulation parameters are summarized in Tab. 7.1.

Table 7.1: Simulation Parameters

Symbol	Parameter Values
ϵ^+, ϵ^-	$[0.7, 0.3]$
$\rho_d^+, \rho_v^+, \rho_{dist}^+$	$[0, 1, 1]$
ρ_v^-, ρ_{dist}^-	$[1, 1]$
p_ψ	0.7
$\eta_{\pm 1}, \eta_0$	$[0.7, 0.5]$
σ	1 m/s ²

The weight on the number of vehicles per lane, ρ_d^+ , is set to zero since the scenario involves only two vehicles. The initial states are set as $\mathbf{x}_{k=0}^C = [150 \text{ m}, 1, 80 \text{ km/h}]$ and $\mathbf{x}_{k=0}^M = [0 \text{ m}, 1, 50 \text{ km/h}]$, with a random initial probability vector $\Pi_{k=0}$. For the Markovian vehicle A_M , the decision threshold is set to $p^* = 1/6$ to capture the variability in human behavior while still assuming rational actions.

Ten simulation runs are performed, and the most frequent kinematic states of the vehicles are presented in terms of velocity (Fig. 7.4) and lane position (Fig. 7.5).

Fig. 7.4 shows that both A_C and A_M accelerate to nearly the maximum reference velocity (dashed black line). The small tracking errors arise from the finite set of feasible accelerations in $\mathcal{A}$. Fig. 7.5 depicts the lane trajectories, with initial positions highlighted by yellow circles. It can be observed that A_C successfully induces a lane change in A_M around $x \approx 1200$ m, consistent with the control objective.

Both figures display the modal kinematic states over multiple simulations to illustrate the typical behavior of the vehicles.

For a deeper insight, Fig. 7.6 shows snapshots from a single simulation at $k = \{32, 33, 34\}$ s, including the positions of A_C and A_M and the corresponding probability vectors of A_M . Probabilities are color-coded from blue (low) to yellow (high). The x -axis

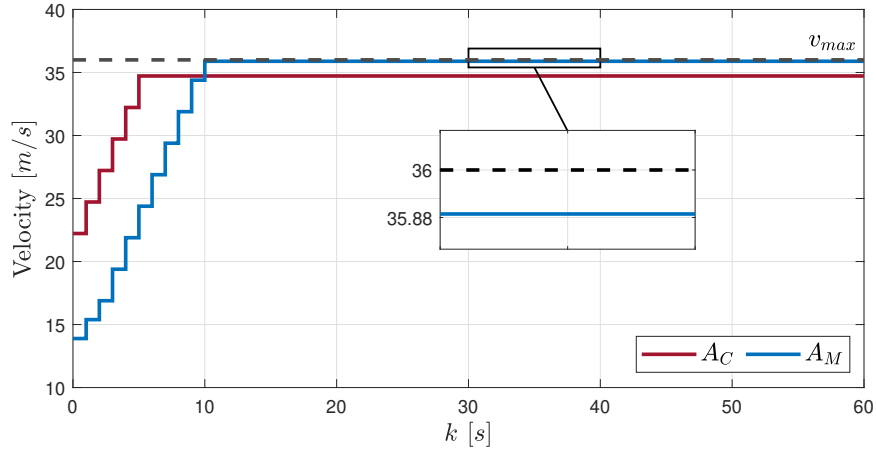


Figure 7.4: Speed evolution of A_C (red) and A_M (blue). The dashed black line represents the reference velocity.

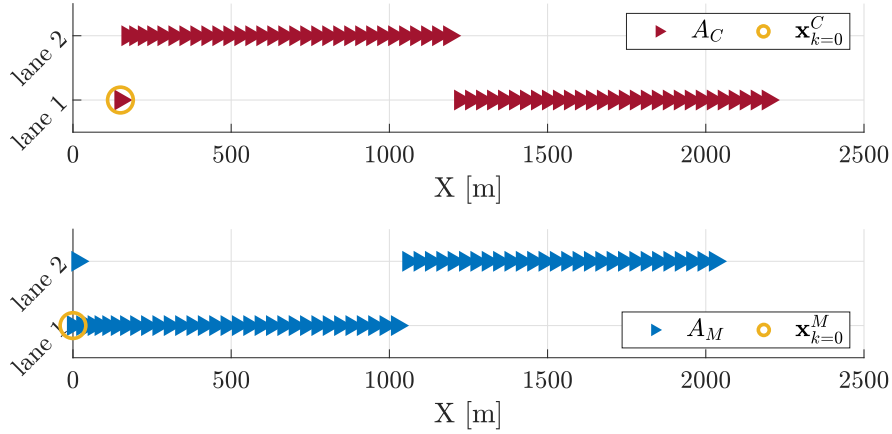


Figure 7.5: Lane trajectories of A_C (red) and A_M (blue), from $k = 0$ (yellow circles) to $k = 60$ s.

represents the discrete accelerations in $\mathcal{A}$ ($a_1 = -5 \text{ m/s}^2$ to $a_6 = 2.5 \text{ m/s}^2$), and the y -axis corresponds to lane moves ($\psi_1 = -1$, $\psi_2 = 0$, $\psi_3 = +1$).

At $k = 32$ s, the lane reference changes from lane 1 to lane 2, and A_M is still in lane 1. Its probability vector indicates a high likelihood of maintaining its current lane and velocity. To induce a lane change, A_C moves into lane 1 at $k = 33$ s, modifying the transition matrix Q of A_M and shifting the highest-probability action to $s = (a = 0 \text{ m/s}^2, \psi = +1)$. By $k = 34$ s, A_M completes the lane change to lane 2, while A_C remains in lane 1. The probability map confirms that the scenario stabilizes, with the next most likely action for A_M being $s = (a = 0 \text{ m/s}^2, \psi = 0)$.

7.4 Discussion

In this chapter, the behavioral motion model designed in Chapter 6.2 has been integrated into a framework for controlling an autonomous vehicle in a highway scenario, serving as a form of safety car intended to influence surrounding vehicles. The interactions between vehicles in this specific environment are modeled to predict how drivers' decisions may change in response to the actions of others.

The scenario-tree MPC has been demonstrated as a suitable control framework to handle these predictions, and the results of a small illustrative example have been

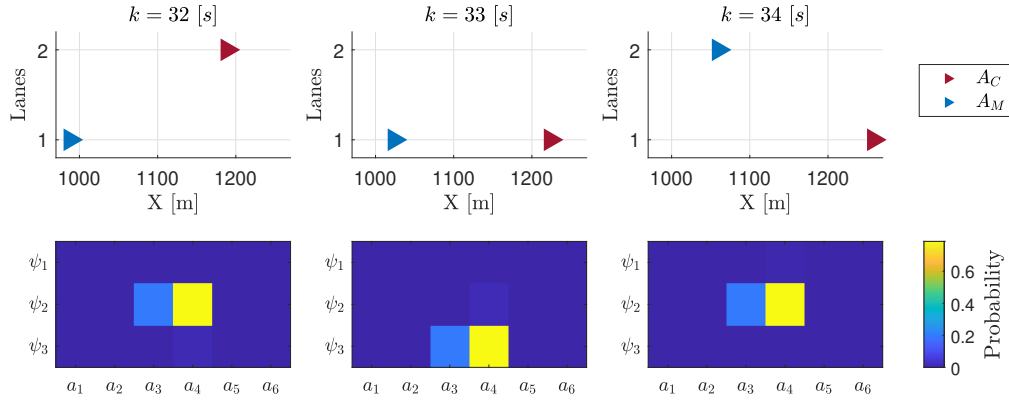


Figure 7.6: Snapshots from a single simulation during the lane reference change from $y^{ref} = 1$ to $y^{ref} = 2$. Top row: positions of A_C (red) and A_M (blue) at $k = \{32, 33, 34\}$ s. Bottom row: corresponding probability vectors of A_M , visualized via a color map.

presented.

It is important to note that this type of algorithm still requires validation with real-world datasets, as different drivers may have varying preferences for actions. Consequently, the parameters of the decision-making model should be tuned accordingly.

It is worth discussing potential applications that motivate this framework. A possible use case is highway traffic management through autonomous vehicles. Such vehicles could operate either as speed regulators or as moving barriers. For example, consider a two-lane highway with upcoming construction that narrows traffic from two lanes to one. An autonomous vehicle could proactively position itself in the second lane, discouraging drivers from occupying it and thereby preventing abrupt lane merging at the bottleneck. In this way, the merging process is anticipated and managed more smoothly. Moreover, with an appropriate coordination algorithm, a fleet of autonomous vehicles could be deployed to regulate and optimize traffic flow in such scenarios.

Social Robotics

Chapter 8

Interactions in Social Robotics: Motion Planners Design for Pedestrian Environments

Overview

The field of social robotics has gained increasing attention over the past decade, reflecting the growing recognition of its potential. Social robotics concerns those robots designed to interact with humans, whether as cooperative or antagonistic agents, or simply as dynamic obstacles in the environment. A classic example is the delivery robot, whose primary task is to transport packages, often within pedestrian environments. More recently, new applications have emerged, particularly in the domain of user assistance, such as support for elderly or disabled individuals, as well as robots designed to aid with household chores.

In the following three chapters, the focus is on the interactions between a general social robot and its surrounding environment, with particular emphasis on the design of efficient and safe motion planning strategies. These planners may also leverage information about human behavior to improve navigation performance and safety.

8.1 Introduction

The social robots have different applications as shown in [73]. Assistance to the elderly and handicapped people is the application that has received much attention from the scientific community [74].

In [75], an overview of the field of social robot navigation is presented. Specifically, the authors highlight the importance of three questions, that should be addressed when planning a motion planner for this type of applications

1. How should a robot plan its motion through a crowd of navigating humans to ensure safety and efficiency?
2. What types of social signs can be inferred from human behavior, and what types of behaviors should a robot exhibit to ensure social norm compliance?
3. How can a social robot navigation system be evaluated?

Starting from the first question, several well-known algorithms have been developed to ensure safety in crowded environments. However, most of these rely on action–reaction strategies, which do not incorporate predictions or account for the effects that the robot’s own motion may have on surrounding humans.

The key challenge, therefore, is how to integrate models of human motion into algorithms for robot decision-making. In the survey, the authors distinguish between two main approaches: decoupled and coupled prediction and planning.

Within decoupled prediction and planning approaches, humans are treated as dynamic obstacles, and it is assumed that the robot’s actions do not influence human decisions. This essentially corresponds to a classical dynamic obstacle avoidance problem, which can be addressed in two ways:

1. Humans are modeled as dynamic, non-responsive obstacles, and the solution is formulated purely in terms of collision avoidance. However, this typically results in a negative user experience.

2. Human motion is characterized by uncertainty. In this case, the main issue is that uncertainty can quickly grow over time, often leading to the so-called freezing robot problem.

In contrast, coupled prediction and planning approaches acknowledge that robot motion affects humans and vice versa. Here, two strategies are typically employed:

1. The explicit approach, which employs explicit predictions of human motion (e.g., in the form of trajectories, probability distributions, or occupancy grids). Game theory and Markov Decision Processes (MDPs) are frequently used in this context.
2. The implicit approach, which leverages techniques such as inverse reinforcement learning to recover models of human navigation objectives.

Both approaches present limitations and open challenges. While decoupled models neglect human aspects, which may lead to the freezing robot problem, since potential cooperation between agents is not considered, coupled approaches are often computationally intractable, as it is not feasible to account for all possible trajectories of each human in the environment. Moreover, in both paradigms, humans are typically modeled homogeneously, without accounting for differences due to age, objectives, or behavior. In the end, designing a motion planner that is robust to different scenarios remains difficult. Indeed, planners are usually tested in specific environments, without considering that individuals' goals and decisions may vary substantially across contexts.

The second question discussed in [75] highlights the importance of understanding the mechanisms behind human-robot interactions. These can be analyzed at three levels of description:

1. Individual level: the concept of individual space is introduced, where the robot should avoid stepping within this area to guarantee a pleasant experience for the human. Alternatively, social repulsive forces can be used to represent personal space: the closer the robot, the stronger the repulsion.
2. Individual and social signs level: the robot communicates its intentions to the humans, for example through explicit signals or behaviors.
3. Macro level: pedestrians are considered in groups, and the robot is aware of group dynamics. This awareness may help to avoid the freezing problem, as it enables the robot to intrude into groups with weak cohesion.

Despite the progress, several limitations and open problems remain. In general, there is still a limited understanding of pedestrian behavior. For example, modeling personal space is not straightforward, as it depends on multiple human features such as age, level of attention, or gender. Another important challenge concerns how robots should communicate their intentions to surrounding agents. While visual displays projecting the robot's planned trajectory are a possible solution, they may fail if pedestrians are not paying attention; conversely, auditory signals may capture attention but risk startling people.

A further open question relates to identifying and negotiating with groups of pedestrians. Beyond the technical challenge of recognizing groups, the robot must also assess their cohesion, in order to determine whether it is socially acceptable to move through them.

So far, research on pedestrian behavior understanding has largely followed a trial-and-error approach. Several models have been proposed and evaluated using datasets. Nevertheless, it is crucial to acknowledge that pedestrians exhibit a wide range of behaviors. Consequently, most algorithms either rely on an average pedestrian model, or require additional studies on human traits to capture inter-individual variability.

The last question concerns the methods for evaluating the proposed algorithms. Several approaches can be considered, ranging from tests on pre-recorded datasets, to simulations, up to real-world implementations. While the use of datasets and simulations in controlled

environments may appear less realistic than real deployments, they provide valuable flexibility: simulations, in particular, can be shaped as desired, allowing the testing of extreme or rare scenarios that would be difficult or unsafe to reproduce in practice.

This part of the dissertation on social robotics is organized into four main chapters. Chapter 8 reviews related work in the field of social robotics, with a focus on the design of motion planners for this application, and highlights the main contributions. Chapter 9 introduces the first motion planning approach, which is based on Game Theory. Chapter 10 presents an alternative approach grounded in Control Barrier Functions, focusing on a single-instant controller. Finally, Chapter 11 extends and improves this approach.

8.2 Related Works

Planning navigation in crowded environments is inherently challenging, as the presence of a large number of dynamic obstacles—namely pedestrians—makes the scenario highly unpredictable. Various frameworks can be leveraged for path planning, ranging from classic approaches to more modern methodologies.

Among the classic approaches, Linear Quadratic Regulator (LQR) and Model Predictive Control (MPC) are the most common. A key aspect of these methods is how they handle the optimization problem, particularly the formulation of the collision avoidance constraint, which, by definition, must be a hard constraint. For instance, in [24], the authors propose an iterative LQR for autonomous vehicle control, where the collision avoidance constraint is modeled as a barrier. Similarly, in [23], collision avoidance is also implemented as a barrier; however, it must be linear and quadratic to be incorporated into the objective function, which somewhat contradicts the strict definition of a hard constraint.

On the other hand, some methods address the collision-avoidance problem by assuming knowledge of the pedestrians' objectives. This can be achieved either by postulating the existence of an energy cost function that governs human reactions when approached by the autonomous agent, or by directly assuming knowledge of the pedestrians' objective functions that dictate their trajectories.

A well-known example is [25], where human decision-making is modeled as an energy minimization process, taking into account the final target, ideal velocity, and interactions with surrounding individuals. While this framework is compelling, it relies on prior knowledge that may not be available in unfamiliar environments with unknown agents.

Other methods, such as those proposed in [27] and [76], introduce alternative prediction strategies but remain constrained by their reliance on pre-recorded trajectories collected within the same environment. In [27], the authors also addressed the problem of classifying pedestrians based on their traits, which are then mapped to a set of social distances to be maintained from the obstacles. However, their framework is limited to the studied case and cannot be generalized to other scenarios. Another example is provided in [77], but once again the approach is based on a pre-recorded dataset, and may therefore fail when applied to different environments.

An alternative approach is based on Game Theory, which focuses on finding Nash equilibria (or ϵ -Nash equilibria) for cooperative agents. While the idea of seeking the best outcome for all participants is appealing, game-theoretical frameworks often rely on simplistic assumptions. These approaches are particularly effective when applied to multiple robots communicating with each other, as demonstrated in [28]. However, their effectiveness diminishes when the controlled agents must navigate an environment that includes humans [78], [79]. Once again, the key challenge lies in predicting human movements. In [29], a game-theoretical framework for robotic motion planning is proposed. While the model accounts for multiple possible human destinations, it relies on the strong assumption that the human objective function is known. Furthermore, the concept of Nash equilibria can sometimes yield misleading results, as it does not always guarantee that humans will act in a way that benefits everyone.

Lastly, there are examples of frameworks that employ a probabilistic version of the CBF. In [31] and [30], the probabilistic constraint is defined as the probability of remaining safe, and the control algorithm is designed to ensure this probability stays within an upper bound. While this approach makes the problem more tractable, thanks to the application of key probabilistic inequality theorems such as Chebyshev's and Cantelli's inequalities, it also implies that the probability of maintaining safety will always be constrained by a predefined upper bound, which typically serves as a tuning parameter. In [32], the authors refine the definition of the probabilistic constraint by replacing the upper bound constraint with a lower bound. However, their approach relies on certain approximations to reformulate the probabilistic constraint into a deterministic one.

8.3 Contributions

As discussed in the previous sections, the design of an efficient and safe motion planner that exploits information about the surrounding environment while minimizing assumptions remains a challenging task. To address this, the development process begins with a Game-Theory-based planner, which serves as an initial design stage. After analyzing the limitations of this framework, an alternative formulation based on Control Barrier Functions is introduced, offering a more robust approach to motion planning for social robots.

To summarize, the contributions of this chapter are:

- The formulation of a Game-Theory-based framework, evaluated in a scenario involving two interacting agents - Chapter 9;
- The design of a motion controller based on the probabilistic version of Control Barrier Functions - Chapter 10;
- The design of a two-layer hierarchical control framework based on Control Barrier Functions, validated in a simulated human-populated environment - Chapter 11.

Chapter 9

Game-Theory-based Motion Planner

Standard motion planners for social robots often treat humans as mere dynamic obstacles. In doing so, they neglect the action-reaction paradigm, failing to account for the fact that the actions of a single agent can influence the surrounding environment [80].

In recent years, several works have emerged aiming to model these intricate interactions between agents, with the ultimate goal of designing motion planners for the controlling agent. The framework commonly employed in these studies is rooted in Game Theory.

There are numerous applications of this approach, including validating human trajectories [78], [79], modeling interactions in traffic scenarios [81], controlling drones [28], [82], [83], [84], and managing robot motion in pedestrian environments [85].

Within the framework of Game Theory, an important concept is the Nash equilibrium. Roughly speaking, a Nash equilibrium represents the optimal "social" configuration, a state where all agents are, in some sense, achieving their best outcomes. If any agent were to unilaterally change their action, it would result in a less favorable scenario for them. It is important to note that the evaluation of win/lose situations is based on a concept known as the payoff. The payoff can be interpreted as a type of reward that an agent receives for performing a specific action, analogous to the concept of payoff in *Markov Decision Process* (MDP).

Given the definition of Nash equilibrium, it is possible to formulate a game between the controlled robot and the surrounding agents, with the goal of seeking a Nash equilibrium. This idea can be illustrated through a common scenario encountered in everyday life. Imagine two pedestrians starting from opposite ends of a hallway, each aiming to reach the opposite side. As they walk toward one another, a mechanism of collision avoidance is naturally considered. However, this mechanism is applied by both agents, and what is typically observed is that they both slightly deviate from their nominal paths to avoid a collision. This deviation from the nominal path can be viewed as physical evidence of a Nash equilibrium: both agents take actions that do not necessarily improve their individual rewards (for example, the new path might increase travel time or distance compared to the nominal path). However, these actions result in the best "social" scenario, where both agents share the "effort" of deviating from their optimal paths to achieve mutual avoidance.

Assume two agents A_1 and A_2 that can perform 3 choices (or take 3 different trajectories), as in Fig. 9.1. Since the goal is collision avoidance, assume that the payoff becomes infinite if the two agents collide (scenario (a_1, a_1)) or if they do not maintain the minimum distance, set as $2\Delta d$ ((a_1, a_2) and (a_2, a_1)). Furthermore, let the payoff for each agent be determined by the deviation from their nominal trajectory a_1 . Under these assumptions, Tab. 9.1 is constructed. In this game, there are three different Nash equilibria: (a_3, a_1) , (a_2, a_2) , and (a_1, a_3) . However, if one considers the payoffs

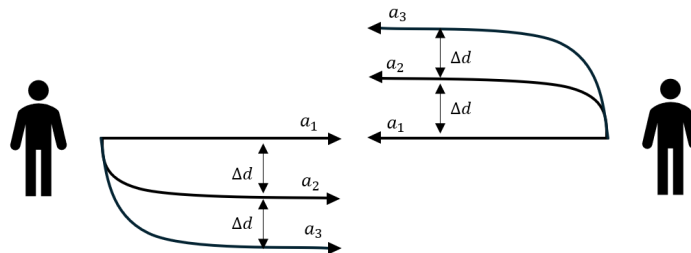


Figure 9.1: Interaction between two players.

obtained by the players and aims to enforce a notion of fairness in the game, (a_2, a_2) emerges as the desired equilibrium. In this scenario, both agents slightly deviate from their nominal paths, ensuring the satisfaction of the collision avoidance constraint while sharing the effort equally.

Table 9.1: Static game with 2 agents representing the game in Fig. 9.1.

	a_1	a_2	a_3
a_1	(∞, ∞)	(∞, ∞)	$(0, 2\Delta d)$
a_2	(∞, ∞)	$(\Delta d, \Delta d)$	$(\Delta d, 2\Delta d)$
a_3	$(2\Delta d, 0)$	$(2\Delta d, \Delta d)$	$(2\Delta d, 2\Delta d)$

This chapter is organized as follows. Section 9.1 defines the problem to be addressed, and Section 9.2 reviews the fundamental concepts required to understand and model the scenario. Simulations are presented in Section 9.3, and the conclusions and limitations of this approach are discussed in Section 9.4.

9.1 Problem definition

The problem of modeling a social game between a set of players, or agents, is here addressed. Let $\mathcal{N} = \{1, \dots, i, \dots, N\}$ denote the set of agent indices. For each agent $i \in \mathcal{N}$, let $\mathcal{X}^i \subseteq \mathbb{R}^{n_i}$ be its state space, where n_i is the state dimension of agent i . The overall state space is then defined as

$$\mathcal{X} = \mathcal{X}^1 \times \dots \times \mathcal{X}^N \subseteq \mathbb{R}^n,$$

where $n = \sum_{i \in \mathcal{N}} n_i$ is the total state dimension of the game.

In this dissertation, the game is assumed to be static. To introduce dynamics, assume that the game evolves over T time steps, with each step of size Δk . The state vector of the game at time step k , $0 \leq k \leq T$, is

$$\mathbf{x}_k = (\mathbf{x}_k^1, \dots, \mathbf{x}_k^N) \in \mathcal{X},$$

where $\mathbf{x}_k^i \in \mathcal{X}^i$ denotes the state vector of agent i at time step k .

For each player $i \in \mathcal{N}$, the action set is denoted by $\mathcal{U}^i \subseteq \mathbb{R}^{m_i}$, where m_i is the dimension of the action space of agent i . The joint action space of the game is

$$\mathcal{U} = \mathcal{U}^1 \times \dots \times \mathcal{U}^N \subseteq \mathbb{R}^m,$$

with $m = \sum_{i \in \mathcal{N}} m_i$ the overall control dimension. The action vector at time step k is

$$\mathbf{u}_k = (\mathbf{u}_k^1, \dots, \mathbf{u}_k^N) \in \mathcal{U},$$

where $\mathbf{u}_k^i \in \mathcal{U}^i$ is the action of agent i .

For each agent i , define

$$\mathcal{U}^{-i} = \prod_{j \neq i} \mathcal{U}^j,$$

which represents the set of actions of all agents except i . Similarly

$$\mathbf{u}_k^{-i} = (\mathbf{u}_k^1, \dots, \mathbf{u}_k^{i-1}, \mathbf{u}_k^{i+1}, \dots, \mathbf{u}_k^N) \in \mathcal{U}^{-i}.$$

The strategy space of agent $i \in \mathcal{N}$ is denoted by Γ^i . A strategy $\gamma^i \in \Gamma^i$ is defined as

$$\gamma^i : \mathcal{X} \times \{0, 1, \dots, T\} \rightarrow \mathcal{U}^i,$$

which maps the game state and time index to the action of agent i . Since a static game repeated over time is considered, and attention is restricted to open-loop strategies, it follows that

$$\gamma^i(\mathbf{x}_0, k) = \mathbf{u}_k^i.$$

The combined strategy profile of all agents is

$$\gamma = (\gamma^1, \dots, \gamma^N) \in \Gamma,$$

where $\Gamma = \Gamma^1 \times \dots \times \Gamma^N$. As with the action vector, it is possible to define

$$\gamma^{-i} = (\gamma^1, \dots, \gamma^{i-1}, \gamma^{i+1}, \dots, \gamma^N),$$

which collects the strategies of all agents except agent i .

Each agent $i \in \mathcal{N}$ is associated with a cost function

$$J^i : \mathcal{X} \times \Gamma \rightarrow \mathbb{R},$$

where $J^i(\cdot)$ represents the cost (or, equivalently, the negative payoff) incurred by agent i .

Within this game-theoretic framework, the objective of each agent is to select a strategy γ^i that minimizes its cost function. However, it is generally not feasible for all agents to simultaneously optimize their costs. Therefore, a common approach involves sequential optimization of agents' cost functions, as demonstrated in [85]. The resulting equilibrium concept is the Nash equilibrium, formally defined below.

Definition 9.1.1 (Nash equilibrium)

An open-loop Nash equilibrium of a game $\mathcal{G} = (\mathcal{N}, \{\Gamma^i\}_{i \in \mathcal{N}}, \{J^i\}_{i \in \mathcal{N}})$ is a feasible strategy profile $\gamma^* \in \Gamma$ such that, for each agent $i \in \mathcal{N}$,

$$J^i(x_0, \gamma^*) \leq J^i(x_0, \gamma^i, \gamma^{-i}), \quad \forall \gamma^i \in \Gamma^i. \quad (9.1)$$

To adapt this framework into a motion planner for an autonomous agent interacting with a pedestrian, a two-player game between the robot and the human is modeled, and a Nash equilibrium—corresponding to the social optimum—is sought. Although the human's actions cannot be directly controlled, it is assumed that the human is willing to reach the equilibrium. The objective is therefore to compute a sequence of control inputs for the robot that actively guides the interaction toward, and remains consistent with, the Nash equilibrium.

9.2 Potential Games

The existence of a Nash equilibrium was established by Nash in [86]. However, this result applies exclusively to mixed-strategy games, in which players select their actions randomly according to a probability distribution over the set of pure strategies, rather than committing to a single deterministic action. In this case, the objective is to design a deterministic control input for the autonomous agent, which acts as a player in the game. Consequently, it is possible to restrict the analysis to pure-strategy games, where players are required to make a deterministic choice from their respective sets of possible actions.

Nevertheless, there exists a category of games for which the existence of the Nash equilibrium has been mathematically proved in pure-strategy [87], namely potential games.

Definition 9.2.1 (Potential Game)

A potential game is a noncooperative game for which there exists a function $\phi : \Gamma \rightarrow \mathbb{R}$ satisfying

$$p^i(\gamma^i, \gamma^{-i}) - p^i(\bar{\gamma}^i, \gamma^{-i}) = \phi(\gamma^i, \gamma^{-i}) - \phi(\bar{\gamma}^i, \gamma^{-i}) \quad (9.2)$$

for every $i \in \mathcal{N}$, $\gamma^i, \bar{\gamma}^i \in \Gamma^i$, $\gamma^{-i} \in \Gamma^{-i}$. The function $p^i(\cdot)$ is the utility function, and the function ϕ is referred to as a potential function of the game.

This definition ensures that the change in the utility of a player who unilaterally deviates to a new strategy, coincides exactly with the corresponding change in the potential function.

An important property of potential games is that the total unilateral utility improvement around a closed path is equal to zero.

Definition 9.2.2 (Path - Closed Path - Improvement Path)

A path ϑ is a collection of strategy profiles $\vartheta = (\gamma_0, \dots, \gamma_M)$, such that γ_ℓ and $\gamma_{\ell+1}$ differ in the strategy of exactly one player. A path is closed if $\gamma_0 = \gamma_M$. A path is simple if no strategy profile other than the first and the last strategy profiles is repeated along the path.

Moreover, the total improvement along the path is defined as

$$I(\vartheta) = \sum_{\ell=1}^M p^{a_\ell}(\gamma_\ell) - p^{a_\ell}(\gamma_{\ell-1}).$$

where a_ℓ indicates the agent that changes strategy from γ_ℓ to $\gamma_{\ell-1}$.

Proposition 9.2.1

A game is a potential game IFF the total utility improvement $I(\gamma) = 0$ for all the simple closed paths γ . $\square$

For the proof of the proposition above, please refer to [87].

Theorem 9.2.1 provides a necessary and sufficient condition for a game to qualify as a potential game. Consequently, satisfying this condition also guarantees the existence of an equilibrium, thereby establishing the feasibility of the proposed problem formulation.

9.3 Game between the autonomous agent and the human

An example of a potential game for modeling interactions with autonomous agents can be found in [28], where the authors formulated a game between drones that must avoid moving obstacles. They proved that, under their formulation, the game is indeed a potential game, thereby guaranteeing the existence of a Nash equilibrium. However, their framework relies on a strong assumption: coupling constraints are shared among agents, and each agent has full knowledge of the individual constraints of the others.

Such an assumption is overly restrictive in realistic scenarios. For instance, when considering pairwise distances, it is unrealistic to assume that the robot's preferred distance from a human coincides with the human's preferred distance from the robot.

Building upon this work, a game-theoretic interaction between an autonomous agent and a human is proposed. While the human's objective function is assumed to be known, the formulation relaxes the previous assumption by allowing each player to define its own safety distance, which may differ between the two agents.

The game as formulated should be demonstrated to be a constrained dynamic potential games, with the constraint to be the collision avoidance constraints.

In [28], the authors stated the definition of constrained dynamic potential games

Definition 9.3.1 (Constrained Potential Game)

A noncooperative constrained dynamic game $\mathcal{G}$ is a constrained dynamic potential game if there exist function $\phi : \Gamma \rightarrow \mathbb{R}$ such that for every agent $i \in \mathcal{N}$, any pair of strategies γ^i and ν^i , and any strategy vector $\gamma^{-i} \in \Gamma^{-i}$, the following is verified for every any initial condition $\mathbf{x}_0$:

$$J^i(\mathbf{x}_0, \gamma^i, \gamma^{-i}) - J^i(\mathbf{x}_0, \nu^i, \gamma^{-i}) = \phi(\mathbf{x}_0, \gamma^i, \gamma^{-i}) - \phi(\mathbf{x}_0, \nu^i, \gamma^{-i}), \quad (9.3)$$

where $\phi(\mathbf{x}_0, \gamma)$ is the potential function, and $J^i(\cdot) = S^i(\mathbf{x}_T, T) + \sum_{k=1}^{T-1} L^i(\mathbf{x}_k^i, \mathbf{u}_k^i, \mathbf{u}_k^{-i}, k)$ is the cost for the agent i , $(\mathbf{x}_k, \mathbf{u}_k) \in \mathcal{C}_k$ for all $k \in \{0, 1, \dots, T-1\}$ and $\mathbf{x}_T \in \mathcal{C}_T$.

Proposition 9.3.1

A game $\mathcal{G}$ is a constrained dynamic potential game if the running costs and terminal costs function of every agent $i \in \mathcal{N}$ can be expressed as the sum of a term that is common to all players plus another term that depends neither on its own action nor on its own state-components:

$$L^i(\mathbf{x}_k^i, \mathbf{u}_k^i, \mathbf{u}_k^{-i}, k) = P(\mathbf{x}_k, \mathbf{u}_k^i, \mathbf{u}_k^{-i}, k) + \Theta^i(\mathbf{x}_k^{-i}, \mathbf{u}_k^{-i}, k) \quad (9.4a)$$

$$S^i(\mathbf{x}_T, T) = R(\mathbf{x}_T, T) + \Phi^i(\mathbf{x}_T^{-i}, T) \quad (9.4b)$$

□

The Definition 9.3.1, together with the Proposition 9.3.1, implies that the change in the running and terminal costs of each agent can be captured by the two functions R and P which are common among all agents. In other words, there exist two functions that can capture the rate of change in each agent's cost when the strategies of all the others are fixed. Thus, R and P together form the potential function. Nevertheless, the constrained aspect is given by the sets $\mathcal{C}_k$ and $\mathcal{C}_T$.

Now, assume a game between two players, one autonomous agent and one human, and consider the case where the utility functions of the two agents are independent of the other agent's state and actions, i.e.,

$$J_1(\mathcal{X}, \mathcal{U}) = J_1(\mathcal{X}^1, \mathcal{U}^1) \quad \text{and} \quad J_2(\mathcal{X}, \mathcal{U}) = J_2(\mathcal{X}^2, \mathcal{U}^2), \quad (9.5)$$

while both players must still satisfy collision avoidance constraints and respect a minimum distance from the antagonist agent.

In [28], the minimum distance is assumed to be identical for all agents. In this setup, named *Game G1*, the constraints are

$$\|x_1 - x_2\|_2 \geq d_{min}^{(i)}, \quad i = \{1, 2\}, \quad \text{with } d_{min}^{(1)} = d_{min}^{(2)}.$$

Starting from here, it is possible to relax this assumption and allow for different safety distances, generating *Game G2*. Without loss of generality, assume $d_{min}^{(1)} < d_{min}^{(2)}$.

In such a setting, the effective distance between the agents should be

$$\|x_1 - x_2\|_2 \geq \max(d_{min}^{(1)}, d_{min}^{(2)}).$$

Intuitively, the agent requiring the larger safety distance ($d_{min}^{(2)}$) is expected to exert more effort and deviate further from its nominal trajectory.

To model this asymmetry, some scaling parameters are introduced in the cost functions:

$$w_1 = \frac{d_{min}^{(2)}}{d_{min}^{(1)} + d_{min}^{(2)}} \quad \text{and} \quad w_2 = \frac{d_{min}^{(1)}}{d_{min}^{(1)} + d_{min}^{(2)}}. \quad (9.6)$$

The weighted cost matrices for each agent then become

$$\tilde{Q}_1 = w_1 Q_1, \quad \tilde{R}_1 = w_1 R_1, \quad \tilde{Q}_T^1 = w_1 Q_T^1, \quad (9.7a)$$

$$\tilde{Q}_2 = w_2 Q_2, \quad \tilde{R}_2 = w_2 R_2, \quad \tilde{Q}_T^2 = w_2 Q_T^2. \quad (9.7b)$$

This formulation enforces asymmetric responsibility in avoiding collisions, depending on the individual safety distance requirements. By introducing this modification, the cost functions remain independent of the actions of the other agent. The results obtained are shown in Fig. 9.2. These results were generated using the *IPOPT* algorithm [88], incorporating both constraints into the optimization problem.

To better highlight the differences between game G_1 and G_2 , the deviations from the nominal trajectories of the two agents in both games are displayed.

From Fig. 9.3, it is possible to observe that the deviation from the nominal trajectory for agent A_1 remains unchanged in both G_1 and G_2 , as $d_{min}^{(1)}$ is constant across the

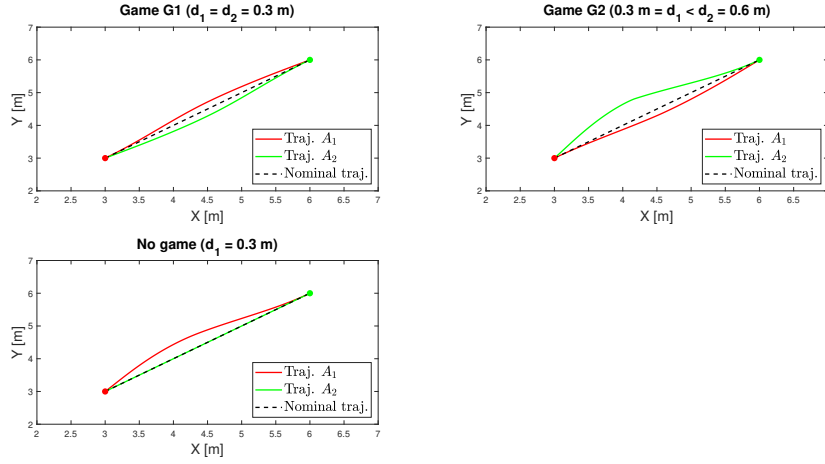


Figure 9.2: Comparison between Game G1 ($d_{min}^{(1)} = d_{min}^{(2)}$), Game G2 ($d_{min}^{(1)} < d_{min}^{(2)}$) and no game.

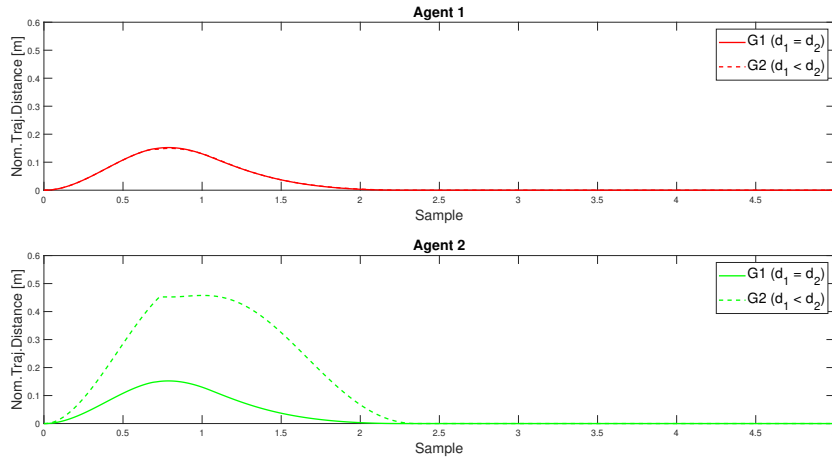


Figure 9.3: Distance to the nominal trajectory.

two games. Conversely, $d_{min}^{(2)}$ changes between G_1 and G_2 (from 0.3 m to 0.6 m). In G_2 , where the constraints differ between the agents, A_2 —the agent requiring a larger distance—exhibits a greater deviation from its nominal trajectory compared to G_1 . Meanwhile, the trajectory of A_1 remains essentially unchanged, as A_1 has no incentive to further deviate.

According to Proposition 9.3.1, scaling the matrices with the respective weights w_1 and w_2 does not alter the fact that the game is a potential game. Indeed, it is still possible to rewrite each agent's cost function as a sum of (i) a term that is common to all agents and (ii) a term that does not depend on the agent's own state or action, namely

$$\begin{aligned}
 J_i(\mathcal{X}, \mathcal{U}) &= (\mathbf{x}_T^i - \mathbf{x}_g^i)^T \tilde{Q}_{T_i} (\mathbf{x}_T^i - \mathbf{x}_g^i) + \sum_{k=0}^{T-1} \left[(\mathbf{x}_k^i - \mathbf{x}_g^i)^T \tilde{Q}_i (\mathbf{x}_k^i - \mathbf{x}_g^i) + (\mathbf{u}_k^i)^T \tilde{R}_i \mathbf{u}_k^i \right] \\
 &= R(\mathbf{x}_T) + \Phi^i(\mathbf{x}_T^{-i}) + \sum_{k=0}^{T-1} \left[P(\mathbf{x}_k, \mathbf{u}_k) + \Theta^i(\mathbf{x}_k^{-i}, \mathbf{u}_k^{-i}) \right],
 \end{aligned} \tag{9.8}$$

where

- $\mathcal{X}, \mathcal{U}$ denote the sequence of states and actions of all agents, respectively;

- $\mathbf{x}_g^i$ is the target position of agent $i \in \mathcal{N}$;
- $\mathcal{X}^{-i}, \mathcal{U}^{-i}$ denote the sequence of states and actions of all agents except i ;
- $P(\mathbf{x}_k, \mathbf{u}_k) = \sum_{i \in \mathcal{N}} [(\mathbf{x}_k^i - \mathbf{x}_g^i)^T \tilde{Q}_i (\mathbf{x}_k^i - \mathbf{x}_g^i) + (\mathbf{u}_k^i)^T \tilde{R}_i \mathbf{u}_k^i]$;
- $\Theta^i(\mathbf{x}_k^{-i}, \mathbf{u}_k^{-i}) = - \sum_{j \in \mathcal{N}, j \neq i} [(\mathbf{x}_k^j - \mathbf{x}_g^j)^T \tilde{Q}_j (\mathbf{x}_k^j - \mathbf{x}_g^j) + (\mathbf{u}_k^j)^T \tilde{R}_j \mathbf{u}_k^j]$;
- $R(\mathbf{x}_T) = \sum_{i \in \mathcal{N}} (\mathbf{x}_T^i - \mathbf{x}_g^i)^T \tilde{Q}_{T_i} (\mathbf{x}_T^i - \mathbf{x}_g^i)$;
- $\Phi^i(\mathbf{x}_T^{-i}) = - \sum_{j \in \mathcal{N}, j \neq i} (\mathbf{x}_T^j - \mathbf{x}_g^j)^T \tilde{Q}_{T_j} (\mathbf{x}_T^j - \mathbf{x}_g^j)$.

According to Theorem 2 in [28], if the cost functions admit such a decomposition, then the solution of the following optimization problem

$$\text{minimize}_{\mathcal{U}} \quad R(\mathbf{x}_T) + \sum_{k=0}^{T-1} P(\mathbf{x}_k, \mathbf{u}_k), \quad (9.9a)$$

$$\text{subject to} \quad \mathbf{x}_{k+1} = f(\mathbf{x}_k, \mathbf{u}_k), \quad \mathbf{x}_0 \text{ given}, \quad (9.9b)$$

$$g(\mathbf{x}_k, \mathbf{u}_k) \leq 0, \quad (9.9c)$$

is a generalized Nash equilibrium of the game. The proof follows the same reasoning as in [28].

Thus, even with heterogeneous constraints and weighted cost functions, the game remains a potential game, and the existence of a Nash equilibrium is guaranteed.

9.4 Discussion

While offering an interesting perspective, the game-theoretical approach presents several limitations. First, the assumption that the objective function of the antagonist player is known is quite strong. In practice, this implies knowledge of the pedestrian's final goal. If this assumption is removed, it is no longer possible to guarantee the existence of a Nash equilibrium. Instead, one may resort to approximate equilibria, such as the ϵ -Nash equilibrium, where the robot determines its control actions under the assumption of an unknown objective function for the other player. A first attempt in this direction was proposed in [89], where the autonomous agent refines its strategy based on iterative updates of its beliefs about the other player.

Another key limitation lies in the concept of Nash equilibrium itself. Although the Nash equilibrium can be interpreted as a social optimum—i.e., a configuration in which no agent can unilaterally reduce its loss—it may still require an agent to deviate from its nominal path or take a longer route. Moreover, this assumption relies on all participants actively engaging in the game, which may not hold in realistic scenarios. For instance, the robot may encounter selfish pedestrians, distracted individuals, or people unwilling to cooperate, thus breaking the equilibrium assumptions.

For these reasons, the game-theoretical approach appears more suited to coordinating fleets of robots than to modeling interactions with humans as obstacles. Consequently, in the next section, an alternative approach that requires fewer assumptions and explicitly accounts for human obstacles will be presented.

Chapter 10

Control Barrier Function for Collision Avoidance in Uncertain Scenarios

As discussed in the previous section, the game-theoretical framework suffers from several limitations, which motivated us to discard it in favor of a new control strategy. In this section, a framework grounded in *Control Barrier Functions* (CBFs) is presented.

First, a formulation where the robot's motion is governed by a quadratic optimization problem with a single-step horizon is introduced. In this case, collision avoidance is enforced through Chance-Constrained CBFs, designed in a manner that differs from conventional approaches in the literature.

Building on this, a more efficient formulation based on a *Model Predictive Control* (MPC) scheme is proposed. Here, the controller optimizes over a finite prediction horizon, and the collision avoidance constraints are derived from CBFs as well, but using a softened formulation, which will be detailed in the following.

Finally, the latter controller is embedded within a broader hierarchical structure, specifically a two-layer cascade framework. This integrated system is tested on a well-known dataset to assess its performance.

In this section, discrete-time CBFs is employed, whose mathematical preliminaries are reported in Appendix E.

The aim is to extend the concept of CBFs to uncertain scenarios, where only partial—or even no—knowledge of the surrounding pedestrian is available. In this setting, multiple possible future trajectories of the pedestrian are considered, each associated with a probability measure.

This chapter is organized as follows. Section 10.1 describes the application scenarios for the framework, and Section 10.2 presents the dynamics of all involved agents. Section 10.3 introduces the probabilistic version of the CBF, and its application is illustrated in Section 10.4. Finally, the conclusions and limitations are discussed in Section 10.5.

10.1 Scenario of application

In this small section, the scenario of application is presented, in a simplistic manner, involving just a single robot interacting with a single pedestrian.

Assume, for simplicity, that there are two possible trajectories for the moving pedestrian, A and B , with associated probabilities p_A and p_B , respectively, such that $p_A + p_B = 1$ and $p_A, p_B \geq 0$. At time k , the pedestrian's current position is recorded as $H_k = (\mathbf{x}_H, v_H, \theta_H)_k$, where θ_H represents the direction of motion. Given this scenario, the pedestrian can follow either trajectory A or B . Assume to have a prediction tool that provides the pedestrian's likely future positions as $\hat{H}_{t=k+1:k+N_p}^{(A)} = (\hat{\mathbf{x}}_H, \hat{v}_H, \hat{\theta}_H)_t$ for trajectory A and $\hat{H}_{t=k+1:k+N_p}^{(B)} = (\hat{\mathbf{x}}_H, \hat{v}_H, \hat{\theta}_H)_t$ for trajectory B . An illustrative sketch of this concept is presented in Fig. 10.1.

Starting from this configuration, different algorithms may generate different trajectories. For instance, a highly conservative algorithm would steer the robot to avoid both potential pedestrian paths, keeping a safe distance from points A and B . In contrast, a probabilistic algorithm might guide the robot to focus only on the most likely trajectory or to perform a risk assessment, selecting the optimal path while taking both A and B into account.

Importantly, the choice of algorithm should also take into account the intended application of the motion planner. For delivery robots, it is reasonable to require the robot to take a longer path to maximize safety. In contrast, for assistive robotics—where the robot may be transporting a human or guiding a person—always demanding that the human sacrifices travel distance or time is undesirable and potentially unfair.

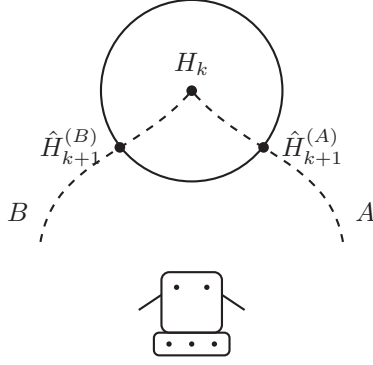


Figure 10.1: Scenario of application. H_k denotes the current position of the pedestrian at time step k . From this state, the pedestrian may follow either trajectory A or trajectory B . The circle around H_k represents the minimum safety distance that should ideally be maintained from the pedestrian.

The framework proposed in the following is primarily designed for assistive robotics. In this context, the robot is not treated as a simple object but as an extension of its user, meaning that effort cannot rest solely on the robot. Nevertheless, the framework is flexible: by appropriately tuning parameters and formulating the objective function, it can be adapted to simpler scenarios, such as hard collision avoidance, which is more suitable for applications like delivery robots.

10.2 Robot and Pedestrian Dynamics

Before detailing the control algorithm itself, this section describes the models of both the robot and an average pedestrian, which will be referenced in subsequent sections.

In this dissertation, unless otherwise specified, the robot is modeled as a single-mass point following a double integrator:

$$\begin{bmatrix} x_{k+1} \\ y_{k+1} \\ v_{k+1}^x \\ v_{k+1}^y \end{bmatrix} = \begin{bmatrix} 1 & 0 & \Delta k & 0 \\ 0 & 1 & 0 & \Delta k \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_k \\ y_k \\ v_k^x \\ v_k^y \end{bmatrix} + \begin{bmatrix} \frac{\Delta k^2}{2} & 0 \\ 0 & \frac{\Delta k^2}{2} \\ \Delta k & 0 \\ 0 & \Delta k \end{bmatrix} \begin{bmatrix} a_k^x \\ a_k^y \end{bmatrix}, \quad (10.1)$$

where $\mathbf{x}_k = [x_k, y_k, v_k^x, v_k^y]^T \in \mathcal{X} \subseteq \mathbb{R}^4$ denotes the system state, $\mathbf{u}_k = [a_k^x, a_k^y]^T \in \mathcal{U}_R \subseteq \mathbb{R}^2$ is the control input, and Δk is the sampling time, with $\mathcal{X}$ and $\mathcal{U}_R$ the set of feasible states and inputs of the robot.

The average pedestrian is instead modeled using a unicycle model. In the following, the pedestrian may also be referred to as an obstacle, since the robot treats it as a moving obstacle in the scenario under consideration. The pedestrian dynamics are given by

$$\begin{bmatrix} o_{k+1}^x \\ o_{k+1}^y \end{bmatrix} = \begin{bmatrix} o_k^x \\ o_k^y \end{bmatrix} + v_k \begin{bmatrix} \cos \theta_k \\ \sin \theta_k \end{bmatrix} \Delta k, \quad (10.2)$$

where $\mathbf{o}_k = [o_k^x, o_k^y]^T \in \mathcal{O} \subseteq \mathbb{R}^2$ represents the pedestrian state, and $v_k \in \mathcal{V} \subseteq \mathbb{R}^+$, $\theta_k \in \mathcal{T} \subseteq \mathbb{R}$ are the inputs of the dynamics, with $\mathcal{O}$, $\mathcal{V}$ and $\mathcal{T}$ being the set of feasible states, speed and motion direction for the obstacle.

The choice of the unicycle model is motivated by the common observation that, in pedestrian environments, people typically walk at an approximately constant speed, and changes in speed are negligible in terms of acceleration. In other words, acceleration variations are often too small to be reliably detected. Consequently, it is common in the literature to use a unicycle model, focusing primarily on variations in speed and motion direction.

10.3 Chance-Constrained CBF

As introduced at the beginning of this section, the first control framework addresses the problem of motion planning and control for an autonomous agent—specifically, a robot governed by dynamics (10.1)—operating in a pedestrian environment. The goal is for the robot to reach a desired target while ensuring collision avoidance with moving obstacles, i.e., pedestrians following dynamics (10.2).

To achieve this, and considering applications in assistive robotics, the approach leverages stochastic predictions of pedestrian motion, which are then incorporated into the controller.

Before detailing the constraint itself, it is useful to briefly introduce the *Control Barrier Function* (CBF) constraint, which enhances collision avoidance. In a straightforward manner, the barrier function $h : \mathcal{X} \times \mathcal{O} \rightarrow \mathbb{R}$ is defined based on the Euclidean distance between the robot and an obstacle:

$$h(\mathbf{x}_k, \mathbf{o}_k) = \|\mathbf{x}_k^{[x,y]} - \mathbf{o}_k\|_2^2 - d_{min}^2, \quad (10.3)$$

where $d_{min} \geq 0$ denotes the minimum safe distance.

Notably, if the robot's state dimension exceeds that of the obstacle, $\mathbf{x}_k$ in (10.3) refers only to the vector containing the x and y coordinates of the robot. For clarity, the following notation is adopted: $\mathbf{x}_k$ represents the full state of the controlled agent, while $\mathbf{x}_k^{[x,y]}$ refers solely to its spatial coordinates.

Recall that a system is considered *safe* when $h(\mathbf{x}_k^{[x,y]}, \mathbf{o}_k) \geq 0$. With this formulation, safety is guaranteed whenever the Euclidean distance between the robot and the obstacle is greater than the minimum threshold, thereby ensuring that no collision occurs.

According to CBF theory, the collision avoidance constraint can be expressed as a *Control Barrier Constraint* (CBC) in the following form:

$$CBC_{k+1|k} = h(\mathbf{x}_{k+1}^{[x,y]}, \mathbf{o}_{k+1}) - (1 - \gamma)h(\mathbf{x}_k^{[x,y]}, \mathbf{o}_k) \geq 0, \quad (10.4)$$

where $\gamma \in (0, 1]$. This can be reformulated as

$$CBC_{k+1|k} = \Delta h(\mathbf{x}_k^{[x,y]}, \mathbf{o}_k) + \gamma h(\mathbf{x}_k^{[x,y]}, \mathbf{o}_k) \geq 0. \quad (10.5)$$

Here, the fundamental assumption is that at time k , $h(\mathbf{x}_k^{[x,y]}, \mathbf{o}_k) \geq 0$.

Rewriting the above

$$h(\mathbf{x}_{k+1}^{[x,y]}, \mathbf{o}_{k+1}) - h(\mathbf{x}_k^{[x,y]}, \mathbf{o}_k) \geq -\gamma h(\mathbf{x}_k^{[x,y]}, \mathbf{o}_k).$$

Analyzing this inequality, two cases can be distinguished:

- **Case 1:** $h(\mathbf{x}_k^{[x,y]}, \mathbf{o}_k) = 0$. At time k , the agent is on the boundary of the safe set. The inequality implies that $h(\mathbf{x}_{k+1}^{[x,y]}, \mathbf{o}_{k+1}) \geq 0$, meaning the agent can either remain on the boundary or move further inside the safe set (in this context, maintaining the minimum distance or increasing it).
- **Case 2:** $h(\mathbf{x}_k^{[x,y]}, \mathbf{o}_k) > 0$. At time k , the agent lies strictly inside the safe set. The condition then becomes

$$h(\mathbf{x}_{k+1}^{[x,y]}, \mathbf{o}_{k+1}) \geq (1 - \gamma)h(\mathbf{x}_k^{[x,y]}, \mathbf{o}_k),$$

where the right-hand side lies in $[0, h(\mathbf{x}_k^{[x,y]}, \mathbf{o}_k))$. Thus, any control input that satisfies the CBF constraint keeps the system safe. The parameter γ directly influences the degree of conservativeness:

- If $\gamma \rightarrow 0$, then $\text{RHS} \rightarrow h(\mathbf{x}_k^{[x,y]}, \mathbf{o}_k)$. The resulting control action is conservative, as it requires the agent to increase its distance from the obstacle relative to time k .

- If $\gamma \rightarrow 1$, then $\text{RHS} \rightarrow 0$. The control action becomes less conservative, essentially enforcing only the standard collision avoidance constraint without reference to the previous position.

Having clarified the role of the parameter γ , now the probabilistic formulation of the constraint is introduced, called *Chance-Constrained Control Barrier Function* (CC-CBF). In this case, the CBC is required to hold with a certain probability, leading to the following condition:

$$\mathbb{P}(CBC_{k+1|k} \leq 0) \leq \rho, \quad (10.6)$$

where $CBC_{k+1|k}$ is defined as in (10.4).

This expression represents the probability of violating the collision avoidance constraint. Intuitively, the objective is to keep this probability as small as possible by tuning the parameter ρ . However, as will be discussed later, the fact that ρ is a tuning parameter may lead to potentially infeasible scenarios.

For the moment, let us assume that the tuning of ρ is not the main concern. In [32], the authors proposed a version of CC-CBF that is applicable only when the system noise follows a Gaussian distribution. In contrast, [30] introduced a slightly different formulation of the constraint (10.6), allowing the application of well-known probabilistic results, such as Chebyshev's inequality.

Building on this State-of-the-Art, the first question to be addressed is whether the probabilistic constraint (10.6) can be reformulated in a way that explicitly exploits knowledge of the underlying probability distribution.

Recalling the pedestrian dynamics in (10.2), the pedestrian's next state depends on the pair (v, θ) , where v is the speed and θ the motion direction. Assuming these two variables are independent, the CC-CBF can be factorized as

$$\mathbb{P}(CBC_{k+1|k} \leq 0) = \sum_{i=1}^{|\Theta|} \mathbb{P}(\theta_k = \theta_i) \mathbb{P}(CBC_{k+1|k}(\theta_i) \leq 0), \quad (10.7)$$

where θ is assumed to have finite support Θ .

By substituting θ_i into the CBC, one obtains an expression that depends only on the speed:

$$CBC(\theta_i) = \left\| f(\mathbf{x}_k, \mathbf{u}_k)^{[x,y]} - \mathbf{o}_k - v_k \begin{bmatrix} \cos \theta_i \\ \sin \theta_i \end{bmatrix} \Delta k \right\|_2^2 - d_{min}^2 - (1 - \gamma)h(\mathbf{x}_k^{[x,y]}, \mathbf{o}_k), \quad (10.8)$$

where the function $f(\cdot)$ is the dynamics of controlled agent (10.1). Notably, here, the only stochastic element is the velocity. By indicating

$$\begin{aligned} \mathbf{a}(\mathbf{u}_k) &= f(\mathbf{x}_k, \mathbf{u}_k)^{[x,y]} - \mathbf{o}_k \\ \mathbf{b} &= \begin{bmatrix} \cos \theta_i \\ \sin \theta_i \end{bmatrix} \Delta k \\ c_k &= d_{min}^2 + (1 - \gamma)h(\mathbf{x}_k^{[x,y]}, \mathbf{o}_k) \end{aligned} \quad (10.9)$$

where $c_k > 0$ if the initial conditions are safe, namely $h(\mathbf{x}_k, \mathbf{o}_k) \geq 0$, one obtains a more general form

$$CBC(\theta_i) = (\mathbf{a}(\mathbf{u}_k) - v_k \mathbf{b})^T (\mathbf{a}(\mathbf{u}_k) - v_k \mathbf{b}) - c_k. \quad (10.10)$$

From here on for simplicity, consider

$$\mathbf{a}(\mathbf{u}_k) = \mathbf{a} \quad (10.11)$$

and thus

$$\begin{aligned}
 \mathbb{P}(CBC(\theta_i) \leq 0) &= \mathbb{P}((\mathbf{a} - v_k \mathbf{b})^T (\mathbf{a} - v_k \mathbf{b}) - c_k \leq 0) = \\
 &= \mathbb{P}(\mathbf{a}^T \mathbf{a} + v_k^2 \mathbf{b}^T \mathbf{b} - 2v_k \mathbf{b}^T \mathbf{a} - c_k \leq 0) = \\
 &= \mathbb{P}(v_k^2 \Delta k^2 - 2v_k \mathbf{b}^T \mathbf{a} + \mathbf{a}^T \mathbf{a} - c_k \leq 0) = \\
 &= \mathbb{P}(v_k^2 \Delta k^2 - 2v_k \Delta k \frac{\mathbf{b}^T \mathbf{a}}{\Delta k} \leq c_k - \mathbf{a}^T \mathbf{a}) = \\
 &= \mathbb{P}(v_k^2 \Delta k^2 - 2v_k \Delta k \frac{\mathbf{b}^T \mathbf{a}}{\Delta k} + \left(\frac{\mathbf{b}^T \mathbf{a}}{\Delta k}\right)^2 - \left(\frac{\mathbf{b}^T \mathbf{a}}{\Delta k}\right)^2 \leq c_k - \mathbf{a}^T \mathbf{a}) = \\
 &= \mathbb{P}\left(\left(v_k \Delta k - \frac{\mathbf{b}^T \mathbf{a}}{\Delta k}\right)^2 \leq c_k + \left(\frac{\mathbf{b}^T \mathbf{a}}{\Delta k}\right)^2 - \mathbf{a}^T \mathbf{a}\right)
 \end{aligned} \tag{10.12}$$

and finally

$$\begin{aligned}
 \mathbb{P}(CBC(\theta_i) \leq 0) &= \mathbb{P}\left(-\sqrt{c_k + \left(\frac{\mathbf{b}^T \mathbf{a}}{\Delta k}\right)^2} - \mathbf{a}^T \mathbf{a} \leq v_k \Delta k - \frac{\mathbf{b}^T \mathbf{a}}{\Delta k} \leq \sqrt{c_k + \left(\frac{\mathbf{b}^T \mathbf{a}}{\Delta k}\right)^2} - \mathbf{a}^T \mathbf{a}\right) = \\
 &= \mathbb{P}\left(\frac{\frac{\mathbf{b}^T \mathbf{a}}{\Delta k} - \sqrt{c_k + \left(\frac{\mathbf{b}^T \mathbf{a}}{\Delta k}\right)^2} - \mathbf{a}^T \mathbf{a}}{\Delta k} \leq v_k \leq \frac{\frac{\mathbf{b}^T \mathbf{a}}{\Delta k} + \sqrt{c_k + \left(\frac{\mathbf{b}^T \mathbf{a}}{\Delta k}\right)^2} - \mathbf{a}^T \mathbf{a}}{\Delta k}\right) = \\
 &= \mathbb{P}\left(\frac{\mathbf{b}^T \mathbf{a} - \sqrt{c_k \Delta k^2 + (\mathbf{a}^T \mathbf{b})(\mathbf{b}^T \mathbf{a})} - \Delta k^2 \mathbf{a}^T \mathbf{a}}{\Delta k^2} \leq v_k \wedge \right. \\
 &\quad \left. v_k \leq \frac{\mathbf{b}^T \mathbf{a} + \sqrt{c_k \Delta k^2 + (\mathbf{a}^T \mathbf{b})(\mathbf{b}^T \mathbf{a})} - \Delta k^2 \mathbf{a}^T \mathbf{a}}{\Delta k^2}\right) = \\
 &= \mathbb{P}\left(\underbrace{\frac{\mathbf{b}^T \mathbf{a} - \sqrt{\mathbf{a}^T M \mathbf{a} + c_k \Delta k^2}}{\Delta k^2}}_{v_1} \leq v_k \leq \underbrace{\frac{\mathbf{b}^T \mathbf{a} + \sqrt{\mathbf{a}^T M \mathbf{a} + c_k \Delta k^2}}{\Delta k^2}}_{v_2}\right)
 \end{aligned} \tag{10.13}$$

where

$$v_1 \triangleq \frac{\mathbf{b}^T \mathbf{a} - \sqrt{\mathbf{a}^T M \mathbf{a} + c_k \Delta k^2}}{\Delta k^2} \tag{10.14a}$$

$$v_2 \triangleq \frac{\mathbf{b}^T \mathbf{a} + \sqrt{\mathbf{a}^T M \mathbf{a} + c_k \Delta k^2}}{\Delta k^2} \tag{10.14b}$$

$$M \triangleq \mathbf{b} \mathbf{b}^T - \Delta k^2 I. \tag{10.14c}$$

In the above formulation, the probability of violating the CBF constraint can be rewritten as the probability that the random variable v_k lies within an interval whose bounds depend on the control input (recall that $\mathbf{a} = \mathbf{a}(\mathbf{u}_k)$ as in (10.11)):

$$\mathbb{P}(CBC(\theta_i) \leq 0) = \mathbb{P}(v_1(\theta_i) \leq v_k \leq v_2(\theta_i)) \leq \rho. \tag{10.15}$$

Consequently, the overall probabilistic constraint becomes

$$\mathbb{P}(CBC_{k+1|k} \leq 0) = \sum_{i=1}^{|\Theta|} \mathbb{P}(\theta_k = \theta_i) \mathbb{P}(v_1(\theta_i) \leq v_k \leq v_2(\theta_i)) \leq \rho. \tag{10.16}$$

Although it is possible to derive a formulation that exploits the distribution of the pedestrian's speed v and motion direction θ , such an approach may still be computationally expensive. In addition, the tuning of the parameter ρ remains a nontrivial issue.

In the following, an example of how this type of constraint can be applied in a one-step-ahead control setting is presented.

10.4 Application of CC-CBF: a wheelchair motion planner

In this section, a possible application of the previously described constraint is proposed. To maintain the focus on social and assistive robotics, the constraint is integrated into a motion planning framework for an autonomous wheelchair designed to transport individuals with walking impairments.

In this application, the wheelchair operates in the presence of a moving obstacle—a human with stochastic behavior, represented, as before, by the pair (v, θ) of speed and motion direction. Assuming that the distribution of (v, θ) is known (as this aspect is not the focus of the present dissertation), a CC-CBF constraint is incorporated into the optimization problem to ensure collision avoidance. The resulting formulation is then compared against the standard hard-constraint case.

Since the previous section highlighted the difficulty of reformulating a probabilistic constraint into deterministic terms, the problem is simplified by assuming that the obstacle moves with constant velocity, at least over the considered time step. This assumption is reasonable in pedestrian scenarios, as their speed typically does not undergo abrupt variations within short time horizons.

Uncertainty remains in the pedestrian's direction of motion, which is modeled as a probability distribution over a discrete set of possible directions $\theta \in \Theta$. Accordingly, θ_k is treated as a random variable with finite support $\Theta_k = \{\theta_{1,k}, \dots, \theta_{\tau,k}\}$, where $|\Theta_k|$ denotes the cardinality of the set.

For simplicity, it is assumed that Θ_k remains constant over time, i.e., $\Theta_k = \Theta$ for all $k = 0, \dots, N-1$, where N is the simulation horizon. This assumption can, however, be relaxed if predictive tools are available that estimate time-varying motion directions.

The probability distribution of the random variable is defined as:

$$\mathbb{P}(\theta = \theta_i) = \begin{cases} p_1, & \text{if } \theta_i = \theta_1, \\ \vdots & \\ p_{|\Theta|}, & \text{if } \theta_i = \theta_{|\Theta|}. \end{cases} \quad \text{s.t.} \quad \sum_{i=1}^{|\Theta|} p_i = 1. \quad (10.17)$$

The objective is to ensure collision avoidance under uncertainty, where the uncertainty arises from the multiple possible motion directions in Θ . To this end, the deterministic constraint in (10.8) is reformulated into a probabilistic form. Following the approach in [32], define

$$\mathbb{P}(CBC_{k+1|k} \geq 0) \geq 1 - \rho, \quad (10.18)$$

which enforces the probability of satisfying the collision avoidance constraint, with $\rho \in [0, 1]$ serving as a tuning parameter that adjusts the acceptable risk level.

By applying the law of total probability and using the distribution defined in (10.17), (10.18) can be factorized as

$$\begin{aligned} \mathbb{P}(CBC_{k+1|k} \geq 0) &= \sum_{i=1}^{|\Theta|} \mathbb{P}(\theta = \theta_i) \cdot \mathbb{P}(CBC_{k+1|k} \geq 0 \mid \theta_i) \\ &= \sum_{i=1}^{|\Theta|} p_i \cdot \mathbb{P}(CBC_{k+1|k} \geq 0 \mid \theta_i). \end{aligned} \quad (10.19)$$

In this expression, $\mathbb{P}(CBC_{k+1|k} \geq 0 \mid \theta_i)$ is the probability of satisfying the collision avoidance constraint given $\theta_k = \theta_i$. Since fixing θ_i removes all randomness, this term becomes deterministic and depends only on the control input $\mathbf{u}_k$. Hence, (10.19) reduces to

$$\mathbb{P}(CBC_{k+1|k} \geq 0) = \sum_{i=1}^{|\Theta|} p_i \cdot \mathbb{I}(CBC_{k+1|k} \mid \theta_i), \quad (10.20)$$

where $\mathbb{I}(\cdot)$ denotes the indicator function, defined as

$$\mathbb{I}(x) \doteq \begin{cases} 1, & \text{if } x \geq 0, \\ 0, & \text{otherwise.} \end{cases} \quad (10.21)$$

Through this formulation, the original probabilistic constraint is expressed as a convex combination of indicator functions, each depending explicitly on the control input.

Before presenting the optimization problem that incorporates this type of constraint, a clarification is needed. While the indicator function in (10.21) provides an exact translation of the probabilistic condition into deterministic terms, it is not differentiable. Consequently, standard optimization solvers may face difficulties in efficiently finding feasible solutions.

To overcome this issue, the indicator function $\mathbb{I}(\cdot)$ is approximated with a smooth sigmoid function, defined as

$$\sigma(x) = \frac{1}{1 + \exp(-\kappa \cdot x)}, \quad (10.22)$$

where $\kappa \in \mathbb{R}$ is a shaping parameter that controls the steepness of the transition between 0 and 1. Larger values of κ yield a closer approximation to the indicator function, at the cost of reduced numerical smoothness.

To evaluate the proposed approach, two different algorithms are compared, both relying on similar optimization formulations.

The robot dynamics are modeled as a discrete-time double integrator, as introduced in (10.1). The robot is assumed to have a final target state

$$\mathbf{x}_{ref} = [x_{ref}, y_{ref}, v_{ref}^x, v_{ref}^y]^T,$$

which represents the desired position and velocity.

The objective function to be minimized is defined as

$$J(\mathbf{x}_k, \mathbf{u}_k) = (\mathbf{x}_k - \mathbf{x}_{ref})^T Q (\mathbf{x}_k - \mathbf{x}_{ref}) + \mathbf{u}_k^T R \mathbf{u}_k, \quad (10.23)$$

where $Q \succeq 0$, $Q \in \mathbb{R}^{4 \times 4}$, and $R \succ 0$, $R \in \mathbb{R}^{2 \times 2}$ are the weighting matrices for the state error and control input, respectively.

Thus, the optimization problem to be solved for all $k = 0, \dots, N - 1$, with $x_0 \in \mathcal{X}$ known, is defined as follows.

Optimization Problem 10.4.1 (OP1)

$$\begin{aligned} \mathbf{u}_k^* &= \operatorname{argmin}_{\mathbf{u} \in \mathcal{U}} J(\mathbf{x}_k, \mathbf{u}_k) \\ \text{s.t. } &\begin{cases} \mathbf{x}_{k+1} = f(\mathbf{x}_k, \mathbf{u}_k), \\ \mathbb{P}(CBC_{k+1|k} \geq 0) \geq 1 - \rho. \end{cases} \end{aligned} \quad (10.24)$$

The behavior of the controlled agent using the controller in (10.24) is compared with an alternative approach that employs a similar optimization problem but implements a different collision avoidance strategy. Specifically, the alternative controller is highly conservative, enforcing a deterministic collision avoidance constraint that accounts for all possible trajectories of the obstacle. In this alternative approach, the optimization problem can be formulated as

Optimization Problem 10.4.2 (OP2)

$$\begin{aligned} \mathbf{u}_k^* &= \operatorname{argmin}_{\mathbf{u} \in \mathcal{U}} J(\mathbf{x}_k, \mathbf{u}_k) \\ \text{s.t. } &\begin{cases} \mathbf{x}_{k+1} = f(\mathbf{x}_k, \mathbf{u}_k), \\ CBC_{k+1|k}(\mathbf{x}_k, \mathbf{u}_k, \mathbf{o}_k) \Big|_{\theta_i} \geq 0, \forall \theta_i \in \Theta. \end{cases} \end{aligned} \quad (10.25)$$

In the following, the optimization problem in (10.24) will be referred to as *OP1*, while the alternative approach in (10.25) will be referred to as *OP2*.

To evaluate and compare the two controllers, the scenario depicted below is considered. The robot is initialized at

$$\mathbf{x}_0 = [0.2, 4, 0, 0]^T,$$

while the obstacle starts from

$$\mathbf{o}_0 = [0, 0]^T,$$

which also corresponds to the robot's final target. The weighting matrices in (10.23) are set as

$$Q = \operatorname{diag}([10, 10, 0.1, 0.1]), \quad R = \operatorname{diag}([0.1, 0.1]).$$

The simulation horizon is 10s, with a sampling time of $\Delta k = 0.1$ s, resulting in $N = 100$ discrete-time steps.

It is assumed that the obstacle can move in three possible directions at each time step.

$$\Theta = \left\{ \frac{\pi}{2} - \alpha; \frac{\pi}{2}; \frac{\pi}{2} + \alpha \right\}, \quad \alpha = \frac{\pi}{4}. \quad (10.26)$$

The distribution of θ is assumed to be constant over time and is defined as a generalized Bernoulli distribution:

$$\mathbb{P}(\theta = \theta_i) = \begin{cases} 0.25, & \text{if } \theta_i = \frac{\pi}{2} - \alpha, \\ 0.5, & \text{if } \theta_i = \frac{\pi}{2}, \\ 0.25, & \text{if } \theta_i = \frac{\pi}{2} + \alpha. \end{cases} \quad (10.27)$$

The obstacle is assumed to move at a constant speed, $v_k = v = 0.5$ m/s, for all $k = 0, \dots, N - 1$, although this assumption can be relaxed if desired.

Given the support in (10.26) and the distribution in (10.27), the obstacle's trajectory is modeled as a random process. The probabilistic constraint parameter is set to $\rho = 0.6$, corresponding to a minimum probability of collision avoidance of 0.4. The minimum distance from the obstacle is set to $d_{min} = 0.5$ m.

Finally, the CBF parameter is set to $\gamma = 0.3$, and the shaping parameter of the sigmoid function in (10.22) is set to $\kappa = 100$.

To enhance clarity in the presentation, all results obtained with *OP1* are plotted in blue, whereas those obtained with *OP2* are plotted in red. First, the trajectories produced by the two methods are compared. In Fig. 10.2, the complete path of the robot in 2D space is shown, while Fig. 10.3 presents selected time snapshots illustrating the robot's behavior as it approaches the obstacle. It is immediately apparent that the robot must adjust its trajectory as the obstacle draws nearer. This behavior arises from the formulation of the optimization problems in (10.24) and (10.25), which consider only a single time instant at a time, preventing the robot from proactively anticipating future motions. Consequently, the trajectory deviations appear abrupt.

Due to the formulation of the optimization problem in (10.25), the red trajectory should be interpreted as a highly conservative path. This arises because the controller, as designed, accounts for all possible directions in Θ , meaning that at each time step it assumes the obstacle could occupy any of three different locations in the next step. Since this approach relies on a deterministic set of constraints, it does not consider differences in probabilities; rather, it ensures safety with respect to all possible outcomes equally.

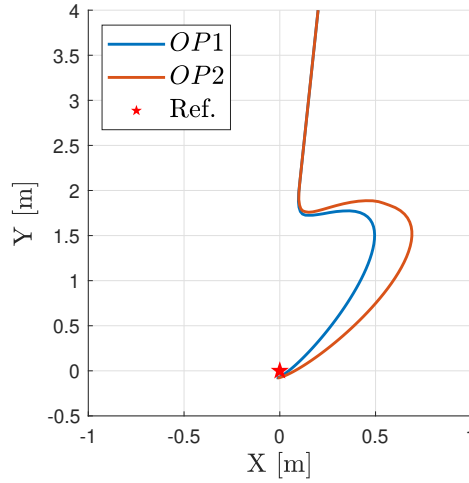


Figure 10.2: Comparison of the obtained trajectories. The red star marks the robot's target.

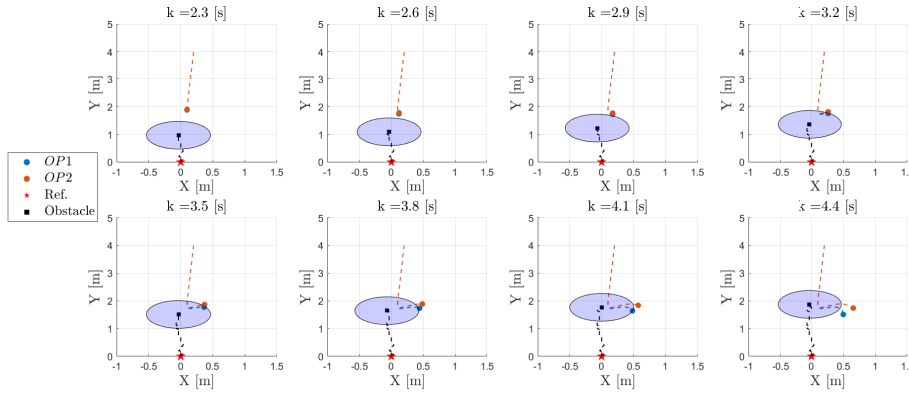


Figure 10.3: Time snapshots of the performed test are shown. The blue and red dots represent the positions of the robot at time step k when controlled using $OP1$ and $OP2$, respectively. The corresponding dashed lines indicate the robot's previous paths, following the same color scheme. The black square represents the obstacle's position, while the pink dashed line traces its past trajectory. The red star marks the robot's target, and the light blue circle highlights the unsafe region, defined as the set of points within an Euclidean distance smaller than d_{min} from the obstacle.

Therefore, it can be asserted that the blue trajectory, again the one derived according to $OP1$, may be less safe compared to the one obtained with $OP2$. However, this does not necessarily compromise the overall safety of the movement, as the controller accounts for the probability distribution of future obstacle motions. If feasible and suitable from an optimization perspective, the controller can still ensure safety with respect to all directions in Θ .

Indeed, by setting the lower bound for the probability of no collision to 0.4 and given the known distribution of θ in (10.27), the solver for $OP1$ should return a solution that guarantees safety either for $\theta_1 = \frac{\pi}{2} - \alpha$ and $\theta_3 = \frac{\pi}{2} + \alpha$ together, for $\theta_2 = \frac{\pi}{2}$ alone, or for any combination of θ_2 with θ_1 and θ_3 .

This allows the control action to be less conservative, sometimes favoring the minimization of the input itself. Moreover, since the control action is recomputed at each time step while accounting for the updated position of the obstacle, the robot can dynamically adjust its path as needed.

The controller still ensures that the safety distance d_{min} is maintained, as illustrated in the snapshots in Fig. 10.3 and in Fig. 10.4, where the Euclidean distance between the robot and the obstacle is displayed.

It is worth noting that there appear to be some time steps between $k = 3s$ and $k = 4s$

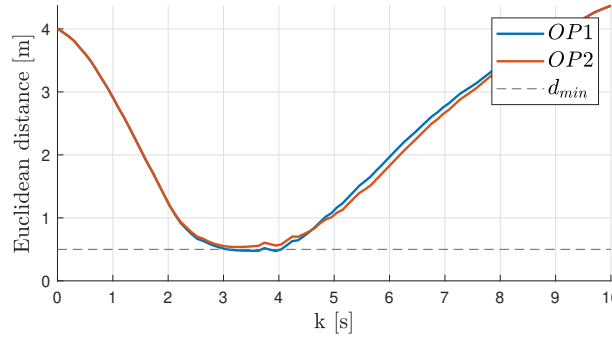


Figure 10.4: Comparison of the Euclidean distance between the robot and the obstacle. The black dashed line represents the minimum allowable distance d_{min} .

where the distance obtained with $OP1$ falls slightly below d_{min} . This discrepancy arises from the approximation of the indicator function with a sigmoid function.

Finally, the differences in the control actions are analyzed, as shown in Fig. 10.5. It can be observed that the inputs derived from the two approaches are nearly identical before and after the interaction with the obstacle, during periods when the collision avoidance constraint is not active. Specifically, the inputs are similar in both approaches from $k = 0$ s to approximately $k \approx 1.5$ s and from $k \approx 7$ s until the end of the simulation. Since the robot is expected to move primarily in the vertical direction to reach its target, a larger acceleration along the y -axis compared to the x -axis is anticipated, which is confirmed by the figure.

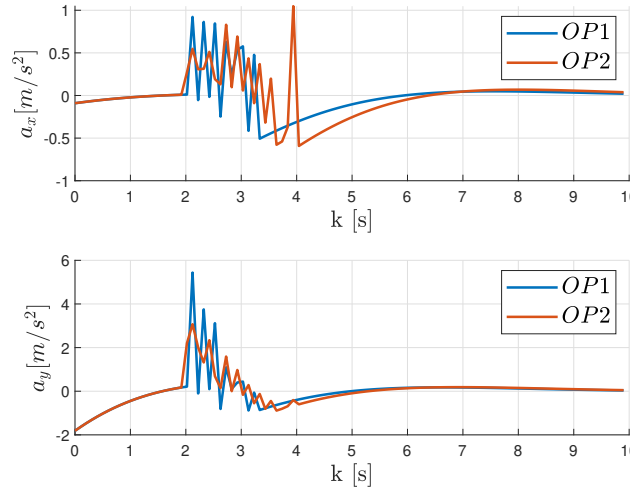


Figure 10.5: Comparison of control inputs: a_x (top) and a_y (bottom).

Both control inputs exhibit irregularities; however, they cannot be directly compared. In the presented simulation, the control action from $OP1$ appears to stabilize earlier than that of $OP2$. This behavior, however, depends on the obstacle's path, which is not controlled. Indeed, a different obstacle trajectory could result in a more irregular control action with $OP1$.

In addition to the comparison with $OP2$, a sensitivity analysis of $OP1$ with respect to the parameter γ is conducted. Figure 10.6 illustrates the different robot trajectories obtained by varying γ , while Fig. 10.7 provides a depiction of the Euclidean distance between the agents.

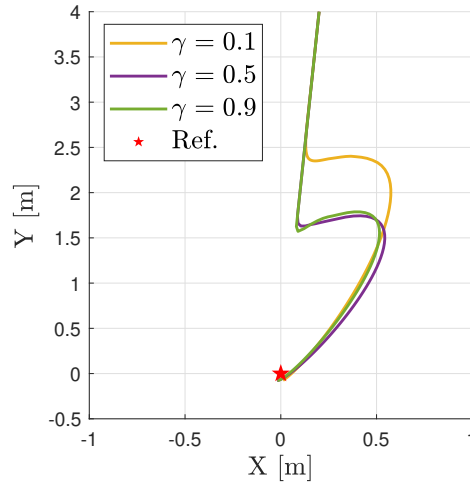


Figure 10.6: Comparison of trajectories obtained with *OP1*, varying γ .

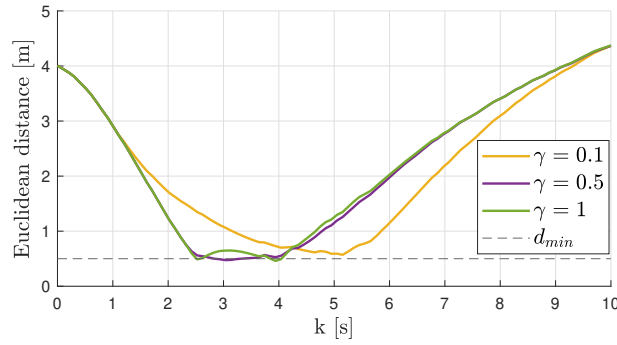


Figure 10.7: Comparison of the Euclidean distance between the robot and the obstacle with *OP1*, varying γ .

From (10.4), it can be intuitively understood that the parameter γ influences the aggressiveness of the collision avoidance constraint. This effect occurs in both the deterministic and probabilistic formulations, as the structure of CBC remains unchanged in the stochastic case. Since the CBF framework assumes that the system starts from safe initial conditions, two cases may arise: either $h(\mathbf{x}_k, \mathbf{o}_k) = 0$ or $h(\mathbf{x}_k, \mathbf{o}_k) > 0$. In the first case, γ has no impact on the controller, as it is multiplied by zero.

Interestingly, γ affects the aggressiveness of the control action when the agent remains in a safe scenario. Specifically, as $\gamma \rightarrow 0$, the next control action considers the previous distance between the robot and the obstacle. Since this distance is positive and multiplied by $(1 - \gamma) > 0$, the constraint enforces maintaining or even increasing the separation. Conversely, as $\gamma \rightarrow 1$, the CBC reduces to a standard collision avoidance constraint that does not account for the previous distance, leading to a more aggressive maneuver. This effect is evident in Fig. 10.6, where the trajectory obtained with $\gamma = 0.9$ deviates from the nominal path later than those obtained with lower γ values.

In conclusion, as $\gamma \rightarrow 0$, the approach becomes more conservative, as illustrated in Figs. 10.6 and 10.7, where the robot deviates from the nominal path earlier. Conversely, for higher values of γ , the approach is more aggressive.

10.5 Discussion

Despite the interesting results obtained, this approach has several limitations. First, it considers only a single time-instant control action, thereby preventing longer-term

predictions of human decisions that could lead to smoother trajectories. This limitation is evident in Fig. 10.2, where the collision avoidance constraint results in an abrupt change of path.

A second limitation lies in the collision avoidance constraint itself, modeled as CC-CBF. The adopted CBF framework considers a probabilistic constraint of the type

$$\mathbb{P}(\varphi_i \geq 0) \leq \rho, \quad i = 1, \dots, n_c, \quad (10.28)$$

with n_c denoting the number of scenarios and $\varphi_i \geq 0$ representing the deterministic constraint.

In this formulation, the probability of remaining within the safe set, expressed by $\varphi_i \geq 0$, must not exceed a predefined threshold ρ . However, this presents a key drawback: satisfying the probabilistic constraint does not necessarily imply safety in the most likely scenario. Safety may instead be guaranteed across several low-probability scenarios which, when combined, fulfill the threshold, without ensuring safety in the most probable outcomes. This undermines the effectiveness of the approach, as the individual likelihoods of each scenario are not explicitly accounted for.

Therefore, an alternative formulation of the CBF constraint is proposed in the following chapter.

Chapter 11

Hierarchical Motion Planner for Autonomous Agents based on Soft Control Barrier Functions

In this chapter, building on the limitations highlighted in Chapter 10, a new version of the CBF is proposed to address uncertain scenarios and to enhance safety with respect to the most likely outcomes. The developed framework is then incorporated into a more general structure that employs a double-layer architecture.

The intended application remains an autonomous wheelchair navigating in human environments. Further details on the experimental setup are provided in Appendix G.

This chapter is organized as follows. Section 11.1 describes the microscopic controller, which has been designed to overcome the limitations of the CC-CBF. Section 11.2 presents the modeling of a macroscopic planner that identifies the most suitable regions for the autonomous robot to move, based on information from the environment. Section 11.3 combines the two planners into a unified control scheme, and the results of its application are presented and discussed in Section 11.4. Finally, conclusions and open issues are discussed in Section 11.5.

The contents of this Chapter are mainly derived from the works presented in [90], [91], although many of the details remain part of ongoing research, especially in Section 11.1.6.

11.1 Microscopic Motion Planner: a Soft CBF Approach for Collision Avoidance

Following the results obtained in Chapter 10, it is observed that the probabilistic constraint in *OP1* does not inherently prioritize any specific trajectory. In other words, when the risk threshold ρ is set equal to the highest individual probability p_i (recall that the constraint requires the probability of safety to be at least $1 - \rho$), the resulting control input may ensure safety with respect to all trajectories, or only a subset of them. However, the formulation provides no guarantee that the trajectory with the highest probability will be more strongly enforced than one with a lower probability. This lack of prioritization motivates a deeper investigation into explicitly incorporating such probabilistic information into the optimization problem in order to favor the most likely trajectories.

Assume, for instance, that

$$\Theta = \{\theta_1, \theta_2, \theta_3\}, \quad (11.1)$$

with probabilities p_1, p_2, p_3 , respectively. Considering a horizon of length N , there are 3^N possible scenarios, each with different probabilities. In the motion planning problem, a key consideration is how to exploit the available probabilistic information regarding the potential future behaviors of obstacles. Rather than adopting a uniformly conservative approach that assumes all trajectories are equally likely, the autonomous agent can reason over the distribution of these possibilities.

One intuitive strategy is to prioritize the most probable trajectories, effectively focusing the collision avoidance constraint on the directions that are statistically more likely to occur. This leads to a probabilistically informed control strategy that balances safety and performance by relaxing constraints associated with low-probability events.

Alternatively, the agent may adopt a fully robust approach, enforcing safety constraints with respect to all possible trajectories in the set. While this guarantees safety under any direction the obstacle may take, it results in significantly more conservative behavior, often causing unnecessary deviations from the optimal path and more aggressive or irregular control inputs.

To assign a form of priority to the trajectories, a collision avoidance constraint based on CBF can be implemented for each possible trajectory i , while introducing a slack variable, referred to as a *relaxation variable*, δ_i . This variable is treated as an optimization variable taking values in $[0, \infty)$, with the conservative case corresponding to $\delta_i = 0$.

For the i^{th} scenario generated by the j^{th} obstacle, where $\mathbb{I}_O$ denotes the set of obstacles, the constraint at a single time step is formulated as

$$CBC_{k+1|k}(\mathbf{x}_k, \mathbf{u}_k, \mathbf{o}_k^{(j)}) \Big|_i \geq -\delta_i, \quad (11.2)$$

where the notation $\mathbf{o}_k^{(j)}$ indicates that the j^{th} obstacle is the one generating the i^{th} scenario, and the barrier function is defined as in (10.4). Notably, when $\delta_i = 0$, the formulation in (11.2) reduces to the robust deterministic case, in which a separate collision avoidance constraint is enforced for each scenario without any relaxation.

The formulation in (11.2) accounts for only a single time step. However, within an MPC framework, predictions are performed over a finite horizon of N steps. Since the environment is highly dynamic due to pedestrian motion, a short-term prediction horizon is both sufficient and appropriate. Pedestrians are modeled using the unicycle dynamics in (10.2); thus, predicting their future states requires consideration of two variables: the speed v_k and the heading angle θ_k . To simplify the theoretical development, the pedestrian's speed is assumed to remain constant over the short-term horizon. This assumption is justified by the observation that, in fully pedestrian environments, individuals typically do not exhibit abrupt changes in velocity over short durations.

Given a fixed v_k , a probabilistic distribution is assumed for θ_k . Ideally, this distribution could be inferred from data, for example by analyzing past portions of the pedestrian's trajectory or using neural networks. However, this is beyond the scope of the present work, as the focus is on the control framework rather than the prediction tools. Therefore, a finite-support distribution for θ_k is assumed for the obstacle j , defined as

$$\Theta_k^{(j)} = \{\theta_{k-1} - \Delta\theta, \theta_{k-1}, \theta_{k-1} + \Delta\theta\}. \quad (11.3)$$

In other words, the next possible motion direction depends on the previous direction θ_{k-1} and may deviate from it by an amount $\Delta\theta$. While the following discussion assumes a fixed probability distribution over these values, nothing precludes the use of a time-varying distribution, which could, for example, be modeled as a Markov chain.

The rationale for assuming the near-constant support in (11.3) is that pedestrians generally follow relatively smooth trajectories, except when deciding to turn left or right; thus deviations are naturally captured by allowing the direction to vary by $\pm\Delta\theta$.

Once the support set $\Theta^{(j)}$ has been defined, all possible scenarios can be generated by combining its values across the N prediction steps. First, all possible sequences of inputs θ , starting from time instant k for the obstacle dynamics (10.2), are collected in τ_k , generated via a Cartesian product as

$$\tau_k^{(j)} = \Theta_k^{(j)} \times \dots \times \Theta_{k+N-1}^{(j)}. \quad (11.4)$$

Once these sequences are defined, the set of possible trajectories (or scenarios) starting from time k is denoted by $\mathcal{T}_k$:

$$\mathcal{T}_k^{(j)} = \{\text{tr}_{1,k}^{(j)}, \text{tr}_{2,k}^{(j)}, \dots, \text{tr}_{N_{\text{tr}}^{(j)},k}^{(j)}\}, \quad (11.5)$$

where $N_{\text{tr}}^{(j)} = |\Theta^{(j)}|^N$ is the number of possible trajectories, and $\text{tr}_{i,k}^{(j)}$ denotes the i^{th} trajectory, expressed as

$$\text{tr}_{i,k}^{(j)} = [\mathbf{o}_k^{(j)}, \mathbf{o}_{k+1}^{(j)}, \dots, \mathbf{o}_{k+N}^{(j)}]. \quad (11.6)$$

The trajectory $\text{tr}_{i,k}^{(j)}$ is obtained by applying the control sequence $\tau_{i,k}^{(j)} = [\theta_{i,k}^{(j)}, \dots, \theta_{i,k+N-1}^{(j)}]^T \in \mathbb{R}^{N-1}$ to the obstacle dynamics, starting from the obstacle state

$\mathbf{o}_k^{(j)}$. Assuming independence of the probability distribution at different time instants, the cumulative probability of each heading sequence $\tau_{i,k}^{(j)}$, and hence of the associated trajectory $\text{tr}_{i,k}^{(j)} \in \mathcal{T}_k^{(j)}$, can be expressed as the product of the probabilities of its elements:

$$\mathbb{P}(\tau_{i,k}^{(j)}) = \mathbb{P}(\text{tr}_{i,k}^{(j)}) = \prod_{z=k}^{k+N-1} \mathbb{P}(\theta_z^j), \quad (11.7)$$

where θ_z^j is the z^{th} heading in the input sequence $\tau_{j,k}^{(j)}$ that generates the trajectory $\text{tr}_{i,k}^{(j)}$, for the obstacle j .

For each identified scenario, a collision avoidance constraint of the type (11.2) must be enforced along the entire predicted obstacle trajectory. This yields the condition

$$CBC_{k+N|k}(\mathbf{x}_k, \mathbf{u}_k, \mathbf{o}_k^{(j)}) \Big|_{\text{tr}_{i,k}^{(j)}} \geq -\delta_i^{(j)}, \quad \forall \text{tr}_{i,k}^{(j)} \in \mathcal{T}_k^{(j)}, \forall j \in \mathbb{I}_O, \quad (11.8)$$

where the notation $\Big|_{\text{tr}_{i,k}^{(j)}}^{(j)}$ indicates that the evolution of the j^{th} obstacle state is determined by the i^{th} trajectory $\text{tr}_{i,k}^{(j)}$.

However, the set of conditions above does not incorporate information regarding the probability distribution of the scenarios. To address this, the following optimization problem is considered by adding the scenario-dependent collision avoidance constraints (11.8) and introducing an additional term in the cost function. The resulting optimization problem is formulated as

Optimization Problem 11.1.1

$$\begin{aligned} \mathcal{U}_{k:k+N-1}^* = \arg\min_{\mathbf{u}_k \in \mathcal{U}, \forall k} & J(\mathbf{x}_{k:k+N}, \mathbf{u}_{k:k+N-1}) + J_{\mathcal{T}}(\boldsymbol{\delta}_k) \\ \text{s.t.} & \begin{cases} \mathbf{x}_{k+1} = f(\mathbf{x}_k, \mathbf{u}_k), \quad \forall k = 1, \dots, N-1, \\ \mathbf{u}_k \in \mathcal{U}_k, \quad \forall k = 1, \dots, N-1 \\ CBC_{k+N|k}(\mathbf{x}_k, \mathbf{u}_k, \mathbf{o}_k^{(j)}) \Big|_{\text{tr}_{i,k}^{(j)}} \geq -\delta_{i,k}^{(j)}, \quad \forall \text{tr}_{i,k}^{(j)} \in \mathcal{T}_k^{(j)}, \forall j \in \mathbb{I}_O \\ \delta_{i,k}^{(j)} \geq 0, \quad \forall i, k, j, \end{cases} \end{aligned} \quad (11.9)$$

The additional cost term is introduced to discourage excessive relaxation of the collision-avoidance constraints for the most likely scenarios, while allowing greater flexibility for less likely ones. Specifically, this term is defined as

$$J_{\mathcal{T}}(\boldsymbol{\delta}_k) = \boldsymbol{\delta}_k^T \mathbf{D}_k \boldsymbol{\delta}_k, \quad (11.10)$$

where $\boldsymbol{\delta}_k = [\boldsymbol{\delta}_k^{(1)}, \dots, \boldsymbol{\delta}_k^{(N_{\text{obs}})}]^T$ denotes the vector of relaxation variables associated to all the obstacles in $\mathbb{I}_O$. Each vector $\boldsymbol{\delta}_k^{(j)} = [\delta_k^{\text{tr}_1}, \dots, \delta_k^{\text{tr}_{N_{\text{tr}}}}]^T \in \mathbb{R}_{\geq 0}^{N_{\text{tr}}}$ contains the slack variables corresponding to the trajectories of the j^{th} obstacle. The matrix $\mathbf{D}_k \in \mathbb{R}^{N_{\text{obs}} N_{\text{tr}} \times N_{\text{obs}} N_{\text{tr}}}$ contains the probability of each trajectory on the principal diagonal, namely

$$\mathbf{D}_k = c \cdot \text{diag}([\mathbb{P}(\text{tr}_{1,k}^{(1)}), \dots, \mathbb{P}(\text{tr}_{N_{\text{tr}},k}^{(N_{\text{obs}})})]), \quad (11.11)$$

where $c \in \mathbb{R}$ is a scaling parameter, for now assumed equal to 1.

In this formulation, relaxation parameters associated with higher-probability scenarios incur stronger penalties. Consequently, the relaxation parameters corresponding to highly probable scenarios remain smaller than those for less likely scenarios, resulting in reduced deviations from the collision-avoidance constraint compared to the fully robust formulation.

Since the formulation above allows multiple pedestrians to be considered for avoidance, it is further assumed, to maintain a realistic setup, that the robot is equipped with an obstacle detection radius denoted by r_{obs} .

Remark 11.1.1

The design of the weighting matrix $\mathbf{D}_k$ in (11.11) can be further enhanced by incorporating contextual information about the obstacles through appropriate tuning of the parameter c . For example, if an external tool is available to estimate a pedestrian's level of attention—such as detecting whether the gaze is directed toward the robot—this information could be used to adapt the values in $\mathbf{D}_k$. In such cases, the robot could reasonably anticipate a certain level of cooperation from attentive humans, allowing a more flexible relaxation of the collision-avoidance constraints while still maintaining overall safety guarantees.

Before moving to the first results obtained by implementing the optimization problem in (11.9), the following lemma is proposed to guarantee mathematical consistence of the constraint.

Lemma 11.1.1

Consider an MPC framework where the collision avoidance constraint is built according to CBF theory (in discrete time). Assume that the goal is to avoid collisions between an autonomous agent and a moving obstacle. Assume that the barrier function designed for this goal is as follows

$$h(\mathbf{x}_k, \mathbf{o}_k) = \|\mathbf{x}_k^{[x,y]} - \mathbf{o}_k^{[x,y]}\|_2^2 - d_{\min}^2, \quad (11.12)$$

where $\mathbf{x}_k^{[x,y]}, \mathbf{o}_k^{[x,y]}$ are the state of the autonomous agent and moving obstacle, respectively, that considers just the spatial coordinates, $d_{\min}$ is the safety distance. Then, if the control barrier function constraint is designed as follows

$$h(\mathbf{x}_{k+1}, \mathbf{o}_{k+1}) \geq (1 - \gamma)h(\mathbf{x}_k, \mathbf{o}_k) - \delta_k, \quad (11.13)$$

where δ_k is the relaxation parameter that can take values in $[0, d_{\min}^2 + (1 - \gamma)h(\mathbf{x}_k, \mathbf{o}_k)]$, and the algorithm starts from safe initial conditions, namely $h(\mathbf{x}_k, \mathbf{o}_k) \geq 0$, it is guaranteed that the lower bound of the squared euclidean distance between the two agents is always positive or equal to zero, i.e., $\|\mathbf{x}_k^{[x,y]} - \mathbf{o}_k^{[x,y]}\|_2^2 \geq 0$. $\square$

Proof. Since an MPC framework is considered, there is a prediction horizon over which collision avoidance constraints must be enforced. Without loss of generality, let N denote the length of the prediction horizon. The collision avoidance constraints, expressed as CBF constraints, are then

$$\begin{aligned} k+1|k & : h(\mathbf{x}_{k+1}, \mathbf{o}_{k+1}) \geq (1 - \gamma)h(\mathbf{x}_k, \mathbf{o}_k) - \delta_k, \\ k+2|k+1 & : h(\mathbf{x}_{k+2}, \mathbf{o}_{k+2}) \geq (1 - \gamma)h(\mathbf{x}_{k+1}, \mathbf{o}_{k+1}) - \delta_{k+1}, \\ & \vdots \\ k+N+1|k+N & : h(\mathbf{x}_{k+N+1}, \mathbf{o}_{k+N+1}) \geq (1 - \gamma)h(\mathbf{x}_{k+N}, \mathbf{o}_{k+N}) - \delta_{k+N}. \end{aligned} \quad (11.14)$$

Consider the first constraint in (11.14). Since $\delta_k \in [0, d_{\min}^2 + (1 - \gamma)h(\mathbf{x}_k, \mathbf{o}_k)]$ and $h(\mathbf{x}_k, \mathbf{o}_k) \geq 0$, the minimum value that the right-hand side can attain is $-d_{\min}^2$, which occurs when $\delta_k = d_{\min}^2 + (1 - \gamma)h(\mathbf{x}_k, \mathbf{o}_k)$. Hence,

$$h(\mathbf{x}_{k+1}, \mathbf{o}_{k+1}) \geq -d_{\min}^2,$$

implying that the Euclidean distance satisfies

$$\|\mathbf{x}_k^{[x,y]} - \mathbf{o}_k^{[x,y]}\|_2^2 - d_{\min}^2 \geq -d_{\min}^2 \quad \Rightarrow \quad \|\mathbf{x}_k^{[x,y]} - \mathbf{o}_k^{[x,y]}\|_2^2 \geq 0.$$

Although the Euclidean distance is guaranteed to be non-negative, this does not necessarily imply that the CBF at time step $k+1$ is non-negative. Two cases may arise: either

$-d_{\min}^2 \leq h(\mathbf{x}_{k+1}, \mathbf{o}_{k+1}) < 0$, or $h(\mathbf{x}_{k+1}, \mathbf{o}_{k+1}) \geq 0$. The safety constraint at $k+2|k+1$ can be analyzed analogously to the first step.

Consider the case where $-d_{\min}^2 \leq h(\mathbf{x}_{k+1}, \mathbf{o}_{k+1}) < 0$. The interval $[0, d_{\min}^2 + (1 - \gamma)h(\mathbf{x}_{k+1}, \mathbf{o}_{k+1})]$ for δ_{k+1} is well-defined, since $d_{\min}^2 + (1 - \gamma)h(\mathbf{x}_{k+1}, \mathbf{o}_{k+1}) > 0$ for any $h(\mathbf{x}_{k+1}, \mathbf{o}_{k+1})$ within the interval.

The right-hand side of the second constraint at $k+2|k+1$ attains its minimum when $\delta_{k+1} = d_{\min}^2 + (1 - \gamma)h(\mathbf{x}_{k+1}, \mathbf{o}_{k+1})$:

$$\begin{aligned} \delta_{k+1} = 0 &\Rightarrow (1 - \gamma)h(\mathbf{x}_{k+1}, \mathbf{o}_{k+1}) - \delta_{k+1} = (1 - \gamma)h(\mathbf{x}_{k+1}, \mathbf{o}_{k+1}), \\ \delta_{k+1} = d_{\min}^2 + (1 - \gamma)h(\mathbf{x}_{k+1}, \mathbf{o}_{k+1}) &\Rightarrow (1 - \gamma)h(\mathbf{x}_{k+1}, \mathbf{o}_{k+1}) - \delta_{k+1} = -d_{\min}^2. \end{aligned} \quad (11.15)$$

If $-d_{\min}^2 \leq h(\mathbf{x}_{k+1}, \mathbf{o}_{k+1}) < 0$ and $\gamma \in (0, 1)$, then

$$-d_{\min}^2 < (1 - \gamma)h(\mathbf{x}_{k+1}, \mathbf{o}_{k+1}).$$

Consequently, the second step satisfies

$$h(\mathbf{x}_{k+2}, \mathbf{o}_{k+2}) \geq -d_{\min}^2,$$

analogously to the first step. $\square$

This result shows that the squared Euclidean distance is always non-negative, and consequently the Euclidean distance itself is non-negative. However, to avoid the edge case in which the distance becomes exactly zero—which may be undesirable or infeasible in practice—the interval for δ_k can be slightly tightened as

$$\delta_k \in [0, d_{\min}^2 + (1 - \gamma)h(\mathbf{x}_k, \mathbf{o}_k) - \epsilon], \quad \epsilon \geq 0.$$

The upper bound of ϵ depends on the specific CBF used, but in practice, a small constant value such as $\epsilon = 10^{-3}$ can be chosen to ensure

$$\|\mathbf{x}_k^{[x,y]} - \mathbf{o}_k^{[x,y]}\|_2^2 \geq \epsilon.$$

This adjustment does not alter the overall proof methodology but ensures feasibility by avoiding zero-distance configurations.

The previous lemma guarantees that the lower bound of the squared Euclidean distance is positive when starting from safe initial conditions. However, in an MPC framework, it is possible to encounter unsafe initial conditions. The following lemma addresses this scenario:

Lemma 11.1.2

Consider an MPC framework in which the collision avoidance constraint and the barrier function are defined as in Lemma 11.1.1. If the relaxation parameter takes values in $[0, d_{\min}^2 + (1 - \gamma)h(\mathbf{x}_k, \mathbf{o}_k)]$ and the system does not start from safe initial conditions, i.e., $h(\mathbf{x}_k, \mathbf{o}_k) < 0$, then the lower bound of the squared Euclidean distance between the two agents is still guaranteed to be non-negative:

$$\|\mathbf{x}_k^{[x,y]} - \mathbf{o}_k^{[x,y]}\|_2^2 \geq 0.$$

$\square$

Proof. If $h(\mathbf{x}_k, \mathbf{o}_k) < 0$, then $-d_{\min}^2 < h(\mathbf{x}_k, \mathbf{o}_k) < 0$. The result then follows directly from the proof of Lemma 11.1.1. $\square$

The proposed controller (11.9) is tested and first compared with the hard-constraint formulation, and subsequently with the CC-CBF formulation presented in Chapter 10.

11.1.1 Comparison with hard CBF

In this section, the distinction between the soft-constraint formulation and a corresponding hard-constraint formulation is highlighted. Specifically, the optimization problem defined in (11.9), which includes the relaxation parameters δ_i as decision variables, is compared with a baseline case in which $\delta_i = 0$ for all i , thereby enforcing hard safety constraints without any relaxation. This comparison demonstrates the benefit of allowing controlled constraint violations, particularly in scenarios where strict feasibility may be difficult to achieve.

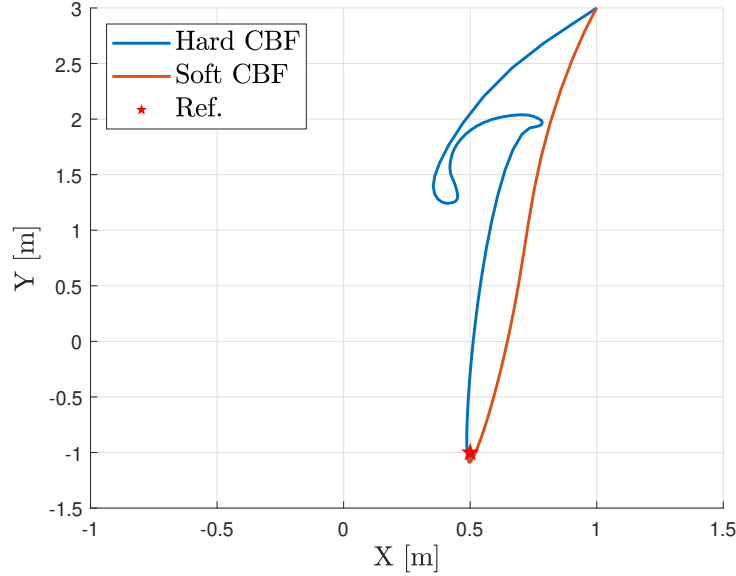


Figure 11.1: Comparison between the trajectories obtained with hard and soft CBF.

In Fig. 11.1, the trajectories obtained using the hard (blue) and soft (red) CBF MPC formulations are plotted. To better visualize the differences, selected snapshots are compared in Figs. 11.2 and 11.3. It is evident that, while the hard formulation strictly prevents the robot from intersecting the safety zones of the obstacles in any situation, the soft formulation allows the robot to approach obstacles more closely if doing so results in a smoother and faster trajectory.

11.1.2 Comparison with CC-CBF

In the CC-MPC setting, the optimization problem is defined as in (10.24), where the collision-avoidance constraint is probabilistic and bounded by $1 - \rho$, with ρ serving as a tuning parameter.

To enable a proper comparison between the two approaches, it is assumed that the stochasticity in the problem arises from the human's decision process. At each time step, the human can choose among three possible directions, denoted by θ , with known probabilities that are assumed to remain constant over time. Under this assumption, the probabilistic constraint can be expressed as

$$\mathbb{P}[CBC_{k+1|k}(\mathbf{x}_k, \mathbf{u}_k, \mathbf{o}_k) \geq 0] = \sum_{i=1}^{|\Theta|} \mathbb{P}[\theta_k = \theta_i] \cdot \mathbb{I}(CBC_{k+1|k}(\mathbf{x}_k, \mathbf{u}_k, \mathbf{o}_k | \theta_i) \geq 0) \geq 1 - \rho. \quad (11.16)$$

A key limitation of the CC-CBF formulation is the lack of prioritization among the probabilistic constraints. To illustrate this, a simple simulation is designed. The threshold of the probabilistic constraint is set to 0.4 (i.e., $\rho = 0.6$), and the obstacle is allowed to

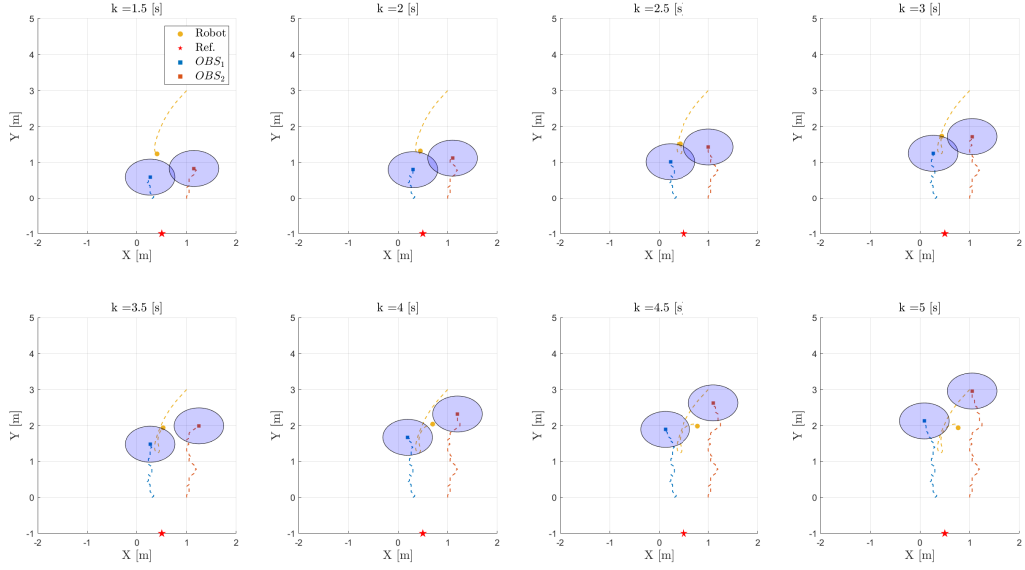


Figure 11.2: Snapshots of the simulation with hard CBF.

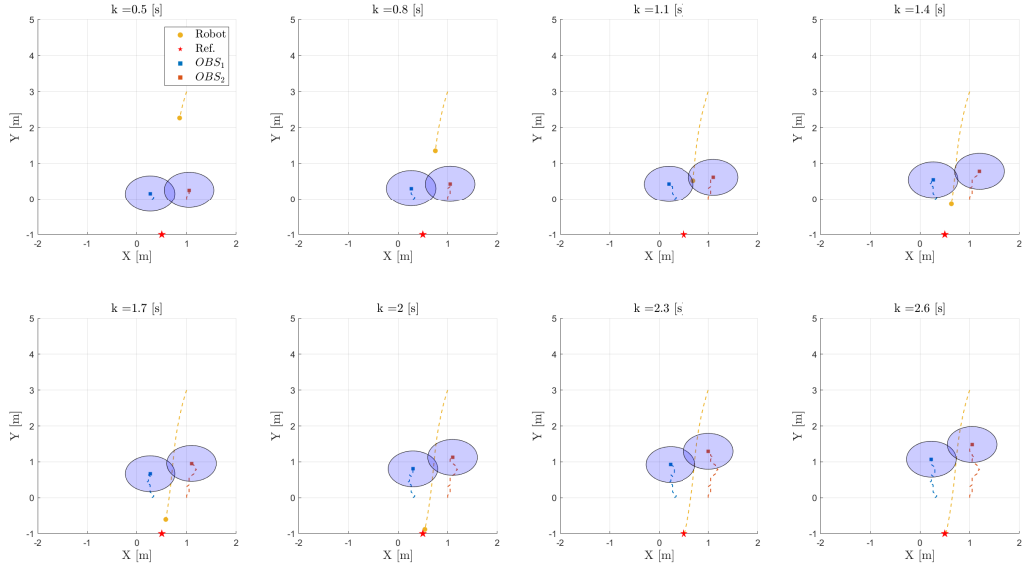


Figure 11.3: Snapshots of the simulation with soft CBF.

choose among three possible directions θ , as defined in (11.1). The probability distribution is defined as

$$\mathbb{P}[\Theta] = \begin{cases} 0.25 & \text{if } \theta = \theta_1, \\ 0.5 & \text{if } \theta = \theta_2, \\ 0.25 & \text{if } \theta = \theta_3. \end{cases} \quad (11.17)$$

Given a safety threshold of 0.4, the robot must ensure safety with respect to different combinations of trajectories:

$$\{(\theta_1, \theta_3), (\theta_2), (\theta_1, \theta_2), (\theta_2, \theta_3), (\theta_1, \theta_2, \theta_3)\}. \quad (11.18)$$

The most conservative option corresponds to enforcing safety for the last combination. However, in the presence of a reference cost in the optimization problem, the behavior

of the autonomous agent may be influenced, potentially leading to a less conservative response.

In the performed simulation, the results are shown in Fig. 11.4.

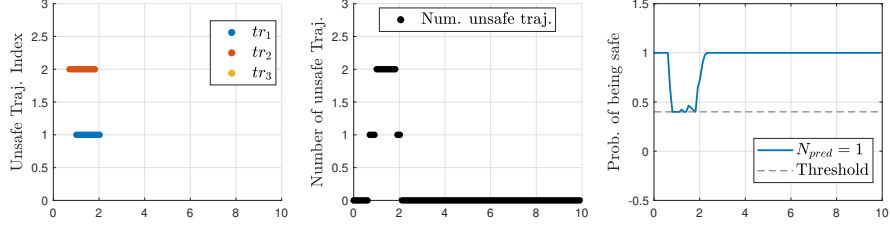


Figure 11.4: CC-CBF simulation.

The figure presents, from left to right, the indices of the unsafe trajectories, the number of unsafe trajectories at each time step, and the probability of satisfying the safety condition, i.e., $\mathbb{P}[CBC \geq 0]$. It can be observed that, at certain time instants, trajectories tr_1 and tr_2 are unsafe. However, this is not explicitly captured in the combinations considered in (11.18). Despite this, the rightmost plot shows that the overall combination still satisfies the probabilistic safety threshold, indicated by the black dashed line.

This apparent discrepancy is due to the implementation details of the optimization problem. In (11.16), the probabilistic constraint is interpreted as the sum of the products between the realization probabilities of θ and the corresponding indicator function $\mathbb{I}(\cdot)$. In practice, the indicator function is approximated by a smooth sigmoid function, as described in (10.22), to ensure differentiability. Consequently, for certain values of the CBC evaluated at specific realizations of θ , the approximate indicator function returns values in the interval $(0, 1)$. As a result, the weighted sum of probabilities may not precisely reflect the true probability of constraint satisfaction.

Nevertheless, the simulation highlights a limitation of the CC-CBF approach: the solution becomes unsafe with respect to tr_2 , which corresponds to the most likely trajectory, rather than avoiding tr_1 or tr_3 . This behavior is influenced by the reference cost in the optimization problem, which may bias the solution toward overall feasibility or performance at the expense of safety for the most probable outcomes.

The behavior observed in Fig. 11.4 is also influenced by the value of ρ , which remains a tuning parameter that may need adjustment based on the scenario. Figs. 11.5, 11.6, and 11.7 illustrate the resulting trajectories, the evolution of the probabilistic constraint along the prediction horizon, and the indices of unsafe trajectories at the first prediction instant, respectively, for different values of the probabilistic threshold.

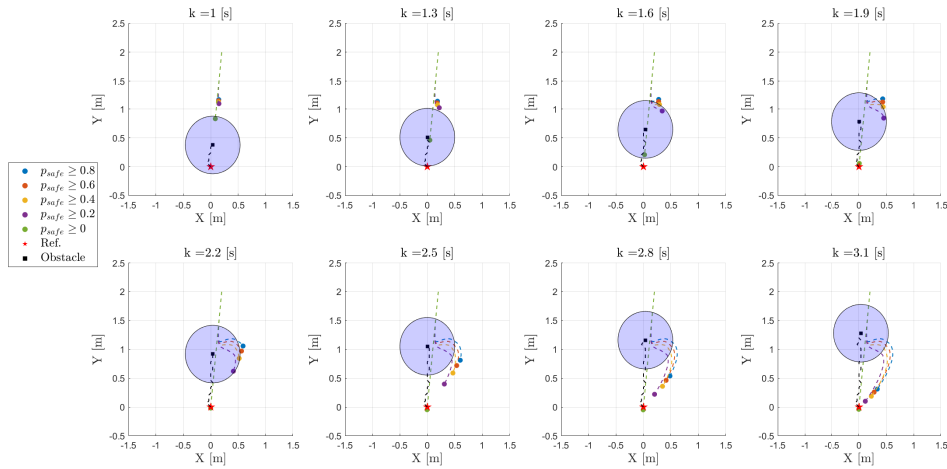


Figure 11.5: Trajectory of the autonomous agent for varying probability thresholds.

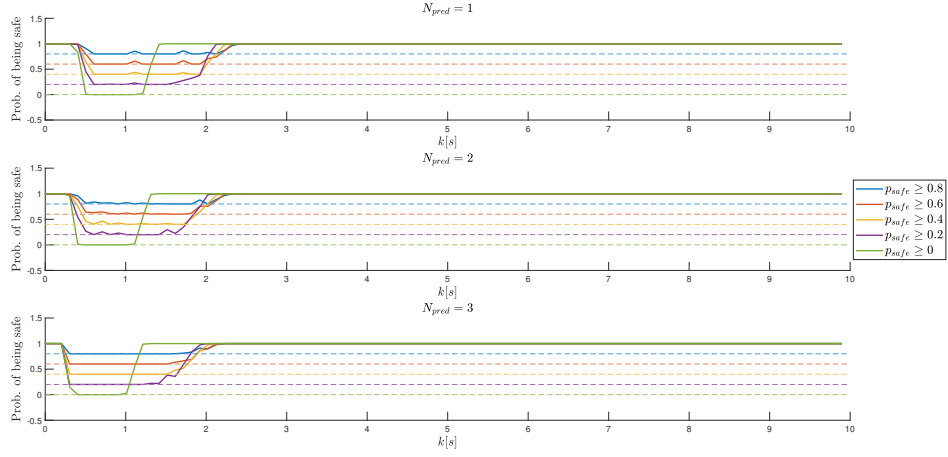


Figure 11.6: Evolution of the probabilistic constraint along the prediction horizon for varying probability thresholds.

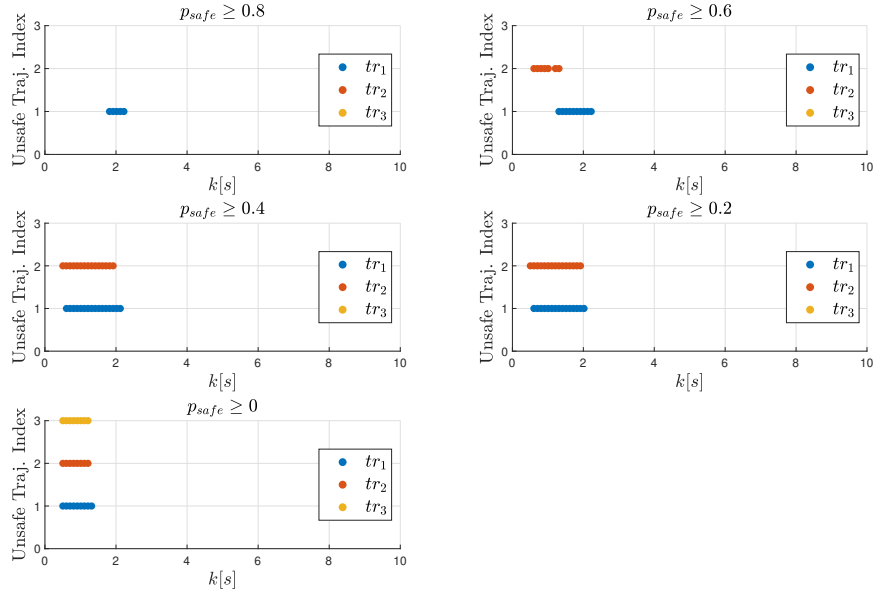


Figure 11.7: Indices of unsafe trajectories at the first prediction instant for varying probability thresholds.

As shown in Fig. 11.7, the optimizer frequently allows violations with respect to the second trajectory (depicted in red), which corresponds to the most likely realization. This further demonstrates the limitation of the chance-constrained approach in distinguishing between high- and low-probability scenarios when enforcing safety constraints.

The lack of prioritization for more probable trajectories is a fundamental limitation of the CC-MPC framework. The proposed approach addresses this issue by introducing relaxation parameters as optimization variables, weighted according to the probability of each trajectory. This formulation explicitly prioritizes safety for the most likely outcomes, ensuring that constraint violations, when they occur, are more likely to affect less probable trajectories, thereby aligning safety enforcement with the underlying uncertainty structure.

11.1.3 Safety improvement

It is useful to revisit a critical issue previously discussed: the necessity of bounding the relaxation variable δ to avoid infeasible or unrealistic scenarios. To this end, Lemma 11.1.1

establishes that

$$\delta_k \in [0, d_{\min}^2 + (1 - \gamma)h(\mathbf{x}_k^{[x,y]}, \mathbf{o}_k^{[x,y]})].$$

However, this interval may lead to the constraint

$$\|\mathbf{x}_k^{[x,y]} - \mathbf{o}_k^{[x,y]}\|_2^2 \geq 0,$$

which, while always satisfied, permits the edge case in which the Euclidean distance between the robot and the obstacle becomes exactly zero. Such a situation is clearly unsafe and must be explicitly avoided in practice.

To address this, the following refined bound can be considered:

$$\delta_k \in [0, d_{\min}^2 + (1 - \gamma)h(\mathbf{x}_k^{[x,y]}, \mathbf{o}_k^{[x,y]}) - \epsilon^2], \quad \epsilon \geq 0.$$

This adjustment must be applied within the prediction step of the MPC framework. Since the first prediction step typically carries the highest certainty and therefore requires the strongest safety guarantees, it is reasonable to adopt a time-varying formulation of ϵ , enforcing stricter margins at earlier prediction steps while allowing greater flexibility further into the horizon.

Let d_{ext} denote the minimum Euclidean distance that must be maintained from the obstacle under extreme safety requirements, and let $i = 1, \dots, N_{\text{pred}}$ index the prediction steps. The parameter is then defined as

$$\epsilon_i = d_{\text{ext}} \cdot e^{-(i-1)}, \quad (11.19)$$

so that at the first prediction instant ($i = 1$) the most conservative margin is enforced with $\epsilon_1 = d_{\text{ext}}$, and this requirement decays exponentially over the horizon. In this way, strong safety guarantees are preserved in the near term while enabling more flexibility for uncertain future predictions.

It is natural to ask whether any value of $0 \leq d_{\text{ext}} \leq d_{\min}$ is admissible. A stricter upper bound for d_{ext} can be established by considering the tightened bound on δ for the first prediction step, i.e., when $\epsilon = d_{\text{ext}}$.

Lemma 11.1.3

The bound

$$\delta_k \in [0, d_{\min}^2 + (1 - \gamma)h(\mathbf{x}_k^{[x,y]}, \mathbf{o}_k^{[x,y]}) - \epsilon^2]$$

is well defined if

$$0 \leq \epsilon < \sqrt{\gamma} d_{\min},$$

where "well defined" refers to the consistency of the bound, i.e.,

$$d_{\min}^2 + (1 - \gamma)h(\mathbf{x}_k^{[x,y]}, \mathbf{o}_k^{[x,y]}) - \epsilon^2 > 0. \quad (11.20)$$

□

Proof. The control barrier function at time k is

$$h(\mathbf{x}_k^{[x,y]}, \mathbf{o}_k^{[x,y]}) = \|\mathbf{x}_k^{[x,y]} - \mathbf{o}_k^{[x,y]}\|_2^2 - d_{\min}^2,$$

which may be positive, zero, or negative.

- If $h(\mathbf{x}_k^{[x,y]}, \mathbf{o}_k^{[x,y]}) \geq 0$, it is straightforward to verify that $\epsilon < \sqrt{\gamma} d_{\min}$ ensures (11.20) is satisfied.
- If $h(\mathbf{x}_k^{[x,y]}, \mathbf{o}_k^{[x,y]}) < 0$, then $h(\mathbf{x}_k^{[x,y]}, \mathbf{o}_k^{[x,y]}) \in [-d_{\min}^2, 0)$. In the worst case, when $h(\mathbf{x}_k^{[x,y]}, \mathbf{o}_k^{[x,y]}) = -d_{\min}^2$, substitution into (11.20) yields

$$d_{\min}^2 + (1 - \gamma)(-d_{\min}^2) - \epsilon^2 > 0,$$

which simplifies to

$$\gamma d_{\min}^2 - \epsilon^2 > 0 \quad \Rightarrow \quad \epsilon < \sqrt{\gamma} d_{\min}.$$

This completes the proof. $\square$

Therefore, the “extreme” distance d_{ext} must satisfy

$$d_{\text{ext}} \in [0, \sqrt{\gamma} d_{\min}),$$

with $\gamma \in (0, 1)$.

11.1.4 Dynamics of relaxation parameters

To ensure that δ converges to zero whenever constraint relaxation is unnecessary, the optimization problem (11.9) can be modified by introducing dynamics for the relaxation parameters. Specifically, the following dynamics are considered:

$$\delta(k+1) = A_\delta \delta(k) + B_\delta \xi(k), \quad (11.21)$$

where A_δ is designed as an asymptotically stable matrix so that $\delta \rightarrow 0$ if $\xi(k) = 0$, with $\xi(k)$ being the input at time instant k . The optimization problem is then reformulated as

Optimization Problem 11.1.2

$$\begin{aligned} \mathcal{U}_{k:k+N}^* = \operatorname{argmin}_{\mathbf{u}_k \in \mathcal{U}, \forall k} & J(\mathbf{x}_{k:k+N}, \mathbf{u}_{k:k+N}) + J_{tr}(\xi_k) \\ \text{s.t.} & \begin{cases} \mathbf{x}_{k+1} = f(\mathbf{x}_k, \mathbf{u}_k), \quad \forall k = 1, \dots, N \\ \mathbf{u}_k \in \mathcal{U}_k, \quad \forall k = 1, \dots, N-1 \\ CBC_{k+N|k}(\mathbf{x}_k, \mathbf{u}_k, \mathbf{o}_k)|_{\text{tr}_{i,k}} \geq -\delta_{i,k}, \quad \forall \text{tr}_{i,k} \in \mathcal{T}_k, \\ \delta_{k+1} = A_\delta \delta_k + B_\delta \xi_k \end{cases} \end{aligned} \quad (11.22)$$

Instead of $J_{tr}(\delta_k)$, the cost term now penalizes the input ξ_k :

$$J_{tr}(\xi_k) = \xi_k^T D' \xi_k, \quad (11.23)$$

where the weighting matrix D' remains the same as in the formulation based on δ_k . Notably, in this optimization problem, where the optimal sequence is computed along a prediction horizon of length N , the evolution of δ is explicitly considered only for one step. This is sufficient because δ already contains all δ_i associated with the horizon.

The eigenvalues of A_δ can be designed to depend on the probabilities associated with each trajectory, thereby influencing the temporal behavior of the corresponding relaxation variables δ_i . Let $\mathbf{p}$ denote the column vector containing the probabilities of all trajectories. One possible construction is

$$A_\delta = I_{n_c \times n_c} - \operatorname{diag}(\mathbf{p}), \quad (11.24)$$

where $I_{n_c \times n_c}$ denotes the identity matrix of appropriate dimension. With this design, relaxation parameters associated with higher-probability trajectories decay more rapidly toward zero, enforcing stronger safety guarantees for those scenarios. The matrix A_δ is always stable and becomes asymptotically stable provided that all trajectory probabilities are strictly positive.

However, the formulation in (11.24) is not time invariant if a prediction tool is introduced to update $\mathbf{p}$ based on past obstacle trajectories. In such cases, an alternative, time-invariant formulation may be adopted:

$$A_\delta = \zeta \cdot I_{n_c \times n_c}, \quad \zeta \in [0, 1). \quad (11.25)$$

The input matrix B_δ can also be designed in different ways:

- Easy formulation: $B_\delta = I_{n_c \times n_c}$,
- Probability-dependent formulation: $B_\delta = \operatorname{diag}\left(\frac{1}{\mathbf{p}}\right)$.

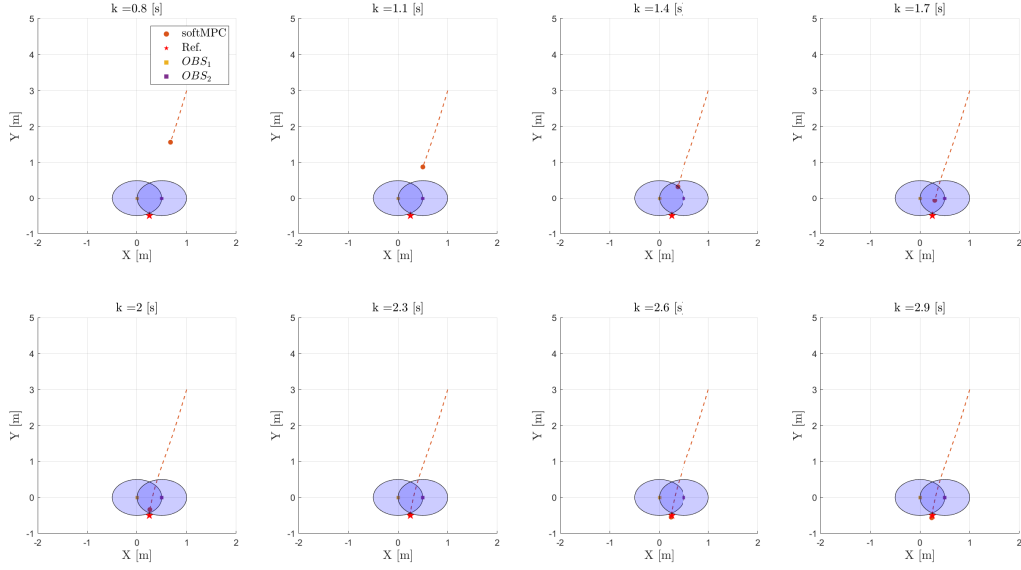


Figure 11.8: Snapshots of the simulation with static obstacles. Algorithm (11.9) (dynamics of δ are not included).

Under the probability-dependent formulation, trajectories with higher probabilities require larger control inputs ξ , reflecting the stricter safety enforcement imposed on the most likely outcomes. Importantly, B_δ must be structured to assign distinct control inputs to each trajectory, ensuring that the optimization framework can independently regulate the control effort ξ across the trajectory set while respecting their associated probabilities.

In the simulations, the following system matrices are employed for δ :

$$A_\delta = \zeta \cdot I_{n_c \times n_c}, \quad B_\delta = I_{n_c \times n_c}.$$

Since (asymptotically) stable dynamics are introduced for δ , the same cost function is preserved in the optimization problem, with the addition of the δ dynamics.

To further justify the necessity of introducing the relaxation variables δ , the behavior of the algorithm is compared with and without their inclusion. A particularly critical scenario arises when the target lies in close proximity to static obstacles. To illustrate this, a test case is considered in which two static obstacles are placed close to one another, while still leaving enough space for the robot to pass between them. This configuration poses a significant challenge for the planner, as it must ensure safety without adopting an overly conservative strategy. The comparison clearly highlights the trade-offs enabled by incorporating δ into the soft constraint formulation.

11.1.5 Including the attentiveness of pedestrians

In Section 11.1, the microscopic controller incorporating relaxation of the collision avoidance constraints was introduced, where the degree of relaxation depends solely on the probability associated with the predicted pedestrian trajectories.

However, the proposed framework is inherently flexible, allowing the integration of additional considerations related to pedestrian behavior into the path planning process. As anticipated by Remark 11.1.1, one particularly relevant aspect is the level of attention of the pedestrians. In real-world scenarios, when one's path intersects with that of an inattentive pedestrian, a larger deviation from the nominal trajectory is typically required compared to when the pedestrian is attentive and likely to cooperate. This observation motivates the inclusion of the attention level in the motion planning problem.

Similar concepts have been explored in the literature, particularly in industrial contexts, such as in [92], [93].

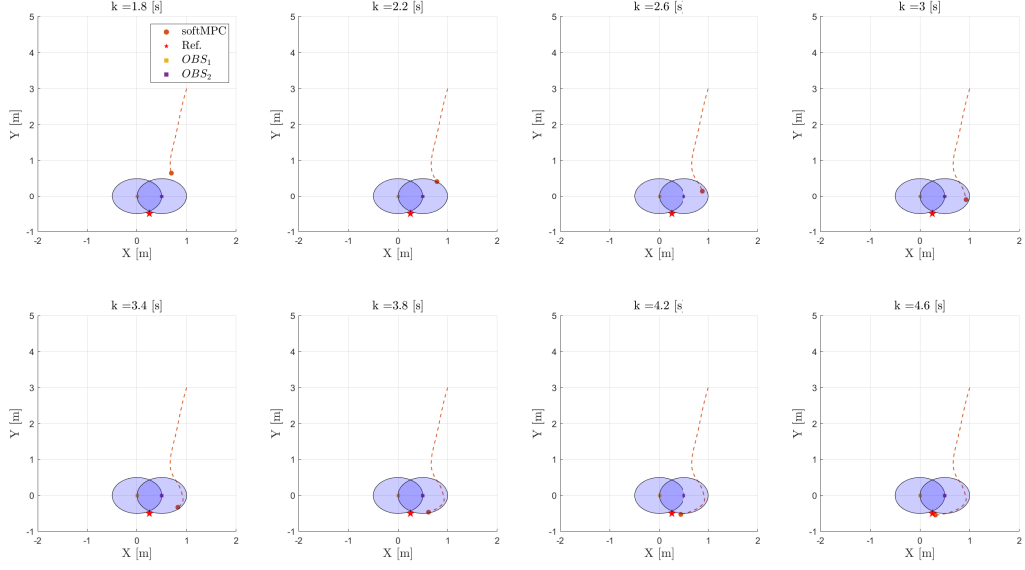


Figure 11.9: Snapshots of the simulation with static obstacles. Algorithm (11.22) (dynamics of δ are included).

This section first introduces a simple attention model used to estimate pedestrian attentiveness from available data and then presents the corresponding results. It is important to note that the proposed attention model is primarily intended for demonstration purposes, as more advanced approaches—often based on NNs—exist in the literature (e.g., [94], [95]). Nevertheless, the presented framework has the advantage of relying on easily obtainable pedestrian features, making it suitable for implementation even on autonomous robots equipped with basic sensors.

Starting from (11.9), the idea is to model how to shape the parameter c in (11.11) as a function of the attentiveness, denoted by η , of the obstacles.

The attentiveness coefficient η is modeled as a continuous variable in the interval $[0, 1]$, rather than as a binary on/off signal. Specifically,

- $\eta = 0$ indicates that the pedestrian is not attentive, and
- $\eta = 1$ indicates that the pedestrian is attentive.

To evaluate η , a simple geometric model is proposed, which exploits information about pedestrians that can be easily obtained using standard sensors, such as LiDARs. However, as stated in the introduction, this model can be replaced by more sophisticated approaches that, for instance, estimate pedestrians' pose and gaze to derive η .

The attentiveness level is evaluated based on:

- the robot heading ψ_k ,
- the pedestrian heading θ_k , and
- the Euclidean distance d_k between them.

The key intuition is that pedestrians tend to be more attentive when oriented toward the robot. This is, of course, a simplifying assumption, since a pedestrian could face the robot yet remain distracted—for example, by looking down at a smartphone. Such aspects could be incorporated into the attention model if appropriate sensors capable of detecting gaze are available. Based on this assumption, the relative heading is computed as

$$\Delta_k = \text{wrap}_{[-\pi, \pi]}(\theta_k - \psi_k),$$

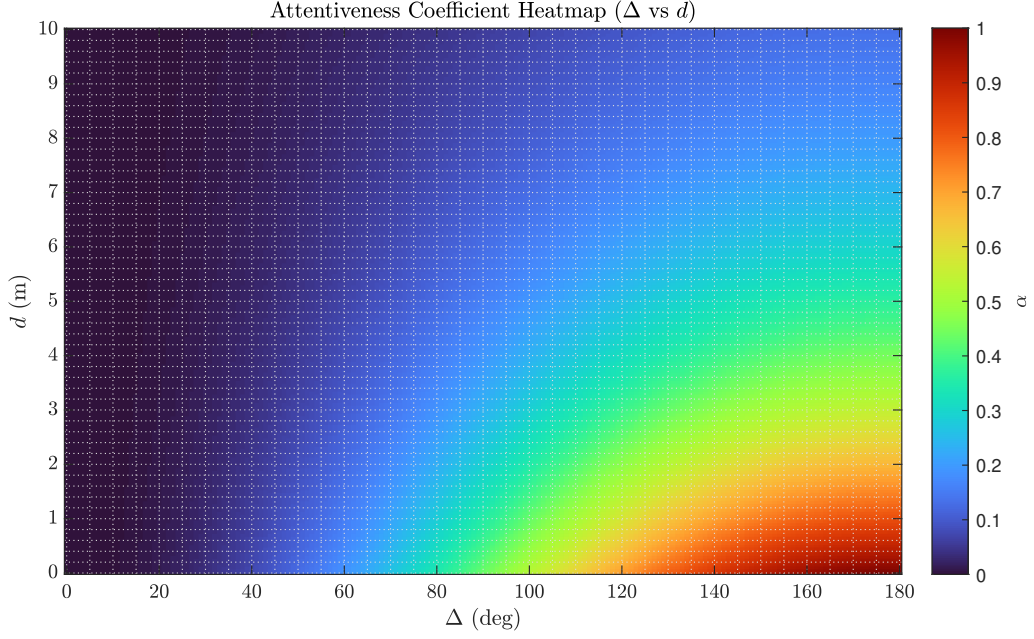


Figure 11.10: Attentiveness coefficient η over relative heading Δ and Euclidean distance d , with $\lambda = 0.2$ and $\beta = 1.2$.

where wrap indicates that the result is rescaled to fit within the interval $[-\pi, \pi]$. Thus, two agents moving in the same direction yield $\Delta_k \approx 0$, whereas two agents facing each other have $\Delta_k \approx \pi$.

Then, a smooth heading score is defined as

$$H(\Delta_k) = \frac{1 - \cos(\Delta_k)}{2}. \quad (11.26)$$

Trivially, $H(\Delta_k) \in [0, 1], \forall \Delta_k$, and when the pedestrian is not attentive, $H(\Delta_k) \approx 0$, whereas in the opposite case, when the pedestrian is fully attentive, $H(\Delta_k) \approx 1$. To adjust the sharpness of this relationship, the score can be raised to a power $\beta \geq 1$.

In addition to the heading difference, attentiveness also depends on the distance between the agents. Intuitively, when the agents are far apart, their trajectories are less likely to influence one another. Since attention typically decreases with distance, the distance score is modeled as an exponential decay:

$$G(d_k) = e^{-\lambda d_k}, \quad (11.27)$$

where $\lambda > 0$ controls the decay rate.

Finally, the overall attentiveness score is obtained by combining (11.26) and (11.27) as

$$\eta_k = (H(\Delta_k)^\beta) \cdot G(d_k). \quad (11.28)$$

Figure 11.10 illustrates a heat map of the attention coefficient η as a function of the relative heading Δ and Euclidean distance d (with $\lambda = 0.2$ and $\beta = 1.2$). The figure shows that attention increases with heading disparity—peaking near opposite headings ($\Delta \approx \pi$)—and decays exponentially with distance, yielding the highest values for small d and large Δ .

The performance of the proposed controller is evaluated in a *MATLAB*-based simulation environment. To initially assess the influence of pedestrian attentiveness on the control algorithm, a scenario involving a single pedestrian is considered.

In the first case (*Scenario A*), the robot and the pedestrian are aligned along the same path and oriented in the same direction. The pedestrian is positioned ahead of the robot

and remains unaware of its presence. This configuration represents a situation in which the pedestrian is inattentive.

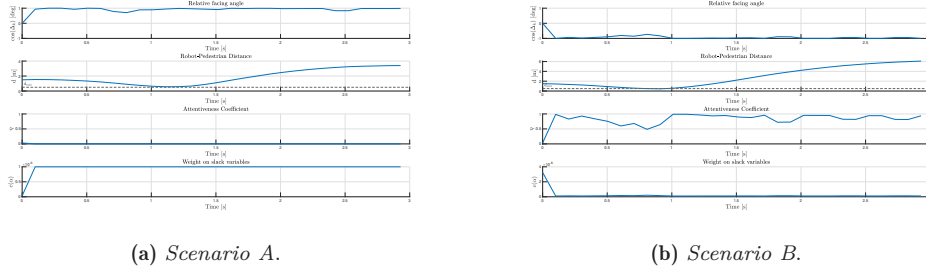


Figure 11.11: Relative facing angle, distance, attentiveness coefficient, and weight on the slack variables.

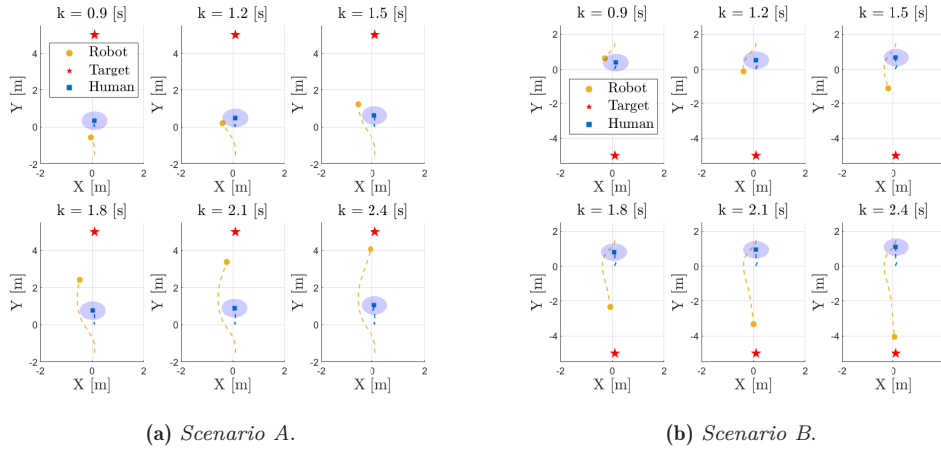


Figure 11.12: Snapshots of the simulation.

Figures 11.11a and 11.12a present partial results from testing the controller in the first scenario. Figure 11.11a shows, from top to bottom: the relative facing angle between the pedestrian and the robot, the distance between the two agents, the attentiveness coefficient η estimated for the pedestrian, and the weight applied to the slack variable $c(\eta)$. As expected, low attentiveness coefficients correspond to high weights on the slack variables. Figure 11.12a shows several snapshots of the simulation, particularly during the robot's approach toward the pedestrian. The blue circle represents the predicted region of possible pedestrian movement over the prediction horizon. It can be observed that, due to the low attentiveness level, the robot follows a relatively conservative trajectory.

To compare the behavior in the presence of an attentive pedestrian, an alternative scenario (*Scenario B*) is considered, in which the robot and the pedestrian face each other. In this case, the pedestrian is assumed to be aware of the robot's presence. The results of this simulation are reported in Figs. 11.11b and 11.12b, where the former shows the evolution of the attentiveness-related quantities, and the latter presents selected snapshots of the simulation.

Figures 11.12a and 11.12b can be compared to appreciate the adaptability of the proposed controller when interacting with pedestrians exhibiting different levels of attention. In *Scenario A*, the robot follows a highly conservative trajectory relative to the pedestrian's predicted positions over the horizon. Conversely, in *Scenario B*, the adaptive algorithm enables a less conservative behavior, allowing the robot to move closer to the predicted pedestrian motion region.

It is important to note that, in these simulations, the pedestrian is modeled as “blind,” meaning that it does not react to the robot's proximity. This simplification is adopted for

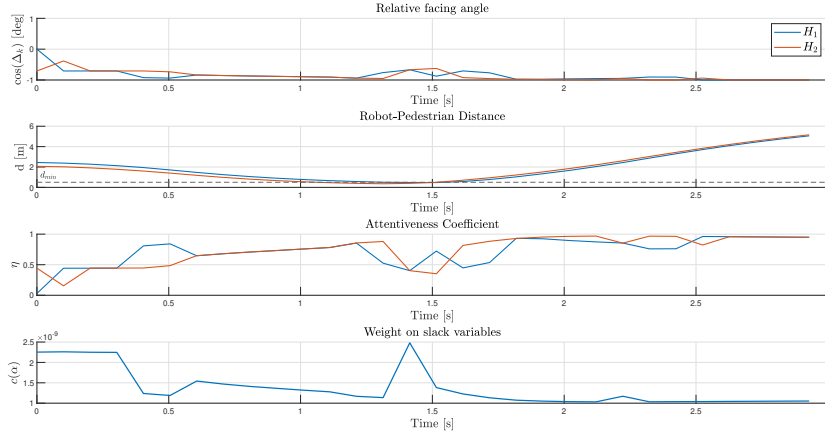


Figure 11.13: Relative facing angle, distance, attentiveness coefficient, and weight on the slack variables for two obstacles scenario.

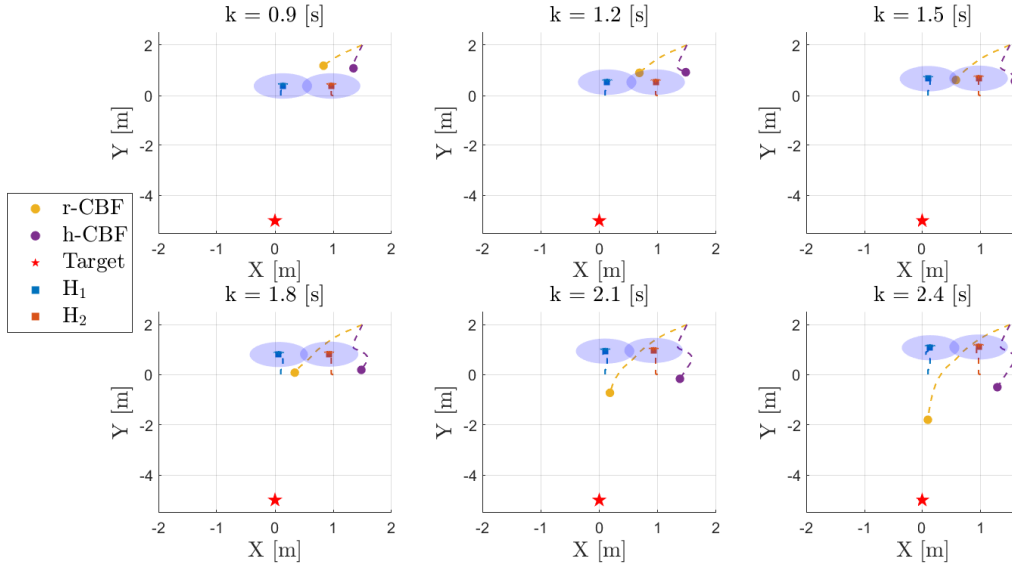


Figure 11.14: Snapshots of the simulation with two obstacles.

two main reasons. First, it provides a clear demonstration of how variations in the detected attention level influence the robot's control behavior. Second, accurately modeling human reactions is inherently difficult; an untuned action–reaction mechanism could lead to misleading conclusions about controller performance. Therefore, as previously mentioned, the pedestrian is assumed to move randomly according to a given probabilistic distribution.

To further assess the effectiveness of the proposed framework, it is compared with its conservative counterpart—an algorithm that employs CBFs without the possibility of relaxation, while still accounting for the predicted motion of obstacles over the prediction horizon. For simplicity, the proposed algorithm is referred to as *r-CBF*, whereas its conservative version is denoted as *h-CBF*. Figures 11.13 and 11.14 illustrate, respectively, the detected levels of attention of the two pedestrians in the environment during the *r-CBF* simulation, and the corresponding snapshots from both simulations.

A clear difference in behavior between the two algorithms can be observed. The conservative algorithm, *h-CBF*, maintains a large safety margin from the obstacles, as passing between them would violate the collision avoidance constraints, which in this

formulation cannot be relaxed—even when pedestrians are detected as attentive.

In contrast, the r -CBF algorithm allows the robot to move between the obstacles, owing to the relaxation of the collision avoidance constraints enabled by the introduction of slack variables.

One might question why r -CBF does not instead select a trajectory farther away from the obstacles, as h -CBF does. The explanation lies in the computation of the optimal path toward the target when obstacle constraints are not considered, and the corresponding deviation from this nominal trajectory. Intuitively, given the robot's dynamics, the nominal path would pass directly between the obstacles. Since such a path would lead to a collision, the algorithm enforces the collision avoidance constraint while allowing the maximum admissible relaxation (which, as noted, still prevents zero-distance scenarios). Consequently, the resulting trajectory passes safely between the obstacles while remaining as close as possible to the nominal path.

It should be noted that this behavior does not always occur. For example, if the robot were to start from a position further to the left, its trajectory would resemble that of the h -CBF, as it would still seek the shortest feasible path that guarantees safety.

11.1.6 Approximated Microscopic Planner

The optimization problem (11.22), which defines the robot's motion planner, integrates all components of the motion objective into a single formulation. While this unified approach enhances readability and conceptual clarity, it comes with a major drawback—the computational burden is significantly high.

Although this aspect remains an ongoing line of research, this section outlines a potential strategy to reduce computational time by means of an approximate optimization scheme.

It is straightforward to observe that the main source of computational complexity arises from the relaxation variables and the collision avoidance constraints. Therefore, the proposed idea is to decompose the original optimization problem into two sequential (cascade) stages. The first stage focuses on determining the optimal values of the slack variables for a given scenario, while the second stage employs these precomputed values within the collision avoidance constraints to compute the control input. A similar hierarchical approach is presented in [96].

Following this approach, the first optimization problem is formulated as

Optimization Problem 11.1.3 (Auxiliary problem)

$$\begin{aligned} \delta_{k:k+N}^* &= \operatorname{argmin}_{\delta_k \in \mathbb{R}_{\geq 0}, \forall k} J_{tr}(\xi_k) \\ \text{s.t.} \quad &\begin{cases} \mathbf{x}_{k+1} = f(\mathbf{x}_k, \mathbf{u}_k), \quad \forall k = 1, \dots, N, \\ \mathbf{u}_k \in \mathcal{U}_k, \quad \forall k = 1, \dots, N-1, \\ CBC_{k+N|k}(\mathbf{x}_k, \mathbf{u}_k, \mathbf{o}_k)|_{\text{tr}_{i,k}} \geq -\delta_{i,k}, \quad \forall \text{tr}_{i,k} \in \mathcal{T}_k, \\ \delta_{k+1} = A_\delta \delta_k + B_\delta \xi_k. \end{cases} \end{aligned} \quad (11.29)$$

The result of this optimization provides the optimal values of the slack variables δ^* for all prediction instants, given the scenario generated from the current robot position and the observed positions of the obstacles. Notably, the objective function depends exclusively on the slack variables, specifically on the control effort required to modify their asymptotic dynamics.

The obtained optimal slack variables are then employed in the second optimization problem, which determines the actual control input for the robot:

Optimization Problem 11.1.4

$$\begin{aligned}
& \mathcal{U}_{k:k+N}^* = \operatorname{argmin}_{\mathbf{u}_k \in \mathcal{U}, \forall k} J(\mathbf{x}_{k:k+N}, \mathbf{u}_{k:k+N}) \\
& \text{s.t.} \quad \begin{cases} \mathbf{x}_{k+1} = f(\mathbf{x}_k, \mathbf{u}_k), \quad \forall k = 1, \dots, N, \\ \mathbf{u}_k \in \mathcal{U}_k, \quad \forall k = 1, \dots, N-1, \\ CBC_{k+N|k}(\mathbf{x}_k, \mathbf{u}_k, \mathbf{o}_k) | \operatorname{tr}_{i,k} \geq -\delta_{i,k}^*, \quad \forall \operatorname{tr}_{i,k} \in \mathcal{T}_k. \end{cases}
\end{aligned} \tag{11.30}$$

By splitting the original problem, an additional advantage is gained, as it provides greater control over the values of the relaxation variables.

A preliminary evaluation of the proposed approach is conducted in a simulated environment with two obstacles exhibiting near-random motion. The approach shows promising results in terms of collision avoidance and path efficiency, as depicted in the snapshots of Figure 11.15.

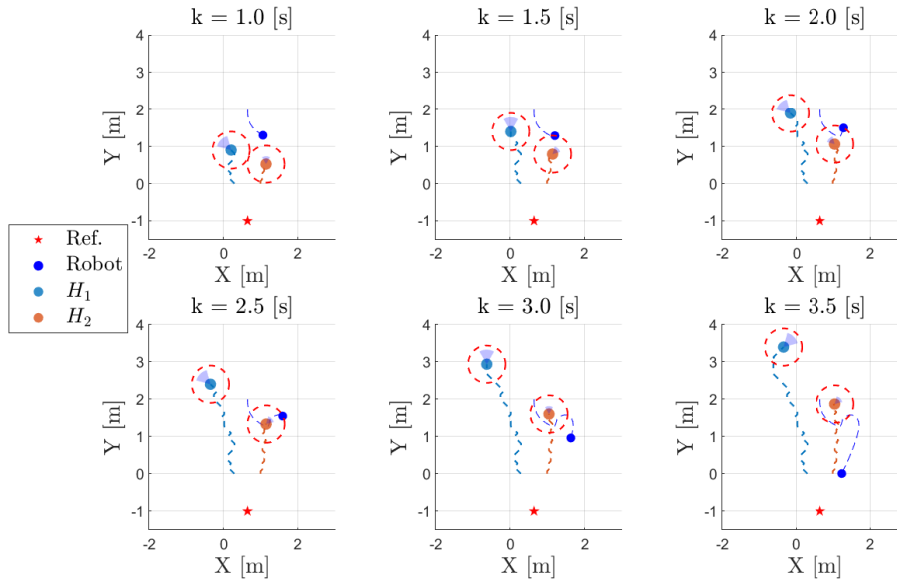


Figure 11.15: Snapshots of the simulation considering the two optimization problems cascade.

Despite these initial results, the ultimate goal is to accelerate the control problem. To achieve this, the auxiliary problem (11.29), which is used to determine the optimal values of the slack variables, is intended to be replaced by a NN. This approximation is expected to reduce the computational time, as discussed in [96], and it will also enable the development of a feasibility proof. These aspects are the focus of ongoing work.

11.2 Macroscopic Motion Planner: how to move through a crowd

The planner described in the previous section 11.1 is devoted to collision avoidance with surrounding pedestrians. However, in highly crowded environments, such a microscopic planner presents limitations, as it merely avoids obstacles without analyzing the situation in a dynamic manner.

Furthermore, in unstructured and dense environments, tracking the individual trajectories of all pedestrians is both impractical and computationally expensive [97].

To address this, macroscopic features of pedestrian groups are employed to describe their collective behavior. Unlike certain approaches in the literature that rely on fluid dynamics, particle-based models, or potential fields [48], [98], [99], [100]—which often require strong assumptions such as knowledge of pedestrians' final goals or cost functions—the proposed method adopts a more straightforward and assumption-light strategy.

To enhance scalability, the sensing area is discretized, and macroscopic quantities inferred from pedestrian groups are exploited to characterize their collective behavior.

Definition 11.2.1 (Discretized sensing area Fig.11.16)

The sensing area refers to the portion of the environment acquired either through the autonomous agent's onboard sensors or via external sensing systems. The discretized sensing area $\mathcal{S}$ is obtained by overlaying a fixed grid onto this sensed region, thereby dividing it into cells of equal size. The set of cells is indicated as $\mathcal{C} = \{C_i, i = 1, \dots, N_{cells}\}$, so that $\mathcal{S} = \bigcup_{C_i \in \mathcal{C}} C_i$ and $C_i \cap C_j = \emptyset, \forall C_i, C_j \in \mathcal{C}, i \neq j$.

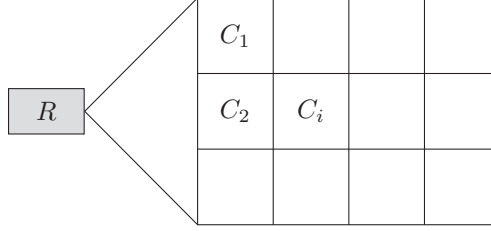


Figure 11.16: Discretized sensing area.

Definition 11.2.2 (Pedestrian group)

Given a discretized sensing area $\mathcal{S}$, a pedestrian group A_i is defined as the set of pedestrians that are located inside a certain cell $C_i \in \mathcal{C}$.

$$A_i = \{a_j \in \mathcal{A} \mid a_j \in C_i\}$$

Therefore, it is defined spatially, rather than based on social or behavioral aggregation.

The assumption regarding the definition of pedestrian groups is primarily introduced to simplify the dissertation and to avoid the challenging problem of identifying a leader or a center of attraction, which is particularly difficult in crowded scenarios.

An alternative to the straightforward squared cell sampling approach is the use of Voronoi cells. However, a major limitation of this method arises from the dynamic nature of Voronoi partitions, which evolve over time according to the pedestrians' trajectories. As a result, it becomes challenging to assign consistent scores to the cells, since their shape and structure are not fixed.

For each macroscopic feature, a distinct spatial layer of the environment is constructed. Then, by combining these layers with appropriately chosen weights, a score is assigned to each cell within the discretized sensing area. This scoring procedure enables the identification of the most favorable regions for the autonomous agent to traverse.

11.2.1 Macroscopic Features

As macroscopic features, three distinct quantities are considered, collectively providing an indication of the state of each cell.

The first is the occupancy level of a single cell C_i , represented by the density D_i , defined as

$$D_i = \frac{|A_i|}{\text{size of } C_i}, \quad (11.31)$$

where $|A_i|$ denotes the number of pedestrians in cell C_i . However, density alone provides an insufficient description of the cell, particularly regarding pedestrian dynamics. Therefore, for each cell C_i , the average velocity K_i and the entropy H_i are also considered.

The average velocity is computed as

$$K_i = \frac{\sum_{a_j \in A_i} v_j}{|A_i|}. \quad (11.32)$$

Entropy is introduced to quantify the variability in pedestrian motion directions within the same cell. In other words, it serves as a measure of disorder or chaos. The Shannon entropy formula $H = -\sum_i p_i \log p_i$ is used, where p_i represents a probability. Since exact probabilities are not available, observed frequencies are used instead. Given that pedestrians may move in different directions, a set of direction clusters $\Theta_j, j = 1, \dots, N_{\text{clusters}}$, is defined, where N_{clusters} denotes the number of motion-direction clusters. Each cluster Θ_j corresponds to a subinterval of $[-\pi, \pi]$, representing a range of possible directions.

The equivalent of the probability p_j in the Shannon formula is computed as

$$p_j = \frac{\text{Number of pedestrians in cluster } \Theta_j}{|A_i|}, \quad (11.33)$$

where $|A_i|$ is the total number of pedestrians in the i^{th} cell. The entropy E_i of cell C_i is then given by

$$E_i = - \sum_{j=1}^{N_{\text{clusters}}} p_j \log p_j, \quad (11.34)$$

with p_j defined as in (11.33).

An example of these macroscopic feature layers, together with the pedestrians' trajectories, is displayed in Fig.11.17.

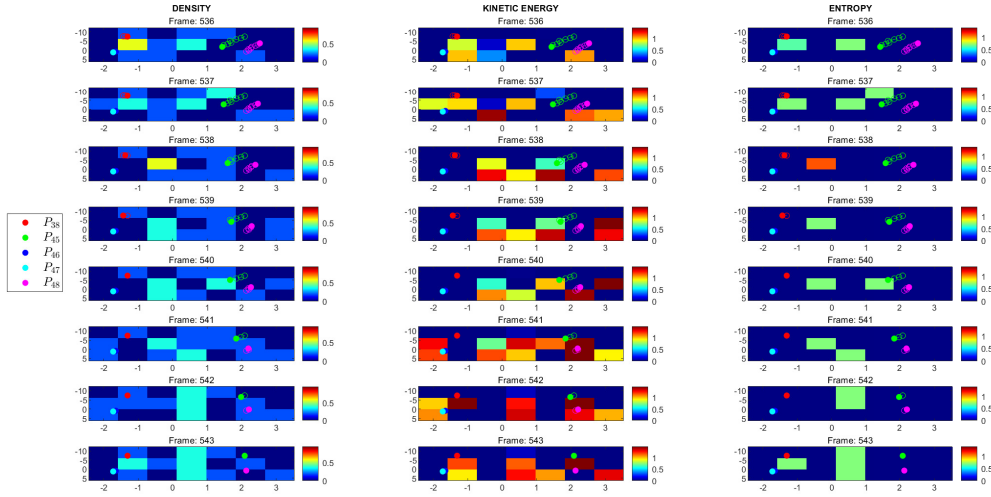


Figure 11.17: Density, kinetic energy and entropy (32 clusters) layers for group description - hotel scene, with 5 pedestrians trajectories included.

11.2.2 Cells Score Function

The three macroscopic features defined in (11.31), (11.32), and (11.34) are intended to characterize the state of each cell in the environment. These features are linearly combined into a score function, which yields a scalar value (*score*) representing the overall condition of the cell.

The score function is time-varying, as the feature values evolve with new sensor detections. It is defined as

$$f_{C_i,k}(D_{i,k}, K_{i,k}, E_{i,k}) = \alpha_D D_{i,k} + \alpha_K K_{i,k} + \alpha_E E_{i,k} + b = s_{i,k}, \quad (11.35)$$

where α_D, α_K , and α_E denote the weights assigned to each macroscopic feature, and b is a bias term. All features should be min-max normalized to ensure comparable scales, so that they contribute equally to the score function when the weights are set to one.

Interestingly, an investigation on the values of the parameters of the score function (11.35) is conducted, exploiting a well-known datasets [25].

To estimate the parameters α_i , $i = \{D, K, E\}$, of the score function (11.35), a simple neural network is employed. The mentioned datasets (that contain two scenarios, the university and the hotel) are used for training and validation.

The environment is discretized into a grid of cells, each assigned a unique index. For each pedestrian and each time frame, cells are labeled with a binary indicator: a value of 0 is assigned to the cell currently occupied by the pedestrian (i.e., the cell through which the pedestrian's trajectory passes at that frame), and a value of 1 is assigned to all other cells. The values of density, kinetic energy, and entropy are then computed for each cell.

The loss function is defined as the Binary Cross-Entropy (BCE) loss:

$$BCE(y, \hat{y}) = -[y \log(\hat{y}) + (1 - y) \log(1 - \hat{y})], \quad (11.36)$$

where y is the true label and $\hat{y}$ is the predicted output:

$$\hat{y} = \sigma(f_{C_i,k}) = \frac{1}{1 + \exp(-f_{C_i,k})}. \quad (11.37)$$

The training is performed for a fixed number of 1500 epochs. The results of a single test are summarized in Table 11.1, where the column labeled "S. ID" indicates whether the dataset includes a label identifying the scenario. The training set corresponds to 80% of the dataset, while the validation set corresponds to the remaining 20%.

# T.	T. Set	V. Set	S. ID	α_D	α_K	α_E	b
1a	Hotel	Hotel	✗	4.31	0.16	-1.24	-2.63
2a	Univ.	Univ.	✗	6.08	6.82	-2.22	-3.44
3a	Hotel + Univ.	Hotel + Univ.	✗	5.54	4.60	-1.58	-2.61
4a	Hotel + Univ.	Hotel + Univ.	✓	5.16	4.73	-1.38	-2.49

Table 11.1: Parameter estimation of score function $f_{C_i,k}$ (single test)

An alternative score function is also considered, incorporating the preferred motion direction of pedestrians to favor cells aligned with their movement:

$$f_{C_i,k}^\theta(\mathcal{C}_{j,k}, D_{i,k}, K_{i,k}, E_{i,k}) = \alpha_D D_{i,k} + \alpha_K K_{i,k} + \alpha_E E_{i,k} - \alpha_\theta \cos(\tilde{\theta}) + b, \quad (11.38)$$

where $\tilde{\theta}$ denotes the angular difference between the current motion direction of the pedestrian and the direction from the pedestrian's current position to the center of the i^{th} cell.

The same training and validation tests are performed, with the results reported in Table 11.2.

# T.	T. Set	V. Set	S. ID	α_D	α_K	α_E	α_θ	b
1b	Hotel	Hotel	✗	4.63	-0.02	-1.46	-0.05	-2.68
2b	Univ.	Univ.	✗	6.25	6.90	-2.32	0.17	-3.49
3b	H. + U.	H. + U.	✗	6.47	3.86	-2.11	-2.11	-3.20
4b	H. + U.	H. + U.	✓	6.42	3.91	-2.08	-2.08	-3.10

Table 11.2: Parameter estimation of score function $f_{C_i,k}^\theta$ (single test)

From Tables 11.1 and 11.2, it is evident that, except for Tests 2a and 2b, the most influential feature for cell selection is the density, as indicated by the consistently largest value of α_D .

For a more robust evaluation, the training and validation process is repeated $N = 100$ times, with the mean and variance of the learned parameters shown in Figs. 11.18 and 11.19.

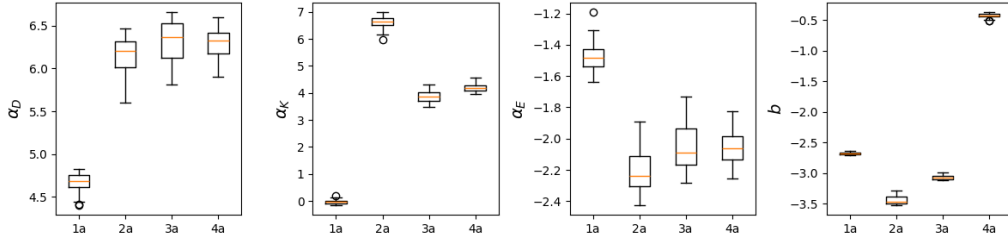


Figure 11.18: Distribution of parameters for score function (11.35)

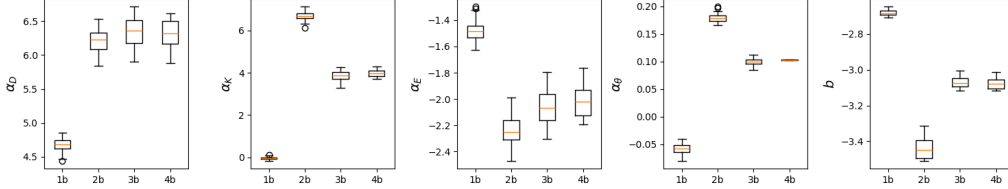


Figure 11.19: Distribution of parameters for score function (11.38)

It is noteworthy that the term associated with the current pedestrian direction has a negligible effect on the next cell decision. This behavior can be attributed to the structure of the environments in the employed datasets, which typically feature two almost fixed zones for the start and end of movements. Consequently, pedestrian flows generally exhibit two predominant, opposing directions.

Moreover, the parameters α_D , α_K , α_E , and b can also be regarded as tuning parameters. In the previous analysis, these values were estimated from the available datasets. However, they can be adjusted to customize the motion of the autonomous agent.

11.2.3 Planning the Macroscopic path

In the previous section, the design of the score function was discussed, which is used to identify the regions through which the autonomous agent should navigate to reach a given static target. Ideally, the agent should select the path that minimizes the cumulative score. However, this is only feasible if the scores of all the cells are known—in other words, if a bird’s-eye-view camera is available to collect data from the entire scenario. This approach is not realistic unless the agent has access to an external camera. A more practical path planning algorithm should instead assume access only to the data in the close neighborhood of the agent’s current position.

Therefore, the algorithm for macroscopic path derivation is presented, together with a comparison between two scenarios: one in which a bird’s-eye-view camera is available, and one in which only onboard sensors (such as Lidars) are available, resulting in a limited field of view.

In both approaches, the robot is intended to navigate through the cells by choosing the most suitable ones in terms of score, which measures the occupancy and chaos level. However, the robot also has a final target to reach, and navigation through low-score cells alone could potentially lead it far from the objective. To address this, a cost function has been formulated that accounts for both the distance to the target and the score of the cell. This cost function is defined as follows:

$$J_{macro}(C_{i,k}) = \beta_1 \|c_i - c_T\|_2^2 + \beta_2 s_{i,k}, \quad (11.39)$$

where c_i is the center of C_i , c_T is the center of the cell containing the final target, $s_{i,k}$ is the score of C_i calculated with (11.35), and β_1, β_2 are the weights assigned to the squared target distance and score, respectively.

To obtain the most suitable path through cells $\mathbf{c}_k$, a Dynamic Algorithm is designed that identifies the path with the minimum cumulative cost, where the cost of a single

cell is defined as in (11.39), starting from the robot's current position and ending at the target. The macroscopic path $\mathbf{c}_k$ is defined as an ordered sequence of visiting points, where each point corresponds to the center of a cell that the robot is required to traverse.

To limit the complexity of the algorithm and reduce the number of possible trajectories to evaluate, a heuristic is employed that restricts the robot's movement toward the goal, preventing backward motion. This constraint effectively limits the number of cells that can be traversed at each step, reducing the overall number of feasible trajectories and enhancing computational efficiency.

The structure of the algorithm remains the same in both approaches. However, the weights β_1 and β_2 are adjusted depending on the available information. In the sensor-view scenario, the scores of cells beyond the sensing radius are unknown. This limitation is incorporated into the algorithm by setting $\beta_2 = 0$ for all cells located outside the sensing radius relative to the robot's current position.

Hence, the different views can be resumed as follows

- **Bird-view full path planning:** the full path to the target is computed at each step using dynamic programming, starting from the target and proceeding backward to the current position;
- **Sensor-based local planning:** in this case, only a limited local view is available (e.g., two rows of cells ahead), defined by the sensing radius r_{sensor} . After evaluating the scores of the visible cells, the robot is assumed to continue moving straight toward the target.

Figures 11.20 and 11.21 illustrate the difference between the two approaches. They separately show the contribution of the cell score, computed using (11.35), and the target score, obtained by penalizing the quadratic distance to the target, i.e., the first term of the cost function in (11.39).

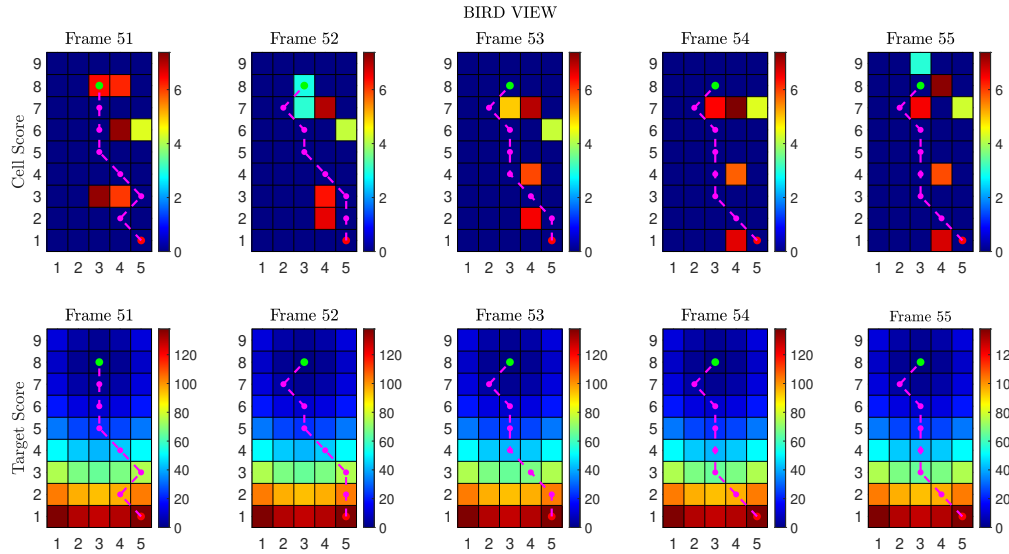


Figure 11.20: Score and target functions of the macro path with bird-view.

Notably, the structure of the cost function (11.39), together with its ability to incorporate different types of perception, makes the framework adaptable to a variety of scenarios and setups. For instance, a robot operating in a highly structured environment may have access to external cameras, thereby obtaining complete knowledge of the environmental conditions. Conversely, in the absence of such external sensors, the robot must rely solely on its onboard perception. Interestingly, it can be observed that, in certain cases, the reduced availability of information does not necessarily result in abrupt changes in the robot's trajectory.

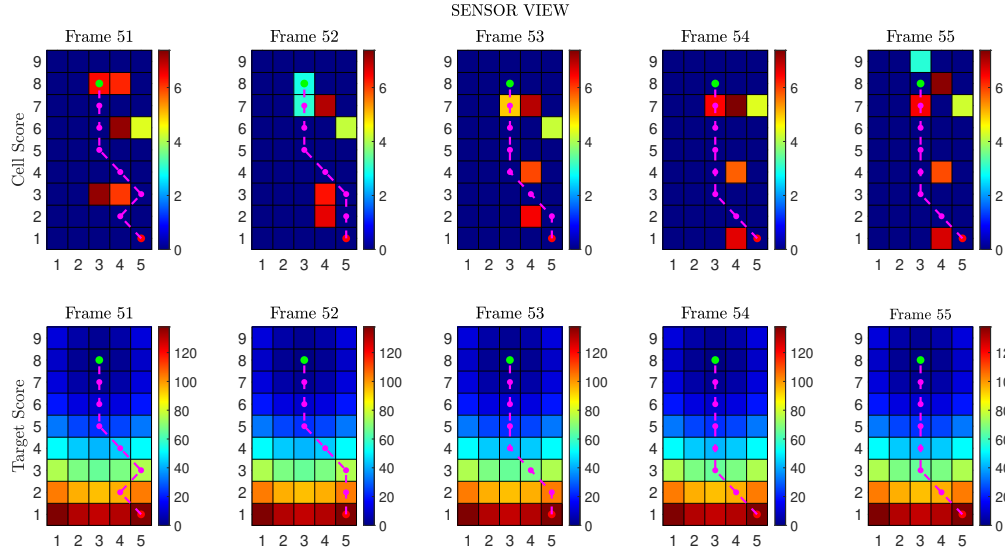


Figure 11.21: Score and target functions of the macro path with sensor-view.

This will be demonstrated in the following section, where the macroscopic controller is integrated into a broader control structure together with the microscopic controller described in Section 11.1.

11.3 Hierarchical double-layer Motion Planner

In this section, the overall control framework is presented. The approach combines the macroscopic controller described in Section 11.2 and the microscopic controller introduced in Section 11.1 (so considering a single optimization problem), arranged in a cascade structure, as shown in Fig.11.22.

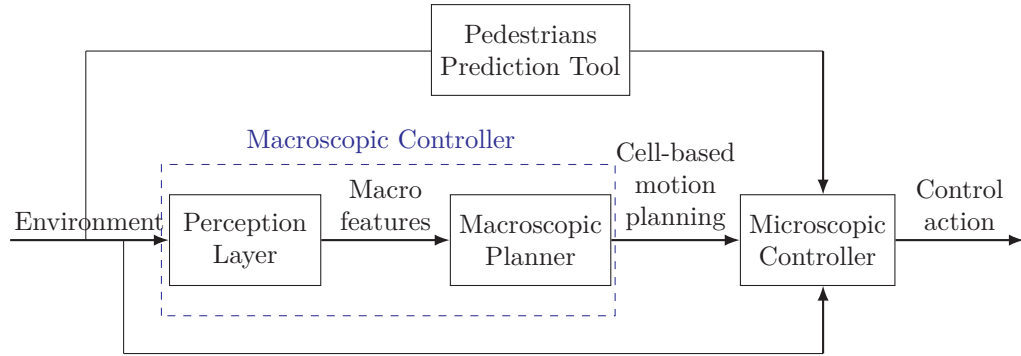


Figure 11.22: Hierarchical Controller block scheme.

Considering the overall framework, the cascade structure becomes evident: the output of the macroscopic controller, highlighted by the blue dashed rectangle, serves as one of the inputs to the microscopic controller.

The macroscopic controller is divided into two main components: the *Perception Layer*, responsible for evaluating the scenario in terms of the score of individual cells, as discussed in Sections 11.2.1 and 11.2.2; and the *Macroscopic Planner*, which is responsible for deriving the macroscopic path, as described in Section 11.2.3.

The input to the macroscopic controller is the environment, which is first analyzed by the Perception Layer. This layer extracts the macroscopic features needed by the Macroscopic Planner, which then designs a suitable path through the cells. Finally,

this path, together with a representation of (part of) the environment, is passed to the microscopic controller, which computes the actual control actions to be applied to the robot's plant.

There are different strategies for incorporating the cell-based motion planning input into the microscopic controller. One possibility is to encode this information as soft constraints, for example by defining a flexible set of reachable states. However, this approach is computationally expensive, as it significantly increases the number of constraints in the optimization problem.

An alternative strategy is to incorporate the information directly into the optimization problem as a path-following objective, which avoids the additional computational burden while still enforcing consistency with the macroscopic planner. This approach consists of representing the cell-based motion path as an ordered sequence of cell centers, which are then interpreted as waypoints to be visited by the robot. The problem of following this suggested path, which should be regarded as a soft constraint—since the robot must retain the ability to deviate from it in case of collision—is incorporated into the optimization problem (11.22) by simply adding an additional term in the cost function.

This term depends on the *Cross-Tracking Error* (CTE), which corresponds to the distance from the current robot position to the closest path segment, where a segment is defined as the line between two consecutive waypoints. Then, the new optimization problem comes as follows:

Optimization Problem 11.3.1

$$\begin{aligned} \mathcal{U}_{k:k+N}^* = \operatorname{argmin}_{\mathbf{u}_k \in \mathcal{U}, \forall k} & J(\mathbf{x}_{k:k+N}, \mathbf{u}_{k:k+N}) + J_{tr}(\boldsymbol{\xi}_k) + J_{CTE}(\mathbf{x}_{k:k+N}, \mathbf{c}_k) \\ \text{s.t.} \quad & \begin{cases} \mathbf{x}_{k+1} = f(\mathbf{x}_k, \mathbf{u}_k), \quad \forall k = 1, \dots, N \\ CBC_{k+N|k}(\mathbf{x}_k, \mathbf{u}_k, \mathbf{o}_k)|_{\text{tr}_{i,k}} \geq -\delta_{i,k}, \quad \forall \text{tr}_{i,k} \in \mathcal{T}_k, \\ \delta_{k+1} = A_\delta \delta_k + B_\delta \boldsymbol{\xi}_k \end{cases} \end{aligned} \quad (11.40)$$

where

$$J_{CTE}(\mathbf{x}_{k:k+N}, \mathbf{c}_k) = e_{CT}(\mathbf{x}_{k:k+N}, \mathbf{c}_k)^T \Lambda_{CT} e_{CT}(\mathbf{x}_{k:k+N}, \mathbf{c}_k). \quad (11.41)$$

Here, the vector $\mathbf{c}_k$ contains the output of the macroscopic planner, i.e., the centers of the cells that should be visited. The term $e_{CT}(\mathbf{x}_{k:k+N}, \mathbf{c}_k)$ denotes the vector of cross-track errors for each prediction step, and $\Lambda_{CT} \in \mathbb{R}^{N \times N}$, $\Lambda_{CT} = \lambda_{CT} \cdot I_{N \times N}$ is a weighting matrix.

11.4 Simulations and Results

The proposed macro–micro control framework described in Section 11.3 is evaluated using the ETH dataset [25] (Hotel scenario). In particular, the robot's trajectory is planned within the pre-recorded pedestrian data.

This dataset provides, for each recorded frame, the position and velocity of every pedestrian. To better approximate a realistic sensing setup, it is further assumed that the robot can detect obstacles only within a limited radius. It is acknowledged that relying on a pre-recorded dataset introduces certain limitations, such as the absence of unexpected pop-up obstacles and the fact that pedestrians do not react to the robot. Nevertheless, in the absence of a real experimental setup, this well-established dataset is employed.

In the following simulations, as discussed throughout this chapter, a distribution over the possible heading changes of pedestrians is considered in order to relax some of the assumptions commonly adopted in the state-of-the-art. In the literature, predicting the future direction of pedestrian motion is a non-trivial task. Consequently, most works assume either knowledge of the pedestrian's final goal or that the current direction of motion remains unchanged.

Since no probabilistic prediction tool is currently employed, a support set Θ for θ is instead defined over time with fixed cardinality. Specifically, $|\Theta| = 3$, and the elements of

Θ are defined as

$$\Theta_k = \left\{ \theta_{k-1} - \frac{\pi}{4}, \theta_{k-1}, \theta_{k-1} + \frac{\pi}{4} \right\}, \quad (11.42)$$

where θ_{k-1} denotes the pedestrian's recorded direction of motion.

The probability distribution over Θ is set as

$$\mathbb{P}(\Theta) = \{0.25, 0.5, 0.25\}, \quad (11.43)$$

which assigns the highest likelihood to the current direction of motion while still accounting for the possibility of left and right turns. It is important to recall that this distribution is manually designed in the present framework, but it could be readily replaced by the output of a prediction tool, for example a NN.

The simulations were conducted over a duration of T time steps, with a prediction horizon of N steps, by solving the optimization problem (11.40). Furthermore, the control inputs are constrained within the bounded set

$$\mathcal{U} = [\underline{a}, \bar{a}]^T,$$

where $\underline{a}$ and $\bar{a}$ denote the lower and upper bounds on the admissible accelerations, respectively. The parameters used are reported in Table F.1 in Appendix F.

In the following, the simulation results obtained with both the bird-view and sensor-view approaches are presented and compared.

Figures 11.23, 11.24, 11.25, 11.26, and 11.27 illustrate, respectively: the evolution of the macroscopic path throughout the simulation, the actual trajectory of the robot in the discretized environment, snapshots highlighting the robot's behavior near obstacles, the robot's speed profile, and its accelerations. All results correspond to the bird-view strategy. A more detailed evolution of the macroscopic path strategy is provided in Appendix F, see Fig. F.1.

For comparison, Figures 11.28, 11.29, 11.30, 11.31, and 11.32 present the same set of results obtained with the sensor-view approach. Similarly, a more detailed evolution of the macroscopic path strategy is reported in Appendix F, see Fig. F.2.

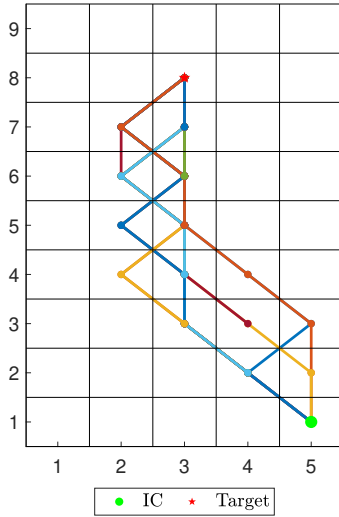


Figure 11.23: Output of the macro planner considering bird-view approach.

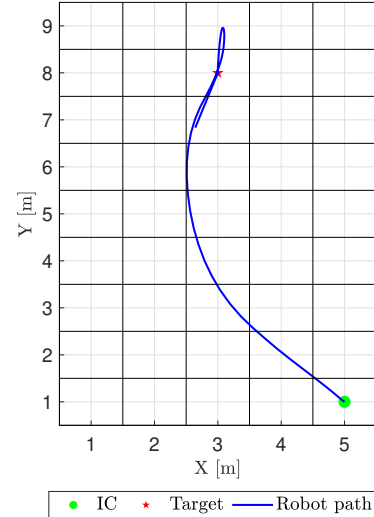


Figure 11.24: Controlled trajectory of the robot considering bird-view approach.

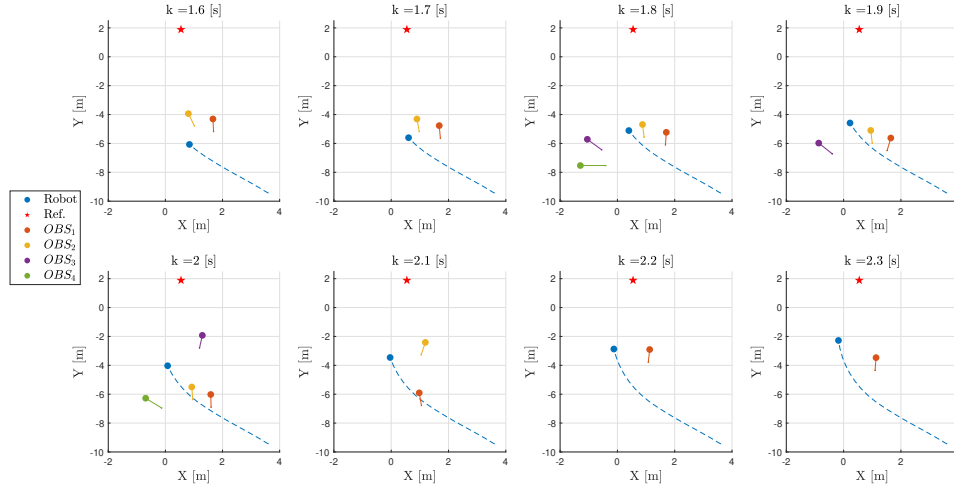


Figure 11.25: Snapshots of simulation considering bird-view approach.

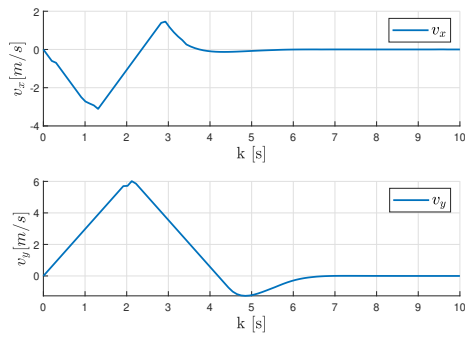


Figure 11.26: Robot's speed considering bird-view approach.

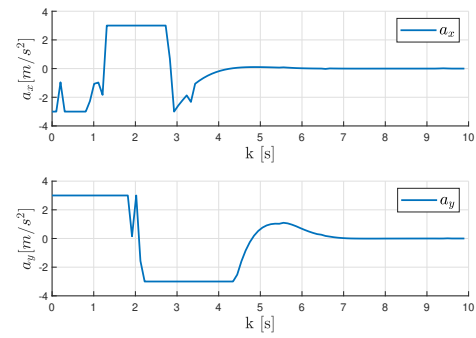


Figure 11.27: Control inputs considering bird-view approach.

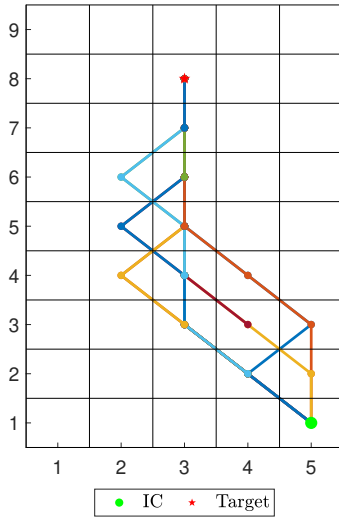


Figure 11.28: Output of the macro planner considering the sensor-view approach.

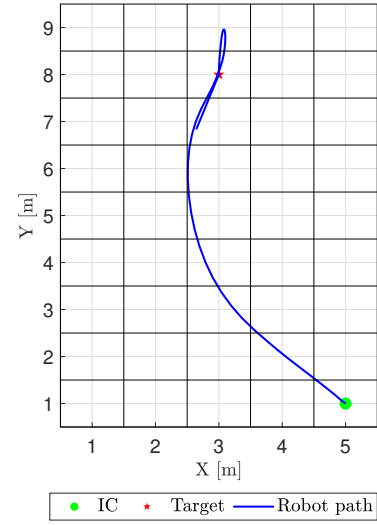


Figure 11.29: Controlled trajectory of the robot considering the sensor-view approach.

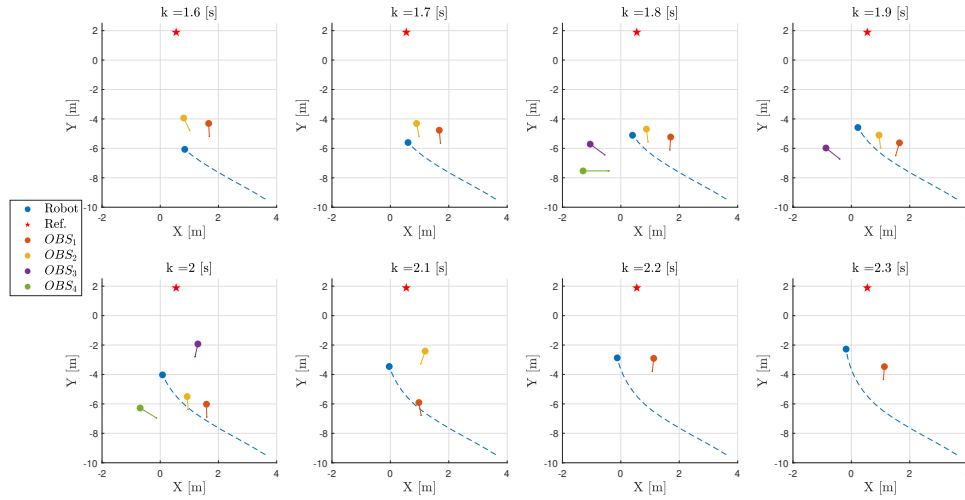


Figure 11.30: Snapshots of simulation considering the sensor-view approach.

11.5 Discussion

In this chapter, a motion planner for controlling an autonomous robot navigating a pedestrian environment has been presented. The discussion begins with a microscopic approach, describing a planner aimed at reaching the final target while avoiding collisions. The main contribution lies in the design of flexible collision-avoidance constraints that can be relaxed based on predictions of obstacle motion.

Next, a macroscopic approach is introduced to identify the most suitable regions in the environment for navigation, taking into account factors such as density, velocity, and entropy.

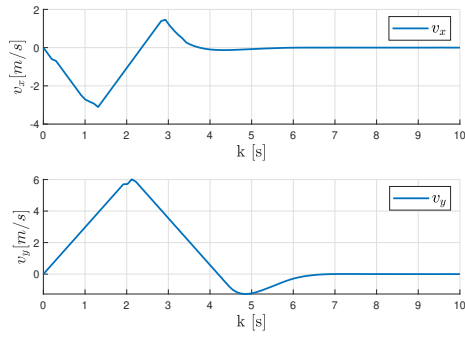


Figure 11.31: Robot's speed considering the sensor-view approach.

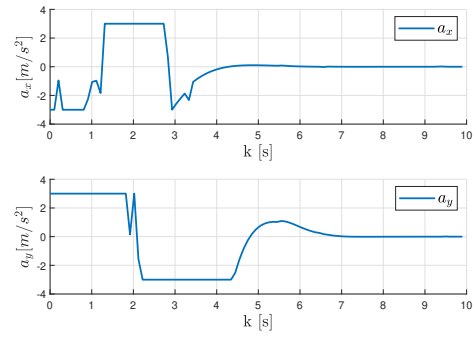


Figure 11.32: Control inputs considering the sensor-view approach.

Finally, these two components are combined into a unified control framework for autonomous navigation and tested using a publicly available dataset.

Ongoing work focuses on developing a faster microscopic algorithm by splitting the optimization problem into two cascade problems and approximating one of them.

Conclusions

Chapter 12

Conclusions

This chapter presents the conclusions of this dissertation and provides an outlook on possible directions for future research.

The central focus of this dissertation is the modeling and prediction of interactions between agents, as well as the exploration of strategies to exploit or steer these interactions toward a more general optimization of the overall scenario.

In Chapters 3 and 4, models describing agent interactions within a social network, based on a simplified Markov chain framework, have been presented.

Specifically, Chapter 3 introduces models in which interactions are represented as social forces acting on the transition matrix of the Markov chain. These forces capture attraction and repulsion effects through rate modifications. The analysis focuses primarily on the repulsive model, which can be characterized as either direct or indirect repulsion. A comparison between these two approaches is conducted to determine their equivalence and identify the limitations that may guide the choice of one model over the other.

Chapter 4 extends this framework to model the interaction between agents and their surrounding environment. A simple control scheme based on MPC is proposed and discussed. This modeling approach demonstrates how intuitive frameworks can be employed to represent everyday phenomena—such as the influence of marketing campaigns or social media advertisements on individuals.

In Chapters 6 and 7, these frameworks are specialized for applications in traffic modeling, with a particular focus on highway scenarios.

Chapter 6 introduces two frameworks based on Markov chain models. The first adopts a macroscopic perspective, representing the highway dynamics through aggregated quantities such as traffic density and flow. The second framework, instead, takes a microscopic viewpoint, modeling how individual human drivers respond to the actions of surrounding vehicles.

Chapter 7 then integrates the previously introduced microscopic prediction model into a control loop for an AV, which operates as a form of safety car. Leveraging the predictive capabilities of the microscopic model, the AV can anticipate likely future traffic configurations resulting from its own maneuvers. This is achieved through a Scenario-Based MPC framework, which constructs a scenario tree where each branch represents a possible evolution of the system.

Although these models face limitations due to the scarcity of real-world data for training and validation, they nonetheless constitute a powerful step toward the emerging paradigm of mixed-autonomy traffic. In such contexts, autonomous vehicles must not only drive safely and handle unexpected events but also adopt driving behaviors that closely resemble those of human drivers, thereby avoiding abrupt reactions from others. Moreover, these frameworks open new possibilities for leveraging autonomous agents as active actuators to improve overall highway efficiency and optimization.

Lastly, Chapters 9, 10, and 11 focus on the interactions between humans and autonomous agents, specifically addressing collision avoidance in pedestrian environments. In this part, the emphasis shifts from the description of interactions to the design of control schemes for the autonomous agent.

Chapter 9 presents a controller based on Game Theory, formulating the interaction as a potential game between the autonomous agent and an adversarial agent. Although this approach offers several advantages—most notably the guaranteed existence of an equilibrium by the very definition of a potential game—it also presents significant

limitations. Chief among these are the assumptions that each agent knows the potential function of the others and that all participants actively engage in the game to seek a Nash equilibrium. These assumptions, while mathematically convenient, render the framework unsuitable for modeling human–robot interactions and for practical motion planning of autonomous agents.

Due to the limitations of the game-theoretical approach, Chapter 10 introduces a preliminary method based on a more classical control strategy, namely the LQR, combined with a probabilistic formulation of CBF to handle collision avoidance constraints. Although this framework shows greater potential than the game-theoretic model, it lacks a mechanism to prioritize trajectories according to their associated probabilities, which limits its effectiveness in uncertain environments.

To address this shortcoming, the final version of the algorithm is presented in Chapter 11, where a hierarchical motion planning framework for autonomous navigation is proposed. This framework is organized in two cascading layers. The first, a macroscopic layer, processes environmental information to generate a high-level path across discrete cells, using the macroscopic features that characterize the state of each cell. The second, a microscopic layer, is responsible for collision avoidance, achieved through a partial relaxation of the safety constraints based on the likelihood of the considered scenarios.

This hierarchical approach has been preliminarily validated using a pre-recorded dataset, demonstrating encouraging results and confirming the feasibility of the proposed strategy. Ongoing work involves testing this framework first in the Gazebo simulation environment [101], and subsequently on a real wheelchair platform, as presented in Appendix G.

In conclusion, this dissertation has addressed two main research questions. The first concerns how to model the decision-making process of human agents in a simple yet effective way, allowing its integration within a control framework. While several methods exist for opinion dynamics based on agent networks—such as consensus dynamics—the options become much more limited when moving to other applications, for instance, traffic scenarios. To overcome this limitation, a decision-making model for human drivers has been developed based on a MC, which offers the strong advantage of being both simple and linear. Starting from intuitive, real-world observations, a model capable of forecasting a driver’s next decision has been formulated.

The second research question focuses on how these models can be exploited within control schemes to govern autonomous agents, such as vehicles or robots. For the control of an AV in a highway scenario—aimed at influencing the behavior of surrounding human-driven vehicles—a Scenario-Based MPC has been designed. This controller uses the predictions generated by the decision-making model of nearby drivers. Through this framework, it is possible not only to control the AV but also, indirectly, to steer the decisions of other drivers, effectively acting as a “safety car” that can enhance overall traffic performance. In a different context, a motion planner for social robots has been developed, where predictions of human motion are used to either relax or tighten the collision avoidance constraints. This enables smoother and less conservative trajectories while maintaining safety. Regarding this second research question, the main contribution lies in how to integrate behavioral predictions within the control framework, rather than in the control structure itself, as both implementations are built upon a standard MPC formulation.

Appendices

Appendix A

Markov Chains

In this Appendix, the essential preliminaries defining Markovian agents and networks are reviewed.

In this dissertation, the decision-making process of agents is modeled as a *Markov Chain* (MC). Hence, decisions are made according to certain probabilities, which evolve over time based on a transition matrix. Agents whose decision processes follow this probabilistic model are referred to as *Markovian agents*. A collection of such agents constitutes a *Markovian agent network*. In the following sections, the mathematical formulation of Markovian agents and networks is presented, both in continuous and discrete time.

Independent of whether the system is defined in continuous or discrete time, let $\mathcal{N}$ denote the set of N agents. The decision-making process of each agent $r \in \mathcal{N}$ is modeled as a MC defined over a finite set of states $\mathcal{S} = \{s^1, \dots, s^M\}$, where M represents the number of possible decision states.

A.0.1 Continuous-Time Markov Chain (CTMC)

A.0.1.1 Continous Markovian Agent

State transitions $s_i \rightarrow s_j$, where $i \neq j$, occur at non-negative real rates $Q^r[i, j]$, defined in the $M \times M$ transition rate matrix Q^r . Each diagonal element $Q^r[i, i]$ is equal to the negated sum of the off-diagonal elements in the corresponding row i . The vector $\mathbf{\Pi}^r(t) = [\pi_1^r(t) \ \pi_2^r(t) \ \dots \ \pi_M^r(t)]^T$ represents a probability distribution over $\mathcal{S}$, describing the likelihood that agent r is in each state at time t . It is obtained by solving

$$\dot{\mathbf{\Pi}}^r(t) = (Q^r)^T \mathbf{\Pi}^r(t), \quad (\text{A.1})$$

where $\dot{\mathbf{\Pi}}^r(t)$ denotes the first time derivative of $\mathbf{\Pi}^r(t)$, from some initial condition $\mathbf{\Pi}^r(0)$. The solution $\mathbf{\Pi}^r(t)$, including its transient behavior, is fully determined by Q^r and $\mathbf{\Pi}^r(0)$, as detailed in [34].

Following the approach in [2] and [14], it is assumed that Q^r is irreducible, which ensures that the associated CTMC is ergodic. This property implies that the agent possesses a nonzero probability of visiting all states in $\mathcal{S}$ and that $\mathbf{\Pi}^r(t)$ converges to a unique stationary state distribution $\bar{\mathbf{\Pi}}^r$.

A.0.1.2 Continuous Markovian Agents Network

For a general network $\mathbb{X}$ composed of Z agents, each with M possible decisions, there exist M^Z network states. Each network state is defined as a tuple $s^{\mathbb{X}} = \langle s^1, \dots, s^r, \dots, s^N \rangle$, where s^r denotes the state of the r th agent. The set of all possible network states is organized as

$$\mathcal{S}^{\mathbb{X}} = \mathcal{S}^1 \times \dots \times \mathcal{S}^r \times \dots \times \mathcal{S}^Z, \quad (\text{A.2})$$

where each $\mathcal{S}^r$ represents the same decision set $\mathcal{S}$ but is indexed to distinguish between agents. Since the probability that two agents transition simultaneously is zero, every transition between two network states is caused by the state transition of a single agent. Consequently, $\mathbb{X}$ can be modeled as a CTMC defined over $\mathcal{S}^{\mathbb{X}}$.

The probability that $\mathbb{X}$ occupies each network state at time t is denoted $\Pi_{\mathbb{X}}(t)$ and evolves according to

$$\dot{\mathbf{\Pi}}^{\mathbb{X}}(t) = (Q^{\mathbb{X}})^T \mathbf{\Pi}^{\mathbb{X}}(t), \quad (\text{A.3})$$

where $\dot{\mathbf{\Pi}}^{\mathbb{X}}(t)$ denotes the first time derivative of $\mathbf{\Pi}^{\mathbb{X}}(t)$, given the initial condition $\mathbf{\Pi}^{\mathbb{X}}(0)$. The network transition rate matrix Q is expressed as

$$Q^{\mathbb{X}} = \sum_{r=1}^Z I_{M^{r-1}} \otimes Q^r \otimes I_{M^{Z-r}}, \quad (\text{A.4})$$

where $\otimes$ denotes the Kronecker product, Q^r is the transition rate matrix of the r th agent, and I represents identity matrices of dimensions M^{r-1} and M^{Z-r} , respectively.

Since the $M^Z \times M^Z$ matrix Q defines a CTMC, it satisfies the same properties as described in Section A.0.1.2, although for an expanded state space. As discussed in [2], this network-level CTMC inherits the ergodicity property of the individual agent models.

A.0.2 Discrete-Time Markov Chain (DTMC)

Alternatively, the discrete-time framework can be employed.

A.0.2.1 Discrete Markovian Agent

State transition probabilities $\Pr(s_{k+1}^r = s_j | s_k^r = s_i) = p_{ij,k}^r$ at the discrete time step $k \in \mathbb{Z}$ are defined in the non-negative, row-stochastic matrix $Q_k^r \in \mathbb{R}^{M \times M}$ for all $i, j \in \{1, \dots, M\}$. The probability vector $\mathbf{\Pi}_k^r = [\pi_{1,k}^r \dots \pi_{M,k}^r]^T$ over $\mathcal{S}$ describes the decision probabilities of r , and is the solution of

$$\mathbf{\Pi}_{k+1}^r = (Q_k^r)^T \mathbf{\Pi}_k^r \quad (\text{A.5})$$

from some initial decision probability $\mathbf{\Pi}_0^r$. The DTMC is said to be *time-homogeneous* if the transition matrix is independent of k , such that $Q_k^r = Q^r \forall k$.

A.0.2.2 Discrete Markovian Agents Network

A network $\mathbb{X}$ of Z discrete-time Markovian agents is defined, where more than one agent may undergo a state transition at each discrete step k . Following the formulation in [2], the network state is denoted as $s^{\mathbb{X}} = \langle s^1, \dots, s^Z \rangle$, and the corresponding state set of the entire network is given by $\mathcal{S}_{\mathbb{X}} = \mathcal{S}^1 \times \dots \times \mathcal{S}^Z$. Since $\mathcal{S}^r$ represents the decision set $\mathcal{S}$ for each agent r , the network contains M^Z possible states. The transition matrix of $\mathbb{X}$ is expressed as

$$Q_k^{\mathbb{X}} = Q_k^1 \otimes \dots \otimes Q_k^r \otimes \dots \otimes Q_k^Z, \quad (\text{A.6})$$

where $\otimes$ denotes the Kronecker product. The evolution of the network decision probabilities is then described by

$$\mathbf{\Pi}_{k+1}^{\mathbb{X}} = (Q_k^{\mathbb{X}})^T \mathbf{\Pi}_k^{\mathbb{X}}, \quad (\text{A.7})$$

given the initial probability distribution $\mathbf{\Pi}_0^{\mathbb{X}}$ of the network decisions.

Appendix B

Interaction forces between agents

In this Appendix, the concept of interaction forces is recalled. To maintain generality, both the attractive and repulsive forces [2],[3],[14] between groups of agents are presented. Trivially, if one is interested in modeling the interaction between two individual agents, it is sufficient to consider groups composed of single agents.

B.0.1 Attraction forces

A group $\mathcal{A}$ consists of a subset of $n \leq N$ agents that interact according to the edges $\mathcal{E}^{\mathcal{A}}$ of the graph $\mathcal{G}_{\mathcal{A}} = (\mathcal{A}, \mathcal{E}^{\mathcal{A}}, \Lambda^{\mathcal{A}})$. The matrix $\Lambda^{\mathcal{A}}$ is a weighted, positive, and row-normalized adjacency matrix whose elements quantify the relative interaction strength between pairs of group members.

The attractive influence $\psi_j^{\mathcal{A}}(r)$ exerted on an agent $r \in \mathcal{A}$ toward state s_j , as a result of the other members' states, is defined as

$$\psi_j^{\mathcal{A}}(r) = \eta^r \lambda^{\mathcal{A}} \sum_{k_1 \in \mathcal{A}} \Lambda_{r,k_1}^{\mathcal{A}} I_j^{k_1}(t). \quad (\text{B.1})$$

The indicator function $I_j^{k_1}(t)$ equals one if member k_1 occupies state s_j at time t , and zero otherwise. The scalar parameter $\eta^r > 0$ characterizes the level of decision uncertainty of agent r , whereas $\lambda^{\mathcal{A}} > 0$ represents the overall influence strength within group $\mathcal{A}$.

B.0.2 Repulsion forces

Agents in a group $\mathcal{A}$ may also experience a repulsive influence on their transition rates, induced by a *repulsive force function* originating from other groups $\mathcal{R}_{\ell}$ within a set of repulsive groups $\mathcal{R}$. This interaction is represented by the graph $\mathcal{G}_{\mathcal{A}\mathcal{R}_{\ell}} = (\mathcal{A} \cup \mathcal{R}_{\ell}, \mathcal{E}^{\mathcal{A}\mathcal{R}_{\ell}}, \Gamma^{\mathcal{A}\mathcal{R}_{\ell}})$, where agents from groups $\mathcal{A}$ and $\mathcal{R}_{\ell}$ interact along the edges $\mathcal{E}^{\mathcal{A}\mathcal{R}_{\ell}}$ with relative interaction strengths specified by the positive, weighted, and row-normalized adjacency matrix $\Gamma^{\mathcal{A}\mathcal{R}_{\ell}}$.

The repulsive force $\xi_j^{\mathcal{A}\mathcal{R}_{\ell}}(r)$ that drives an agent r away from state s_j depends on the number of agents in $\mathcal{R}_{\ell}$ currently occupying that state. This interaction is defined as a *reactive force*

$$\xi_j^{\mathcal{A}\mathcal{R}_{\ell}}(r) = \eta^r \gamma^{\mathcal{A}\mathcal{R}_{\ell}} \frac{1}{|\mathcal{R}_{\ell}|} \sum_{k_2 \in \mathcal{R}_{\ell}} \Gamma_{r,k_2}^{\mathcal{A}\mathcal{R}_{\ell}} (1 - I_j^{k_2}(t)), \quad (\text{B.2})$$

where the positive scalar $\gamma^{\mathcal{A}\mathcal{R}_{\ell}}$ quantifies the repulsion strength exerted by agents in $\mathcal{R}_{\ell}$ on those in $\mathcal{A}$, and the number of repulsive groups $|\mathcal{R}_{\ell}|$ serves as a normalization factor. By expressing the repulsive force in this form, it avoids the introduction of negative transition rates toward state s_j .

B.0.3 Marginalized Model

In this small section, the complete proof of Theorem 3.3.1 is discussed.

Theorem B.0.1

Each row in the marginalized model describes the probability that the n :th agent is in s_j as

$$\dot{\pi}_j^n = \sum_{i=1}^M Q^n[i,j] \pi_i^n + \pi_j^n \sum_{\mathcal{R}_{\ell} \in \mathcal{R}} \gamma - \sum_{\mathcal{R}_{\ell} \in \mathcal{R}} \gamma \sum_k \Gamma_{\mathcal{R}_{\ell}}^{\mathcal{A}}[n,k] \pi_j^k. \quad (\text{B.3})$$

□

Proof. It is assumed that $\mathbb{E}[\mathbf{1}(t)] = \pi(t)$, where $\mathbb{E}[\cdot]$ denotes the expected value of a random variable. The probability that the n :th agent is in s_j after a time interval δt is expressed using the infinitesimal definition of the CTMC as

$$\mathbb{E}[\mathbf{1}_j^n(t + \delta t) | s^X] = \mathbf{1}_j^n(1 - \delta t \mathbb{Q}_O) + (1 - \mathbf{1}_j^n) \delta t \mathbb{Q}_I, \quad (\text{B.4})$$

where $\mathbb{Q}_O$ and $\mathbb{Q}_I$ represent the total outgoing and incoming rates from s_j , respectively. Taking the expected value of the right-hand side yields

$$\frac{\mathbb{E}[\mathbf{1}_j^n(t + \delta t) | s^X] - \mathbb{E}[\mathbf{1}_j^n]}{\delta t} = \mathbb{E}[-\mathbf{1}_j^n(\mathbb{Q}_O + \mathbb{Q}_I) + \mathbb{Q}_I], \quad (\text{B.5})$$

and the left-hand side approaches $\dot{\pi}_j^n$ as $\delta t \rightarrow 0$. The quantities $\mathbb{Q}_I$ and $\mathbb{Q}_O$ are then separated into two parts: isolated and repulsive rates. The following derivations describe the contribution of each part to $\dot{\pi}_j^n$.

First, by letting $\mathbb{Q}_O = \sum_{i \neq j} Q_n[j, i]$ and $\mathbb{Q}_I = \sum_{i \neq j} Q_n[i, j] \mathbf{1}_i^n$ in the right-hand side of (B.5), it follows that

$$\mathbb{E}\left[-\mathbf{1}_j^n \left(\sum_{i \neq j} Q_n[j, i] + \sum_{i \neq j} Q_n[i, j] \mathbf{1}_i^n \right) + \sum_{i \neq j} Q_n[i, j] \mathbf{1}_i^n \right]. \quad (\text{B.6})$$

Since $\sum_{i \neq j} Q_n[j, i] = -Q_n[j, j]$ and $\mathbf{1}_j^n \mathbf{1}_i^n = 0$ for all t , this expression simplifies to

$$\mathbb{E}\left[Q_n[j, j] \mathbf{1}_j^n + \sum_{i \neq j} Q_n[i, j] \mathbf{1}_i^n\right] = \mathbb{E}\left[\sum_{i=1}^M Q_n[i, j] \mathbf{1}_i^n\right]. \quad (\text{B.7})$$

The contribution to $\dot{\pi}_j^n$ from the isolated rates is therefore

$$\sum_{i=1}^M Q_n[i, j] \pi_i^n, \quad (\text{B.8})$$

which corresponds to the first term in (B.3).

Next, by setting $\mathbb{Q}_O = -\sum_{\mathcal{R}_\ell \in \mathcal{R}} \sum_{i \neq j}^{-\xi_i^n(\mathcal{R}_\ell, \gamma)}$ and $\mathbb{Q}_I = -\sum_{\mathcal{R}_\ell \in \mathcal{R}}^{-\xi_j^n(\mathcal{R}_\ell, \gamma)}$, the rate contributions from the direct repulsion force in (3.3) can be evaluated. Assuming that γ is state-independent, but possibly dependent on n , $\mathcal{A}$, and $\mathcal{R}_\ell$, the right-hand side of (B.5) becomes

$$\mathbb{E}\left[\mathbf{1}_j^n \sum_{\mathcal{R}_\ell \in \mathcal{R}} \sum_{i=1}^M -\xi_i^n(\mathcal{R}_\ell, \gamma) - \sum_{\mathcal{R}_\ell \in \mathcal{R}} -\xi_j^n(\mathcal{R}_\ell, \gamma)\right]. \quad (\text{B.9})$$

Because γ is independent of s_i and s_j , it follows that

$$\mathbb{E}\left[\mathbf{1}_j^n \sum_{\mathcal{R}_\ell \in \mathcal{R}} \gamma \sum_{i=1}^M \sum_k \Gamma_{\mathcal{R}_\ell}^{\mathcal{A}}[n, k] \mathbf{1}_i^k - \sum_{\mathcal{R}_\ell \in \mathcal{R}} \gamma \sum_k \Gamma_{\mathcal{R}_\ell}^{\mathcal{A}}[n, k] \mathbf{1}_j^k\right]. \quad (\text{B.10})$$

For each agent in $\mathcal{R}_\ell$, indexed by k , summing over all indicator functions across possible states $i = 1, \dots, M$ gives $\mathbf{1}_i^k = 1$ for exactly one i . Moreover, since $\Gamma_{\mathcal{R}_\ell}^{\mathcal{A}}$ is row-normalized, it follows that

$$\mathbb{E}\left[\mathbf{1}_j^n \sum_{\mathcal{R}_\ell \in \mathcal{R}} \gamma - \sum_{\mathcal{R}_\ell \in \mathcal{R}} \gamma \sum_k \Gamma_{\mathcal{R}_\ell}^{\mathcal{A}}[n, k] \mathbf{1}_j^k\right], \quad (\text{B.11})$$

which, when expressed in terms of π , becomes

$$\pi_j^n \sum_{\mathcal{R}_\ell \in \mathcal{R}} \gamma - \sum_{\mathcal{R}_\ell \in \mathcal{R}} \gamma \sum_k \Gamma_{\mathcal{R}_\ell}^{\mathcal{A}}[n, k] \pi_j^k, \quad (\text{B.12})$$

corresponding to the second and third terms in (B.3). $\square$

Appendix C

Cell-Transmission Model

In this appendix, the *Cell-Transmission Model*, a widely used modeling framework that describes highway traffic [102]-[103], is recalled. Additionally, it is discussed how the vehicle arrivals into the model can be described in this setting.

C.0.1 Cell Transmission Model

The CTM builds upon the decomposition of the considered road stretch into a series of short sections, referred to as *cells* and denoted as $\mathcal{C} = \{c_1, \dots, c_L\}$. The CTM is based on the following set of two equations:

$$n_{c,k+1} = n_{c,k} + d_{c,k} - d_{c+1,k}, \quad (\text{C.1a})$$

$$d_{c,k} = \min \left\{ n_{c-1,k}, D_{c,k}, N_{c,k} - n_{c,k} \right\}, \quad (\text{C.1b})$$

where $n_{c,k}$ represents the number of vehicles in cell c , $d_{c,k}$ is the inflow to cell c , and $d_{c+1,k}$ is the outflow from the same cell. Additionally, $N_c(k)$ represents the maximum number of vehicles that cell c can contain, and $D_{c,k}$ denotes the capacity flow into cell c , which corresponds to the maximum inflow of vehicles allowed in that cell. The term $N_{c,k} - n_{c,k}$ represents the available space within cell c . Moreover, the parameters $N_{c,k}$ and $D_{c,k}$ can also be constant over time.

Regardless of the number of cells, a CTM always includes *source*, *gate*, and *sink* cells. Specific properties hold for each cell type, recalled next. The source cell is assumed to contain a potentially infinite initial number of vehicles (i.e., $n_{\text{source},0} = \infty$), which enter the gate cell, thus resulting in an infinitely large cell ($N_{\text{gate}}(k) = \infty$). After traveling through the series of cells, the vehicles eventually arrive at the sink cell, which has an infinite capacity ($N_{\text{sink}}(k) = \infty$) and can accommodate all entering vehicles.

The source-gate and sink cells are the input and output cells respectively, which in turn represent the unknown road network structure outside the scenario's boundaries, namely the segments of road before and after the modeled road stretch, that are not of interest in the analysis.

C.0.2 Modeling the input to the traffic scenario

In the previous subsection, the infinite capacity of the gate cell in the CTM was mentioned. However, the literature does not clearly explain the transition of vehicles from the gate to the first cell of the road. That is, the function $n_{\text{gate},k}$, which influences the inflow $q_{1,k}$ of vehicles into the first cell:

$$d_{1,k} = \min \left\{ n_{\text{gate},k}, D_{1,k}, N_{1,k} - n_{1,k} \right\}, \quad (\text{C.2})$$

is not explicitly defined.

Nevertheless, in traffic scenario modeling, the process of arrivals in the gate cell is often assumed to follow a Poisson Process [104]. This approach is inspired by queuing theory from the telecommunications field [105], where the one-way road can be seen as a queue. The Poisson Process assumes infinite capacity, with the arrival inflow as a stochastic variable. The probability mass function of a Poisson distributed variable $n_{\text{gate},k}$, defined as

$$\mathbb{P}(n_{\text{gate},k} = a) = \frac{(\lambda k)^a}{a!} e^{-\lambda k},$$

describes the probability of having a arrivals at step k , where λk is the average number of arrivals. Therefore, the outflow from the gate cell to the first cell is a random variable.

Appendix D

Scenario-Tree MPC

This appendix provides a detailed description of the Scenario-Tree MPC introduced in Chapter 7. Figure D.1 presents a schematic illustration of the algorithm's operation.

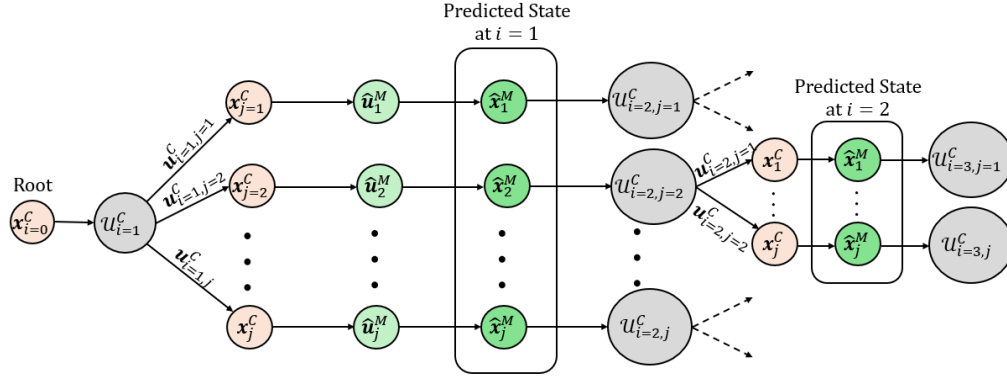


Figure D.1: Scenario-Tree Model Predictive Control.

The algorithm is based on the construction of a scenario tree, which accounts for different possible evolutions of the system depending on the control actions of the controller vehicle A_C .

The root of the tree corresponds to the current state of the controller agent A_C . At the first time instant, the control set is $\mathcal{U}_{i=1}^C$, from which A_C can select an action. Each different control input $u_{i=1}^C$ results in a distinct next state x^C . Using the predictive model that links the actions of A_C to the responses of A_M , the corresponding predicted input $\hat{u}^M$ and state $\hat{x}^M$ are computed.

This procedure is then repeated, taking the next state of A_C as the root for new branches of the tree.

The tree is expanded for N_c steps. Subsequently, the cost of each branch is evaluated according to the cost function (7.1), and the optimal scenario is identified.

Finally, the control sequence corresponding to the optimal scenario is applied, with only the first control input implemented. The algorithm is then repeated at the next time step until the end of the simulation.

Appendix E

Control Barrier Functions

This appendix collects the fundamental preliminaries on Control Barrier Function theory, that are applied or considered in Chapter 10 and 11. Here, the discrete version of CBF is provided, as it is treated in this dissertation. For the continuous version, please refer to [106].

Consider the following generic affine dynamics for the controlled agent

$$\mathbf{x}_{k+1} = f(\mathbf{x}_k, \mathbf{u}_k) = \phi(\mathbf{x}_k) + \psi(\mathbf{x}_k)\mathbf{u}_k \quad (\text{E.1})$$

where $\mathbf{x}_k \in \mathcal{X} \subseteq \mathbb{R}^{n_R}$, $\mathbf{u}_k \in \mathcal{U} \subseteq \mathbb{R}^{m_R}$ the state and the input respectively, and the functions $\phi : \mathbb{R}^{n_R} \rightarrow \mathbb{R}^{n_R}$ and $\psi : \mathbb{R}^{n_R} \rightarrow \mathbb{R}^{n_R \times m_R}$ locally Lipschitz.

Consider a set $\mathcal{D}_k$ defined as *superlevel set* of a function $h : \mathcal{X} \times \mathcal{O} \subseteq \mathbb{R}^{n_R \times n_O} \rightarrow \mathbb{R}$ and $\partial\mathcal{D}_k$ and $\text{Int}(\mathcal{D}_k)$ denote the boundary and interior of this set, respectively. In other words,

$$\begin{aligned} \mathcal{D}_k &= \{(\mathbf{x}_k, \mathbf{o}_k) \in \mathcal{X} \times \mathcal{O} : h(\mathbf{x}_k, \mathbf{o}_k) \geq 0\} \\ \partial\mathcal{D}_k &= \{(\mathbf{x}_k, \mathbf{o}_k) \in \mathcal{X} \times \mathcal{O} : h(\mathbf{x}_k, \mathbf{o}_k) = 0\} \\ \text{Int}(\mathcal{D}_k) &= \{(\mathbf{x}_k, \mathbf{o}_k) \in \mathcal{X} \times \mathcal{O} : h(\mathbf{x}_k, \mathbf{o}_k) > 0\} \end{aligned} \quad (\text{E.2})$$

It refers to $\mathcal{D}_k$ as the *safe set* and assumes it is nonempty.

Definition E.0.1 (Discrete-time CBF [32])

Consider the discrete-time system (E.1). Given a set $\mathcal{D}_k$ defined as in (E.2) for a function $h : \mathcal{X} \times \mathcal{O} \subseteq \mathbb{R}^{n_R \times n_O} \rightarrow \mathbb{R}$, the function h is a CBF defined on set $\mathcal{X}$ if there exists a function $\gamma \in \mathcal{K}_\infty$ such that

$$\inf_{\mathbf{u} \in \mathcal{U}} \Delta h(\mathbf{x}_k) \geq -\gamma(h(\mathbf{x}_k)),$$

where $\Delta h(\mathbf{x}_k) := h(\mathbf{x}_{k+1}) - h(\mathbf{x}_k)$.

Without loss of generality, assume that the $\mathcal{K}_\infty$ class function $\gamma(h(\mathbf{x}_k))$ to be $\gamma h(\mathbf{x}_k)$, $0 < \gamma \leq 1$.

Then the discrete-time control barrier function constraint is defined as

$$CBC_{k+1|k}(\mathbf{x}_k, \mathbf{u}_k, \mathbf{o}_k) = h(\mathbf{x}_{k+1}, \mathbf{o}_{k+1}) - (1 - \gamma)h(\mathbf{x}_k, \mathbf{o}_k). \quad (\text{E.3})$$

and, according to Definition E.0.1, it must be enforced to be greater than or equal to zero.

Appendix F

Hierarchical Motion Planner Additional Simulation Results

In this appendix, additional details on the simulations of the hierarchical motion planner, presented in Chapter 11, are provided.

Parameter	Value	Unit of Measure
$\mathbf{x}_0$	$[3.61, -9.44, 0, 0]^T$	$[\text{m}, \text{m}, \text{m/s}, \text{m/s}]^T$
$\mathbf{x}_{ref}$	$[0.54, 1.88, 0, 0]^T$	$[\text{m}, \text{m}, \text{m/s}, \text{m/s}]^T$
γ	0.3	—
α_D	5	$1/\text{m}^2$
α_K	3	$1/\text{s}^2$
α_E	2	—
b	0	—
d_{min}	0.5	m
Q	$\text{diag}([10, 10, 1, 1])$	$\text{diag}([1/\text{m}^2, 1/\text{m}^2, \text{s}^2/\text{m}^2, \text{s}^2/\text{m}^2])$
R	$\text{diag}([0.1, 0.1])$	$\text{diag}([\text{s}^4/\text{m}^2, \text{s}^4/\text{m}^2])$
d_W	10	—
$\bar{\mathbf{a}}$	$[3, 3]^T$	$[\text{m}/\text{s}^2, \text{m}/\text{s}^2]^T$
$\underline{\mathbf{a}}$	$[-3, -3]^T$	$[\text{m}/\text{s}^2, \text{m}/\text{s}^2]^T$
r_{sensor}	2.23	m
β_1	1	$1/\text{m}^2$
β_2	10	—
λ_{CT}	100	$1/\text{m}^2$
Δk	0.1	s
N	3	—
T	10	s
r_{obs}	3	m

Table F.1: Parameters employed in the simulations of the hierarchical controller.

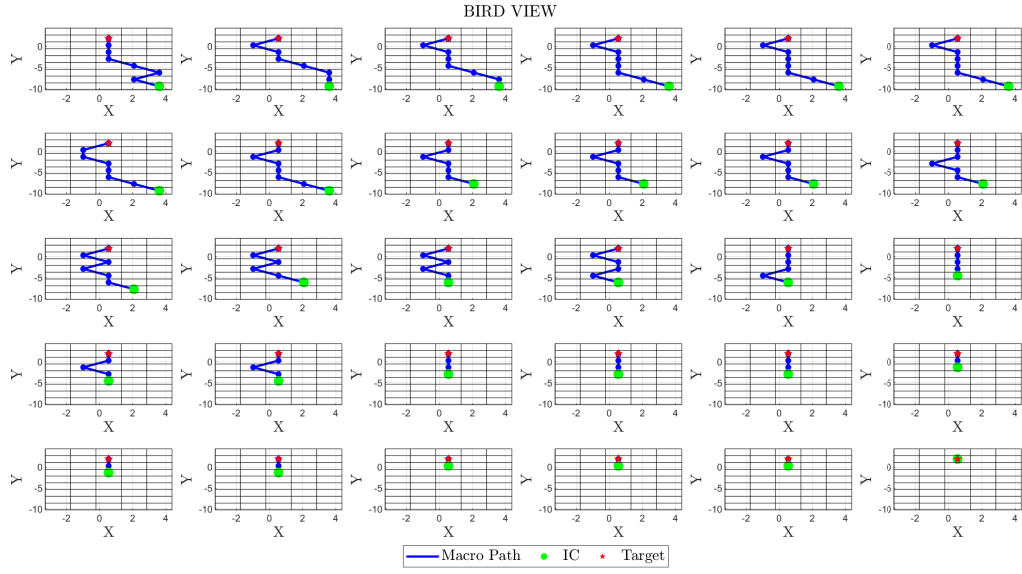


Figure F.1: Macroscopic path evolution during the simulation considering the bird-view approach.

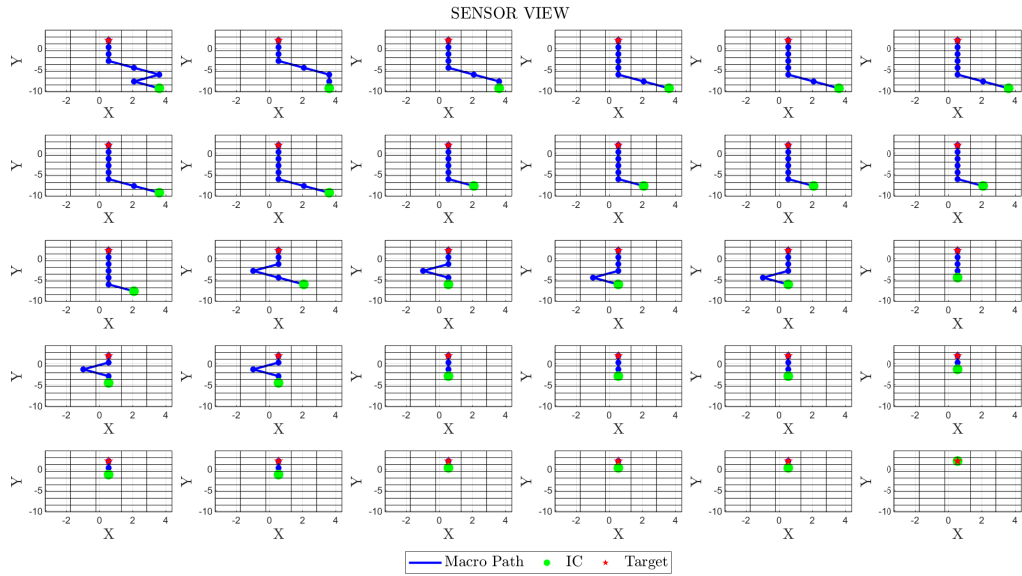


Figure F.2: Macroscopic path evolution during the simulation considering the sensor-view approach.

Appendix G

Autonomous Wheelchair System Description

In this appendix, the autonomous wheelchair used as the reference platform for the experimental tests presented in Chapter 11 is described.

Figure G.1 illustrates the control architecture of the wheelchair.

The system consists of two main components: a high-level controller, namely the motion planner (described in detail in Chapter 11), and a low-level controller. The motion planner is responsible for determining the optimal trajectory based on the perceived scenario, while the low-level controller translates the motion planner's directives into actual control inputs for the wheel motors.

Since the operation of the motion planner has been thoroughly discussed in the main text, this appendix focuses on providing additional details regarding the implementation and functioning of the low-level controller.

A differential-drive platform with wheel radius r and axle length L is considered. The wheelchair is modeled as a unicycle system, whose kinematic equations are given by:

$$\dot{x} = v \cos \theta, \quad \dot{y} = v \sin \theta, \quad \dot{\theta} = \omega. \quad (\text{G.1})$$

Let $\mathbf{x} = [x, y, \theta]^T$ denote the pose in the world frame, where θ is measured counterclockwise from the x -axis. The linear and angular velocities are denoted by (v, ω) , while (ω_L, ω_R) represent the left and right wheel angular velocities, respectively.

The forward kinematic map, which relates the wheel angular velocities to the body velocities, is expressed as:

$$\begin{aligned} v &= \frac{r}{2}(\omega_R + \omega_L), \\ \omega &= \frac{r}{L}(\omega_R - \omega_L). \end{aligned} \quad (\text{G.2})$$

Conversely, the inverse kinematic map, which determines the wheel angular velocities

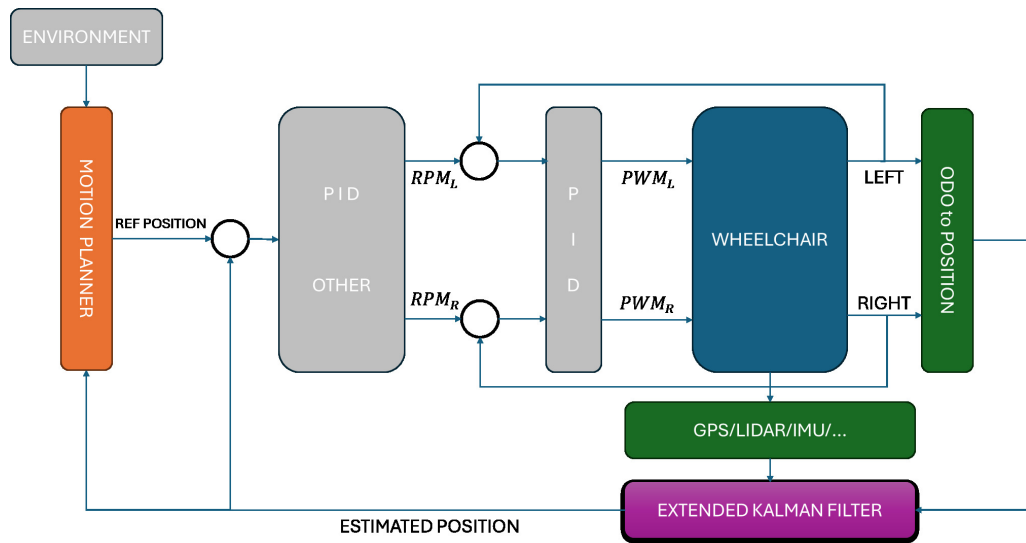


Figure G.1: Block scheme of the wheelchair.

from the desired body velocities, is given by:

$$\omega_L = \frac{1}{r} \left(v - \frac{L}{2} \omega \right), \quad \omega_R = \frac{1}{r} \left(v + \frac{L}{2} \omega \right). \quad (\text{G.3})$$

The wheel speeds can be expressed either as linear velocities or, if required, converted into revolutions per minute (RPM). The linear velocities at the left and right wheel rims are given by:

$$v_L = v - \frac{L}{2} \omega, \quad v_R = v + \frac{L}{2} \omega. \quad (\text{G.4})$$

The corresponding angular velocities and RPM values are obtained as:

$$\omega_{L,R} = \frac{v_{L,R}}{r}, \quad \text{RPM}_{L,R} = \omega_{L,R} \frac{60}{2\pi}. \quad (\text{G.5})$$

Each wheel is controlled in speed using a PID regulator with anti-windup compensation.

The robot is guided toward a planar reference position $\mathbf{p}^* = [x^*, y^*]^\top$ through two independent PID loops operating on the Cartesian position errors. The outputs of these controllers are combined to define a desired motion *direction* and *speed*, which are subsequently converted into body-frame commands (v^*, ω^*) .

The Cartesian errors are defined as:

$$e_x := x^* - x, \quad e_y := y^* - y. \quad (\text{G.6})$$

Each error drives a PID controller that produces “virtual” velocity commands u_x, u_y :

$$u_x(t) = \text{PID}_x\{e_x(t)\}, \quad u_y(t) = \text{PID}_y\{e_y(t)\}. \quad (\text{G.7})$$

In discrete time, with sampling period T_s and generic error e_k , the PID formulation is:

$$I_k = I_{k-1} + K_i T_s e_k, \quad D_k = \frac{e_k - e_{k-1}}{T_s}, \quad (\text{G.8})$$

$$u_k^{\text{raw}} = K_p e_k + I_k + K_d D_k, \quad u_k = \text{clip}(u_k^{\text{raw}}, u_{\min}, u_{\max}). \quad (\text{G.9})$$

An optional anti-windup compensation term can be included as:

$$I_k \leftarrow I_k + K_{aw} (u_k - u_k^{\text{raw}}). \quad (\text{G.10})$$

The pair (u_x, u_y) encodes the desired planar translation, from which a speed magnitude and heading direction are extracted as:

$$v_{\text{fb}} = \sqrt{u_x^2 + u_y^2}, \quad (\text{G.11})$$

$$\theta_{\text{fb}}^* = \text{atan2}(u_y, u_x). \quad (\text{G.12})$$

A (possibly time-varying) feedforward cruise speed v_{ff} , provided by the planner or operator, is added to the feedback component and limited as:

$$v^* = \text{clip}(v_{\text{ff}} + v_{\text{fb}}, -v_{\max}, v_{\max}). \quad (\text{G.13})$$

The heading error is defined as:

$$e_\theta = \text{wrap}_\pi(\theta_{\text{fb}}^* - \theta), \quad (\text{G.14})$$

and is regulated by a dedicated heading PID controller that produces the yaw rate command:

$$\omega^* = \text{PID}_\theta\{e_\theta\}, \quad \omega^* \leftarrow \text{clip}(\omega^*, -\omega_{\max}, \omega_{\max}). \quad (\text{G.15})$$

If a feedforward yaw rate ω_{ff} is available from the planner, the combined control law becomes:

$$\omega^* \leftarrow \text{clip}(\omega_{\text{ff}} + \text{PID}_\theta\{e_\theta\}, -\omega_{\max}, \omega_{\max}). \quad (\text{G.16})$$

Finally, the desired body-frame commands (v^*, ω^*) are converted into wheel angular velocities and RPM setpoints as:

$$\omega_L^* = \frac{1}{r} \left(v^* - \frac{L}{2} \omega^* \right), \quad (\text{G.17a})$$

$$\omega_R^* = \frac{1}{r} \left(v^* + \frac{L}{2} \omega^* \right), \quad (\text{G.17b})$$

$$\text{RPM}_{L,R}^* = \omega_{L,R}^* \frac{60}{2\pi}. \quad (\text{G.17c})$$

These RPM references are then tracked by the individual wheel speed PID loops described previously.

Figure G.2 shows photographs of the original autonomous wheelchair employed in the experiments.



Figure G.2: Pictures of the autonomous wheelchair.

References

- [1] Bolzern, P., Colaneri, P., and De Nicolao, G., “Opinion Dynamics in Social Networks with Heterogeneous Markovian Agents,” in *2018 IEEE Conference on Decision and Control (CDC)*, Miami Beach, FL: IEEE, Dec. 2018, pp. 6180–6185. DOI: [10.1109/CDC.2018.8619269](https://doi.org/10.1109/CDC.2018.8619269). Accessed: Oct. 1, 2025. [Online]. Available: <https://ieeexplore.ieee.org/document/8619269/>.
- [2] Bolzern, P., Colaneri, P., and Nicolao, G. D., “Opinion influence and evolution in social networks: A Markovian agents model,” *Automatica*, vol. 100, pp. 219–230, 2019. DOI: <https://doi.org/10.1016/j.automatica.2018.11.023>.
- [3] Bolzern, P., Colaneri, P., and De Nicolao, G., “Opinion Dynamics in Social Networks: The Effect of Centralized Interaction Tuning on Emerging Behaviors,” *IEEE Transactions on Computational Social Systems*, vol. 7, no. 2, pp. 362–372, 2020. DOI: [10.1109/TCSS.2019.2962273](https://doi.org/10.1109/TCSS.2019.2962273).
- [4] Ravazzi, C., Hojjatinia, S., Lagoa, C. M., and Dabbene, F., “Ergodic Opinion Dynamics Over Networks: Learning Influences From Partial Observations,” *IEEE Transactions on Automatic Control*, vol. 66, no. 6, pp. 2709–2723, 2021. DOI: [10.1109/TAC.2021.3056362](https://doi.org/10.1109/TAC.2021.3056362).
- [5] De Pasquale, G. and Elena Valcher, M., “Consensus Problems on Clustered Networks,” in *2020 59th IEEE Conference on Decision and Control (CDC)*, Jeju, Korea (South): IEEE, Dec. 2020, pp. 3675–3680. DOI: [10.1109/CDC42340.2020.9303877](https://doi.org/10.1109/CDC42340.2020.9303877). Accessed: Feb. 23, 2024.
- [6] Clark, A., Alomair, B., Bushnell, L., and Poovendran, R., “Leader selection in multi-agent systems for smooth convergence via fast mixing,” in *2012 IEEE 51st IEEE Conference on Decision and Control (CDC)*, ISSN: 0743-1546, Dec. 2012, pp. 818–824. DOI: [10.1109/CDC.2012.6426323](https://doi.org/10.1109/CDC.2012.6426323).
- [7] Pasqualetti, F., Zampieri, S., and Bullo, F., “Controllability metrics, limitations and algorithms for complex networks,” in *2014 American Control Conference*, ISSN: 2378-5861, Jun. 2014, pp. 3287–3292. DOI: [10.1109/ACC.2014.6858621](https://doi.org/10.1109/ACC.2014.6858621).
- [8] Dietrich, F., Martin, S., and Jungers, M., “Control via Leadership of Opinion Dynamics with State and Time-Dependent Interactions,” *IEEE Transactions on Automatic Control*, vol. 63, no. 4, pp. 1200–1207, Apr. 2018. DOI: [10.1109/TAC.2017.2742139](https://doi.org/10.1109/TAC.2017.2742139).
- [9] Puterman, M. L., “Chapter 8 Markov decision processes,” in *Stochastic Models*, ser. Handbooks in Operations Research and Management Science, vol. 2, ISSN: 0927-0507, Elsevier, 1990, pp. 331–434. DOI: [https://doi.org/10.1016/S0927-0507\(05\)80172-0](https://doi.org/10.1016/S0927-0507(05)80172-0). [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0927050705801720>.
- [10] Bauso, D., Tembine, H., and Başar, T., “Opinion Dynamics in Social Networks through Mean-Field Games,” in *SIAM Journal on Control and Optimization*, vol. 54, no. 6, pp. 3225–3257, Jan. 2016. DOI: [10.1137/140985676](https://doi.org/10.1137/140985676). Accessed: Feb. 23, 2024.
- [11] Bolzern, P., Colombo, A., and Piccardi, C., “Manipulating Opinions in Social Networks With Community Structure,” *IEEE Transactions on Network Science and Engineering*, vol. 11, no. 1, pp. 185–196, Jan. 2024. DOI: [10.1109/TNSE.2023.3293233](https://doi.org/10.1109/TNSE.2023.3293233). Accessed: Feb. 23, 2024.

- [12] Oumaima, E. J., Jalel, B. O., and Veronique, V., “Continuous Time Markov Chain Traffic Model for Urban Environments,” in *GLOBECOM 2020 - 2020 IEEE Global Communications Conference*, Taipei, Taiwan: IEEE, Dec. 2020, pp. 1–6. DOI: [10.1109/GLOBECOM42002.2020.9348256](https://doi.org/10.1109/GLOBECOM42002.2020.9348256). Accessed: Oct. 1, 2025. [Online]. Available: <https://ieeexplore.ieee.org/document/9348256/>.
- [13] Joubari, O. E., Othman, J. B., and Vèque, V., “Markov Chain Mobility Model for Multi-lane Highways,” en, *Mobile Networks and Applications*, vol. 27, no. 3, pp. 1286–1298, Jun. 2022. DOI: [10.1007/s11036-021-01893-4](https://doi.org/10.1007/s11036-021-01893-4). Accessed: Oct. 1, 2025. [Online]. Available: <https://link.springer.com/10.1007/s11036-021-01893-4>.
- [14] Heiker, C.-J. and Falcone, P., “Decision Modeling in Markovian Multi-Agent Systems,” in *2022 IEEE 61st Conference on Decision and Control (CDC)*, 2022, pp. 7235–7240. DOI: [10.1109/CDC51059.2022.9993134](https://doi.org/10.1109/CDC51059.2022.9993134).
- [15] Moridpour, S., Sarvi, M., and Rose, G., “Lane changing models: A critical review,” en, *Transportation Letters*, vol. 2, no. 3, pp. 157–173, Jul. 2010. DOI: [10.3328/TL.2010.02.03.157-173](https://doi.org/10.3328/TL.2010.02.03.157-173). Accessed: Feb. 14, 2024.
- [16] R.C. Carlson, I. Papamichail, and M. Papageorgiou, “Local feedback-based mainstream traffic flow control on motorways using variable speed limits,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 12, 2011.
- [17] Gray, A., Gao, Y., Hedrick, J. K., and Borrelli, F., “Robust Predictive Control for semi-autonomous vehicles with an uncertain driver model,” in *2013 IEEE Intelligent Vehicles Symposium (IV)*, Gold Coast City, Australia: IEEE, Jun. 2013, pp. 208–213. DOI: [10.1109/IVS.2013.6629472](https://doi.org/10.1109/IVS.2013.6629472). Accessed: Mar. 5, 2024.
- [18] Kamal, M. A. S., Taguchi, S., and Yoshimura, T., “Efficient vehicle driving on multi-lane roads using model predictive control under a connected vehicle environment,” in *2015 IEEE Intelligent Vehicles Symposium (IV)*, Seoul, South Korea: IEEE, Jun. 2015, pp. 736–741. DOI: [10.1109/IVS.2015.7225772](https://doi.org/10.1109/IVS.2015.7225772). Accessed: Mar. 5, 2024.
- [19] Y. Zhang and P. Ioannou, “Combined Variable Speed Limit and Lane Change Control for Highway Traffic,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 18, 2016.
- [20] Karlsson, J., Murgovski, N., and Sjöberg, J., “Optimal trajectory planning and decision making in lane change maneuvers near a highway exit,” in *2019 18th European Control Conference (ECC)*, Naples, Italy: IEEE, Jun. 2019, pp. 3254–3260. DOI: [10.23919/ECC.2019.8796010](https://doi.org/10.23919/ECC.2019.8796010). Accessed: Mar. 5, 2024.
- [21] Gottschalk, S., Burger, M., and Gerdts, M., “A projected primal-dual gradient optimal control method for deep reinforcement learning,” en, *Journal of Mathematics in Industry*, vol. 10, no. 1, p. 9, Dec. 2020. DOI: [10.1186/s13362-020-00075-3](https://doi.org/10.1186/s13362-020-00075-3). Accessed: Apr. 30, 2024.
- [22] Gottschalk, S. and Gerdts, M., “Constrained Training for Guided Feedback Controller,” en, *IFAC-PapersOnLine*, vol. 55, no. 20, pp. 97–102, 2022. DOI: [10.1016/j.ifacol.2022.09.078](https://doi.org/10.1016/j.ifacol.2022.09.078). Accessed: Apr. 30, 2024.
- [23] Chen, J., Zhan, W., and Tomizuka, M., “Constrained iterative LQR for on-road autonomous driving motion planning,” in *2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC)*, Yokohama: IEEE, Oct. 2017. DOI: [10.1109/itsc.2017.8317745](https://doi.org/10.1109/itsc.2017.8317745).
- [24] Pan, Y., Lin, Q., Shah, H., and Dolan, J. M., “Safe Planning for Self-Driving Via Adaptive Constrained ILQR,” in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Las Vegas, NV, USA: IEEE, Oct. 2020. DOI: [10.1109/iros45743.2020.9340886](https://doi.org/10.1109/iros45743.2020.9340886).
- [25] Pellegrini, S., Ess, A., Schindler, K., and Van Gool, L., “You’ll never walk alone: Modeling social behavior for multi-target tracking,” in *2009 IEEE 12th International Conference on Computer Vision*, Kyoto: IEEE, Sep. 2009, pp. 261–268. DOI: [10.1109/ICCV.2009.5459260](https://doi.org/10.1109/ICCV.2009.5459260).

- [26] Pellegrini, S., Ess, A., Tanaskovic, M., and Van Gool, L., “Wrong turn - No dead end: A stochastic pedestrian motion model,” in *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition - Workshops*, San Francisco, CA, USA: IEEE, Jun. 2010, pp. 15–22. DOI: [10.1109/CVPRW.2010.5543166](https://doi.org/10.1109/CVPRW.2010.5543166). Accessed: Apr. 4, 2024. [Online]. Available: <http://ieeexplore.ieee.org/document/5543166/>.
- [27] Bera, A., Randhavane, T., Prinja, R., and Manocha, D., “SocioSense: Robot navigation amongst pedestrians with social and psychological constraints,” in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Vancouver, BC: IEEE, Sep. 2017, pp. 7018–7025. DOI: [10.1109/IROS.2017.8206628](https://doi.org/10.1109/IROS.2017.8206628).
- [28] Bhatt, M., Jia, Y., and Mehr, N., “Efficient Constrained Multi-Agent Trajectory Optimization Using Dynamic Potential Games,” in *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Detroit, MI, USA: IEEE, Oct. 2023, pp. 7303–7310. DOI: [10.1109/IROS55552.2023.10342328](https://doi.org/10.1109/IROS55552.2023.10342328).
- [29] Peters, L. et al., “Contingency Games for Multi-Agent Interaction,” *IEEE Robotics and Automation Letters*, vol. 9, no. 3, pp. 2208–2215, Mar. 2024. DOI: [10.1109/LRA.2024.3354548](https://doi.org/10.1109/LRA.2024.3354548).
- [30] Wang, C., Bahreinian, M., and Tron, R., “Chance Constraint Robust Control with Control Barrier Functions,” in *2021 American Control Conference (ACC)*, New Orleans, LA, USA: IEEE, May 2021, pp. 2315–2322. DOI: [10.23919/ACC50511.2021.9482973](https://doi.org/10.23919/ACC50511.2021.9482973).
- [31] Zhu, H., Zhu, X., and Yao, W., “Probabilistic Multi-Robot Collision Avoidance Using Chance-Constrained Safety Barrier Certificates,” in *2023 42nd Chinese Control Conference (CCC)*, Tianjin, China: IEEE, Jul. 2023, pp. 5817–5822. DOI: [10.23919/CCC58697.2023.10241128](https://doi.org/10.23919/CCC58697.2023.10241128).
- [32] Li, M., Sun, Z., Liao, Z., and Weiland, S., *Moving Obstacle Collision Avoidance via Chance-Constrained MPC with CBF*, arXiv:2304.01639, Apr. 2023. DOI: [10.48550/arXiv.2304.01639](https://doi.org/10.48550/arXiv.2304.01639).
- [33] Heiker, C.-J., Gaetan, E., Giarré, L., and Falcone, P., “Repulsive Markovian models for opinion dynamics,” *Systems & Control Letters*, vol. 185, p. 105720, 2024. DOI: <https://doi.org/10.1016/j.sysconle.2024.105720>.
- [34] Cassandras, C. G. and Lafortune, S., Eds., *Introduction to Discrete Event Systems*, en. Boston, MA: Springer US, 2008. DOI: [10.1007/978-0-387-68612-7](https://doi.org/10.1007/978-0-387-68612-7). Accessed: Jun. 10, 2024. [Online]. Available: <http://link.springer.com/10.1007/978-0-387-68612-7>.
- [35] James, M., “The generalised inverse,” en, *The Mathematical Gazette*, vol. 62, no. 420, pp. 109–114, Jun. 1978. DOI: [10.2307/3617665](https://doi.org/10.2307/3617665). Accessed: Oct. 1, 2025. [Online]. Available: https://www.cambridge.org/core/product/identifier/S0025557200086460/type/journal_article.
- [36] Gaetan, E., Giarré, L., Sacone, S., and Falcone, P., “Control of decisions in stochastic multi-agent systems,” in *2024 European Control Conference (ECC)*, 2024, pp. 2356–2361. DOI: [10.23919/ECC64448.2024.10591205](https://doi.org/10.23919/ECC64448.2024.10591205).
- [37] Gaetan, E., Giarré, L., and Falcone, P., “Feedback control in multi-agent markovian networks,” in *2024 IEEE 20th International Conference on Automation Science and Engineering (CASE)*, 2024, pp. 1264–1269. DOI: [10.1109/CASE59546.2024.10711515](https://doi.org/10.1109/CASE59546.2024.10711515).
- [38] Kalman, R., “On the general theory of control systems,” *IRE Transactions on Automatic Control*, vol. 4, no. 3, pp. 110–110, 1959. DOI: [10.1109/TAC.1959.1104873](https://doi.org/10.1109/TAC.1959.1104873).
- [39] García, C. E., Prett, D. M., and Morari, M., “Model predictive control: Theory and practice—A survey,” *Automatica*, vol. 25, no. 3, pp. 335–348, 1989. DOI: [https://doi.org/10.1016/0005-1098\(89\)90002-2](https://doi.org/10.1016/0005-1098(89)90002-2).

- [40] Borrelli, F., Bemporad, A., and Morari, M., *Predictive Control for Linear and Hybrid Systems*. Cambridge University Press, 2017. [Online]. Available: <https://books.google.it/books?id=7NUoDwAAQBAJ>.
- [41] Cox, D. R., *Principles of Statistical Inference*. Cambridge University Press, 2006. DOI: [10.1017/CBO9780511813559](https://doi.org/10.1017/CBO9780511813559).
- [42] Laine, F. and Tomlin, C., “Efficient Computation of Feedback Control for Equality-Constrained LQR,” in *2019 International Conference on Robotics and Automation (ICRA)*, 2019, pp. 6748–6754. DOI: [10.1109/ICRA.2019.8793566](https://doi.org/10.1109/ICRA.2019.8793566).
- [43] Laine, F. and Tomlin, C., “Parallelizing LQR Computation Through Endpoint-Explicit Riccati Recursion,” in *2019 IEEE 58th Conference on Decision and Control (CDC)*, Nice, France: IEEE, Dec. 2019, pp. 1395–1402. DOI: [10.1109/CDC40024.2019.9029974](https://doi.org/10.1109/CDC40024.2019.9029974). Accessed: Jan. 29, 2024.
- [44] “Transport: Increasing Oil Consumption and Greenhouse Gas Emissions hamper EU Progress Towards Environment and Climate Objective,” European Environment Agency, Tech. Rep., 2020.
- [45] R. Filippova and N. Buchoud, “A Handbook on Sustainable Urban Mobility and Spatial Planning,” *Promoting Active Mobility*, vol. No. ECE/TRANS/298, 2020.
- [46] S. Siri, C. Pasquale, S. Sacone, and A. Ferrara, “Freeway traffic control: A survey,” *Automatica*, vol. 130, 2021.
- [47] C. Pasquale, S. Sacone, S. Siri, and A. Ferrara, “Traffic control for freeway networks with sustainability-related objectives: Review and future challenges,” *Annual Reviews in Control*, vol. 48, 2019.
- [48] Sun, Y. and Liu, H., “Crowd evacuation simulation method combining the density field and social force model,” en, *Physica A: Statistical Mechanics and its Applications*, vol. 566, p. 125 652, Mar. 2021. DOI: [10.1016/j.physa.2020.125652](https://doi.org/10.1016/j.physa.2020.125652). Accessed: Jun. 11, 2025. [Online]. Available: <https://linkinghub.elsevier.com/retrieve/pii/S037843712030950X>.
- [49] Sun, Z., Bebis, G., and Miller, R., “On-road vehicle detection using optical sensors: A review,” in *Proceedings. The 7th International IEEE Conference on Intelligent Transportation Systems (IEEE Cat. No.04TH8749)*, Washington, WA, USA: IEEE, 2004, pp. 585–590. DOI: [10.1109/ITSC.2004.1398966](https://doi.org/10.1109/ITSC.2004.1398966). Accessed: Mar. 26, 2024.
- [50] Ignatious, H. A., Sayed, H.-E., and Khan, M., “An overview of sensors in Autonomous Vehicles,” en, *Procedia Computer Science*, vol. 198, pp. 736–741, 2022. DOI: [10.1016/j.procs.2021.12.315](https://doi.org/10.1016/j.procs.2021.12.315). Accessed: Mar. 26, 2024.
- [51] Ferrara, A., Sacone, S., and Siri, S., “Event-triggered strategies for the networked control of freeway traffic systems,” in *2014 European Control Conference (ECC)*, Strasbourg, France: IEEE, Jun. 2014, pp. 2594–2599. DOI: [10.1109/ECC.2014.6862470](https://doi.org/10.1109/ECC.2014.6862470). Accessed: Mar. 7, 2024.
- [52] Ojeda, L. L., Kibangou, A. Y., and De Wit, C. C., “Adaptive Kalman filtering for multi-step ahead traffic flow prediction,” in *2013 American Control Conference*, Washington, DC: IEEE, Jun. 2013, pp. 4724–4729. DOI: [10.1109/ACC.2013.6580568](https://doi.org/10.1109/ACC.2013.6580568). Accessed: Mar. 27, 2024.
- [53] Messaoud, K., Yahiaoui, I., Verroust-Blondet, A., and Nashashibi, F., “Attention Based Vehicle Trajectory Prediction,” *IEEE Transactions on Intelligent Vehicles*, vol. 6, no. 1, pp. 175–185, Mar. 2021. DOI: [10.1109/TIV.2020.2991952](https://doi.org/10.1109/TIV.2020.2991952). Accessed: Mar. 26, 2024.
- [54] Kuhnt, F., Schulz, J., Schamm, T., and Zollner, J. M., “Understanding interactions between traffic participants based on learned behaviors,” in *2016 IEEE Intelligent Vehicles Symposium (IV)*, Gotenburg, Sweden: IEEE, Jun. 2016, pp. 1271–1278. DOI: [10.1109/IVS.2016.7535554](https://doi.org/10.1109/IVS.2016.7535554). Accessed: Mar. 27, 2024.

- [55] Li, X., Liu, K., Tseng, H. E., Girard, A., and Kolmanovsky, I., *Interaction-Aware Decision-Making for Autonomous Vehicles in Forced Merging Scenario Leveraging Social Psychology Factors*, arXiv:2309.14497, Sep. 2023. Accessed: May 1, 2024.
- [56] Merabet, G. H. et al., “Applications of Multi-Agent Systems in Smart Grids: A survey,” in *2014 International Conference on Multimedia Computing and Systems (ICMCS)*, Marrakech, Morocco: IEEE, Apr. 2014, pp. 1088–1094. DOI: [10.1109/ICMCS.2014.6911384](https://doi.org/10.1109/ICMCS.2014.6911384). Accessed: Mar. 5, 2024.
- [57] Gaetan, E., Giarré, L., Sacone, S., Falcone, P., and Heiker, C.-J., “Modelling traffic scenarios via markovian opinion dynamics,” in *2023 IEEE 26th International Conference on Intelligent Transportation Systems (ITSC)*, 2023, pp. 1072–1077. DOI: [10.1109/ITSC57777.2023.10422274](https://doi.org/10.1109/ITSC57777.2023.10422274).
- [58] Maiworm, M., Bähge, T., and Findeisen, R., “Scenario-based Model Predictive Control: Recursive Feasibility and Stability,” en, *IFAC-PapersOnLine*, vol. 48, no. 8, pp. 50–56, 2015. DOI: [10.1016/j.ifacol.2015.08.156](https://doi.org/10.1016/j.ifacol.2015.08.156). Accessed: Mar. 4, 2024.
- [59] Bernardini, D. and Bemporad, A., “Scenario-based model predictive control of stochastic constrained linear systems,” in *Proceedings of the 48th IEEE Conference on Decision and Control (CDC) held jointly with 2009 28th Chinese Control Conference*, Shanghai: IEEE, Dec. 2009, pp. 6333–6338. DOI: [10.1109/CDC.2009.5399917](https://doi.org/10.1109/CDC.2009.5399917). Accessed: Mar. 4, 2024.
- [60] De La Penad, D., Bemporad, A., and Alamo, T., “Stochastic Programming Applied to Model Predictive Control,” in *Proceedings of the 44th IEEE Conference on Decision and Control*, Seville, Spain: IEEE, 2005, pp. 1361–1366. DOI: [10.1109/CDC.2005.1582348](https://doi.org/10.1109/CDC.2005.1582348). Accessed: Mar. 4, 2024.
- [61] Tran, A.-T., Muraleedharan, A., Okuda, H., and Suzuki, T., “Scenario-based stochastic MPC for vehicle speed control considering the interaction with pedestrians,” en, *IFAC-PapersOnLine*, vol. 53, no. 2, pp. 15 325–15 331, 2020. DOI: [10.1016/j.ifacol.2020.12.2341](https://doi.org/10.1016/j.ifacol.2020.12.2341). Accessed: Mar. 4, 2024.
- [62] Gaetan, E., Giarré, L., and Falcone, P., “A behavioral model for interacting vehicles in highway scenarios*,” in *2024 IEEE 27th International Conference on Intelligent Transportation Systems (ITSC)*, 2024, pp. 633–638. DOI: [10.1109/ITSC58415.2024.10919737](https://doi.org/10.1109/ITSC58415.2024.10919737).
- [63] Singer, P., Helic, D., Taraghi, B., and Strohmaier, M., “Detecting Memory and Structure in Human Navigation Patterns Using Markov Chain Models of Varying Order,” *PLOS ONE*, vol. 9, no. 7, pp. 1–21, Jul. 2014, Publisher: Public Library of Science. DOI: [10.1371/journal.pone.0102070](https://doi.org/10.1371/journal.pone.0102070).
- [64] Fleming, T. R. and Harrington, D. P., “Estimation for discrete time nonhomogeneous Markov chains,” *Stochastic Processes and their Applications*, vol. 7, no. 2, pp. 131–139, Jun. 1978.
- [65] Craig, B. A. and Sendi, P. P., “Estimation of the transition matrix of a discrete-time Markov chain,” *Health economics*, vol. 11 1, pp. 33–42, 2002.
- [66] Anderson, T. W. and Goodman, L. A., “Statistical Inference about Markov Chains,” *The Annals of Mathematical Statistics*, vol. 28, no. 1, pp. 89–110, 1957, Publisher: Institute of Mathematical Statistics. Accessed: Apr. 24, 2023.
- [67] Boel, R. and Mihaylova, L., “A compositional stochastic model for real time freeway traffic simulation,” *Transportation Research Part B: Methodological*, vol. 40, no. 4, pp. 319–334, 2006. DOI: <https://doi.org/10.1016/j.trb.2005.05.001>.
- [68] Cowles, M. K. and Carlin, B. P., “Markov Chain Monte Carlo Convergence Diagnostics: A Comparative Review,” *Journal of the American Statistical Association*, vol. 91, no. 434, pp. 883–904, 1996, Publisher: Taylor & Francis. eprint: <https://www.tandfonline.com/doi/pdf/10.1080/01621459.1996.10476956>. DOI: [10.1080/01621459.1996.10476956](https://doi.org/10.1080/01621459.1996.10476956).

- [69] Karimi, S. and Vahidi, A., “Receding Horizon Motion Planning for Automated Lane Change and Merge Using Monte Carlo Tree Search and Level-K Game Theory,” in *2020 American Control Conference (ACC)*, 2020, pp. 1223–1228. DOI: [10.23919/ACC45564.2020.9147369](https://doi.org/10.23919/ACC45564.2020.9147369).
- [70] Knowles Flanagan, S., Tang, Z., He, J., and Yusoff, I., “Investigating and Modeling of Cooperative Vehicle-to-Vehicle Safety Stopping Distance,” *Future Internet*, vol. 13, no. 3, 2021. DOI: [10.3390/fi13030068](https://doi.org/10.3390/fi13030068).
- [71] Drożdżiel, P., Tarkowski, S., Rybicka, I., and Wrona, R., “Drivers’ reaction time research in the conditions in the real traffic,” en, *Open Engineering*, vol. 10, no. 1, pp. 35–47, Jan. 2020. DOI: [10.1515/eng-2020-0004](https://doi.org/10.1515/eng-2020-0004). Accessed: Feb. 12, 2024.
- [72] Gaetan, E., Giarré, L., and Falcone, P., “Scenario-tree model predictive control for vehicle interactions in highway setting,” *IEEE Control Systems Letters*, vol. 8, pp. 1162–1167, 2024. DOI: [10.1109/LCSYS.2024.3408035](https://doi.org/10.1109/LCSYS.2024.3408035).
- [73] Sheridan, T. B., “A review of recent research in social robotics,” en, *Current Opinion in Psychology*, vol. 36, pp. 7–12, Dec. 2020. DOI: [10.1016/j.copsyc.2020.01.003](https://doi.org/10.1016/j.copsyc.2020.01.003).
- [74] Balatti, P. et al., “Robot-Assisted Navigation for Visually Impaired through Adaptive Impedance and Path Planning,” in *2024 IEEE International Conference on Robotics and Automation (ICRA)*, Yokohama, Japan: IEEE, May 2024, pp. 2310–2316. DOI: [10.1109/ICRA57147.2024.10611071](https://doi.org/10.1109/ICRA57147.2024.10611071). Accessed: Nov. 6, 2025. [Online]. Available: <https://ieeexplore.ieee.org/document/10611071/>.
- [75] Mavrogiannis, C. et al., “Core Challenges of Social Robot Navigation: A Survey,” en, *ACM Transactions on Human-Robot Interaction*, vol. 12, no. 3, pp. 1–39, Sep. 2023. DOI: [10.1145/3583741](https://doi.org/10.1145/3583741).
- [76] Van Den Berg, J., Ming Lin, and Manocha, D., “Reciprocal Velocity Obstacles for real-time multi-agent navigation,” in *2008 IEEE International Conference on Robotics and Automation*, Pasadena, CA, USA: IEEE, May 2008, pp. 1928–1935. DOI: [10.1109/ROBOT.2008.4543489](https://doi.org/10.1109/ROBOT.2008.4543489).
- [77] Shu-Yun Chung and Han-Pang Huang, “A mobile robot that understands pedestrian spatial behaviors,” in *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Taipei: IEEE, Oct. 2010, pp. 5861–5866. DOI: [10.1109/IROS.2010.5649718](https://doi.org/10.1109/IROS.2010.5649718). Accessed: May 16, 2024. [Online]. Available: <http://ieeexplore.ieee.org/document/5649718/>.
- [78] Turnwald, A., Althoff, D., Wollherr, D., and Buss, M., “Understanding Human Avoidance Behavior: Interaction-Aware Decision Making Based on Game Theory,” en, *International Journal of Social Robotics*, vol. 8, no. 2, pp. 331–351, Apr. 2016. DOI: [10.1007/s12369-016-0342-2](https://doi.org/10.1007/s12369-016-0342-2). Accessed: May 9, 2024. [Online]. Available: <http://link.springer.com/10.1007/s12369-016-0342-2>.
- [79] Turnwald, A. and Wollherr, D., “Human-Like Motion Planning Based on Game Theoretic Decision Making,” en, *International Journal of Social Robotics*, vol. 11, no. 1, pp. 151–170, Jan. 2019. DOI: [10.1007/s12369-018-0487-2](https://doi.org/10.1007/s12369-018-0487-2). Accessed: May 9, 2024. [Online]. Available: <http://link.springer.com/10.1007/s12369-018-0487-2>.
- [80] Mozzarelli, L., Corno, M., and Savaresi, S. M., “Socially Aware Local Planning: A Dynamic Window-based Approach,” in *2024 European Control Conference (ECC)*, Stockholm, Sweden: IEEE, Jun. 2024, pp. 3255–3260. DOI: [10.23919/ECC64448.2024.10591069](https://doi.org/10.23919/ECC64448.2024.10591069). Accessed: Jul. 25, 2024. [Online]. Available: <https://ieeexplore.ieee.org/document/10591069/>.
- [81] Fridovich-Keil, D., Ratner, E., Peters, L., Dragan, A. D., and Tomlin, C. J., “Efficient Iterative Linear-Quadratic Approximations for Nonlinear Multi-Player General-Sum Differential Games,” in *2020 IEEE International Conference on Robotics and Automation (ICRA)*, Paris, France: IEEE, May 2020. DOI: [10.1109/icra40945.2020.9197129](https://doi.org/10.1109/icra40945.2020.9197129). Accessed: Sep. 16, 2024. [Online]. Available: <https://ieeexplore.ieee.org/document/9197129/>.

- [82] Kavuncu, T., Yaraneri, A., and Mehr, N., “Potential iLQR: A Potential-Minimizing Controller for Planning Multi-Agent Interactive Trajectories,” in *Robotics: Science and Systems XVII*, Robotics: Science and Systems Foundation, Jul. 2021. DOI: 10.15607/rss.2021.xvii.084. Accessed: Sep. 16, 2024. [Online]. Available: <http://www.roboticsproceedings.org/rss17/p084.pdf>.
- [83] Williams, Z., Chen, J., and Mehr, N., *Distributed Potential iLQR: Scalable Game-Theoretic Trajectory Planning for Multi-Agent Interactions*, Version Number: 1, 2023. DOI: 10.48550/ARXIV.2303.04842. Accessed: Sep. 16, 2024. [Online]. Available: <https://arxiv.org/abs/2303.04842>.
- [84] Bhatt, M. and Mehr, N., *Strategic Decision-Making in Multi-Agent Domains: A Weighted Potential Dynamic Game Approach*, Version Number: 2, 2023. DOI: 10.48550/ARXIV.2308.05876. Accessed: Oct. 29, 2024. [Online]. Available: <https://arxiv.org/abs/2308.05876>.
- [85] Galati, G., Primatesta, S., Grammatico, S., Macrì, S., and Rizzo, A., “Game theoretical trajectory planning enhances social acceptability of robots by humans,” en, *Scientific Reports*, vol. 12, no. 1, p. 21 976, Dec. 2022. DOI: 10.1038/s41598-022-25438-1. Accessed: Apr. 4, 2024. [Online]. Available: <https://www.nature.com/articles/s41598-022-25438-1>.
- [86] Nash Jr, J., *Essays on Game Theory*. Edward Elgar Publishing, Dec. 1996. DOI: 10.4337/9781781956298. Accessed: Sep. 18, 2024. [Online]. Available: <https://www.elgaronline.com/view/book/9781781956298/9781781956298.xml>.
- [87] Monderer, D. and Shapley, L. S., “Potential Games,” en, *Games and Economic Behavior*, vol. 14, no. 1, pp. 124–143, May 1996. DOI: 10.1006/game.1996.0044. Accessed: Sep. 18, 2024. [Online]. Available: <https://linkinghub.elsevier.com/retrieve/pii/S0899825696900445>.
- [88] Wächter, A. and Biegler, L. T., “On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming,” en, *Mathematical Programming*, vol. 106, no. 1, pp. 25–57, Mar. 2006. DOI: 10.1007/s10107-004-0559-y. Accessed: Sep. 10, 2025. [Online]. Available: <http://link.springer.com/10.1007/s10107-004-0559-y>.
- [89] Bianchin, F., Lefringhausen, R., Gaetan, E., Tesfazgi, S., and Hirche, S., “A set-theoretic robust control approach for linear quadratic games with unknown counterparts*,” 2025. arXiv: 2504.02679 [eess.SY]. [Online]. Available: <https://arxiv.org/abs/2504.02679>.
- [90] Gaetan, E., Falcone, P., Bayen, A. M., and Giarré, L., “Soft Control Barrier Functions for autonomous robots moving in human populated environments,” *IEEE Transactions on Systems, Man and Cybernetics (submitted)*,
- [91] Gaetan, E., Jeddi, M., Falcone, P., and Giarré, L., “Human-Attention-Aware Motion Planning for Social Robots,” in *European Control Conference (submitted)*, 2026.
- [92] Hosseinzadeh, M., Sinopoli, B., and Bobick, A. F., “Toward Safe and Efficient Human–Robot Interaction via Behavior-Driven Danger Signaling,” *IEEE Transactions on Control Systems Technology*, vol. 32, no. 1, pp. 214–224, Jan. 2024. DOI: 10.1109/TCST.2023.3305100. Accessed: Oct. 27, 2025. [Online]. Available: <https://ieeexplore.ieee.org/document/10229973/>.
- [93] Hosseinzadeh, M., Sinopoli, B., and Bobick, A. F., “Robot Action Planning in the Presence of Careless Humans,” *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 54, no. 9, pp. 5400–5411, Sep. 2024. DOI: 10.1109/TSMC.2024.3404346. Accessed: Oct. 27, 2025. [Online]. Available: <https://ieeexplore.ieee.org/document/10551849/>.

- [94] Chakraborty, P., Ahmed, S., Yousuf, M. A., Azad, A., Alyami, S. A., and Moni, M. A., "A Human-Robot Interaction System Calculating Visual Focus of Human's Attention Level," *IEEE Access*, vol. 9, pp. 93 409–93 421, 2021. DOI: [10.1109/ACCESS.2021.3091642](https://doi.org/10.1109/ACCESS.2021.3091642). Accessed: Sep. 24, 2025. [Online]. Available: <https://ieeexplore.ieee.org/document/9462086/>.
- [95] Mao, W., Liu, M., Salzmann, M., and Li, H., "Multi-level Motion Attention for Human Motion Prediction," en, *International Journal of Computer Vision*, vol. 129, no. 9, pp. 2513–2535, Sep. 2021. DOI: [10.1007/s11263-021-01483-7](https://doi.org/10.1007/s11263-021-01483-7). Accessed: Sep. 24, 2025. [Online]. Available: <https://link.springer.com/10.1007/s11263-021-01483-7>.
- [96] Quan, Y. S., Jeddi, M., Prignoli, F., and Falcone, P., "Priority-Driven Constraints Softening in Safe MPC for Perturbed Systems," *IEEE Control Systems Letters*, vol. 9, pp. 1069–1074, 2025. DOI: [10.1109/LCSYS.2025.3580494](https://doi.org/10.1109/LCSYS.2025.3580494). Accessed: Oct. 30, 2025. [Online]. Available: <https://ieeexplore.ieee.org/document/11072918/>.
- [97] Martinez-Gil, F., Lozano, M., García-Fernández, I., and Fernández, F., "Modeling, Evaluation, and Scale on Artificial Pedestrians: A Literature Review," en, *ACM Computing Surveys*, vol. 50, no. 5, pp. 1–35, Sep. 2018. DOI: [10.1145/3117808](https://doi.org/10.1145/3117808). Accessed: Jun. 11, 2025. [Online]. Available: <https://dl.acm.org/doi/10.1145/3117808>.
- [98] Jiang, Y., Chen, B., Li, X., and Ding, Z., "Dynamic navigation field in the social force model for pedestrian evacuation," en, *Applied Mathematical Modelling*, vol. 80, pp. 815–826, Apr. 2020. DOI: [10.1016/j.apm.2019.10.016](https://doi.org/10.1016/j.apm.2019.10.016). Accessed: Jun. 11, 2025. [Online]. Available: <https://linkinghub.elsevier.com/retrieve/pii/S0307904X19306043>.
- [99] Jian, X.-X., Wong, S., Zhang, P., Choi, K., Li, H., and Zhang, X., "Perceived cost potential field cellular automata model with an aggregated force field for pedestrian dynamics," en, *Transportation Research Part C: Emerging Technologies*, vol. 42, pp. 200–210, May 2014. DOI: [10.1016/j.trc.2014.01.018](https://doi.org/10.1016/j.trc.2014.01.018). Accessed: Jun. 11, 2025. [Online]. Available: <https://linkinghub.elsevier.com/retrieve/pii/S0968090X14000321>.
- [100] Liang, H., Du, J., and Wong, S., "A Continuum model for pedestrian flow with explicit consideration of crowd force and panic effects," en, *Transportation Research Part B: Methodological*, vol. 149, pp. 100–117, Jul. 2021. DOI: [10.1016/j.trb.2021.05.006](https://doi.org/10.1016/j.trb.2021.05.006). Accessed: Jun. 11, 2025. [Online]. Available: <https://linkinghub.elsevier.com/retrieve/pii/S0191261521000849>.
- [101] Koenig, N. and Howard, A., "Design and use paradigms for gazebo, an open-source multi-robot simulator," in *2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (IEEE Cat. No.04CH37566)*, vol. 3, Sendai, Japan: IEEE, 2004, pp. 2149–2154. DOI: [10.1109/IROS.2004.1389727](https://doi.org/10.1109/IROS.2004.1389727). Accessed: Oct. 30, 2025. [Online]. Available: <http://ieeexplore.ieee.org/document/1389727/>.
- [102] Daganzo, C. F., "The cell transmission model: A dynamic representation of highway traffic consistent with the hydrodynamic theory," *Transportation Research Part B: Methodological*, vol. 28, no. 4, pp. 269–287, 1994. DOI: [https://doi.org/10.1016/0191-2615\(94\)90002-7](https://doi.org/10.1016/0191-2615(94)90002-7).
- [103] Daganzo, C. F., "The cell transmission model, part II: Network traffic," *Transportation Research Part B: Methodological*, vol. 29, no. 2, pp. 79–93, 1995. DOI: [https://doi.org/10.1016/0191-2615\(94\)00022-R](https://doi.org/10.1016/0191-2615(94)00022-R).
- [104] Chandrasekaran, B., "Survey of network traffic models," *Washington University in St. Louis CSE*, vol. 567, 2009.
- [105] Cooper, R. B., *Introduction to queueing theory [by] Robert B. Cooper*, English. Macmillan New York, 1972, Type: Book.

- [106] Ames, A. D., Grizzle, J. W., and Tabuada, P., “Control barrier function based quadratic programs with application to adaptive cruise control,” in *53rd IEEE Conference on Decision and Control*, Los Angeles, CA, USA: IEEE, Dec. 2014, pp. 6271–6278. DOI: [10.1109/CDC.2014.7040372](https://doi.org/10.1109/CDC.2014.7040372). Accessed: Jan. 23, 2025. [Online]. Available: <http://ieeexplore.ieee.org/document/7040372/>.

