Robust and Decentralized Control of Multi-agent Systems under High-level Tasks

ALEXANDROS NIKOU

PhD Thesis
Stockholm, Sweden, 2019

KTH Royal Institute of Technology
School of Electrical Engineering
and Computer Science
Division of Decision and Control Systems
SE-100 44 Stockholm
Sweden

TRITA-EECS-AVL-2019:72
ISBN: 978-91-7873-319-4

Akademisk avhandling som med tillstånd av Kungl Tekniska högskolan framlägges till offentlig granskning för avläggande av teknologie doktorsexamen i electro- och systemteknik fredagen den 6 December 2019 klockan 9.00 i Kollegiesalen, Kungliga Tekniska högskolan, Brinellvägen 8, Stockholm, Sweden.

Tryck: Universitetsservice US AB

Abstract

Decentralized control of multi-agent systems is an active topic of research, with many practical applications arising in multi-robot systems, autonomous driving, transportation systems and robotic manipulation. The contributions of this thesis lie in the scope of three topics: formation control, robust decentralized tube-based nonlinear Model Predictive Control and time-constrained cooperative planning of multi-agent systems.

In the first part of the thesis, given a team of rigid bodies, we propose model-free and decentralized control protocols such that a desired distance and orientation-based formation between neighboring agents is achieved. Inter-agent collisions are guaranteed to be avoided by the proposed control scheme. Furthermore, the connectivity between agents that are initially connected is preserved. The transient and steady state responses are solely determined by certain designer-specified performance functions.

In the second part of the thesis, the problem of robust navigation of a multi-agent system to predefined states of the workspace while using only local information is addressed, under certain distance and control input constraints. The agents are modeled by nonlinear continuous-time dynamics with additive and bounded disturbances. In order to address this problem, decentralized tube-based nonlinear Model Predictive Control protocols are proposed. In particular, the feedback control law contains a portion that is calculated offline and a portion which is the outcome of an online optimal control problem.

In the third part of the thesis, a team of agents operating in a bounded workspace is considered. Each agent is assigned with high-level tasks given in Metric Interval Temporal Logic. First, by providing novel decentralized abstraction design techniques, the motion of each agent is captured through a weighted transition system. Then, we propose decentralized control methodologies and high-level algorithms that guarantee the satisfaction of the desired tasks of each agent. The proposed approach can handle couplings as well as transient constraints of each agent in a novel way.

Sammanfattning

Reglering och planering för fleragentssystem är ett aktivt forskningsfält motiverat av ett antal praktiska tillämpningar: självkörande bilar, situationer där ett antal robotar behöver utföra en gemensam uppgift, transport- och logistiksystem, etc. Den här avhandlingens huvudsakliga bidrag faller inom tre delområden: formationsreglering, robust decentraliserad tubbaserad icke-linjär modellprediktiv reglering (eng: model predictive control), samt tidsbegränsad kooperativ planering.

I den första delen av avhandlingen föreslås ett decentraliserat modellfritt reglerprotokoll som garanterar att avstånds- och orientationsangivelser för en önskad gruppformation uppnås. Det föreslagna protokollet garanterar att kollisioner mellan agenter undviks samt att agenter som ursprungligen är sammankopplade förblir så. De transienta och stationära förloppen bestäms utifrån användarspecificerade funktioner.

I den andra delen av avhandlingen behandlas följande problem: en grupp agenter vill uppnå en förbestämd konfiguration men har bara tillgång till lokal sensorinformation samt har begränsningar på sina styrsignaler. Agenterna modelleras med icke-linjär dynamik i kontinuerlig tid med additiva störsignaler. Vi föreslår ett modellprediktivt reglerprotokoll som tillåter en decentraliserad implementation. Återkopplingen i protokollet består av två delar: en del som kan beräknas offline, och en del som resulterar från ett optimalt styrproblem som löses i realtid.

I den tredje och sista delen av avhandlingen ges varje agent en övergripande uppgift specificerad i temporallogiken MITL (eng: metric interval temporal logic). Vi förslår en ny abstraheringsalgoritm: varje agents rörelse modelleras med hjälp av ett viktat transitionssystem. Baserat på denna abstraktion kan vi sedan konstruera decentraliserade reglerprotokoll som garanterar att varje agents uppgift genomförs. Våra algoritmer tillåter att transienta bivillkor samt kopplingar mellan agenter kan tas i beaktning.

Acknowledgements

First of all, I would like to express my gratitude to my supervisor Prof. Dimos Dimarogonas for his valuable support, guidance and encouragement, both in my life and my work; his continuous feedback and inspiration has made this thesis possible, and it has been an excellent experience to work with him. I would also like to acknowledge my co-supervisor Prof. Jana Tumova for her great insight, knowledge and assistance; my colleagues Dimitris Boskos, Christos Verginis, Shahab Heshmati-alamdari and Dionysis Palimeris for the continuous offer of assistance and inspiration to me, in the projects that we are working together, and for being good friends.

During the past four years, Division of Decision and Control Systems has been a joyful place to stay and work. I would like through this text to thank first of all Robert Mattila for being a good friend and collaborator and for his continuous support for everything that I needed. Among the great people I have met at the department, I would like to thank, in particular Lars Lindemann, Pedro Pereira, Vahan Petrosyan, Damianos Tranos, Pedro Roque, Linnea Persson, Jezdimir Milosevic, Peter Varnai, Rodrigo Vidal, David Umsonst, Valerio Turri and Manne Held for not only being my colleagues but also part of my life. Sofie Ahlberg, Alexandros Filotheou and Yu Wang for the great collaboration and work during their master thesis. Many thanks also to the rest of my colleagues, the professors and the administrators of the department.

Finally, I would like to thank my family and my girlfriend Linnea Holm for always believing in me and for their constant moral support.

Alexandros Nikou

Stockholm, Sweden
November 2019.

Contents

Abstract **3**

Sammanfattning **5**

Acknowledgements **7**

List of Abbreviations **11**

List of Symbols **14**

List of Figures **17**

1 Introduction **21**
1.1 Motivating Applications 22
1.2 Thesis Outline and Contributions 25
1.3 Related Work 33

2 Notation and Preliminary Background **35**
2.1 Dynamical Systems 37
2.2 Prescribed Performance Control (PPC) 39
2.3 Nonlinear Model Predictive Control (NMPC) 39
2.4 Graph Theory 42
2.5 Time Sequence, Timed Run and Weighted Transition System . . 44
2.6 Metric Interval Temporal Logic (MITL) 45
2.7 Timed Büchi Automata 47

3 Formation Control of Multiple Rigid Bodies **49**
3.1 Introduction 49
3.2 Notation 51
3.3 Problem Formulation 51
3.4 Feedback Control Design 54
3.5 Stability Analysis 62
3.6 Simulation Results 64

3.7 Conclusions 65

4 Decentralized Tube-based NMPC of Uncertain Multi-agent Systems 69
4.1 Introduction 69
4.2 Problem Formulation 72
4.3 Main Results 74
4.4 Simulation Results 81
4.5 Summary 87

5 Time-constrained High-Level Planning of Multi-agent Systems 89
5.1 Introduction 89
5.2 Problem Formulation 91
5.3 Proposed Solution 96
5.4 An Illustrative Example 104
5.5 Conclusions 106

6 Decentralized Abstractions for Time-constrained Planning 107
6.1 Introduction 107
6.2 Notation and Preliminaries 108
6.3 Problem Formulation 109
6.4 Specification 109
6.5 Proposed Solution 113
6.6 Simulation Results 125
6.7 Conclusions 127

7 Scalable Time-constrained Planning of Multi-robot Systems 129
7.1 Introduction 130
7.2 Problem Formulation 131
7.3 Problem Solution 134
7.4 Experimental Results 143
7.5 Summary 153

8 Summary and Future Research Directions 155
8.1 Summary 155
8.2 Future Research Directions 156

A Proofs of Chapter 3 159
A.1 Proof of Theorem 3.1 159
A.2 Proof of Auxiliary Lemma 165

B Proofs of Chapter 4 167
B.1 Proof of Lemma 4.1 167
B.2 Proof of Theorem 4.1 169

C Proofs of Chapter 6 **173**
C.1 Proof of Theorem 6.1 173
C.2 Auxiliary Lemma 176
C.3 Sufficient Conditions for Well-possessedness of the Abstraction . 177

D Tube-based NMPC for Second-order Nonlinear Dynamics **179**

E Tube-based NMPC for Nonholonomic Systems **191**

Bibliography **199**

List of Abbreviations

AUV	Autonomous Underwater Vehicle
UAV	Unmanned Aerial Vehicle
CTL	Computational Tree Logic
DFHOCP	Decentralized Finite Horizon Optimal Control Problem
FHOCP	Finite Horizon Optimal Control Problem
IoT	Internt of Things
ISS	Input-to-State Stable
LTL	Linear Temporal Logic
MILP	Mixed-Integer Linear Programming
MTL	Metric Temporal Logic
MITL	Metric Interval Temporal Logic
NMPC	Nonlinear Model Predictive Control
RoI	Regions of Interest
SML	Smart Mobility Lab
TBA	Timed Büchi Automata
TS	Transition System
VRP	Vehicle Routing Problem
WTS	Weighted Transition System

List of Symbols

$\mathbb{N}$	Set of natural numbers
$\mathbb{Q}$	Set of rational numbers
$\mathbb{R}$	Set of real numbers
N	Number of agents
L	Lipschitz constant
F_I, F_B	Inertial and body-fixed frames
$\mathcal{F}$	Terminal set introduced to guarantee the stability of nominal NMPC
$\mathcal{V} \coloneqq \{1, \ldots, N\}$	Labeling set of the N agents
$\mathcal{N}_i$	Neighboring set of agent $i \in \mathcal{V}$
$\mathcal{G} = (\mathcal{V}, \mathcal{E})$	A graph $\mathcal{G}$ with set of nodes $\mathcal{V}$ and edge set $\mathcal{E}$
$\mathcal{M} \coloneqq \{1, \ldots, M\}$	Labeling set of the edges 1, …, M of a graph $\mathcal{G}$
$\mathscr{L}(\mathcal{G}) \in \mathbb{R}^{N \times N}$	Laplacian matrix of a graph $\mathcal{G}$
$\mathscr{D}(\mathcal{G}) \in \mathbb{R}^{N \times M}$	Incidence matrix of a graph $\mathcal{G}$
$t \in \mathbb{R}_{\geq 0}$	Variable modeling the time
$\mathfrak{d}_i > 0$	Sensing radius of agent i
$\mathfrak{r}_i > 0$	Radius of the ball that covers the volume of agent i
$\mathcal{B}(p, y)$	A ball with center p and radius y
$v \in \mathscr{V}$	Set of velocity constraints
$u \in \mathcal{U}$	Set of control input constraints
$\mathcal{W} \subseteq \mathbb{R}^n$	The workspace that the agents are operating in
$\mathcal{Z} \coloneqq \{1, \ldots, Z\}$	Labeling set for the Regions of Interest (RoI) of the workspace $\mathcal{W}$
$\mathcal{R} \coloneqq \bigcup_{z \in \mathcal{Z}} \mathcal{R}_z$	Union of the Region of Interest (RoI) of the workspace $\mathcal{W}$
$\mathcal{L} : \mathcal{R} \to 2^{\Sigma}$	Labeling function
$A \otimes B$	Kronecker product of the matrices $A, B \in \mathbb{R}^{m \times n}$
$\mathcal{T}_1 \circledast \mathcal{T}_2$	Product between Transition Systems $\mathcal{T}_1$ and $\mathcal{T}_2$

$\delta \in \Delta$	Variable modeling external disturbances and uncertainties
Σ_i	Set of atomic propositions of agent i.
φ_i	Desired MITL high-level task for agent i.
$\tau(l)$, $l \geq 0$	Time stamp at position $l \geq 0$
$r^t = (r(0), \tau(0)) \ldots$	Timed run
$w^t = (w(0), \tau(0)) \ldots$	Timed word
$\mathfrak{t} :\longrightarrow\!\!\!\!\rightarrow \mathbb{Q}_{\geq 0}$	A map that assigns a positive weight to each transition
CL	Finite set of clocks of a TBA
RS	Finite set of clocks to be reset over time of a TBA
FS	Accepted states of a TBA
$\Phi(\text{CL})$	Set of clock constraints
$\mathfrak{g} \in \Phi(\text{CL})$	Guard of an edge of a TBA
$\|x\|_2 \coloneqq \sqrt{x^\top x}$	Euclidean norm
$\|x\|_A \coloneqq \sqrt{x^\top A x}$	Weighted norm
$\|A\| \coloneqq \max\{\|Ax\|_2 : \|x\|_2 = 1\}$	Induced norm of matrix $A \in \mathbb{R}^{n \times n}$
$\mathbb{1}_n \in \mathbb{R}^n$	The column vector with all entries 1
$I_n \in \mathbb{R}^{n \times n}$	The unit matrix of dimension n
$0_{m \times n} \in \mathbb{R}^{m \times n}$	The $m \times n$ matrix with all entries zeros
P, Q, R	Positive definite weight matrices used in running cost and terminal penalty cost functions of NMPC
$T > 0$	Prediction horizon of NMPC
$t_k, k \in \mathbb{N}$	Sampling time step
$h > 0$	Constant sampling period with $t_{k+1} = t_k + h$
$\mathfrak{e} \coloneqq e - \overline{e}$	Deviation between the real error state e and the nominal error state $\overline{e}$
Ω	A set defining the invariant tube centered along the nominal trajectory of an uncertain nonlinear system

List of Figures

1.1 Automated platooning [1]. 22
1.2 Autonomous cars in a highway [2]. 23
1.3 A team of heterogeneous robots operating in a warehouse of a company [3]. 24
1.4 The research fields on which this thesis lies on are the following: robotics, control theory and formal verification. 26

2.1 ISS combines overshot and asymptotic behavior. 37
2.2 Graphical illustration of the prescribed performance definition. . . . 40
2.3 The difference between open-loop prediction and close-loop behavior [4]. 42
2.4 An example of a WTS with 3 states. 46

3.1 Illustration of two agents $i, j \in \mathcal{V}$ in the workspace; F_I is the inertial frame, F_i, F_j are the frames attached to the agents' center of mass, $p_i, p_j \in \mathbb{R}^3$ are the positions of the center of mass with respect to F_I; $\mathfrak{r}_i$, $\mathfrak{r}_j$ are the radii of the agents and $\mathfrak{d}_i > \mathfrak{d}_j$ are their sensing ranges. 53
3.2 The evolution of the distance errors $e_k^p(t)$, along with the performance bounds imposed by $\rho_k^p(t), \forall k \in \{1, 2, 3\}$. 66
3.3 The evolution of the orientation errors $e_{k_n}^q(t)$, along with the performance bounds imposed by $\rho_k^q(t), \forall k, n \in \{1, 2, 3\}$. 66
3.4 The distance between neighboring agents along with the collision and connectivity constraints. 67
3.5 The evolution of the velocity errors $e_{i_m}^v(t)$, along with the performance bounds imposed by $\rho_{i_m}^v(t), \forall i \in \{1, \ldots, 4\}, m \in \{1, \ldots, 6\}$. 67
3.6 The resulting control input signals $u_i(t), i \in \{1, \ldots, 4\}$. 68

4.1 The hyper-tube of agent i centered along the trajectory $\bar{e}_i(t)$ (depicted by blue line). Under the proposed control law, the real trajectory $e_i(t)$ (depicted with red line) lies inside the hyper-tube for all times, i.e., $\mathfrak{e}_i(t) \in \Omega_i$, $\forall t \in \mathbb{R}_{\geq 0}$. 76

4.2 The evolution of the real error signal $e_1(t)$ and the nominal error signal $\overline{e}_1(t)$ over the time interval $[0, 10]$ sec. 83
4.3 The evolution of the trajectory of agent 1 in the workspace $\mathcal{W}$ over the time interval $[0, 10]$ sec. The solid and the dashed lines represent the real and the nominal trajectory, respectively. The gray circles represent the tube. 83
4.4 The evolution of the real error signals $e_2(t)$ and the nominal error signals $\overline{e}_2(t)$ over the time interval $[0, 10]$ sec. 84
4.5 The evolution of the trajectory of agent 2 in the workspace $\mathcal{W}$ over the time interval $[0, 10]$ sec. The solid and the dashed lines represent the real and the nominal trajectory, respectively. The gray circles represent the tube. 84
4.6 The evolution of the real error signals $e_3(t)$ and the nominal error signals $\overline{e}_3(t)$ over the time interval $[0, 10]$ sec. 85
4.7 The evolution of the trajectory of agent 3 in the workspace $\mathcal{W}$ over the time interval $[0, 10]$ sec. The solid and the dashed lines represent the real and the nominal trajectory, respectively. The gray circles represent the tube. 85
4.8 The workspace $\mathcal{W}$ along with the trajectories of all agents. 86
4.9 The distance between the neighboring agents $1-2$ and $2-3$. The distance remains below the threshold $d_1 = d_2 = d_3 = 5$ for all times, i.e., the connectivity of the neighboring agents is preserved for all times. 86
4.10 The control input signals $u_i(t)$, $i \in \mathcal{V}$ satisfying the constraints $u_i(t) \in \mathcal{U}_i$, $\forall i \in \mathcal{V}$, $t \in \mathbb{R}_{\geq 0}$. 87

5.1 Two WTSs $\mathcal{T}_1, \mathcal{T}_2$ representing two agents in $\mathcal{Z}$ with $\Pi_1 = \{\pi_1^1, \pi_2^1, \pi_3^1\}$, $\Pi_1^{\text{init}} = \{\pi_1^1\}$, $\Pi_2 = \{\pi_1^2, \pi_2^2, \pi_3^2\}, \Pi_2^{\text{init}} = \{\pi_1^2\}$. The transitions are depicted with arrows which are annotated with the corresponding weights. 94
5.2 A graphic illustration of the proposed framework 97
5.3 An illustrative example with 2 robots evolving in a common workspace 104
5.4 The accepting runs $\widetilde{r}_1^t, \widetilde{r}_2^t$, the collective run $\widetilde{r}_G^t$ and the corresponding timed stamps. We denote with red dashed lines the times that both agents have the same time stamps 105

6.1 An example of two agents performing in a partitioned workspace. . . 112
6.2 Illustration of a space-time discretization which is well-posed for system (i) but non-well posed for system (ii). 115
6.3 An example with a given cell decomposition $S = \{S_l\}_{l \in \{1,\ldots,6\}}$ from Problem 6.1 and a non-compliant cell decomposition $\bar{S} = \{\bar{S}_l\}_{l \in \bar{\mathbb{l}} = \{1,\ldots,6\}}$ which is the outcome of the proposed abstraction technique. 118
6.4 The resulting compliant cell decomposition $\hat{S} = \{\hat{S}_{\hat{l}}\}_{\hat{l} \in \hat{\mathbb{l}} = \{1,\ldots,15\}}$ of the Example 6.2 which serves as solution to Problem 1. 119

6.5 Timed runs of the agents i, j_1, j_2 122
6.6 A graphic illustration of the proposed framework 124
6.7 A simulation scenario with $N = 5$ agents and $\lambda = 0.10$, $\bar{d}_{\max} = 0.25$. The figure shows the evolution of the agents' reachable cells up to time $t = 15\delta t$. 125

7.1 A graphic illustration of the proposed framework. 141
7.2 A Nexus 10011 mobile robot with an attached 3 Degrees of Freedom (DoF) manipulator. 144
7.3 The experimental setup demonstrating the proposed framework. Three Nexus 10011 mobile robots, in the workspace of Smart Mobility Lab (SML) [5] that contains 5 RoI. 145
7.4 A panoramic view of the workspace with the 5 RoI. 147
7.5 The trajectory of robot 1 in the workspace. 147
7.6 The trajectory of robot 2 in the workspace. 148
7.7 The trajectory of robot 3 in the workspace. 148
7.8 The evolution of the control signal $u_{1,1}$ or robot 1 over the time. . . 149
7.9 The evolution of the control signal $u_{1,2}$ or robot 1 over the time. . . 149
7.10 The evolution of the control signal $u_{1,3}$ or robot 1 over the time. . . 150
7.11 The evolution of the control signal $u_{2,1}$ or robot 2 over the time. . . 150
7.12 The evolution of the control signal $u_{2,2}$ or robot 2 over the time. . . 151
7.13 The evolution of the control signal $u_{2,3}$ or robot 2 over the time. . . 151
7.14 The evolution of the control signal $u_{3,1}$ or robot 3 over the time. . . 152
7.15 The evolution of the control signal $u_{3,2}$ or robot 3 over the time. . . 152
7.16 The evolution of the control signal $u_{3,3}$ or robot 3 over the time. . . 153

D.1 The hyper-tube centered along the trajectory $\overline{e}(t)$ (depicted by blue line) with radius $\frac{\widetilde{\delta}}{\min\{\alpha_1,\alpha_2\}}$. Under the proposed control law, the real trajectory $e(t)$ (depicted with red line) lies inside the hyper-tube for all times, i.e., $\|\mathfrak{e}(t)\|_2 \leq \frac{\widetilde{\delta}}{\min\{\alpha_1,\alpha_2\}}$, $\forall t \in \mathbb{R}_{\geq 0}$. 185

E.1 The nonholonomic underwater vehicle. Blue color indicates the actuated degrees of freedom. 192

Chapter 1

Introduction

Autonomous robots such as cars, drones, underwater vehicles and domestic service robots appear in our everyday life more and more frequently. In recent years, the technical development and manufacturing of industrial and domestic robots have been boosted by an unprecedented evolution of digital processing. Robots and embedded computers have become more powerful in terms of speed and capacity, and at the same time more affordable. An essential feature of autonomous robots is that they are expected to comprehend tasks specified by non-expert end-users, reason about them, figure out a plan and more importantly execute the plan to accomplish the tasks with or without minimal human intervention.

Wireless communication technology enables autonomous robots to be connected with each other and with internal or external sensors. The emerging Internet of Things (IoT) allows robots to have more accurate and up-to-date information about their operation space. Even better, cloud-computing platforms provide them on-demand access to computing and storage resources for a vast amount of real-time data.

The aforementioned increase in connectivity leads to networked autonomous system paradigms where the focus is placed specifically on on the interplay of robots, introducing new opportunities and challenges. A group of coordinated autonomous robots can be more efficient and achieve more complex tasks than a collection of independent single robots [6]. For example, four coordinated robots can together carry a heavy object which cannot be carried by them independently. The effectiveness of a multi-robot system is closely related to the complexity and flexibility of the underlying coordination scheme. Constraints and additionally imposed rules are often specified as well and need to be respected by such systems. Local and global task specifications can be distributed among robots such that they are subject to local as well as collaborative tasks. Such an example is automated truck platooning [1] aiming at optimizing the fuel consumption of trucks while respecting delivery deadlines of each single truck (local task) and at improving traffic safety as shown in Fig. 1.1. The agents (trucks) are required to remain within certain distances with the neighboring trucks (global task).

Figure 1.1: Automated platooning [1].

Moreover, uncertainty in the control design process needs to be taken into consideration. Thus, the core problem for any multi-robot application is to design a suitable coordination framework that can improve the overall performance, i.e., satisfying local and global tasks if possible, while keeping the cost and complexity of coordination low.

All issues mentioned above call for a framework for the modeling, design and analysis of interconnected multi-robot systems under complex individual tasks. In the remainder of this chapter, we provide some motivating applications of this thesis, which are followed by the main contributions and the thesis outline.

1.1 Motivating Applications

In this section, some real-life applications that motivate the topics addressed in this thesis are provided.

Autonomous cars, also known as self-driving cars, are capable of sensing the traffic environment and navigating to desired destinations without human guidance (Fig. 1.2). The last decade has witnessed an increasing interest in autonomous driving technologies from both academia and industry. Almost every major car manufacturer around the world is developing autonomous or driver assistance technology to upgrade its existing models, e.g., adaptive cruise control, automatic parking, collision avoidance and highway platooning. The prototype of Google's self-driving car has been tested on road for over 1 million miles [7]. Note in particular the nature of such connected network of cars, which requires avoiding collisions at any cost while satisfying local task objectives like:

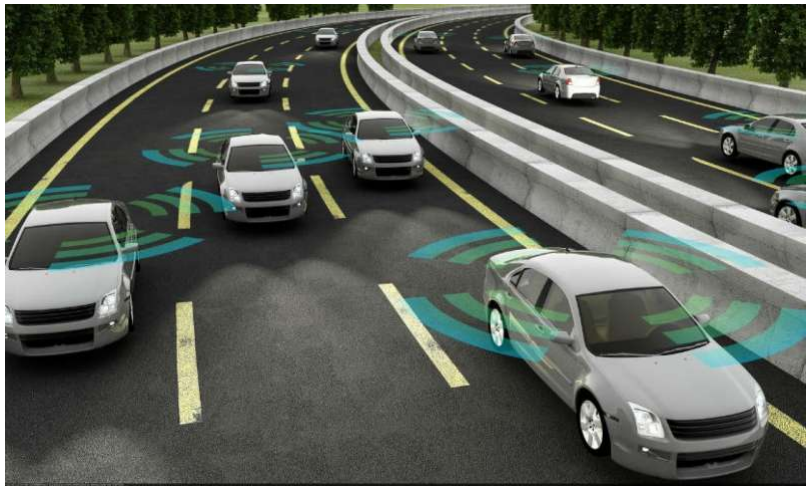

Figure 1.2: Autonomous cars in a highway [2].

- "eventually arrive at home before the evening, but on the way pass by the supermarket, and always avoid highways";

- "go to the pharmacy before the weekend and park in a close-by area with free parking".

These tasks are clearly much more complex than just "going from A to B", but still quite common in our daily driving routine. They all have a temporal characteristic for different locations to visit and there are properties of interest attached to these locations. To put on another layer of complexity, as in particular analyzed in this thesis, these tasks may include quantitative timing constraints, i.e., a task should be satisfied within a certain time interval. This means that an extra layer of logical reasoning and planning is required to find a higher-level plan as a sequence of locations to visit and then merge this plan with navigation techniques.

In order to achieve coordinated group behavior under local and global tasks, new coordination paradigms are needed, as will be introduced in the thesis. In particular, consider a scenario with a team of heterogeneous robots (such as manipulators, mobile robots, quadrotors, etc.) operating in a warehouse of a company (see Fig. 1.3). The robots are governed by uncertain nonlinear dynamics and their tasks may be given as follows:

Figure 1.3: A team of heterogeneous robots operating in a warehouse of a company [3].

- local high-level tasks such us "place the specific box at a specific location of the warehouse within certain time constraints and return to the base to await new delivery tasks";
- global tasks such as connectivity maintenance with neighboring robots so that teams of robots are collaborating in order to be able to carry heavy-weight loads;
- global tasks such collision avoidance among the robots in the warehouse.

Motivated by the above applications, the main contribution of this thesis is to propose novel methodologies and algorithms which solve the following three classes of problems.

- **Problem 1** Consider a multi-agent system modeled by Newton-Euler dynamics. The goal is to design model-free feedback control strategies that use only local information for the neighboring agents such that a predefined distance- and orientation-based formation between the initially connected agents, is achieved. In parallel to that, inter-agent collisions should be avoided.

- **Problem 2** Consider an uncertain multi-agent dynamical system under the presence of state and input constraints. Then, design novel robust feedback control strategies for navigation of the agents to predefined configurations of

the workspace. Moreover, ensure that connectivity of the initially connected neighboring agents is preserved.

- **Problem 3** Consider a multi-agent system in which individual as well as collaborative high-level tasks are assigned to each agent. By taking into account that the assigned high-level tasks should be fulfilled within certain time constraints, design decentralized control laws such that each agent fulfills the desired specifications.

Taking the aforementioned into consideration, this thesis is divided into three parts. Each part deals with novel methodologies and algorithms for solving Problems 1-3, respectively. The research areas that this thesis lies in are robotics, control theory and formal verification, as shown in Figure 1.4. Finally, the work developed in this thesis is embedded in the European Research Projects BUCOPHSYS [8] and CO4ROBOTS [9]. The next section considers the outline of this thesis.

1.2 Thesis Outline and Contributions

In this section, we provide the outline of the thesis and indicate the contributions of each chapter. Chapter 2 is devoted to the notation that will be adopted in this thesis and preliminary background knowledge. The thesis is divided into *three main parts* which address Problems 1-3 that were previously mentioned.

- The *first part* consists of Chapter 3. In this part, we propose a novel decentralized control protocol for each agent of a multi-agent system consisting of N rigid bodies governed by Newton-Euler dynamics. The proposed model-free feedback control scheme guarantees distance- and orientation-based formation of the neighbors of the initial graph, inter-agent collision avoidance, as well as connectivity maintenance with the initially connected agents.

- The *second part* consists of Chapter 4. In this part, we investigate robust Nonlinear Model Predictive Control (NMPC) methodologies and their application to multi-agent systems. In particular, by considering a team of agents with uncertain nonlinear dynamics, we propose decentralized control protocols such that the agents are navigated to predefined configurations while transient constraints such as connectivity maintenance are imposed to the system.

- The *third part* consists of Chapters 5, 6 and 7. Here we deal with the problem of high-level planning of multi-agent systems under time constraints. By

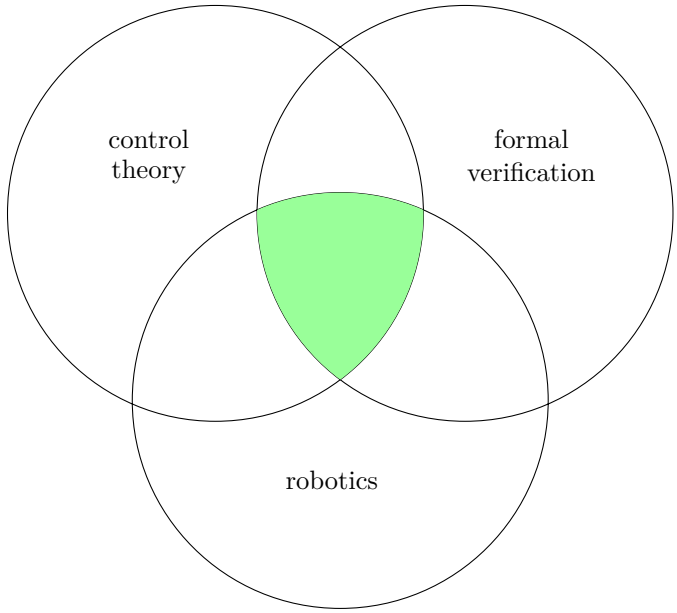

Figure 1.4: The research fields on which this thesis lies on are the following: robotics, control theory and formal verification.

providing suitable decentralized abstraction techniques, each agent's motion is captured through a Weighted Transition System (WTS). In the sequel, we propose control synthesis tools to guarantee the satisfaction of the Metric Interval Temporal Logic (MITL) formulas which are assigned to each agent. In this part, we work both at continuous and discrete levels.

Chapter 3

In this chapter, we address the problem of distance- and orientation-based formation control of a class of second order nonlinear multi-agent systems in a 3D workspace, under static and undirected communication topologies. More specifically, we design a decentralized model-free control protocol in the sense that each agent uses only local information from its neighbors to calculate its own control signal, without incorporating any knowledge of the model nonlinearities and exogenous disturbances. Moreover, the transient and steady state responses are solely determined by certain designer-specified performance functions and are fully decoupled by the agents' dynamic model, the control gain selection, the underlying graph topology as well as the initial conditions. Additionally, by introducing certain inter-agent distance

constraints, we guarantee collision avoidance and connectivity maintenance between neighboring agents. The covered material is based on the following contribution [10]:

- [C8] [1] Alexandros Nikou, Christos K. Verginis and Dimos V. Dimarogonas, "Robust Distance-Based Formation Control of Multiple Rigid Bodies with Orientation Alignment", 20th World Congress of the International Federation of Automatic Control (IFAC WC), Volume 50, Issue 1, Pages 15458-15463, Toulouse, France, July 2017.

and related work has appeared in the following papers [11, 12]:

- [C9] Christos K. Verginis, Alexandros Nikou and Dimos V. Dimarogonas, "Position and Orientation Based Formation Control of Multiple Rigid Bodies with Collision Avoidance and Connectivity Maintenance", 56th IEEE Conference on Decision and Control (CDC), Pages 411-416, Melbourne, Australia, December 2017.
- [J4] Christos K. Verginis, Alexandros Nikou and Dimos V. Dimarogonas, "Robust Formation Control in SE(3) for Tree-graph Structures with Prescribed Transient and Steady State Performance", Automatica, Volume 103, Pages 538-548, May 2019.

Chapter 4

In this chapter, we investigate the problem of decentralized tube-based NMPC for a general class of uncertain nonlinear continuous-time multi-agent systems with additive and bounded exogenous disturbances/uncertainties. In particular, the problem of robust navigation of a multi-agent system to predefined states of the workspace while using only local information is addressed, under certain distance and control input constraints. We propose a decentralized feedback control protocol that consists of two terms: a nominal control input, which is computed online and is the outcome of a Decentralized Finite Horizon Optimal Control Problem (DFHOCP) that each agent solves at every sampling time step, for its nominal system dynamics; and an additive state feedback law which is computed offline and guarantees that the real trajectories of each agent belong to a hyper-tube centered along the nominal trajectory. The volume of the hyper-tube depends on the upper bound of the disturbances as well as the bounds of the derivatives of the dynamics. In addition, by introducing certain distance constraints, the proposed scheme guarantees that the initially connected agents remain connected for all times. Under standard assumptions that arise in nominal NMPC schemes, controllability assumptions as well as communication capabilities between the agents, it is guaranteed that the

[1] The notations C, J and T stand for conference, journal and thesis publications, respectively, enumerated by chronological order of publication date.

multi-agent system is Input-to-State Stable (ISS) with respect to the disturbances, for all initial conditions satisfying the state constraints. The aforementioned results are based on the following publications [13–18]:

- [J5] Alexandros Nikou and Dimos V. Dimarogonas, "Decentralized Tube-Based Robust Model Predictive Control for Uncertain Nonlinear Multi-Agent Systems", International Journal of Robust and Nonlinear Control (IJRNC), Volume 29, Issue 10, Pages 2799-2818, July 2019.
- [C14] Alexandros Nikou and Dimos V. Dimarogonas, "Robust Tube-based Model Predictive Control for Timed Constrained Robot Navigation", American Control Conference (ACC), Pages 1152-1157, Philadelphia, USA, July 2019.
- [C13] Alexandros Nikou, Christos K. Verginis and Dimos V. Dimarogonas, "A Tube-based Nonlinear MPC Scheme for Interaction Control of Underwater Vehicle Manipulator Systems", IEEE OES Autonomous Underwater Vehicle Symposium (AUVS), Pages 1-6, Porto, Portugal, November 2018.
- [C15] Alexandros Nikou, Shahab Heshmati-almadari and Dimos V. Dimarogonas, "Design and Experimental Validation of Tube-based MPC for Timed-constrained Robot Planning", IEEE International Conference on Automation Science and Engineering (CASE), Pages 1181-1186, Vancouver, Canada, August 2019.
- [C16] Shahab Heshmati-almadari, Alexandros Nikou and Dimos V. Dimarogonas, "Robust Trajectory Tracking Control for Underactuated Autonomous Underwater Vehicles", 58th IEEE Conference on Decision and Control (CDC), Nice, France, December 2019 (to appear).
- [J6] Shahab Heshmati-alamdari, Alexandros Nikou and Dimos V. Dimarogonas, "Robust Trajectory Tracking Control for Underctuated Autonomous Underwater Vehicles in Uncertain Environments", IEEE Transactions on Automation Science and Engineering (TASE), 2019 (under review).

Chapter 5

In this chapter, the problem of cooperative task planning of multi-agent systems subject to imposed time constraints is investigated. We consider time constraints given by Metric Interval Temporal Logic (MITL). We propose a method for automatic control synthesis in a two-stage systematic procedure. Under the proposed method, it is guaranteed that all the agents satisfy their own individual task specifications as well as that the team satisfies a team task specification. The covered material is based on the following contributions [19–21]:

- [C1] Alexandros Nikou, Jana Tumova and Dimos V. Dimarogonas, "Cooperative Task Planning Synthesis for Multi-Agent Systems Under Timed Temporal Specifications", American Control Conference (ACC), Pages 13-19, Boston, USA, July 2016.
- [C5] Sofie Andersson, Alexandros Nikou and Dimos V. Dimarogonas, "Control Synthesis for Multi-Agent Systems under Metric Interval Temporal Logic Specifications", 20th World Congress of the International Federation of Automatic Control (IFAC WC), Volume 50, Issue 1, Pages 2397-2402, Toulouse, France, July 2017.
- [T1] Alexandros Nikou, "Cooperative Planning Control and Formation Control of Multi-agent Systems", Licentiate Dissertation, KTH Royal Institute of Technology, June 2017.

Chapter 6

In this chapter, a fully automated procedure for controller synthesis for multi-agent systems under coupling constraints is presented. Each agent is modeled with dynamics consisting of two terms: the first one models the coupling constraints and the second one is an additional bounded control input. We aim to design these inputs so that each agent meets an individual high-level specification defined using a Metric Interval Temporal Logic (MITL). First, a decentralized abstraction that provides a space and time discretization of the multi-agent system is designed. Second, by utilizing this abstraction and techniques from formal verification, we propose an algorithm that computes the individual runs which provably satisfy the high-level tasks. The overall approach is demonstrated in a simulation example conducted in the MATLAB environment. The covered material is based on the following contributions [22–24]:

- [J2] Alexandros Nikou, Dimitris Boskos, Jana Tumova and Dimos V. Dimarogonas, "On the Timed Temporal Logic Planning of Coupled Multi-Agent Systems", Volume 97, Pages 339-345, Automatica, November 2018.
- [C3] Alexandros Nikou, Dimitris Boskos, Jana Tumova and Dimos V. Dimarogonas, "Cooperative Planning Synthesis for Coupled Multi-Agent Systems Under Timed Temporal Specifications", American Control Conference (ACC), Pages 1847-1852, Seattle, USA, July 2017.
- [C10] Alexandros Nikou, Shahab Heshmati-alamdari, Christos K. Verginis and Dimos V. Dimarogonas, "Decentralized Abstractions and Timed Constrained Planning of a General Class of Coupled Multi-Agent Systems", 56th IEEE Conference on Decision and Control (CDC), Pages 990-995, Melbourne, Australia, December 2017.

Chapter 7

In this chapter, we consider a scalable procedure for time-constrained planning of a class of nonlinear multi-robot systems. In particular, we consider N robotic agents operating in a workspace which contains Regions of Interest (RoI), in which atomic propositions for each agent are assigned. The main goal is to design decentralized control laws so that each agent meets an individual high-level specification given in MITL, while using only local information based on a limited sensing radius. Furthermore, the agents need to fulfill certain desired transient constraints such as collision avoidance between them. The controllers, which guarantee the transition between RoI, are the solution of a Decentralized Finite-Horizon Optimal Control Problem (DFHOCP) and serve as actions for the individual WTS of each agent. The time duration required for the transition between RoI is modeled by a weight. The DFHOCP is solved at every sampling time step by each agent and then necessary information is exchanged between neighboring agents. The proposed approach is scalable since it does not require a product computation among the WTS of the agents. The proposed framework is experimentally tested and the results show that the approach is promising for solving real-life robotic as well as industrial applications. The covered material is based on the following contributions [16, 25–28]:

- [J8] Alexandros Nikou, Shahab Heshmati-alamdari and Dimos V. Dimarogonas, "Scalable Time-constrained Planning of Multi-robot Systems", Autonomous Robots, 2019 (under review).

- [J7] Alexandros Nikou, Christos K. Verginis, Dimos V. Dimarogonas, "A Robust Nonlinear MPC Framework for Control of Underwater Vehicle Manipulator Systems under High-Level Tasks", IET Control Theory and Applications, 2019 (under review).

- [C15] Alexandros Nikou, Shahab Heshmati-almadari and Dimos V. Dimarogonas, "Design and Experimental Validation of Tube-based MPC for Timed-constrained Robot Planning", IEEE International Conference on Automation Science and Engineering (CASE), Pages 1181-1186, Vancouver, Canada, August 2019.

- [J1] Alexandros Filotheou, Alexandros Nikou and Dimos V. Dimarogonas, "Robust Decentralized Navigation of Multi-Agent Systems with Collision Avoidance and Connectivity Maintenance Using Model Predictive Controllers", International Journal of Control (IJC), Pages 1-15, August 2018.

- [C11] Alexandros Filotheou, Alexandros Nikou and Dimos V. Dimarogonas, "Decentralized Control of Uncertain Multi-Agent Systems with Connectivity Maintenance and Collision Avoidance", European Control Conference (ECC), Pages 8-13, Limassol, Cyprus, June 2018.

Chapter 8

In this chapter, the thesis is concluded with a summary and discussion of the results, where some future research directions are also presented.

Appendixes

Appendix A, Appendix B and Appendix C are devoted to technical proofs that are required in Chapter 3, Chapter 4 and Chapter 6, respectively. Appendix D deals with tube-based NMPC design for second-order uncertain nonlinear systems, while Appendix E deals with tube-based NMPC for nonholonomic systems. In particular, the tube-based methodology that is derived in previous chapters is applied to solve a stabilization task to a desired configuration of the workspace of a nonholonomic underwater vehicle.

Contributions not included in this thesis

The following publications are not covered in this thesis, but are related to the work presented here [29–34]:

- [C4] Alexandros Nikou, Jana Tumova and Dimos V. Dimarogonas, "Probabilistic Plan Synthesis for Coupled Multi-Agent Systems", 20th World Congress of the International Federation of Automatic Control (IFAC WC), Volume 50, Issue 1, Pages 10766-10771, Toulouse, France, July 2017.
- [J3] Shahab Heshmati-alamdari, Charalampos P. Bechlioulis, George C. Karras, Alexandros Nikou, Dimos V. Dimarogonas, Kostas J. Kyriakopoulos, "A Robust Interaction Control Approach for Underwater Vehicle Manipulator Systems", Annual Reviews in Control, Volume 46, Pages 315-325, January 2018.
- [C8] Alexandros Nikou, Christos K. Verginis, Shahab Heshmati-alamdari and Dimos V. Dimarogonas, "A Nonlinear Model Predictive Control Scheme for Cooperative Manipulation with Singularity and Collision Avoidance", 25th IEEE Mediterranean Conference on Control and Automation (MED), Pages 707-712, Valletta, Malta, June 2017.
- [C12] Christos K. Verginis, Alexandros Nikou and Dimos V. Dimarogonas, "Communication-based Decentralized Cooperative Object Transportation Using Nonlinear Model Predictive Control", European Control Conference (ECC), Pages 733-738, Limassol, Cyprus, June 2018.
- [C6] Shahab Heshmati-alamdari, Alexandros Nikou, Kostas J. Kyriakopoulos and Dimos V. Dimarogonas, "A Robust Control Approach for Underwater Vehicle Manipulator Systems in Interaction with Compliant Environments", 20th World Congress of the International Federation of Automatic Control (IFAC WC), Volume 50, Issue 1, Toulouse, France, July 2017.
- [C1] Alexandros Nikou, Georgios Gavridis and Kostas J. Kyriakopoulos, "Mechanical Design, Modelling and Control of a Novel Aerial Manipulator", IEEE International Conference on Robotics and Automation (ICRA), Pages 4698 - 4703, Seattle, USA, May 2015.

Author's Contributions

Wherever listed as the first author of a publication, the author of this thesis had the most significant role in developing the results, and has completed the majority of the writing.

1.3 Related Work

In this section, we provide a review of some related work to the three problems considered in this thesis.

Problem 1

The literature in formation control is traditionally categorized in single or double integrator agent dynamics and directed or undirected communication topologies (see e.g. [35–50]). Orientation-based formation control has been addressed in [51–54], whereas the authors in [53, 55, 56] have considered the combination of distance- and orientation-based formation.

Problem 2

NMPC has been proven to be a powerful control framework for dealing with the problem of stabilization of dynamical systems under state and input constraints [57–65]. One of the main challenges in NMPC is the treatment of potential uncertainties due to imperfect modeling and/or disturbances that may affect the system.

The problem of robust NMPC has been extensively investigated over the last years. Authors in [66, 67] proposed a method of constraint sets tightening for guaranteeing robust stability. These approaches are relatively conservative since the constraints are tightened proportionally with the length of the horizon. Thus, they cannot be applied in problems where a larger prediction horizon T is required, since this can lead to infeasibility of constraint satisfaction. A robust MPC approach for solving a min-max optimization problem online in the presence of uncertainties has been proposed in [68–70]. We argue that such schemes are computationally intractable since the complexity of the resulting optimization problem grows exponentially with the increase of the predicted horizon. A promising robust strategy, originally proposed for discrete-time linear systems in [71–73], is the so-called tube-based approach.

Tube-based approaches for nonlinear discrete-time systems have been considered in [74–78]. The authors in [79] have addressed the linear continuous-time case. In [80, 81], regarding the computation of the offline feedback controller, the discrepancy between the nominal nonlinear system with the corresponding linear system has been considered.

Problem 3

The specification language that has extensively been used to express desired tasks for robots is Linear Temporal Logic (LTL) (see, e.g., [82–90]). LTL has proven a valuable tool for controller synthesis, because it provides a compact mathematical

formalism for specifying desired behaviors of a system [91–97]. There is a rich body of literature containing algorithms for verification and synthesis of multi-agent systems under temporal logic specifications [98].

Explicit time constraints in the system modeling have been included e.g., in [84], where a method of automated planning of optimal paths of a group of agents satisfying a common high-level mission specification was proposed. The mission was given in LTL and the goal was the minimization of a cost function that captures the maximum time between successive satisfactions of the formula. Authors in [99, 100] used a different approach, representing the motion of each agent in the environment with a timed automaton. The composition of the team automaton was achieved through synchronization and the UPPAAL verification tool ([101]) was utilized for specifications given in Computational Tree Logic (CTL). In the same direction, authors in [102] modeled the multi-robot framework with timed automata and weighted transition systems considering LTL specifications and then, an optimal motion of the robots satisfying instances of the optimizing proposition was proposed.

Most of the previous works on multi-agent planning consider temporal properties which essentially treat time in a qualitative manner. For real applications, a multi-agent team might be required to perform a specific task within a certain time bound, rather than at some arbitrary time in the future (quantitative manner). Controller synthesis under timed specifications for the single-agent case has been considered in [103–109].

Abstractions for both single- and multi-agent systems can be found in [110–116]. Compositional frameworks are provided in [114] for safety specifications of discrete time systems, and [115], which is focused on feedback linearizable systems with a cascade interconnection. In addition, local invariant sets for discrete time coupled linear systems are considered in [117] and are leveraged for control synthesis.

Chapter 2

Notation and Preliminary Background

In this chapter, the notation that will be used in this thesis as well as the necessary preliminary background are provided.

We denote by $\mathbb{R}$, $\mathbb{Q}$ and $\mathbb{N}$ the set of real, rational and natural numbers including 0, respectively; $\mathbb{R}_{\geq 0}$ and $\mathbb{R}_{>0}$ are the sets of nonnegative and positive real numbers, respectively. Given a set $\mathcal{S}$, denote by $|\mathcal{S}|$ its cardinality, by $\mathcal{S}^n \coloneqq \mathcal{S} \times \cdots \times \mathcal{S}$ its n-fold Cartesian product, and by $2^{\mathcal{S}}$ the set of all its subsets; $\partial\mathcal{S}$ stands for the boundary of the set $\mathcal{S}$;

$$\|x\|_2 \coloneqq \sqrt{x^\top x} \quad \text{and} \quad \|x\|_A \coloneqq \sqrt{x^\top A x}, \quad A \geq 0,$$

stand for the Euclidean and the weighted norm of a vector $y \in \mathbb{R}^n$, respectively; $\|A\| \coloneqq \max\{\|Ax\| : \|x\| = 1\}$ stands for the induced norm of a matrix $A \in \mathbb{R}^{n\times n}$; $\lambda_{\min}(A)$ ($\lambda_{\max}(A)$) stands for the minimum (maximum) absolute value of the real part of the eigenvalues of $A \in \mathbb{R}^{n\times n}$. Define by $\mathbb{1}_n \in \mathbb{R}^n$, $I_n \in \mathbb{R}^{n\times n}$ and $0_{m\times n} \in \mathbb{R}^{m\times n}$ the column vector with all entries 1, the unit matrix and the $m \times n$ matrix with all entries zeros, respectively. The notation $\operatorname{diag}\{A_1, \ldots, A_n\}$ stands for the block diagonal matrix with the matrices A_1, …, A_n in the main diagonal; $A \otimes B$ denotes the Kronecker product of the matrices A, $B \in \mathbb{R}^{m\times n}$ (see [118]). The set

$$\mathcal{B}(p, \mathfrak{r}) \coloneqq \big\{y \in \mathbb{R}^n : \|y - p\|_2 \leq \mathfrak{r}\big\},$$

represents the n-dimensional ball with center $x \in \mathbb{R}^n$ and radius $\mathfrak{r} \in \mathbb{R}_{>0}$. Define a vector of a canonical basis of $\mathbb{R}^n$ by:

$$\mathfrak{l}_n^i \coloneqq \Big[0, \ldots, 0, \underbrace{1}_{i-\text{th element}}, 0, \ldots, 0\Big]^\top \in \mathbb{R}^n. \tag{2.1}$$

Definition 2.1. Given two vectors x, $y \in \mathbb{R}^n$, the *convex hull* of the set $\{x, y\}$ is defined by:

$$\operatorname{Co}(x, y) \coloneqq \big\{\theta x + (1-\theta)y,\ \theta \in [0, 1]\big\}.$$

Lemma 2.1. *For any constant $\varrho > 0$, vectors x, $y \in \mathbb{R}^n$ and positive semi-definite matrix $A \in \mathbb{R}^{n\times n}$, it holds that:*

$$x^\top Ay \leq \frac{1}{4\varrho} x^\top Ay + \varrho\ x^\top Ay.$$

Proof. By using the fact that $\varrho > 0$ and A is positive semi-definite the following equivalences hold:

$$\begin{aligned}
& (x - 2\varrho y)^\top A\ (x - 2\varrho y) \geq 0 \\
\iff\ & x^\top Ax - 2\varrho x^\top Ay - 2\varrho y^\top Ax + (2\varrho)^2 y^\top Ay \geq 0 \\
\iff\ & 2\varrho \left(x^\top Ay + y^\top Ax \right) \leq x^\top Ax + (2\varrho)^2 y^\top Ay \\
\iff\ & 4\varrho\ x^\top Ay \leq x^\top Ax + (2\varrho)^2 y^\top Ay \\
\iff\ & x^\top Ay \leq \frac{1}{4\varrho} x^\top Ay + \varrho\ x^\top Ay,
\end{aligned}$$

which concludes the proof. □

Definition 2.2. [119] Given the sets $\mathcal{S}_1$, $\mathcal{S}_2 \subseteq \mathbb{R}^n$, the *Minkowski addition* and the *Pontryagin difference* are respectively defined by:

$$\begin{aligned}
\mathcal{S}_1 \oplus \mathcal{S}_2 &\coloneqq \{s_1 + s_2 \in \mathbb{R}^n : s_1 \in \mathcal{S}_1,\ s_2 \in \mathcal{S}_2\}, \\
\mathcal{S}_1 \ominus \mathcal{S}_2 &\coloneqq \{s_1 \in \mathbb{R}^n : s_1 + s_2 \in \mathcal{S}_1,\ \forall s_2 \in \mathcal{S}_2\}.
\end{aligned}$$

Proposition 1. [120, 121] Consider a vector valued function $f : \mathbb{R}^n \to \mathbb{R}^m$. Assume that f is differentiable on an open set $\mathcal{S} \subseteq \mathbb{R}^n$. Let x, y be two vectors of $\mathcal{S}$ such that $\mathrm{Co}(x, y) \subseteq \mathcal{S}$. Then, there exist vectors $\xi_1, \ldots, \xi_m \in \mathrm{Co}(x, y)$, $\xi_i \neq x$ and $\xi_i \neq y$, $\forall i = 1, \ldots, m$, such that:

$$f(x) - f(y) = \left[\sum_{k=1}^{m} \sum_{j=1}^{n} \mathbb{I}_m^k (\mathbb{I}_n^j)^\top \frac{\partial f_k(\xi_k)}{\partial x_j} \right] (x - y). \tag{2.2}$$

Definition 2.3. ([122]) A continuous function $\boldsymbol{\alpha} : [0, a) \to \mathbb{R}_{\geq 0}$ is said to belong to *class* $\mathcal{K}$, if it is strictly increasing and $\boldsymbol{\alpha}(0) = 0$. It is said to belong to class $\mathcal{K}_\infty$ if $a = \infty$ and $\boldsymbol{\alpha}(r) \to \infty$, as $r \to \infty$.

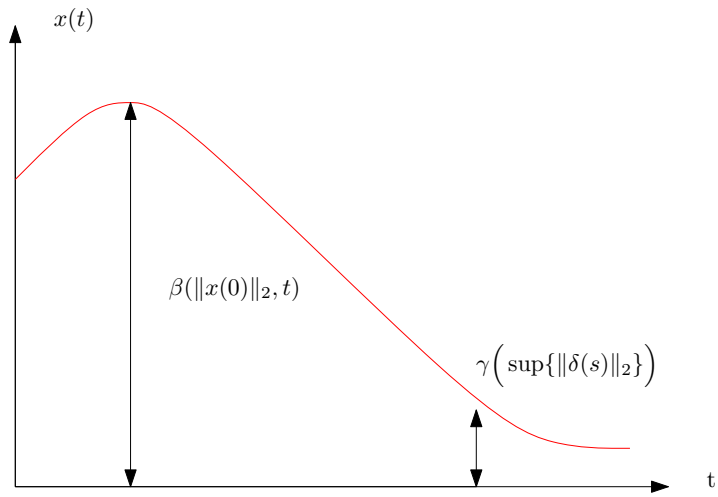

Figure 2.1: ISS combines overshot and asymptotic behavior.

Definition 2.4. ([122]) A continuous function $\boldsymbol{\beta} : [0, a) \times \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$ is said to belong to *class* $\mathcal{KL}$, if:

- for each fixed s, $\boldsymbol{\beta}(r, s) \in \mathcal{K}$ with respect to r;
- for each fixed r, $\boldsymbol{\beta}(r, s)$ is decreasing with respect to s and $\boldsymbol{\beta}(r, s) \to 0$, at $s \to \infty$.

2.1 Dynamical Systems

Definition 2.5. [123] Consider a dynamical system:

$$\dot{x} = f(x, u, \delta),\ x \in \mathcal{X},\ u \in \mathcal{U},\ \delta \in \mathcal{D},$$

with initial condition $x(0) \in \mathcal{X}$. A set $\mathcal{X}' \subseteq \mathcal{X}$ is a *Robust Control Invariant (RCI) set* for the system, if there exists a feedback control law $u := \kappa(x) \in \mathcal{U}$, such that for all $x(0) \in \mathcal{X}'$ and for all disturbances $\delta \in \mathcal{D}$ it holds that $x(t) \in \mathcal{X}'$ for all $t \geq 0$, along every solution $x(t)$.

Definition 2.6. [124] A nonlinear system

$$\dot{x} = f(x, u, \delta),\ x \in \mathcal{X},\ u \in \mathcal{U},\ \delta \in \mathcal{D},$$

with initial condition $x(0) \in \mathcal{X}$ is said to be *Input-to-State Stable (ISS)* with respect to $\delta \in \mathcal{D}$, if there exist functions $\boldsymbol{\beta} \in \mathcal{KL}$, $\boldsymbol{\gamma} \in \mathcal{K}$ such that for any initial condition $x(0) \in \mathcal{X}$, the solution $x(t)$ exists for all $t \geq 0$ and satisfies:

$$\|x(t)\|_2 \leq \boldsymbol{\beta}\big(\|x(0)\|_2, t\big) + \boldsymbol{\gamma}\left(\sup_{0 \leq s \leq t} \|\delta(s)\|_2\right).$$

Remark 2.1. Intuitively, the definition of ISS requires that, as t grows, the size of the state must be bounded by some function of the amplitude of the disturbance because $\boldsymbol{\beta}\big(\|x(0)\|_2, t\big) \to 0$ as t grows. On the other hand, the $\boldsymbol{\beta}\big(\|x(0)\|_2, t\big)$ term may dominate for small t, and this serves to quanfity the magnitude of the transient (overshoot) behavior as a function of the size of the initial state $x(0)$ (Fig. 2.1).

Consider the initial value problem:

$$\dot{x} = f(t, x),\ x(0) \in \mathcal{X}, \tag{2.3}$$

with $f : \mathbb{R}_{\geq 0} \times \mathcal{X} \to \mathbb{R}^n$, where $\mathcal{X} \subseteq \mathbb{R}^n$ is a non-empty open set.

Definition 2.7. ([125]) A solution $x(t)$ of the initial value problem (2.3) is *maximal* if it has no proper right extension that is also a solution of (2.3).

Theorem 2.1. *([125]) Consider the initial value problem* (2.3)*. Assume that* $f(t, x)$ *is:*

- *locally Lipschitz in* x *for almost all* $t \in \mathbb{R}_{\geq 0}$*;*
- *piecewise continuous in* t *for each fixed* $x \in \mathcal{X}$*;*
- *locally integrable in* t *for each fixed* $x \in \mathcal{X}$*.*

Then, there exists a maximal solution $x(t)$ *of* (2.3) *on the time interval* $[0, \tau_{\max})$*, with* $\tau_{\max} \in \mathbb{R}_{>0}$ *such that* $x(t) \in \mathcal{X},\ \forall t \in [0, \tau_{\max})$*.*

Proposition 2. ([125]) Assume that the hypotheses of Theorem 2.1 hold. For a maximal solution $x(t)$ on the time interval $[0, \tau_{\max})$ with $\tau_{\max} < \infty$ and for any compact set $\mathcal{X}' \subseteq \mathcal{X}$, there exists a time instant $t' \in [0, \tau_{\max})$ such that $x(t') \notin \mathcal{X}'$.

2.2 Prescribed Performance Control (PPC)

Prescribed Performance Control (PPC), originally proposed in [126, 127], describes the behavior where a tracking error $e(t) : \mathbb{R}_{\geq 0} \to \mathbb{R}$ evolves strictly within a predefined region that is bounded by certain functions of time, achieving prescribed transient and steady state performance. The mathematical expression of prescribed performance is given by the following inequalities:

$$-\rho_L(t) < e(t) < \rho_U(t), \quad \forall t \in \mathbb{R}_{\geq 0},$$

where $\rho_L(t), \rho_U(t)$ are smooth and bounded decaying functions of time, satisfying

$$\lim_{t\to\infty} \rho_L(t) > 0 \quad \text{and} \quad \lim_{t\to\infty} \rho_U(t) > 0,$$

called performance functions (see Fig. 2.2). Specifically, for the exponential performance functions

$$\rho_i(t) = (\rho_{i0} - \rho_{i\infty})e^{-l_i t} + \rho_{i\infty},$$

with $\rho_{i0}, \rho_{i\infty}, l_i \in \mathbb{R}_{>0}, i \in \{U, L\}$, appropriately chosen constants, $\rho_{L_0} = \rho_L(0)$, $\rho_{U_0} = \rho_U(0)$ are selected such that $\rho_{U_0} > e(0) > \rho_{L0}$ and the constants

$$\rho_{L_\infty} = \lim_{t\to\infty} \rho_L(t) < \rho_{L_0}, \quad \rho_{U_\infty} = \lim_{t\to\infty} \rho_U(t) < \rho_{U0},$$

represent the maximum allowable size of the tracking error $e(t)$ at steady state, which may be set arbitrarily small to a value reflecting the resolution of the measurement device, thus achieving practical convergence of $e(t)$ to zero. Moreover, the decreasing rate of $\rho_L(t), \rho_U(t)$, which is affected by the constants l_L, l_U in this case, introduces a lower bound on the required speed of convergence of $e(t)$. Therefore, the appropriate selection of the performance functions $\rho_L(t), \rho_U(t)$ imposes performance characteristics on the tracking error $e(t)$.

2.3 Nonlinear Model Predictive Control (NMPC)

Nonlinear Model Predictive Control (NMPC) has become an attractive feedback control strategy in applications where the stabilization or trajectory tracking of a system is required while there exist certain state and input constraints [4, 58, 60]. In general, the NMPC is formulated as solving at each sampling time step an online Finite Horizon Optimal Control Problem (FHOCP) subject to system dynamics and constraints involving states and controls. Based on measurements obtained at each sampling time step, the controller predicts the dynamic behavior of the system over a predictive horizon in the future and determines the input such that a predetermined open-loop performance objective is minimized. In order to incorporate feedback, the optimal open-loop input is implemented only until the next sampling time step.

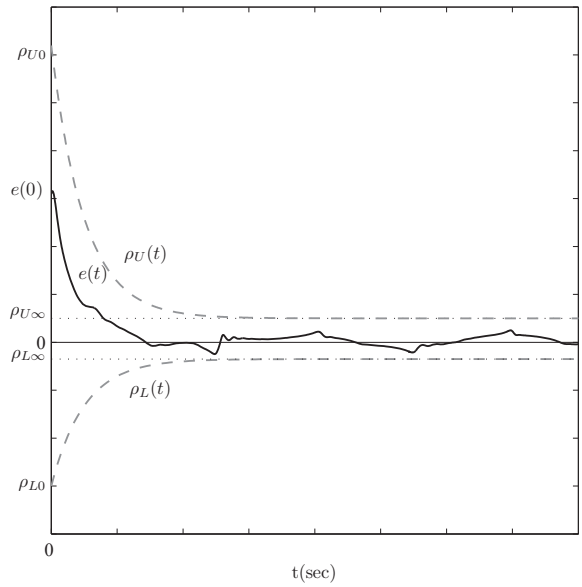

Figure 2.2: Graphical illustration of the prescribed performance definition.

Using the new system state at the next sampling time step, the whole procedure - prediction and optimization - is repeated, moving the control and prediction horizon forward. Summarizing, a standard NMPC scheme works as follows.

1. Obtain estimates of the states of the system.
2. Calculate a constraint-conforming optimal input minimizing the desired cost function over the prediction horizon using the system model and the current state estimate for prediction.
3. Implement the first part of the optimal input until the next sampling time step.
4. Continue with 1.

More technically, consider the stabilization problem for a class of systems described by the following nonlinear dynamical system:

$$\dot{x} = f(x, u), \tag{2.4}$$

with $f : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^n$. The system is subject to the input and state constraints:

$$u \in \mathcal{U} \subseteq \mathbb{R}^m, \quad x \in \mathcal{X} \subseteq \mathbb{R}^n,$$

respectively. Denote by $h > 0$, $T > h$ the sampling step and the finite prediction horizon. Consider a sequence of sampling times $\{t_k\}$, $k \in \mathbb{N}$. Then, at every sampling time step t_k the following FHOCP is solved:

$$\min_{\overline{u}(\cdot)} \left\{ \|\overline{x}(t_k + T)\|_P^2 + \int_{t_k}^{t_k+T} \left[\|\overline{x}(\mathfrak{s})\|_Q^2 + \|\overline{u}(\mathfrak{s})\|_R^2 \right] d\mathfrak{s} \right\} \tag{2.5a}$$

subject to:

$$\dot{\overline{x}}(\mathfrak{s}) = f(\overline{x}(\mathfrak{s}), \overline{u}(\mathfrak{s})), \quad \overline{x}(t_k) = x(t_k), \tag{2.5b}$$

$$\overline{x}(\mathfrak{s}) \in \mathcal{X}, \quad \overline{u}(\mathfrak{s}) \in \mathcal{U}, \quad \mathfrak{s} \in [t_k, t_k + T], \tag{2.5c}$$

$$\overline{x}(t_k + T) \in \mathcal{F}, \tag{2.5d}$$

where P, Q and R are positive definite weight matrices. Due to the fact that a finite prediction horizon is used, the actual closed-loop input and state trajectories will differ from the predicted open-loop trajectories, even if no model plant mismatch and no disturbances are present. This fact is depicted in Fig. 2.3 where the system can only move inside the shaded area as state constraints of the form $x(t) \in \mathcal{X}$. This makes the key difference between standard control strategies, where the feedback law is obtained *a priori* and NMPC where the feedback law is obtained online. Thus, the notation $\overline{\cdot}$ is used to denote the predicted variables internal to the controller. Since a finite horizon approach is used, the terminal set $\mathcal{F}$ is introduced and it is appropriately designed as described in [4, 58] in order to guarantee the stability of the closed-loop system. The solution to FHOCP (2.5a)-(2.5d) is denoted by $\overline{u}^\star(\cdot; x(t_k))$. It defines the open-loop input that is applied to the system until the next sampling time step t_{k+1} as:

$$u(\mathfrak{s}; x(t_k)) = \overline{u}^\star(\mathfrak{s}; x(t_k)), \ \mathfrak{s} \in [t_k, t_{k+1}).$$

The control input $u(\mathfrak{s}; x(t_k))$ is a feedback control law, since it is recalculated at each sampling instant using the new state information. The following theorem states the stability results for the NMPC.

Theorem 2.2. *([4]) Suppose that:*

- *f is twice continuously differentiable and satisfies $f(0_{n\times 1}, 0_{m\times 1}) = 0_{n\times 1}$;*
- *$\mathcal{U} \subseteq \mathbb{R}^m$ is compact, $\mathcal{X} \subseteq \mathbb{R}^n$ is connected and $(0_{n\times 1}, 0_{m\times 1}) \in \mathcal{X} \times \mathcal{U}$;*
- *the Jacobian linearization of the nonlinear system (2.4) is stabilizable;*
- *the FHOCP (2.5a)-(2.5d) has a feasible solution at time $t = 0$.*

Then, the closed-loop system is asymptotically stable with the region of attraction being the sets of states for which the FHOCP (2.5a)-(2.5d) has a feasible solution.

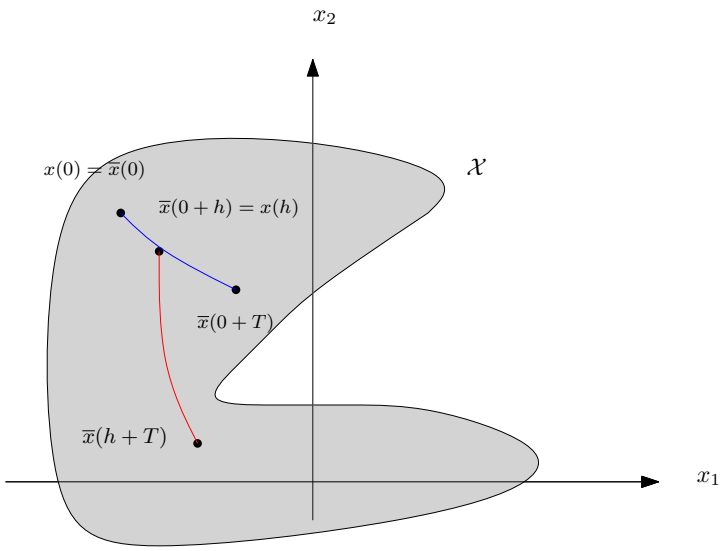

Figure 2.3: The difference between open-loop prediction and close-loop behavior [4].

Remark 2.2. Assumption $f(0_{n\times 1}, 0_{m\times 1}) = 0_{n\times 1}$ is not very restrictive, since if for example $f(x_s, u_s) = 0_{n\times 1}$, one can always shift the origin of the system to the configuration (x_s, u_s).

2.4 Graph Theory

An *undirected graph* $\mathcal{G}$ is a pair $(\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ is a finite set of nodes, representing a team of agents, and

$$\mathcal{E} \subseteq \{\{i, j\} : i, j \in \mathcal{V},\ i \neq j\},$$

with $M = |\mathcal{E}|$, being the set of edges that model the communication capability between neighboring agents. For each agent, its neighbors' set $\mathcal{N}_i$ is defined as:

$$\begin{aligned}\mathcal{N}_i &\coloneqq \{j \in \mathcal{V} : \{i, j\} \in \mathcal{E}\} \\ &= \{j_1, \ldots, j_{N_i}\},\end{aligned}$$

where $N_i = |\mathcal{N}_i|$.

If there is an edge $\{i, j\} \in \mathcal{E}$, then i, j are called *adjacent*. A *path* of length r from vertex i to vertex j is a sequence of $r + 1$ distinct vertices, starting with i and ending with j, such that consecutive vertices are adjacent. For $i = j$, the path is called a cycle. If there is a path between any two vertices of the graph $\mathcal{G}$, then $\mathcal{G}$ is called *connected*. A connected graph is called a *tree* if it contains no cycles.

The *adjacency matrix* $\mathscr{A}(\mathcal{G}) = [a_{ij}] \in \mathbb{R}^{N\times N}$ of graph $\mathcal{G}$ is defined by $a_{ij} = a_{ji} = 1$, if $\{i,j\} \in \mathcal{E}$, and $a_{ij} = 0$ otherwise. The *degree* $\deg(i)$ of vertex i is defined as the number of its neighboring vertices, i.e. $\deg(i) = N_i, i \in \mathcal{V}$. Let also:

$$\mathrm{Deg}(\mathcal{G}) = \mathrm{diag}\{[\deg(i)]_{i\in\mathcal{V}}\} \in \mathbb{R}^{N\times N},$$

be the *degree matrix* of the system. Consider an arbitrary orientation of $\mathcal{G}$, which assigns to each edge $\{i,j\} \in \mathcal{E}$ precisely one of the ordered pairs (i,j) or (j,i). When selecting the pair (i,j), we say that i is the tail and j is the head of the edge $\{i,j\}$. By considering a numbering $k \in \mathcal{M} = \{1,...,M\}$ of the graph's edge set, we define the $N \times M$ *incidence matrix* $\mathscr{D}(\mathcal{G}) = [d_{ij}]$, where: $d_{ij} = 1$, if i is the head of edge j; $d_{ij} = -1$, if i is the tail of edge j; and $d_{ij} = 0$, otherwise. For more details we refer to [128]. The *Laplacian matrix* $\mathscr{L}(\mathcal{G}) \in \mathbb{R}^{N\times N}$ of the graph $\mathcal{G}$ is defined as

$$\mathscr{L}(\mathcal{G}) = \mathrm{Deg}(\mathcal{G}) - \mathscr{A}(\mathcal{G}) = \mathscr{D}(\mathcal{G})\mathscr{D}(\mathcal{G})^\top.$$

Lemma 2.2. *[40, Section III] Assume that the graph $\mathcal{G}$ is a tree. Then, the matrix $\mathscr{D}^\top(\mathcal{G})\mathscr{D}(\mathcal{G})$ is positive definite.*

If we consider an ordering:

$$0 = \lambda_1(\mathcal{G}) \leq \lambda_2(\mathcal{G}) \leq \ldots \leq \lambda_N(\mathcal{G}) = \lambda_{\max}(\mathcal{G}),$$

of the eigenvalues of $\mathscr{L}(\mathcal{G})$, then we have that $\lambda_2(\mathcal{G}) > 0$ iff $\mathcal{G}$ is connected ([128, Chapter 2]). Denote by $\widetilde{x} \in \mathbb{R}^{|\mathcal{E}|n}$ the stack column vector of the vectors $x_i - x_j, \{i,j\} \in \mathcal{E}$ with the edges ordered as in the case of the incidence matrix $\mathscr{D}(\mathcal{G})$.

2.5 Time Sequence, Timed Run and Weighted Transition System

In the next three sections, we include definitions from computer science that are required to analyze Problem 3 addressed in this thesis.

An infinite sequence of elements of a set S is called an *infinite word* over this set and it is denoted by $w = w(0)w(1)\ldots$. The l-th element of a sequence is denoted by $w(l)$.

Definition 2.8. An *atomic proposition* σ is a statement that is either True ($\top$) or False ($\bot$).

Definition 2.9. ([129]) A *time sequence* $\tau = \tau(0)\tau(1)\ldots$ is an infinite sequence of time values $\tau(j) \in \mathbb{Q}_{\geq 0}$, satisfying the following properties:

- monotonicity: $\tau(l) < \tau(l+1)$ for all $l \geq 0$;
- progress: For every $t \in \mathbb{Q}_{\geq 0}$, there exists $l \geq 1$, such that $\tau(l) > t$.

Definition 2.10. ([129]) Let Σ be a finite set of atomic propositions. A *timed word* w over Σ, is an infinite sequence $w^t = (w(0), \tau(0))(w(1), \tau(1))\ldots$ where $w(0)w(1)\ldots$ is an infinite word over 2^Σ, and $\tau(0)\tau(1)\ldots$ is a time sequence with $\tau(l) \in \mathbb{Q}_{\geq 0},\ l \geq 0$.

Definition 2.11. A *Weighted Transition System* (WTS) is a tuple $(S, S^{\text{init}}, \text{Act}, \longrightarrow, \mathfrak{t}, \Sigma, \mathcal{L})$ where:

- S is a finite set of states;
- $S^{\text{init}} \subseteq S$ is a set of initial states;
- Act is a set of actions;
- $\longrightarrow \subseteq S \times \text{Act} \times S$ is a transition relation;
- $\mathfrak{t} : \longrightarrow \to \mathbb{Q}_{\geq 0}$ is a map that assigns a positive weight to each transition;
- Σ is a finite set of atomic propositions;
- $\mathcal{L} : S \to 2^\Sigma$ is a labeling function, which maps each state with the atomic propositions that are true in this state.

The notation $s \xrightarrow{\alpha} s'$ is used to denote that $(s, \alpha, s') \in \longrightarrow$ for $s, s' \in S$ and $\alpha \in \text{Act}$. For every $s \in S$ and $\alpha \in \text{Act}$ define:

$$\text{Post}(s, \alpha) \coloneqq \{s' \in S : (s, \alpha, s') \in \longrightarrow\}.$$

Hereafter, $\longrightarrow$ will denote transitions and $\rightarrow$ will denote function mappings.

Definition 2.12. A *timed run* of a WTS is an infinite sequence of the form:

$$r^t = (r(0), \tau(0))(r(1), \tau(1))(r(2), \tau(2)) \ldots,$$

such that $r(0) \in S^{\text{init}}$, and for all $l \geq 0$, it holds that $r(l) \in S$ and $(r(l), \alpha(l), r(l+1)) \in \longrightarrow$ for a sequence of actions $\alpha(0)\alpha(1)\alpha(2)\ldots$ with $\alpha(l) \in \text{Act}$, $\forall l \geq 0$. The *time stamps* $\tau(l), l \geq 0$ are inductively defined as:

1. $\tau(0) = 0$;
2. $\tau(l+1) = \tau(l) + \mathfrak{t}((r(l), \alpha(l), r(l+1)))$, $\forall l \geq 0$.

Definition 2.13. Every timed run r^t of a WTS generates a *timed word*

$$w^t = (w(0), \tau(0))(w(1), \tau(1))(w(2), \tau(2)) \ldots,$$

over the set 2^{Σ}, where $w(l) = \mathcal{L}(r(l))$, $\forall\ l \geq 0$ is the subset of atomic propositions that are true in state $r(l)$.

2.6 Metric Interval Temporal Logic (MITL)

The syntax of *Metric Interval Temporal Logic (MITL)* over the set of atomic propositions Σ is defined by the grammar:

$$\varphi := \top \mid \sigma \mid \neg\varphi \mid \varphi_1 \wedge \varphi_2 \mid \Diamond_I \varphi \mid \Box_I \varphi \mid \varphi_1\ \mathcal{U}_I\ \varphi_2,$$

where $\sigma \in \Sigma$ is an atomic proposition; $\neg$, $\wedge$ are boolean negation and conjunction operators, respectively; and $\Diamond$, $\Box$ and $\mathcal{U}$ are the eventually (eventually in the future), always (now and forever in the future) and until temporal operators, respectively, as they are defined in [130, Chapter 5]; $I = [a, b]$ where a, $b \in \mathbb{Q}_{\geq 0} \cup \{\infty\}$ with $a < b$, is a non-empty timed interval. The MITL formulas are interpreted over timed words like the ones produced by a WTS as is given in Definition 2.13.

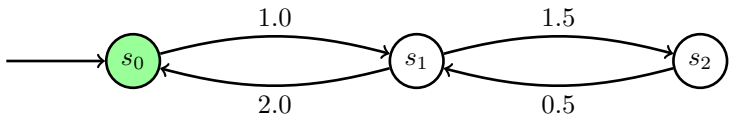

Figure 2.4: An example of a WTS with 3 states.

Definition 2.14. ([131], [132], [133]) Given a timed word:

$$w^t = (w(0), \tau(0))(w(1), \tau(1)) \ldots,$$

an MITL formula φ and a position l in the timed word, the satisfaction relation $(w^t, l) \models \varphi$, for $l \geq 0$ (read w^t satisfies φ at position l) is inductively defined as follows:

$$\begin{aligned}
&(w^t, l) \models \sigma \Leftrightarrow \sigma \in w(l),\\
&(w^t, l) \models \neg\varphi \Leftrightarrow (w^t, l) \not\models \varphi,\\
&(w^t, l) \models \varphi_1 \wedge \varphi_2 \Leftrightarrow (w^t, l) \models \varphi_1 \text{ and } (w^t, l) \models \varphi_2,\\
&(w^t, l) \models \Diamond_I \varphi \Leftrightarrow \exists l' \geq l, \text{ such that } (w^t, l') \models \varphi, \tau(l') - \tau(l) \in I,\\
&(w^t, l) \models \Box_I \varphi \Leftrightarrow \forall l' \geq l,\ \tau(l') - \tau(l) \in I \Rightarrow (w^t, l') \models \varphi,\\
&(w^t, l) \models \varphi_1\ \mathcal{U}_I\ \varphi_2 \Leftrightarrow \exists l' \geq l, \text{ such that } (w^t, l') \models \varphi_2,\\
&\qquad\qquad \tau(l') - \tau(l) \in I \text{ and } (w^t, l'') \models \varphi_1, \forall\ l \leq l'' < l'.
\end{aligned}$$

We say that w^t satisfies φ, denoted by $w^t \models \varphi$, if $(w^t, 0) \models \varphi$. Furthermore, we say that a timed run $r^t = (r(0), \tau(0))(r(1), \tau(1)) \ldots$ satisfies the MITL formula φ if and only if the corresponding timed word $w^t = (w(0), \tau(0))(w(1), \tau(1)) \ldots$ with $w(l) = L(r(l)),\ \forall l \geq 0$, satisfies the MITL formula $(w^t \models \varphi)$. We say that r^t satisfies φ, denoted by $r^t \models \varphi$, if $(r^t, 0) \models \varphi$.

Example 2.1. Consider the WTS with:

$$\begin{aligned}
S = \{s_0, s_1, s_2\},\quad S^{\text{init}} = \{s_0\},\quad \text{Act} = \emptyset,\\
\longrightarrow = \{(s_0, \emptyset, s_1), (s_1, \emptyset, s_2), (s_1, \emptyset, s_0), (s_2, \emptyset, s_1)\},\\
\mathfrak{t}((s_0, \emptyset, s_1)) = 1.0,\quad \mathfrak{t}((s_1, \emptyset, s_2)) = 1.5,\\
\mathfrak{t}((s_1, \emptyset, s_0)) = 2.0,\quad \mathfrak{t}((s_2, \emptyset, s_1)) = 0.5,\\
\Sigma = \{\text{green}\},\quad \mathcal{L}(s_0) = \{\text{green}\},\quad \mathcal{L}(s_1) = \mathcal{L}(s_2) = \emptyset.
\end{aligned}$$

depicted in Figure 2.4. Let two timed runs of the system be:

$$\begin{aligned}
r_1^t &= (s_0, 0.0)(s_1, 1.0)(s_0, 3.0)(s_1, 4.0)(s_0, 6.0) \ldots,\\
r_2^t &= (s_0, 0.0)(s_1, 1.0)(s_2, 2.5)(s_1, 3.0)(s_0, 5.0) \ldots,
\end{aligned}$$

and two MITL formulas:

$$\varphi_1 = \Diamond_{[2,5]}\{\text{green}\}, \quad \varphi_2 = \Box_{[0,5]}\{\text{green}\}.$$

According to the MITL semantics of Definition 2.14, it follows that the timed run r_1^t satisfies φ_1 ($r_1^t \models \varphi_1$), since at the time stamp $3.0 \in [2,5]$ we have that $\mathcal{L}(s_0) = \{green\}$. Thus, the atomic proposition green occurs at least once in the given interval. On the other hand, the timed run r_2^t does not satisfy φ_2 ($r_2^t \not\models \varphi_2$) since the atomic proposition green does not hold at every time stamp of the run r_2^t (it holds only at the time stamp 0.0).

2.7 Timed Büchi Automata

Timed Büchi Automata (TBA) were originally introduced in [129]. In this work, we adopt the notation from [134, 135]. Let CL be a finite set of *clocks*. The set of *clock constraints* $\Phi(\text{CL})$ is defined by the grammar:

$$\phi := \top \mid \neg\phi \mid \phi_1 \wedge \phi_2 \mid c \bowtie \psi,$$

where $c \in \text{CL}$ is a clock, $\psi \in \mathbb{Q}_{\geq 0}$ is a clock constant and $\bowtie \in \{<, >, \geq, \leq, =\}$. A clock *valuation* is a function $\nu : \text{CL} \to \mathbb{Q}_{\geq 0}$ that assigns a value to each clock. A clock c_i has valuation ν_i for $i \in \{1, \ldots, |\text{CL}|\}$, and $\nu = (\nu_1, \ldots, \nu_{|\text{CL}|})$. We denote by $\nu \models \phi$ the fact that the valuation ν satisfies the clock constraint ϕ.

Definition 2.15. A *Timed Büchi Automaton* is a tuple (Q, Q^{init}, CL, Inv, E, FS, Σ) where

- Q is a finite set of locations;
- $Q^{\text{init}} \subseteq Q$ is the set of initial locations;
- CL is a finite set of clocks;
- $\text{Inv} : Q \to \Phi(\text{CL})$ is an invariant function that labels each location $q \in Q$ with a subset of clock constraints in $\Phi(\text{CL})$;
- $E \subseteq Q \times \Phi(\text{CL}) \times 2^{\text{CL}} \times 2^{\Sigma} \times Q$ gives the set of edges of the form $\mathfrak{E} = (q, \mathfrak{g}, \text{RS}, \sigma, q')$, where q, q' are the source and target states; $\mathfrak{g} \in \Phi(\text{CL})$ is the guard of the edge; RS is a set of clocks to be reset upon executing the edge, i.e. if a $c_i \in \text{RS}$ then $\nu_i = 0, i \in \{1, \ldots, |\text{CL}|\}$ and if $c_i \notin \text{RS}$, ν_i remains unchanged; and $\sigma \in 2^{\Sigma}$ is an input action;
- $\text{FS} \subseteq Q$ is a set of accepting locations; and
- Σ is a finite set of atomic propositions.

A state of a TBA is a pair (q, ν) where $q \in Q$ and ν satisfies the *invariant* $\mathrm{Inv}(q)$, i.e., $\nu \models \mathrm{Inv}(q)$. The initial state is $(q(0), (0, \ldots, 0))$, where $q(0) \in Q^{\mathrm{init}}$. Given two states (q, ν) and (q', ν') and an edge $e = (q, \mathfrak{g}, \mathrm{RS}, \sigma, q')$, there exists a *discrete transition* $(q, \nu) \xrightarrow{e} (q', \nu')$ iff $\nu \models \mathfrak{g}$, $\nu' \models \mathrm{Inv}(q')$, and RS is the *reset set*, i.e., $\nu'_i = 0$ for $c_i \in \mathrm{RS}$ and $\nu'_i = \nu_i$ for $c_i \notin \mathrm{RS}$. Given a $\eta \in \mathbb{Q}_{\geq 0}$, there exists a *time transition* $(q, \nu) \xrightarrow{\eta} (q', \nu')$ iff $q = q', \nu' = \nu + \eta$ (η is summed component-wise) and $\nu' \models \mathrm{Inv}(q)$. We write $(q, \nu) \xrightarrow{\eta} \xrightarrow{e} (q', \nu')$ if there exists q'', ν'' such that $(q, \nu) \xrightarrow{\eta} (q'', \nu'')$ and $(q'', \nu'') \xrightarrow{e} (q', \nu')$ with $q'' = q$.

An infinite run of a TBA starting at state $(q(0), \nu(0))$ is an infinite sequence of time and discrete transitions

$$(q(0), \nu(0)) \xrightarrow{\eta_0} (q(0)', \nu(0)') \xrightarrow{e_0} (q(1), \nu(1)) \xrightarrow{\eta_1} (q(1)', \nu(1)') \ldots,$$

where $(q(0), \nu(0))$ is an initial state. This run produces the timed word

$$w^t = (\mathcal{L}(q(0)), \tau(0))(\mathcal{L}(q(1)), \tau(1)) \ldots,$$

with $\tau(0) = 0$ and $\tau(i+1) = \tau(i) + \eta_i$, $\forall\ i \geq 1$. The run is called *accepting* if $q(i) \in \mathrm{FS}$ for infinitely many times. A timed word is *accepted* if there exists an accepting run that produces it.

Any MITL formula φ over Σ can be algorithmically translated into a TBA with the alphabet 2^{Σ}, such that the language of timed words that satisfy φ is the language of the accepting timed words produced by the TBA ([136–138]).

Chapter 3

Formation Control of Multiple Rigid Bodies

This chapter addresses the problem of distance- and orientation-based formation control of a class of second-order nonlinear multi-agent systems in 3D space, under static and undirected communication topologies. More specifically, we design a decentralized model-free control protocol in the sense that each agent uses only local information from its neighbors to calculate its own control signal, without incorporating any knowledge of the model nonlinearities and exogenous disturbances. Moreover, the transient and steady state response are solely determined by certain designer-specified performance functions and is fully decoupled by the agents' dynamic model, the control gain selection, the underlying graph topology as well as the initial conditions. Additionally, by introducing certain inter-agent distance constraints, we guarantee collision avoidance and connectivity maintenance between neighboring agents. Finally, simulation results verify the performance of the proposed controllers.

3.1 Introduction

During the last decades, decentralized control of networked multi-agent systems has gained a significant amount of attention due to the great variety of its applications, including multi-robot systems, transportation, multi-point surveillance and biological systems. The main focus of multi-agent systems is the design of distributed control protocols in order to achieve global tasks, such as consensus [139–142], and at the same time fulfill certain properties, e.g., network connectivity [143, 144].

A particular multi-agent problem that has been considered in the literature is the formation control problem, where the agents represent robots that aim to form a prescribed geometrical shape, specified by a certain set of desired relative configurations between the agents. The main categories of formation control that have been studied in the related literature are ([145]) position-based control, displacement-based control, distance-based control and orientation-based control. Distance- and orientation-based control constitute the topics in this work.

In distance-based formation control, inter-agent distances are actively controlled to achieve a desired formation, dictated by desired inter-agent distances. Each agent is assumed to be able to sense the position of its neighboring agents. When orientation alignment is considered as a control design goal, the problem is known as orientation-based (or bearing-based) formation control. The orientation-based control steers the agents to configurations that achieve desired relative orientation angles. In this work, we aim to design a decentralized control protocol such that both distance- and orientation-based formation is achieved.

The literature in distance-based formation control is rich, and is traditionally categorized in single or double integrator agent dynamics and directed or undirected communication topologies (see e.g. [35–50, 145])

Orientation-based formation control has been addressed in [51–54], whereas the authors in [53, 55, 56] have considered the combination of distance- and orientation-based formation.

In most of the aforementioned works in formation control, the two-dimensional case with simple dynamics and point-mass agents has been dominantly considered. In real applications, however, the engineering systems have nonlinear second order dynamics and are usually subject to exogenous disturbances and modeling errors. Other important issues concern the connectivity maintenance, the collision avoidance between the agents and the transient and steady state response of the closed loop system, which have not been taken into account in the majority of related woks. Thus, taking all the above into consideration, the design of robust distributed control schemes for the multi-agent formation control problem becomes a challenging task.

Motivated by this, we aim to address here the distance-based formation control problem with orientation alignment for a team of rigid bodies operating in 3D space, with unknown second-order nonlinear dynamics and external disturbances. We propose a purely decentralized control protocol that guarantees distance formation, orientation alignment as well as collision avoidance and connectivity maintenance between neighboring agents and in parallel ensures the satisfaction of prescribed transient and steady state performance. The prescribed performance control framework has been incorporated in multi-agent systems in [146], where first order dynamics have been considered without taking into account the problem of orientation alignment.

This chapter is organized as follows. Section 3.2 provides some mathematical notation that will be used in this Chapter. Section 3.3 provides the system dynamics under consideration and the problem statement. Section 3.4 discusses the technical details of the proposed feedback control design. In Section 3.5 the stability analysis of the proposed control law is provided and Section 3.6 is devoted to a simulation example. Finally, the conclusions of this chapter are discussed in Section 3.7.

3.2 Notation

The vector connecting the origins of coordinate frames F_A and F_B expressed in frame F_C coordinates in 3D space is denoted by $p^C_{B/A} \in \mathbb{R}^3$. We further denote by $q_{B/A} = [\phi_{B/A}, \theta_{B/A}, \psi_{B/A}]^\tau$ the Euler angles representing the orientation of frame F_B with respect to frame F_A, with $-\pi \leq \phi_{A/B}, \psi_{A/B} \leq \pi$ and $-\frac{\pi}{2} \leq \theta_{A/B} \leq \frac{\pi}{2}$. Define also $\mathbb{T} \coloneqq [-\pi, \pi] \times \left[-\frac{\pi}{2}, \frac{\pi}{2}\right] \times [-\pi, \pi]$. The angular velocity of frame F_B with respect to F_A, expressed in frame F_C coordinates, is denoted by $\omega^C_{B/A} \in \mathbb{R}^3$. We also use the notation $\mathbb{M} \coloneqq \mathbb{R}^3 \times \mathbb{T}$. For notational brevity, when a coordinate frame corresponds to an inertial frame of reference F_I, we will omit its explicit notation (e.g., $p_B = p^0_{B/0}, \omega_B = \omega^0_{B/0}$). All vector and matrix differentiations are derived with respect to the inertial frame F_I, unless otherwise stated.

3.3 Problem Formulation

3.3.1 System Model

Consider a set of N rigid bodies, with labeling set $\mathcal{V} = \{1, 2, \ldots, N\}$, operating in a workspace $\mathcal{W} \subseteq \mathbb{R}^3$, with coordinate frames F_i, $i \in \mathcal{V}$, attached to their centers of mass. We consider that each agent occupies a ball $\mathcal{B}(p_i(t), \mathfrak{r}_i)$, where $p_i : \mathbb{R}_{\geq 0} \to \mathbb{R}^3$ is the position of the agent's center of mass and $\mathfrak{r}_i$ is the agent's radius (see Fig. 3.1). We also denote by $q_i : \mathbb{R}_{\geq 0} \to \mathbb{T}, i \in \mathcal{V}$, the Euler angles representing the agents' orientation with respect to an inertial frame F_I, with $q_i \coloneqq [\phi_i, \theta_i, \psi_i]^\top$. By defining $x_i : \mathbb{R}_{\geq 0} \to \mathbb{M}, v_i : \mathbb{R}_{\geq 0} \to \mathbb{R}^6$, with:

$$x_i \coloneqq [p_i^\top, q_i^\top]^\top,$$
$$v_i \coloneqq [\dot{p}_i^\top, \omega_i^\top]^\top,$$

we model each agent's motion with the second order dynamics:

$$\dot{x}_i(t) = J_i(x_i) v_i(t), \tag{3.1a}$$
$$M_i(x_i)\dot{v}_i(t) + C_i(x_i, \dot{x}_i)v_i(t) + g_i(x_i) + \delta_i(x_i, \dot{x}_i, t) = u_i, \tag{3.1b}$$

where $J_i : \mathbb{M} \to \mathbb{R}^{6\times 6}$ is a Jacobian matrix that maps the Euler angle rates to v_i, given by:

$$J_i(x_i) \coloneqq \begin{bmatrix} I_3 & 0_{3\times 3} \\ 0_{3\times 3} & J_q(x_i) \end{bmatrix},$$
$$J_q(x_i) \coloneqq \begin{bmatrix} 1 & \sin(\phi_i)\tan(\theta_i) & \cos(\phi_i)\tan(\theta_i) \\ 0 & \cos(\phi_i) & -\sin(\phi_i) \\ 0 & \dfrac{\sin(\phi_i)}{\cos(\theta_i)} & \dfrac{\cos(\phi_i)}{\cos(\theta_i)} \end{bmatrix},$$

for which we make the following assumption:

Assumption 3.1. The angle θ_i satisfies the inequality:

$$-\frac{\pi}{2} < \theta_i(t) < \frac{\pi}{2}, \quad \forall i \in \mathcal{V}, \quad t \in \mathbb{R}_{\geq 0}.$$

The aforementioned assumption guarantees that J_i is always well-defined and invertible, since $\det(J_i) = \frac{1}{\cos\theta_i}$. Furthermore, $M_i : \mathbb{M} \to \mathbb{R}^{6\times 6}$ is the positive definite inertia matrix, $C_i : \mathbb{M} \times \mathbb{R}^6 \to \mathbb{R}^{6\times 6}$ is the Coriolis matrix, $g_i : \mathbb{M} \to \mathbb{R}^6$ is the gravity vector, and $\delta_i : \mathbb{M} \times \mathbb{R}^6 \times \mathbb{R}_{\geq 0} \to \mathbb{R}^6$ is a bounded vector representing model uncertainties and external disturbances. We consider that the aforementioned vector fields are unknown and continuous. Finally, $u_i \in \mathbb{R}^6$ is the control input vector representing the 6D generalized forces/torques acting on the agent.

The dynamics (3.1) can be written in vector form as:

$$\dot{x}(t) = J(x)v(t), \tag{3.2a}$$

$$\overline{M}(x)\dot{v}(t) + \overline{C}(x,\dot{x})v(t) + \overline{g}(x) + \overline{\delta}(x,\dot{x},t) = u, \tag{3.2b}$$

where $x = [x_1^\top, \ldots, x_N^\top]^\top : \mathbb{R}_{\geq 0} \to \mathbb{M}^N, v = [v_1^\top, \ldots, v_N^\top]^\top : \mathbb{R}_{\geq 0} \to \mathbb{R}^{6N}, u = [u_1^\top, \ldots, u_N^\top]^\top \in \mathbb{R}^{6N}$, and

$$\begin{aligned} J &\coloneqq \operatorname{diag}\{[J_i]_{i\in\mathcal{V}}\} \in \mathbb{R}^{6N\times 6N}, \\ \overline{M} &\coloneqq \operatorname{diag}\{[M_i]_{i\in\mathcal{V}}\} \in \mathbb{R}^{6N\times 6N}, \\ \overline{C} &\coloneqq \operatorname{diag}\{[C_i]_{i\in\mathcal{V}}\} \in \mathbb{R}^{6N\times 6N}, \\ \overline{g} &\coloneqq [g_1^\top, \ldots, g_N^\top]^\top \in \mathbb{R}^{6N}, \\ \overline{\delta} &\coloneqq [\delta_1^\top, \ldots, \delta_N^\top]^\top \in \mathbb{R}^{6N}. \end{aligned}$$

It is further assumed that each agent can measure its own $p_i, q_i, \dot{p}_i, v_i, i \in \mathcal{V}$, and has a limited sensing range of:

$$\mathfrak{d}_i > \max\{\mathfrak{r}_i + \mathfrak{r}_j : i, j \in \mathcal{V}\}.$$

Therefore, by defining the neighboring set:

$$\mathcal{N}_i(t) \coloneqq \{j \in \mathcal{V} : p_j(t) \in \mathcal{B}(p_i(t), \mathfrak{d}_i)\},$$

agent i knows at each time instant t all $p^i_{j/i}(t), q_{j/i}(t)$ and, since it knows its own $p_i(t), q_i(t)$, it can compute all $p_j(t)$, $q_j(t)$, $\forall j \in \mathcal{N}_i(t), t \in \mathbb{R}_{\geq 0}$.

The topology of the multi-agent network is modeled through the graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, with $\mathcal{V} = \{1, \ldots, N\}$ and:

$$\mathcal{E} \coloneqq \Big\{\{i,j\} \in \mathcal{V}\times\mathcal{V} : j \in \mathcal{N}_i(0) \text{ and } i \in \mathcal{N}_j(0)\Big\}.$$

The latter implies that at $t = 0$ the graph is undirected, i.e.,

$$\|p_{\ell_k}(0) - p_{m_k}(0)\|_2 < d_{k,\text{con}}, \quad \forall\{\ell_k, m_k\} \in \mathcal{E},$$

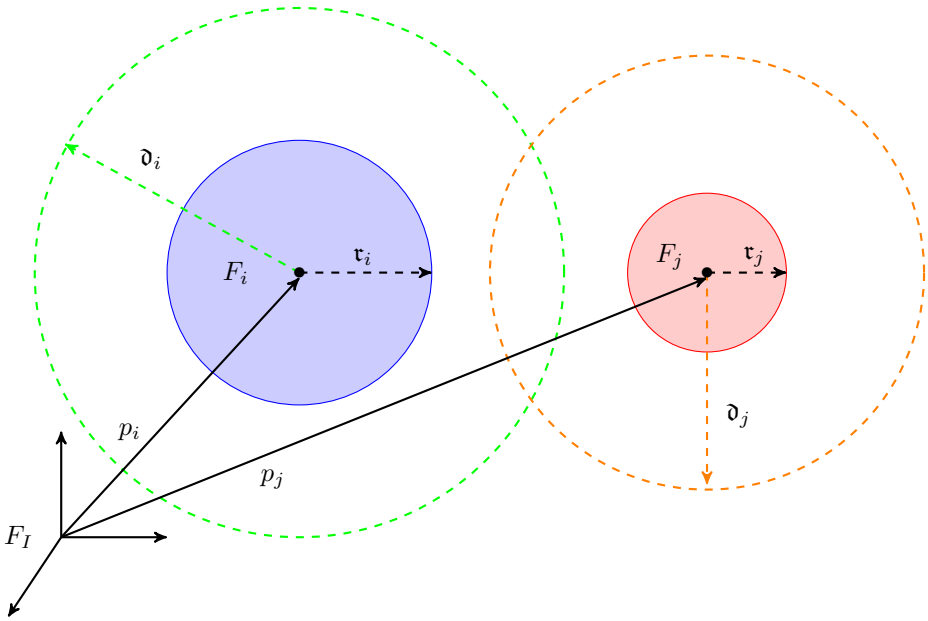

Figure 3.1: Illustration of two agents $i, j \in \mathcal{V}$ in the workspace; F_I is the inertial frame, F_i, F_j are the frames attached to the agents' center of mass, $p_i, p_j \in \mathbb{R}^3$ are the positions of the center of mass with respect to F_I; $\mathfrak{r}_i$, $\mathfrak{r}_j$ are the radii of the agents and $\mathfrak{d}_i > \mathfrak{d}_j$ are their sensing ranges.

with:

$$d_{k,\text{con}} \coloneqq \min\{\mathfrak{d}_{\ell_k}, \mathfrak{d}_{m_k}\}, \quad \ell_k, m_k \in \mathcal{V}, \quad \forall k \in \mathcal{M}.$$

We also consider that $\mathcal{G}$ is static in the sense that no edges are added to the graph. We do not exclude, however, edge removal through connectivity loss between initially neighboring agents, which it will be guaranteed to be avoided by the proposed control law, as it will be presented in the sequel. It is also assumed that at $t = 0$ the neighboring agents are at a collision-free configuration, i.e.,

$$d_{k,\text{col}} < \|p_{\ell_k}(0) - p_{m_k}(0)\|_2, \quad \forall\{\ell_k, m_k\} \in \mathcal{E},$$

with $d_{k,\text{col}} \coloneqq r_{\ell_k} + r_{m_k}$. Hence, we conclude that:

$$d_{k,\text{col}} < \|p_{\ell_k}(0) - p_{m_k}(0)\|_2 < d_{k,\text{con}}, \quad \forall\{\ell_k, m_k\} \in \mathcal{E}. \tag{3.3}$$

Moreover, given the desired formation constants $d_{k,\text{des}}$, $q_{k,\text{des}}$ for the edge $k \in \mathcal{M}$, the formation configuration is called *feasible* if the set:

$$\Big\{x \in \mathbb{M}^N : \quad \|p_{\ell_k} - p_{m_k}\|_2 = d_{k,\text{des}}, \quad q_{\ell_k} - q_{m_k} = q_{k,\text{des}}, \quad \forall\{\ell_k, m_k\} \in \mathcal{E}\Big\},$$

with $\ell_k, m_k \in \mathcal{V}, \forall k \in \mathcal{M}$, is nonempty.

3.3.2 Problem Statement

Due to the fact that the agents are not dimensionless and their communication capabilities are limited, the control protocol, except from achieving a desired inter-agent formation, it should also guarantee for all $t \in \mathbb{R}_{\geq 0}$ that:

- the neighboring agents avoid collision with each other;
- and, all the initial edges are maintained, i.e., connectivity maintenance.

Therefore, all pairs $\{\ell_k, m_k\} \in \mathcal{V} \times \mathcal{V}$ of agents that initially form an edge must remain within distance greater than $d_{k,\text{col}}$ and less than $d_{k,\text{con}}$. We also make the following assumption that are required on the graph topology:

Assumption 3.2. The communication graph $\mathcal{G}$ is initially a tree.

Formally, the robust formation control problem under the aforementioned constraints is formulated as follows:

Problem 3.1. Given N agents governed by the dynamics (3.1), under the Assumptions 3.1 - 3.2 and given the desired inter-agent distances and angles $d_{k,\text{des}}, q_{k,\text{des}}$, with:

$$d_{k,\text{col}} < d_{k,\text{des}} < d_{k,\text{con}}, \quad \forall \{\ell_k, m_k\} \in \mathcal{E}, \quad \ell_k, m_k \in \mathcal{V}, \quad k \in \mathcal{M},$$

design decentralized control laws $u_i \in \mathbb{R}^6, i \in \mathcal{V}$ such that $\forall\ \{\ell_k, m_k\} \in \mathcal{E}, k \in \mathcal{M}$, the following hold:

1. $\lim_{t\to\infty} \|p_{\ell_k}(t) - p_{m_k}(t)\|_2 = d_{k,\text{des}}$,
2. $\lim_{t\to\infty} [q_{m_k}(t) - q_{\ell_k}(t) - q_{k,\text{des}}] = 0_{3\times 1}$,
3. $d_{k,\text{col}} < \|p_{\ell_k}(t) - p_{m_k}(t)\|_2 < d_{k,\text{con}}, \quad \forall t \in \mathbb{R}_{\geq 0}$.

3.4 Feedback Control Design

3.4.1 Error Derivation

Let $p = [p_1^\top, \ldots, p_N^\top]^\top : \mathbb{R}_{\geq 0} \to \mathbb{R}^{3N}$, $q = [q_1^\top, \ldots, q_N^\top]^\top : \mathbb{R}_{\geq 0} \to \mathbb{T}^N$ be the stack vectors of all the agent positions and Euler angles. We denote by $\tilde{p}, \tilde{q} : \mathbb{R}_{\geq 0} \to \mathbb{R}^{3M}$ the stack column vectors of:

$$p_{\ell_k, m_k}(t) \coloneqq p_{\ell_k}(t) - p_{m_k}(t),$$
$$q_{\ell_k, m_k}(t) \coloneqq q_{\ell_k}(t) - q_{m_k}(t),$$

respectively, $\forall\{\ell_k, m_k\} \in \mathcal{E}$, with the edges ordered as in the case of the incidence matrix $\mathscr{D}(\mathcal{G})$. Thus, the following holds:

$$\begin{aligned}\tilde{p}(t) &= \begin{bmatrix} p_{\ell_1,m_1}(t) \\ \vdots \\ p_{\ell_M,m_M}(t) \end{bmatrix} \\ &= \begin{bmatrix} p_{\ell_1}(t) - p_{m_1}(t) \\ \vdots \\ p_{\ell_M}(t) - p_{m_M}(t) \end{bmatrix} \\ &= \left(\mathscr{D}^\top(\mathcal{G}) \otimes I_3\right) p(t), \qquad (3.4a) \\ \tilde{q}(t) &= \begin{bmatrix} q_{\ell_1}(t) - q_{m_1}(t) \\ \vdots \\ q_{\ell_M}(t) - q_{m_M}(t) \end{bmatrix} \\ &= \left(\mathscr{D}^\top(\mathcal{G}) \otimes I_3\right) q(t). \qquad (3.4b)\end{aligned}$$

Next, we introduce the errors $e_k^p : \mathbb{R}_{\geq 0} \to \mathbb{R}, e_k^q = [e_{k_1}^q\ , e_{k_2}^q,\ e_{k_3}^q]^\top : \mathbb{R}_{\geq 0} \to \mathbb{T}$:

$$\begin{aligned} e_k^p(t) &\coloneqq \|p_{\ell_k}(t) - p_{m_k}(t)\|_2^2 - d_{k,\mathrm{des}}^2, \\ e_k^q(t) &\coloneqq q_{m_k}(t) - q_{\ell_k}(t) - q_{k,\mathrm{des}}, \end{aligned}$$

for all distinct edges $\{\ell_k, m_k\} \in \mathcal{E}, k \in \mathcal{M}$, in the numbered order they appear in the edge set $\mathcal{E}$.
Taking the time derivative of e_k^p, e_k^p, the following are obtained:

$$\dot{e}_k^p(t) = 2p_{\ell_k,m_k}^\top(t)\dot{p}_{\ell_k,m_k}(t), \qquad (3.5a)$$
$$\dot{e}_k^q(t) = \dot{q}_{m_k}(t) - \dot{q}_{\ell_k}(t). \qquad (3.5b)$$

Also, by defining the vectors:

$$\begin{aligned} e^p(t) &\coloneqq [e_1^p(t), \ldots, e_M^p(t)]^\top \in \mathbb{R}^M, \\ e^q(t) &\coloneqq [(e_1^q(t))^\top, \ldots, (e_M^q(t))^\top]^\top \in \mathbb{T}^M, \end{aligned}$$

and employing (3.4a), (3.4b), (3.5a) and (3.5b), the time derivatives of the vectors

$e^p(t)$ and $e^q(t)$ can be written in vector form as:

$$\begin{aligned}
\dot{e}^p(t) &= \begin{bmatrix} \dot{e}^p_1(t) \\ \vdots \\ \dot{e}^p_M(t) \end{bmatrix} \\
&= \begin{bmatrix} 2p^\top_{\ell_1,m_1}(t)\dot{p}_{\ell_1,m_1}(t) \\ \vdots \\ 2p^\top_{\ell_M,m_M}(t)\dot{p}_{\ell_M,m_M}(t) \end{bmatrix} \\
&= 2\begin{bmatrix} p^\top_{\ell_1,m_1}(t) & \dots & 0_{1\times 3} \\ \vdots & \ddots & \vdots \\ 0_{1\times 3} & \dots & p^\top_{\ell_M,m_M}(t) \end{bmatrix} \begin{bmatrix} \dot{p}_{\ell_1,m_1}(t) \\ \vdots \\ \dot{p}_{\ell_M,m_M}(t) \end{bmatrix} \\
&= \mathbb{F}_p(x)\left(\mathscr{D}^\top(\mathcal{G}) \otimes I_3\right)\dot{p}, \qquad (3.6a) \\
\dot{e}^q(t) &= \begin{bmatrix} \dot{e}^q_1(t) \\ \vdots \\ \dot{e}^q_M(t) \end{bmatrix} \\
&= \begin{bmatrix} \dot{q}_{\ell_1} - \dot{q}_{m_1} \\ \vdots \\ \dot{q}_{\ell_M} - \dot{q}_{m_M} \end{bmatrix} \\
&= \left(\mathscr{D}^\top(\mathcal{G}) \otimes I_3\right)\dot{q}, \qquad (3.6b)
\end{aligned}$$

where $\mathbb{F}_p : \mathbb{M}^N \to \mathbb{R}^{M\times 3M}$, with:

$$\mathbb{F}_p(x) \coloneqq 2\begin{bmatrix} p^\top_{\ell_1,m_1}(t) & \dots & 0_{1\times 3} \\ \vdots & \ddots & \vdots \\ 0_{1\times 3} & \dots & p^\top_{\ell_M,m_M}(t) \end{bmatrix}.$$

By introducing the stack error vector $e(t) \coloneqq [(e^p(t))^\top, (e^q(t))^\top]^\top \in \mathbb{R}^{4M}$, (3.6) can be written as:

$$\dot{e}(t) = \overline{\mathbb{F}}_p(x)\overline{\mathscr{D}}^\top(\mathcal{G})\begin{bmatrix} \dot{p} \\ \dot{q} \end{bmatrix}, \qquad (3.7)$$

where:

$$\overline{\mathbb{F}}_p(x) \coloneqq \begin{bmatrix} \mathbb{F}_p(x) & 0_{M\times 3M} \\ 0_{3M\times 3M} & I_{3M} \end{bmatrix} \in \mathbb{R}^{4M\times 6M}, \qquad (3.8a)$$

$$\overline{\mathscr{D}}(\mathcal{G}) \coloneqq \begin{bmatrix} \mathscr{D}(\mathcal{G}) \otimes I_3 & 0_{3N\times 3M} \\ 0_{3N\times 3M} & \mathscr{D}(\mathcal{G}) \otimes I_3 \end{bmatrix} \in \mathbb{R}^{6N\times 6M}. \qquad (3.8b)$$

Finally, we obtain from (3.2a):

$$\begin{bmatrix}\dot{p}\\ \dot{q}\end{bmatrix} = \underbrace{\left[\begin{array}{ccc|ccc} I_3 & \dots & 0_{3\times3} & 0_{3\times3} & \dots & 0_{3\times3}\\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots\\ 0_{3\times3} & \dots & I_3 & 0_{3\times3} & \dots & 0_{3\times3}\\ \hline 0_{3\times3} & \dots & 0_{3\times3} & J_q(x_1) & \dots & 0_{3\times3}\\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots\\ 0_{3\times3} & \dots & 0_{3\times3} & 0_{3\times3} & \dots & J_q(x_N) \end{array}\right]}_{\underline{J}(x)} \underbrace{\begin{bmatrix}\dot{p}_1\\ \vdots\\ \dot{p}_N\\ \omega_1\\ \vdots\\ \omega_N\end{bmatrix}}_{\underline{v}(t)}$$

$$= \underline{J}(x)\underline{v}(t), \tag{3.9}$$

and thus, (3.7) can be written as:

$$\dot{e}(t) = \overline{\mathbb{F}}_p(x)\overline{\mathscr{D}}^{\top}(\mathcal{G})\underline{J}(x)\underline{v}(t). \tag{3.10}$$

3.4.2 Performance Functions

The concepts and techniques of prescribed performance control (see Section 2.2) are adapted in this work in order to:

- achieve predefined transient and steady state response for the distance and orientation errors $e_k^p, e_k^q, \forall k \in \mathcal{M}$;
- avoid the violation of the collision and connectivity constraints between neighboring agents, as presented in Section 3.3.

The mathematical expressions of prescribed performance are given by the inequality objectives:

$$-C_{k,\text{col}}\rho_k^p(t) < e_k^p(t) < C_{k,\text{con}}\rho_k^p(t), \tag{3.11a}$$
$$-\rho_k^q(t) < e_{k_n}^q(t) < \rho_k^q(t), \tag{3.11b}$$

$\forall k \in \mathcal{M}, n \in \{1,2,3\}$, where:

$$\rho_k^p(t) \coloneqq \left[1 - \frac{\rho_{k,\infty}^p}{\max\{C_{k,\text{con}}, C_{k,\text{col}}\}}\right] e^{-l_k^p t} + \frac{\rho_{k,\infty}^p}{\max\{C_{k,\text{con}}, C_{k,\text{col}}\}},$$
$$\rho_k^q(t) \coloneqq (\rho_{k,0}^q - \rho_{k,\infty}^q)e^{-l_k^q t} + \rho_{k,\infty}^q,$$

are designer-specified, smooth, bounded, and decreasing functions of time, where $l_k^p, l_k^q, \rho_{k,\infty}^p, \rho_{k,\infty}^q \in \mathbb{R}_{>0}, \forall k \in \mathcal{M}$, incorporate the desired transient and steady state performance specifications respectively, as presented in Section 2.2; and $C_{k,\text{col}}$,

$C_{k,\mathrm{con}} \in \mathbb{R}_{>0}, \forall k \in \mathcal{M}$, are associated with the collision and connectivity constraints. In particular, we select:

$$C_{k,\mathrm{col}} \coloneqq d_{k,\mathrm{des}}^2 - d_{k,\mathrm{col}}^2, \tag{3.12a}$$

$$C_{k,\mathrm{con}} \coloneqq d_{k,\mathrm{con}}^2 - d_{k,\mathrm{des}}^2, \tag{3.12b}$$

$\forall k \in \mathcal{M}$, which, since the desired formation is compatible with the collision and connectivity constraints (i.e., $d_{k,\mathrm{col}} < d_{k,\mathrm{des}} < d_{k,\mathrm{con}}, \forall k \in \mathcal{M}$), ensures that $C_{k,\mathrm{col}}, C_{k,\mathrm{con}} \in \mathbb{R}_{>0}, \forall k \in \mathcal{M}$ and consequently, in view of (3.3), that:

$$-C_{k,\mathrm{col}}\rho_k^p(0) < e_k^p(0) < \rho_k^p(0)C_{k,\mathrm{con}}, \tag{3.13a}$$

$\forall k \in \mathcal{M}$. Moreover, by choosing

$$\rho_{k,0}^q = \rho_k^q(0) > \max_{n\in\{1,2,3\}} |e_{k_n}^q(0)|, \tag{3.13b}$$

it is also guaranteed that:

$$-\rho_k^q(0) < e_{k_n}^q(0) < \rho_k^q(0), \tag{3.13c}$$

$\forall k \in \mathcal{M}, n \in \{1,2,3\}$. Hence, if we guarantee prescribed performance via (3.11), by employing the decreasing property of $\rho_k^p(t), \rho_k^q(t), \forall k \in \mathcal{M}$, we obtain:

$$-C_{k,\mathrm{col}} < e_k^p(t) < C_{k,\mathrm{con}},$$
$$-\rho_k^q(t) < e_{k_n}^q(t) < \rho_k^q(t),$$

and, consequently, owing to (3.12):

$$d_{k,\mathrm{col}} < \|p_{\ell_k}(t) - p_{m_k}(t)\|_2 < d_{k,\mathrm{con}},$$

$\forall k \in \mathcal{M}, t \in \mathbb{R}_{\geq 0}$, providing, therefore, a solution to Problem 3.1.

In the sequel, we propose a decentralized control protocol that does not incorporate any information on the agents' dynamic model and guarantees (3.11) for all $t \in \mathbb{R}_{\geq 0}$.

3.4.3 Feedback Control Design

Given the errors $e^p(t), e^q(t)$ defined in Section 3.4.1:

Step I-a: Select the corresponding functions $\rho_k^p(t), \rho_k^q(t)$ and positive parameters $C_{k,\mathrm{con}}, C_{k,\mathrm{col}}, k \in \mathcal{M}$, following (3.11), (3.13b), and (3.12), respectively, in order to incorporate the desired transient and steady state performance specifications as well as the collision and connectivity constraints, and define the *normalized errors* $\xi_k^p : \mathbb{R}_{\geq 0} \to \mathbb{R}, \xi_k^q = [\xi_{k_1}^q, \xi_{k_2}^q, \xi_{k_3}^q]^\top : \mathbb{R}_{\geq 0} \to \mathbb{R}^3$:

$$\xi_k^p(t) \coloneqq (\rho_k^p(t))^{-1} e_k^p(t), \tag{3.14a}$$

$$\xi_k^q(t) \coloneqq \mathrm{diag}\Big\{\rho_1^q(t), \rho_2^q(t), \rho_3^q(t)\Big\}^{-1} e_k^q(t), \tag{3.14b}$$

$\forall k \in \mathcal{M}$, as well as the stack vector forms:

$$\begin{aligned}
\xi^p(t) &\coloneqq [\xi_1^p(t), \ldots, \xi_M^p(t)]^\top \\
&\coloneqq (\rho^p(t))^{-1} e^p(t), \\
\xi^q(t) &\coloneqq [(\xi_1^q(t))^\top, \ldots, (\xi_M^q(t))^\top]^\top \\
&\coloneqq (\rho^q(t))^{-1} e^q(t), \\
\xi(t) &\coloneqq [(\xi^p(t))^\top, (\xi^q(t))^\top]^\top \\
&\coloneqq (\rho(t))^{-1} e(t) \in \mathbb{R}^{4M},
\end{aligned} \tag{3.15}$$

where:

$$\begin{aligned}
\rho^p(t) &\coloneqq \operatorname{diag}\{[\rho_k^p(t)]_{k\in\mathcal{M}}\} \in \mathbb{R}^{M\times M}, \\
\rho^q(t) &\coloneqq \operatorname{diag}\{[\rho_k^q(t) I_3]_{k\in\mathcal{M}}\} \in \mathbb{R}^{3M\times 3M}, \\
\rho(t) &\coloneqq \operatorname{diag}\{\rho^p(t), \rho^q(t)\} \in \mathbb{R}^{4M\times 4M}.
\end{aligned}$$

Step I-b: Define the *transformed errors* $\varepsilon_k^p : \mathbb{R} \to \mathbb{R}, \varepsilon_k^q : \mathbb{R}^3 \to \mathbb{R}^3$ and the signals $r_k^p : \mathbb{R} \to \mathbb{R}, r_k^q : \mathbb{R}^3 \to \mathbb{R}^{3\times 3}$ as:

$$\varepsilon_k^p(\xi_k^p) \coloneqq \ln\left(\left(1 + \frac{\xi_k^p}{C_{k,\mathrm{col}}}\right)\left(1 - \frac{\xi_k^p}{C_{k,\mathrm{con}}}\right)^{-1}\right), \tag{3.16a}$$

$$\varepsilon_k^q(\xi_k^q) \coloneqq \left[\ln\left(\frac{1+\xi_{k_1}^q}{1-\xi_{k_1}^q}\right), \ln\left(\frac{1+\xi_{k_2}^q}{1-\xi_{k_2}^q}\right), \ln\left(\frac{1+\xi_{k_3}^q}{1-\xi_{k_3}^q}\right)\right]^\top, \tag{3.16b}$$

$$\begin{aligned}
r_k^p(\xi_k^p) &\coloneqq \frac{\partial \varepsilon_k^p(\xi_k^p)}{\partial \xi_k^p} \\
&= \frac{C_{k,\mathrm{col}} + C_{k,\mathrm{con}}}{(C_{k,\mathrm{col}} + \xi_k^p)(C_{k,\mathrm{con}} - \xi_k^p)}, \\
r_k^q(\xi_k^q) &\coloneqq \frac{\partial \varepsilon_k^q(\xi_k^q)}{\partial \xi_k^q} \\
&= \operatorname{diag}\left\{\left[r_{k_n}^q(\xi_{k_n}^q)\right]_{n\in\{1,2,3\}}\right\} \\
&= \operatorname{diag}\left\{\left[\frac{2}{1-(\xi_{k_n}^q)^2}\right]_{n\in\{1,2,3\}}\right\},
\end{aligned}$$

and design the decentralized reference velocity vector for each agent $v_{i,\mathrm{des}} = [\dot{p}_{i,\mathrm{des}}^\top, \omega_{i,\mathrm{des}}^\top]^\top : \mathbb{R}^{4M} \times \mathbb{R}_{\geq 0} \to \mathbb{R}^6$ as:

$$v_{i,\mathrm{des}}(\xi, t) \coloneqq -J_i^{-1}(x_i)\begin{bmatrix} \sum_{j\in\mathcal{N}_i(0)} (\rho_{k_{ij}}^p(t))^{-1} r_{k_{ij}}^p(\xi_{k_{ij}}^p)\varepsilon_{k_{ij}}^p(\xi_{k_{ij}}^p) p_{i,j}(t) \\ \sum_{j\in\mathcal{N}_i(0)} (\rho_{k_{ij}}^q(t))^{-1} r_{k_{ij}}^q(\xi_{k_{ij}}^q)\varepsilon_{k_{ij}}^q(\xi_{k_{ij}}^q) \end{bmatrix}, \tag{3.17}$$

where $k_{ij} \in \mathcal{M}$ is the edge between agents $i, j \in \mathcal{N}_i(0)$, i.e., $\{\ell_{k_{ij}}, m_{k_{ij}}\} \in \mathcal{E}$ and $\ell_{k_{ij}} = i, m_{k_{ij}} = j$. The desired velocities (3.17) can be written in vector form as:

$$\begin{aligned}\underline{v}_{\text{des}}(\xi, t) &= \begin{bmatrix}\dot{p}_{\text{des}}(\xi^p, t) \\ \omega_{\text{des}}(\xi^q, t)\end{bmatrix} \\ &= -\underline{J}^{-1}(x)\overline{\mathscr{D}}(\mathcal{G})\overline{\mathbb{F}}_p^{\top}(x) r(\xi)(\rho(t))^{-1}\varepsilon(\xi), \end{aligned} \tag{3.18}$$

where

$$\begin{aligned}\dot{p}_{\text{des}} &\coloneqq [\dot{p}_{1,\text{des}}^{\top}, \ldots, \dot{p}_{N,\text{des}}^{\top}]^{\top}, \\ \omega_{\text{des}} &\coloneqq [\omega_{1,\text{des}}^{\top}, \ldots, \omega_{N,\text{des}}^{\top}]^{\top} \in \mathbb{R}^{3N}, \\ \varepsilon &\coloneqq [(\varepsilon^p)^{\top}, (\varepsilon^q)^{\top}]^{\top} \\ &= [\varepsilon_1^p, \ldots, \varepsilon_M^p, (\varepsilon_1^q)^{\top}, \ldots, (\varepsilon_M^q)^{\top}]^{\top} \in \mathbb{R}^{4M},\end{aligned}$$

and $\underline{J}(x), \overline{\mathscr{D}}(\mathcal{G}), \overline{\mathbb{F}}_p$ as they were defined in (3.8) and (3.9), respectively. Moreover,

$$\begin{aligned} r &\coloneqq \begin{bmatrix} r^p & 0_{M\times 3M} \\ 0_{3M\times M} & r^q \end{bmatrix} \in \mathbb{R}^{4M\times 4M}, \\ r^p &\coloneqq \text{diag}\{[r_k^p]_{k\in\mathcal{M}}\} \in \mathbb{R}^{M\times M}, \\ r^q &\coloneqq \text{diag}\{[r_k^q]_{k\in\mathcal{M}}\} \in \mathbb{R}^{3M\times 3M}. \end{aligned}$$

It should be noted that $\underline{J}^{-1}(x)$ is always well-defined due to Assumption 1.

Step II-a: Define the velocity errors $e^v : \mathbb{R}^{4M} \times \mathbb{R}_{\geq 0} \to \mathbb{R}^{6N}$, with:

$$\begin{aligned} e^v(\xi, t) &\coloneqq [(e_1^v)^{\top}, \ldots, (e_N^v)^{\top}]^{\top} \\ &\coloneqq v(t) - v_{\text{des}}(\xi, t), \end{aligned}$$

where

$$\begin{aligned} e_i^v(\xi, t) &= [e_{i_1}^v(\xi, t), \ldots, e_{i_6}^v(\xi, t)]^{\top} \\ &= [\dot{p}_i^{\top}(t) - \dot{p}_{i,\text{des}}^{\top}(\xi^p, t), \omega_i^{\top}(t) - \omega_{i,\text{des}}^{\top}(\xi^q, t)]^{\top} \\ &= v_i(t) - v_{i,\text{des}}(\xi, t), \end{aligned}$$

and select the corresponding performance functions $\rho_{i_m}^v : \mathbb{R}_{\geq 0} \to \mathbb{R}_{>0}$, with:

$$\begin{aligned} \rho_{i_m}^v(t) &\coloneqq (\rho_{i_m,0}^v - \rho_{i_m,\infty}^v)e^{-l_{i_m}^v t} + \rho_{i_m,\infty}^v, \\ \rho_{i_m,0}^v &= \rho_{i_m}^v(0) > |e_{i_m}^v(0)|, \\ l_{i_m}^v, \rho_{i_m,\infty}^v &\in \mathbb{R}_{>0}, \quad \rho_{i_m,\infty}^v < \rho_{i_m,0}^v, \quad \forall i \in \mathcal{V}, \quad m \in \{1, \ldots, 6\}. \end{aligned}$$

Moreover, define the *normalized velocity errors* $\xi_i^v = [\xi_{i_1}^v, \dots, \xi_{i_6}^v]^\top : \mathbb{R}^{4M} \times \mathbb{R}_{\geq 0} \to \mathbb{R}^6$ as:

$$\xi_i^v(\xi, t) \coloneqq (\rho_i^v(t))^{-1} e_i^v(\xi, t), \tag{3.19}$$

with $\rho_i^v(t) \coloneqq \operatorname{diag}\{[\rho_{i_m}^v(t)]_{m\in\{1,\dots,6\}}\} \in \mathbb{R}^{6\times 6}$, which is written in vector form as:

$$\begin{aligned}\xi^v(\xi, t) &\coloneqq [(\xi_1^v(\xi, t))^\top, \dots, (\xi_N^v(\xi, t))^\top]^\top \\ &\coloneqq (\rho^v(t))^{-1} e^v(\xi, t) \in \mathbb{R}^{6N},\end{aligned} \tag{3.20}$$

with $\rho^v(t) \coloneqq \operatorname{diag}\left\{[\rho_i^v(t)]_{i\in\mathcal{V}}\right\} \in \mathbb{R}^{6N\times 6N}$.

Step II-b: Define the *transformed velocity errors* $\varepsilon_i^v : \mathbb{R}^6 \to \mathbb{R}^6$ and the signal $r_i^v : \mathbb{R}^6 \to \mathbb{R}^{6\times 6}$ as:

$$\varepsilon_i^v(\xi_i^v) \coloneqq \left[\ln\left(\frac{1+\xi_{i_1}^v}{1-\xi_{i_1}^v}\right), \cdots, \ln\left(\frac{1+\xi_{i_6}^v}{1-\xi_{i_6}^v}\right)\right]^\top, \tag{3.21a}$$

$$\begin{aligned} r_i^v(\xi_i^v) &\coloneqq \frac{\partial \varepsilon_i^v(\xi_i^v)}{\partial \xi_i^v} \\ &= \operatorname{diag}\{\left[r_{i_m}^v(\xi_{i_m}^v)\right]_{m\in\{1,\dots,6\}}\} \\ &= \operatorname{diag}\left\{\left[\frac{2}{(1-(\xi_{i_m}^v)^2)}\right]_{m\in\{1,\dots,6\}}\right\}, \end{aligned} \tag{3.21b}$$

and design the decentralized control protocol $u_i : \mathbb{R}^6 \times \mathbb{R}_{\geq 0} \to \mathbb{R}^6$ for each agent $i \in \mathcal{V}$ as:

$$u_i(\xi_i^v, t) \coloneqq -\gamma_i (\rho_i^v(t))^{-1} r_i^v(\xi_i^v) \varepsilon_i^v(\xi_i^v), \tag{3.22}$$

with $\gamma_i \in \mathbb{R}_{>0}, \forall i \in \mathcal{V}$, which can be written in vector form as:

$$u(\xi^v, t) \coloneqq -\Gamma (\rho^v(t))^{-1} r^v(\xi^v) \varepsilon^v(\xi^v), \tag{3.23}$$

where:

$$\begin{aligned}\Gamma &\coloneqq \operatorname{diag}\{[\gamma_i I_6]_{i\in\mathcal{V}}\} \in \mathbb{R}^{6N\times 6N}, \\ \varepsilon^v &\coloneqq [(\varepsilon_1^v)^\top, \dots, (\varepsilon_N^v)^\top]^\top \in \mathbb{R}^{6N}, \\ r^v &\coloneqq \operatorname{diag}\{[r_i^v]_{i\in\mathcal{V}}\} \in \mathbb{R}^{6N\times 6N}.\end{aligned}$$

Remark 3.1. Note that the selection of $C_{k,\mathrm{col}}, C_{k,\mathrm{con}}$ according to (3.12) and of $\rho_k^q(t), \rho_{i_m}^v(t)$ such that $\rho_{k,0}^q = \rho_k^q(0) > \max_{n\in\{1,2,3\}} |e_{k_n}^q(0)|, \rho_{i_m,0}^v = \rho_{i_m}^v(0) > |e_{i_m}^v(0)|$ along with (3.3), guarantee that:

$$\begin{aligned} \xi_k^p(0) &\in (C_{k,\mathrm{col}}, C_{k,\mathrm{con}}), \\ \xi_{k_n}^q(0) &\in (-1, 1), \\ \xi_{i_m}^v(\xi(0), 0) &\in (-1, 1),\ \forall k \in \mathcal{M},\ n \in \{1,2,3\},\ m \in \{1,\dots,6\},\ i \in \mathcal{V}. \end{aligned}$$

The prescribed performance control technique enforces these normalized errors $\xi_k^p(t), \xi_{k_n}^q(t)$ and $\xi_{i_m}^v(t)$ to remain strictly within the sets $(-C_{k,\text{col}}, C_{k,\text{con}}), (-1,1)$, and $(-1,1)$, respectively, $\forall k \in \mathcal{M}, n \in \{1,2,3\}, m \in \{1,\ldots,6\}, i \in \mathcal{V}, t \geq 0$, guaranteeing thus a solution to Problem 3.1. It can be verified that this can be achieved by maintaining the boundedness of the modulated errors $\varepsilon^p(\xi^p(t)), \varepsilon^q(\xi^q(t))$ and $\varepsilon^v(\xi^v(t)), \forall t \geq 0$.

Remark 3.2. Notice by (3.17) and (3.22) that the proposed control protocols are decentralized in the sense that each agent uses only local information to calculate its own signal. In that respect, regarding every edge k_{ij}, with $\{\ell_{k_{ij}}, m_{k_{ij}}\} = \{i,j\}$, the parameters $\rho_{k_{ij},\infty}^p$, $\rho_{k_{ij},\infty}^q$, $l_{k_{ij}}^p$, $l_{k_{ij}}^q$, as well as the sensing radii $\mathfrak{d}_j$, $\forall j \in \mathcal{N}_i(0)$, which are needed for the calculation of the performance functions $\rho_{k_{ij}}^p, \rho_{k_{ij}}^q$, can be transmitted offline to each agent $i \in \mathcal{V}$. It should also be noted that the proposed control protocol (3.22) depends exclusively on the velocity of each agent and not on the velocity of its neighbors. Moreover, the proposed control law does not incorporate any prior knowledge of the model nonlinearities/disturbances, enhancing thus its robustness. Furthermore, the proposed methodology results in a *low complexity*. Notice that no hard calculations (neither analytic nor numerical) are required to output the proposed control signal.

Remark 3.3. Regarding the construction of the performance functions, we stress that the desired performance specifications concerning the transient and steady state response as well as the collision and connectivity constraints are introduced in the proposed control schemes via $\rho_k^p(t), \rho_k^q(t)$ and $C_{k,\text{col}}, C_{k,\text{con}}$, $k \in \mathcal{M}$. In addition, the velocity performance functions $\rho_{i_m}^v(t)$, impose prescribed performance on the velocity errors $e_i^v = v_i - v_{i,\text{des}}$, $i \in \mathcal{V}$. In this respect, notice that $v_{i,\text{des}}$ acts as a reference signal for the corresponding velocities v_i, $i \in \mathcal{V}$. However, it should be stressed that although such performance specifications are not required (only the neighborhood position and orientation errors need to satisfy predefined transient and steady state performance specifications), their selection affects both the evolution of the errors within the corresponding performance envelopes as well as the control input characteristics (magnitude and rate). Nevertheless, the only hard constraint attached to their definition is related to their initial values. More specifically:

$$\rho_{k,0}^q = \rho_k^q(0) > \max_{n\in\{1,2,3\}} |e_{k_n}^q(0)|,$$
$$\rho_{i_m,0}^v = \rho_{i_m}^v(0) > |e_{i_m}^v(0)|, \ \forall k \in \mathcal{M}, \ n \in \{1,2,3\}, \ m \in \{1,\ldots,6\}, \ i \in \mathcal{V}.$$

3.5 Stability Analysis

The main results of this chapter is summarized in the following theorem.

Theorem 3.1. *Consider a system of N rigid bodies aiming at establishing a formation described by the desired distances $d_{k,des}$ and orientation angles $q_{k,des}, k \in \mathcal{M}$,*

while satisfying the collision and connectivity constraints between neighboring agents, represented by $d_{k,col}$ and $d_{k,con}$, respectively, with $d_{k,col} < d_{k,des} < d_{k,con}, k \in \mathcal{M}$. Then, under Assumptions 3.1, 3.2, the decentralized control protocol (3.22)-(3.23) *guarantees that:*

$$-C_{k,col}\rho_k^p(t) < e_k^p(t) < C_{k,con}\rho_k^p(t),$$
$$-\rho_k^q(t) < e_{k_n}^q(t) < \rho_k^q(t),$$

$\forall k \in \mathcal{M}, n \in \{1,2,3\}, t \geq 0$, *as well as the boundedness of all closed loop signals.*

Proof. The proof can be found in Appendix A.1. □

Remark 3.4. Notice that (A.6) and (A.7) hold no matter how large the finite bounds $\overline{\varepsilon}, \overline{\varepsilon}^v$ are. Therefore, there is no need to render $\overline{\varepsilon}^v$ arbitrarily small by adopting extreme values of the control gains γ_i. In the same spirit, large uncertainties involved in the nonlinear model (3.1) can be compensated, as they affect only the size of $\overline{\varepsilon}^v$ through $\overline{B}_2$, but leave unaltered the achieved stability properties. Hence, the actual performance of the system becomes isolated against model uncertainties, thus enhancing the robustness of the proposed control schemes.

Remark 3.5. The transient and steady state performance of the closed loop system is explicitly and solely determined by appropriately selecting the parameters l_k^p, l_k^q, $\rho_{k,\infty}^p, \rho_{k,\infty}^q, \rho_{k,0}^p$ and $C_{k,\text{col}}$, $C_{k,\text{con}}$, $k \in \mathcal{M}$. In that respect, the performance attributes of the proposed control protocols are selected *a priori*, in accordance to the desired transient and steady state performance specifications. In this way, the selection of the control gains $\gamma_i, i \in \mathcal{V}$, that has been isolated from the actual control performance, is significantly simplified to adopting those values that lead to reasonable control effort. Nonetheless, it should be noted that their selection affects both the quality of evolution of the errors inside the corresponding performance envelopes as well as the control input characteristics. Hence, fine tuning might be needed in real-time scenarios, to retain the required control input signals within the feasible range that can be implemented by real actuators. Similarly, the control input constraints impose an upper bound on the required speed of convergence of $\rho_k^p(t)$, and $\rho_k^q(t), k \in \mathcal{M}$, as obtained by the exponentials $e^{-l_k^p t}, e^{-l_k^q t}$. Therefore, the selection of the control gains γ_i can have positive influence on the overall closed loop system response. More specifically, notice that (A.4)-(A.7) provide bounds on $\varepsilon, \varepsilon^v$ and r, r^v that depend on the constants $\overline{B}_1, \overline{B}_2$. Therefore, in the special case that bounds on the model nonlinearities/disturbances are known, we can design the control gains γ_i via (3.22) such that the control signals u_i are retained within certain bounds.

Remark 3.6. Regarding Assumption 1, we stress that, by choosing the initial conditions $\theta_i(0), \forall i \in \mathcal{V}$ as well as the desired formation constants $\theta_{k,\text{des}} = q_{k_2,\text{des}}, \forall k \in \mathcal{M}$ close to zero, the condition $-\frac{\pi}{2} < \theta_i(t) < \frac{\pi}{2}$ will not be violated, since the agents will be mostly operating near the point $\theta_i = 0, \forall i \in \mathcal{V}$. This is a reasonable assumption for real applications, since the angle θ_i represents the pitch angle of agent i and is desired to be as close to zero as possible (consider, e.g., aerial vehicles). Furthermore, notice that the proposed control scheme guarantees collision avoidance only for the initially neighboring agents (at $t = 0$), since that is how the edge set $\mathcal{E}$ is defined. Inter-agent collision avoidance with all possible agent pairs is left as future work by employing potential fields-based performance control laws.

3.6 Simulation Results

To demonstrate the efficiency of the proposed control protocol, we consider a simulation example with:

$$N = 4, \quad \mathcal{V} = \{1, 2, 3, 4\},$$

spherical agents of the form (3.1), with $\mathfrak{r}_i = 0.5\text{m}$ and $\mathfrak{d}_i = 4\text{m}, \forall i \in \mathcal{V}$. We select the exogenous disturbances as:

$$\delta_i = A_i \sin(\omega_{c,i} t)(a_{i_1} x_i - a_{i,2} \dot{x}_i),$$

where the parameters $A_i, \omega_{c,i}, a_{i_1}, a_{i_2}$ as well as the dynamic parameters of the agents were randomly chosen in $[0, 1]$. The initial conditions were taken as:

$$\begin{aligned} p_1(0) &= [0, 0, 0]^\top \text{ m}, \quad p_2(0) = [2, 2, 2]^\top \text{ m}, \\ p_3(0) &= [2, 4, 4]^\top \text{ m}, \quad p_4(0) = [2, 3, 2.5]^\top \text{ m}, \\ q_1(0) &= q_2(0) = q_3(0) = q_4(0) = [0, 0, 0]^\top \text{ rad}, \end{aligned}$$

which imply the initial edge set $\mathcal{E} = \{\{1, 2\}, \{2, 3\}, \{2, 4\}\}$. The desired graph formation was defined by the constants

$$d_{k,\text{des}} = 2.5\text{m}, \quad q_{k,\text{des}} = [\tfrac{\pi}{4}, 0, \tfrac{\pi}{3}]^\top \text{ rad}, \quad \forall k \in \{1, 2, 3\},$$

and it also holds that:

$$d_{k,\text{col}} = 1, \quad d_{k,\text{con}} = 4.$$

By invoking (3.12), we also chose $C_{k,\text{col}} = 5.25\text{m}$ and $C_{k,\text{con}} = 9.75\text{m}$. Moreover, the parameters of the performance functions were chosen as:

$$\begin{aligned} \rho^p_{k,\infty} &= 0.1, \\ \rho^q_{k,0} &= \tfrac{\pi}{2} > \max\{e^q_{k_1}(0), e^q_{k_2}(0), \\ e^q_{k_3}(0)\} &= \tfrac{\pi}{3}, \\ l^p_k = l^q_k &= 1, \quad \forall k \in \{1, 2, 3\}. \end{aligned}$$

In addition, we chose:

$$\rho^v_{i_m,0} = 2|e^v_{i_m}(0)| + 0.5, \quad l^v_{i_m} = 1, \quad \rho^v_{i_m,\infty} = 0.1.$$

Finally, γ_i is set to 5 in order to produce reasonable control signals that can be implemented by real actuators. The simulation results are depicted in Fig. 3.2-Fig. 3.6. In particular, Fig. 3.2 and Fig. 3.3 show the evolution of $e^p_k(t)$ and $e^q_{k_n}(t)$ along with $\rho^p_k(t)$ and $\rho^q_k(t)$, respectively, $\forall k \in \{1,2,3\}, n \in \{1,2,3\}$. Furthermore, the distances $\|p_{1,2}\|_2$, $\|p_{2,3}\|_2$, $\|p_{2,4}\|_2$ along with the collision and connectivity constraints are depicted in Fig. 3.4. Finally, the velocity errors $e^v_{i_m}(t)$ along $\rho^v_{i_m}(t)$ and the control signals u_i are illustrated in Figs. 3.5 and Fig. 3.6, respectively. As it was predicted by the theoretical analysis, the formation control problem with prescribed transient and steady state performance is solved with bounded closed loop signals, despite the unknown agent dynamics and the presence of external disturbances.

3.7 Conclusions

In this chapter, we have proposed a robust and model-free decentralized control protocol for distance- and orientation-based formation control for a class of multi-agent systems modeled by Newton-Euler dynamics. Collision avoidance as well as connectivity maintenance is guaranteed to be satisfied by the proposed feedback control scheme. A simulation example with $N = 4$ agents has verified the efficiency of the proposed approach.

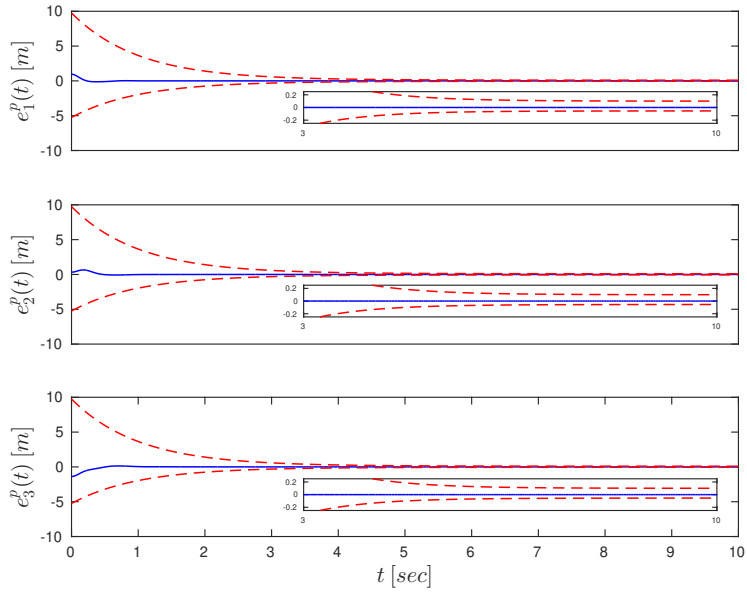

Figure 3.2: The evolution of the distance errors $e_k^p(t)$, along with the performance bounds imposed by $\rho_k^p(t), \forall k \in \{1, 2, 3\}$.

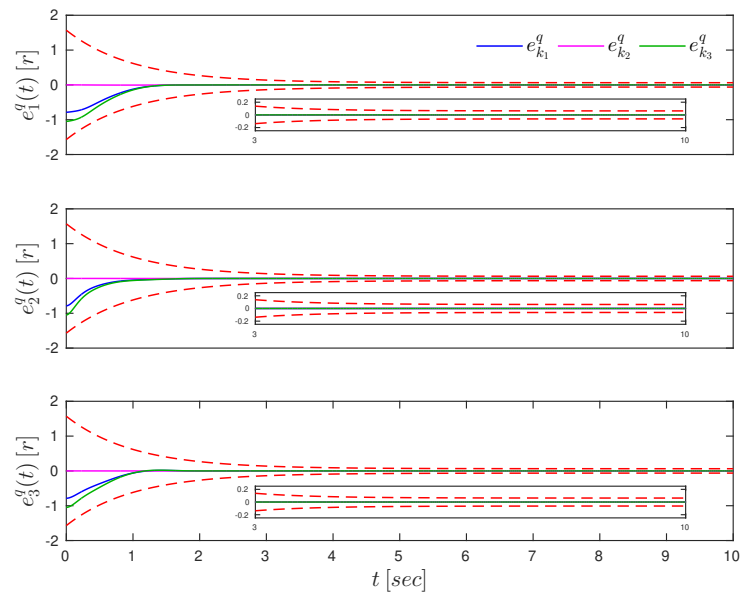

Figure 3.3: The evolution of the orientation errors $e_{k_n}^q(t)$, along with the performance bounds imposed by $\rho_k^q(t), \forall k, n \in \{1, 2, 3\}$.

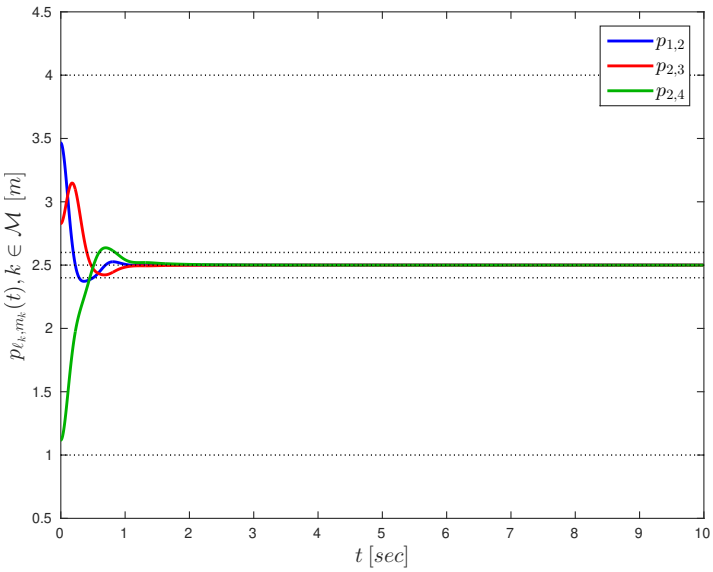

Figure 3.4: The distance between neighboring agents along with the collision and connectivity constraints.

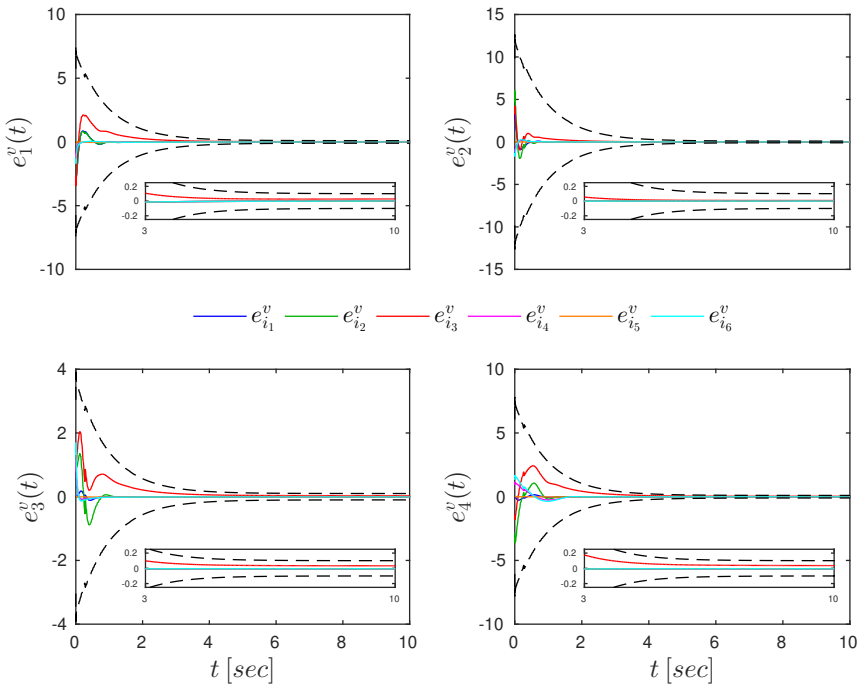

Figure 3.5: The evolution of the velocity errors $e_{i_m}^v(t)$, along with the performance bounds imposed by $\rho_{i_m}^v(t), \forall i \in \{1, \ldots, 4\}, m \in \{1, \ldots, 6\}$.

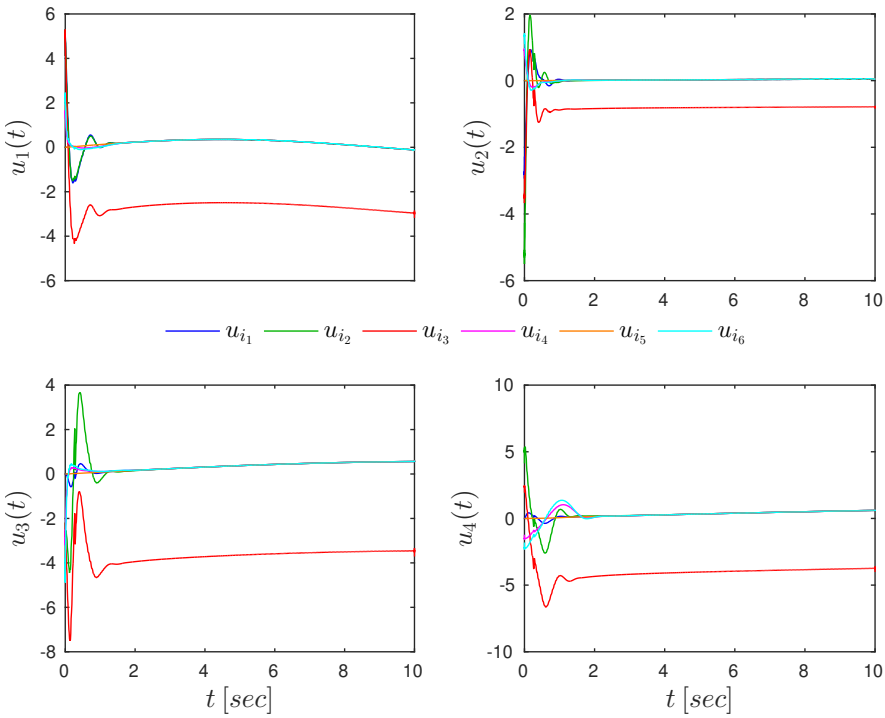

Figure 3.6: The resulting control input signals $u_i(t), i \in \{1, \ldots, 4\}$.

Chapter 4

Decentralized Tube-based NMPC of Uncertain Multi-agent Systems

This chapter addresses the problem of decentralized tube-based NMPC for a general class of uncertain nonlinear continuous-time multi-agent systems with additive and bounded disturbance. In particular, the problem of robust navigation of a multi-agent system to predefined states of the workspace while using only local information is addressed, under certain distance and control input constraints. We propose a decentralized feedback control protocol that consists of two terms: a nominal control input, which is computed online and is the outcome of a Decentralized Finite Horizon Optimal Control Problem (DFHOCP) that each agent solves at every sampling time step, for its nominal system dynamics; and an additive state feedback law which is computed offline and guarantees that the real trajectories of each agent belong to a hyper-tube centered along the nominal trajectory, for all times. The volume of the hyper-tube depends on the upper bound of the disturbances as well as the bounds of the derivatives of the dynamics. In addition, by introducing certain distance constraints, the proposed scheme guarantees that the initially connected agents remain connected for all times. Under standard assumptions that arise in nominal NMPC schemes, controllability assumptions as well as communication capabilities between the agents, it is guaranteed that the multi-agent system is Input-to-State Stable (ISS) with respect to the disturbances, for all initial conditions satisfying the state constraints. Simulation results verify the correctness of the proposed framework.

4.1 Introduction

During the last decades, NMPC has been proven to be a powerful control framework for dealing with the problem of stabilization of dynamical systems under state and input constraints [57–65]. One of the main challenges in NMPC is the treatment of potential uncertainties due to imperfect modeling and/or disturbances that may affect the system. In parallel to that, *decentralized control of multi-agent systems*

has gained significant attention due to its great variety of applications including multi-robot systems, transportation and biological systems [140]. In this chapter, we aim to exploit a novel robust NMPC framework in order to solve a collaborative multi-agent navigation problem under certain distance and input constraints.

The problem of robust NMPC has been extensively investigated over the last years. Authors in [66, 67] proposed a method of constraint sets tightening for guaranteeing robust stability. These approaches are relatively conservative since the constraints are tightened proportionally with the length of the horizon. Thus, they cannot be applied in problems where a larger prediction horizon T is required, since this can lead to infeasibility of constraint satisfaction. A robust MPC approach for solving a min-max optimization problem online in the presence of uncertainties has been proposed in [68–70]. We argue that such schemes are computationally intractable since the complexity of the resulting optimization problem grows exponentially with the increase of the predicted horizon. A promising robust strategy, originally proposed for discrete-time linear systems in [71–73], is the so called tube-based approach. In tube-based MPC a Finite Horizon Optimal Control Problem (FHOCP) is solved online for the nominal system, while the real trajectory is guaranteed to remain in a bounded tube for all times. Tube-based NMPC is a powerful robust strategy for the following reasons:

1. the tightening of the constraints does not depend on the length of the horizon, since it is conducted offline. Hence, the framework can handle disturbances/uncertainties of larger amplitude;

2. the complexity of the FHOCP is the same with the complexity of the nominal NMPC, due to the fact that only the nominal dynamics are considered while the FHOCP is solved [123];

3. the size of the tube depends of the upper bound of the disturbances as well as the bounds of the dynamics of the system. Furthermore, it can be adjusted offline by appropriately tuning the gain of the state feedback law.

Tube-based approaches for nonlinear discrete-time systems have been considered in [74–78]. The authors in [79] have addressed the linear continuous-time case. In [80, 81], regarding the computation of the offline feedback controller, the discrepancy between the nominal nonlinear system with the corresponding linear system has been considered. In [123], sufficient conditions for affine in the control continuous-time nonlinear systems with constant matrices multiplying the control input vectors have been proposed, which we aim to extend here in order to cover a larger class of nonlinear systems, and in particular decentralized multi-agent systems.

Multi-agent navigation in an important field in both the robotics and the control communities, due to the need for autonomous control of multiple robotic agents in the same workspace [147–149]. Important applications of multi-agent navigation arise also in the fields of air-traffic management and autonomous driving for guaranteeing collision avoidance with other cars and obstacles. In this work, we study the problem of decentralized navigation for nonlinear multi-agent systems with network connectivity maintenance. Decentralized MPC schemes for multi-agent navigation have been investigated in [27, 150–154]. The authors in [150] and [151] have considered specific dynamics (unicycles and flying robots in $\mathbb{SE}(3)$, respectively) and a potential robustness analysis has not been considered. In [152], a decentralized robust linear MPC formulation for multi-agent systems with coupled constraints is studied. The authors in [153] have proposed a decentralized receding horizon protocol for formation control of linear multi-agent systems. In [154], the authors considered path-following problems for multiple Unmanned Aerial Vehicles (UAVs) in which a decentralized optimization method is proposed through linearization of the dynamics of the UAVs. In our previous work [27], the robust analysis relied on constraint tightening at every sampling time step which may be conservative and not suitable methodology for problems that a larger prediction horizon T is required.

Decentralized tube-based MPC frameworks for linear multi-agent systems in discrete time have been considered in [155, 156]. However, these approaches cannot be extended to nonlinear non-affine multi-agents in a straightforward way. Furthermore, the approach in [155] requires restrictive assumptions regarding the sets involved in the proposed nested control framework.

Motivated by the aforementioned, the contribution of this chapter is to propose a decentralized tube-based feedback control protocol for a general class of uncertain nonlinear continuous-time multi-agent systems, i.e., a more general class of systems than the ones that have already been studied in the literature. More specifically, each agent solves a DFHOCP and exchanges its open-loop predicted trajectory with its neighbors in order to reach a predefined state, under certain distance and inputs constraints. The proposed control law consists of two parts: the first part is the online solution to a nominal DFHOCP, which is solved at every constant sampling time step; the second part is a state feedback law which is calculated offline in order to guarantee that the error between the real and the nominal trajectory remains in a bounded hyper-tube, for all times. Under standard nominal NMPC and communication capabilities assumptions (see [58, 157]) between neighboring agents, we show that the proposed control law renders the closed multi-agent system ISS with respect to the disturbances. The contributions of this chapter are summarized as follows:

- we develop a systematic control design methodology for tube-based NMPC which guarantees ISS for uncertain nonlinear continuous-time systems; and
- the above results are exploited and extended for solving a constrained navigation multi-agent problem under coupled constraints in a decentralized manner

without any assumptions regarding the communication configuration.

The remainder of the chapter is organized as follows: Section 4.2 provides the system dynamics under consideration and the problem statement. Section 4.3 discusses the technical details of the proposed solution and Section 4.4 is devoted to a simulation example. Finally, the summary of this chapter is provided in Section 4.5.

4.2 Problem Formulation

4.2.1 System Model

Consider a set $\mathcal{V}$ of N agents, $\mathcal{V} = \{1, \ldots, N\}$, operating in a bounded workspace $\mathcal{W} \subseteq \mathbb{R}^n$; The *uncertain nonlinear continuous dynamics* of each agent $i \in \mathcal{V}$ are given by:

$$\dot{x}_i = f_i(x_i, u_i) + \delta_i, \tag{4.1}$$

where $x_i \in \mathcal{W}$ denotes the state of each agent; $u_i \in \mathbb{R}^n$ denotes the control input; $f_i : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}^n$ is a continuous nonlinear vector valued function; and the term $\delta_i \in \mathbb{R}^n$ represents external disturbances/uncertainties as well as unmodeled dynamics. The state x_i of each agent $i \in \mathcal{V}$ is assumed to be available for measurement. The control inputs are assumed to satisfy $u_i(t) \in \mathcal{U}_i \subseteq \mathbb{R}^n$, for every $t \in \mathbb{R}_{\geq 0}$, where $\mathcal{U}_i$ are convex sets containing the origin. Denote by $x_i(0) \in \mathcal{W}$ the initial condition of (4.1). Assume also that the uncertainties are bounded i.e., there exist finite constants $\widetilde{\delta}_i$ such that:

$$\delta_i \in \Delta_i \coloneqq \left\{ \delta_i \in \mathbb{R}^n : \|\delta_i\|_2 \leq \widetilde{\delta}_i \right\}, \quad \forall t \in \mathbb{R}_{\geq 0}, \quad i \in \mathcal{V}. \tag{4.2}$$

For the system (4.1), define the *nominal dynamics* (without disturbances) by:

$$\dot{\overline{x}}_i = f_i(\overline{x}_i, \overline{u}_i), \tag{4.3}$$

where $\delta_i = 0_{n\times 1}$, $\overline{x}_i(t) \in \mathcal{W}$ and $\overline{u}_i(t) \in \mathcal{U}_i$, for every $t \in \mathbb{R}_{\geq 0}$, $i \in \mathcal{V}$. Hereafter, we shall denote by $\bar{\cdot}$ all the nominal signals.

Assumption 4.1. The nonlinear functions $f_i : \mathcal{W} \times \mathcal{U}_i \to \mathbb{R}^n$ are *continuously differentiable* with respect both to x_i and u_i in $\mathcal{W} \times \mathcal{U}_i$ with $f_i(0_{n\times 1}, 0_{n\times 1}) = 0_{n\times 1}$, $\forall i \in \mathcal{V}$.

Assumption 4.2. The linear systems $\dot{\overline{x}}_i(t) = A_i\overline{x}_i(t) + B_i\overline{u}_i(t)$, that are the outcome of the Jacobian linearization of the nominal systems (4.3) around the equilibrium states $x_i = 0$ are stabilizable.

Define the function $J_i : \mathcal{W} \times \mathcal{U}_i \to \mathbb{R}^n \times \mathbb{R}^n$ by:

$$J_i(x_i, u_i) \coloneqq \sum_{k=1}^{n} \sum_{j=1}^{n} \mathfrak{l}_n^k \left(\mathfrak{l}_n^j\right)^\top \frac{\partial f_{i,k}(x_i, u_i)}{\partial u_j}, \tag{4.4}$$

where $f_{i,k}$ stands for the k-th component of the function f_i and the vectors $\mathfrak{l}_n^k$, $\mathfrak{l}_n^j$ as defined in (2.1).

Assumption 4.3. It is assumed that there exist constants $\underline{J}_i$ such that:

$$\lambda_{\min}\left[\frac{J_i(x_i, u_i) + J_i^\top(x_i, u_i)}{2}\right] \geq \underline{J}_i > 0, \quad \forall x_i \in \mathcal{W}, \quad u_i \in \mathcal{U}_i. \tag{4.5}$$

Remark 4.1. Assumptions 4.1, 4.2 are standard assumptions required for the NMPC nominal stability to be guaranteed (see [58]). Assumption 4.3 is a sufficient controllability condition for nonlinear systems in non-affine form (see e.g., [158]).

4.2.2 Objectives

Given the aforementioned modeling, the objective of each agent $i \in \mathcal{V}$ is to be steered to a predefined desired configuration $x_{i,d} \in \mathcal{W}$ of the workspace from any initial conditions $x_i(0) \in \mathcal{W}$. Moreover, motivated by practical applications in which agents need to stay sufficiently close in order to execute collaborative tasks, it is desired to introduce connectivity maintenance coupled constraints between the agents. For this reason, assume that:

- over time $t \in \mathbb{R}_{\geq 0}$, each agent $i \in \mathcal{V}$ occupies a ball $\mathcal{B}(x_i(t), \mathfrak{r}_i)$, where $\mathfrak{r}_i \in \mathbb{R}_{>0}$ stands for the radius of the ball;
- each agent $i \in \mathcal{V}$ has *communication capabilities* within a limited sensing range $\mathfrak{d}_i \in \mathbb{R}_{>0}$ such that:
$$\mathfrak{d}_i > \max_{i,j \in \mathcal{V}, i \neq j} \{\mathfrak{r}_i + \mathfrak{r}_j\}. \tag{4.6}$$

The latter implies that each agent has sufficiently large sensing radius so as to measure the agent with the biggest volume in its vicinity, due to the fact that the agents' radii are not the same. Taking the above into consideration, the neighboring set of agent $i \in \mathcal{V}$ is defined by:

$$\mathcal{N}_i \coloneqq \Big\{ j \in \mathcal{V} \backslash \{i\} : \|x_i(0) - x_j(0)\|_2 < \mathfrak{d}_i \Big\}.$$

The set $\mathcal{N}_i$ is composed of indices of agents $j \in \mathcal{N}_i$ which are within the sensing range of agent i at time $t = 0$. The proposed decentralized feedback control laws

$u_i \in \mathcal{U}_i$ need to guarantee that the neighboring agents $j \in \mathcal{N}_i$ with $j \neq i$ remain within distance $\mathfrak{d}_i$ from agent i for all times, i.e., the connectivity of all initially connected agents is preserved. We assume that the workspace $\mathcal{W}$ is sufficiently large so that the agents are able to perform the desired navigation task.

Definition 4.1. The desired configurations $x_{i,d} \in \mathcal{W}$, $i \in \mathcal{V}$ are called *feasible*, if they do not result in violation of the connectivity maintenance constraint between the agents, i.e.,

$$\|x_{i,d} - x_{j,d}\|_2 < \mathfrak{d}_i, \quad \forall i \in \mathcal{V}, \quad j \in \mathcal{N}_i.$$

4.2.3 Problem Statement

Formally, the control design problem considered in this chapter, is formulated as follows:

Problem 4.1. Given N agents governed by dynamics as in (4.1), under Assumptions 4.1- 4.3, modeled by the balls $\mathcal{B}(x_i, \mathfrak{r}_i)$, $i \in \mathcal{V}$, and operating in a bounded workspace $\mathcal{W}$. The agents have communication capabilities according to sensing radii $\mathfrak{d}_i$, as given in (4.6). Then, given desired feasible configurations $x_{i,d} \in \mathcal{W}$, $i \in \mathcal{V}$ according to Definition 4.1, the problem lies in designing *decentralized feedback control* laws $u_i \in \mathcal{U}_i$, such that for every $i \in \mathcal{V}$ and for all initial conditions satisfying $x_i(0) \in \mathcal{W}$ the following specifications are satisfied:

1. navigation to a neighborhood of the desired configuration $x_{i,d}$ is achieved;

2. connectivity between the neighboring agents is preserved:

$$\|x_i(t) - x_j(t)\|_2 < \mathfrak{d}_i, \quad \forall j \in \mathcal{N}_i, \quad t \in \mathbb{R}_{\geq 0}.$$

3. the agents remain in the workspace: $x_i(t) \in \mathcal{W}$, $\forall t \in \mathbb{R}_{\geq 0}$.

4.3 Main Results

In this section, a systematic solution to Problem 4.1 is introduced. Due to the fact that we aim to minimize the terms $\|x_i(t) - x_{i,d}\|_2$, as $t \to \infty$, subject to distance and control input constraints imposed by Problem 4.1, we seek a solution which is the outcome of a decentralized optimization. Section 4.3.1 is devoted to the proposed feedback control design; and lastly in Section 4.3.2, we deal with the online nominal NMPC design.

4.3.1 Feedback Control Design

Define the uncertain error and nominal error signals $e_i : \mathbb{R}_{\geq 0} \to \mathbb{R}^n$, $\overline{e}_i : \mathbb{R}_{\geq 0} \to \mathbb{R}^n$ by:

$$e_i(t) \coloneqq x_i(t) - x_{i,d},$$
$$\overline{e}_i(t) \coloneqq \overline{x}_i(t) - x_{i,d},$$

respectively. Then, the corresponding *uncertain error dynamics* and *nominal error dynamics* are given by:

$$\dot{e}_i(t) = f_i(e_i(t) + x_{i,d}, u_i(t)) + \delta_i(t), \tag{4.7a}$$
$$\dot{\overline{e}}_i(t) = f_i(\overline{e}_i(t) + x_{i,d}, \overline{u}_i(t)). \tag{4.7b}$$

Consider the following *feedback control law:*

$$u_i \coloneqq \overline{u}_i + \kappa_i(e_i, \overline{e}_i), \tag{4.8}$$

which consists of a nominal control action $\overline{u}_i \in \mathcal{U}_i$ and a state feedback law $\kappa_i : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}^n$. As it will be presented hereafter, $\overline{u}_i$ will be the outcome of a nominal DFHOCP solved at each sampling time step by each agent $i \in \mathcal{V}$; and the feedback law $\kappa_i(e_i, \overline{e}_i)$ is used to guarantee that the real trajectories $e_i(t)$ remain in bounded hyper-tubes centered among the nominal trajectories $\overline{e}_i(t)$, for all times. We show that the volume of the hyper-tubes depends on the upper bound of the disturbances $\widetilde{\delta}_i$ as given in (4.2) as well as the bounds of the derivatives of functions f_i. For each agent $i \in \mathcal{V}$, denote by:

$$\mathfrak{e}_i(t) \coloneqq e_i(t) - \overline{e}_i(t), \tag{4.9}$$

the deviation between the real states $e_i(t)$ of the uncertain system (4.7a) and the states $\overline{e}_i(t)$ of the nominal system (4.7b). Note that it holds that $\mathfrak{e}_i(0) = 0_{n\times 1}$ for every $i \in \mathcal{V}$. It is proven hereafter that the states $\mathfrak{e}_i(t)$ remain invariant in certain compact sets. By using (4.7a), (4.7b) and (4.9), the dynamics of $\mathfrak{e}_i(t)$ for each agent $i \in \mathcal{V}$ are given by:

$$\begin{aligned}
\dot{\mathfrak{e}}_i &= \dot{e}_i - \dot{\overline{e}}_i \\
&= f_i(e_i + x_{i,d}, u_i) + \delta_i - f_i(\overline{e}_i + x_{i,d}, \overline{u}_i) \\
&= f_i(e_i + x_{i,d}, u_i) - f_i(\overline{e}_i + x_{i,d}, u_i) + f_i(\overline{e}_i + x_{i,d}, u_i) \\
&\qquad\qquad - f_i(\overline{e}_i + x_{i,d}, \overline{u}_i) + \delta_i \\
&= g_i(e_i, \overline{e}_i, u_i) + f_i(\overline{e}_i + x_{i,d}, u_i) - f_i(\overline{e}_i + x_{i,d}, \overline{u}_i) + \delta_i.
\end{aligned} \tag{4.10}$$

In the latter, the functions $g_i : \mathcal{W} \times \mathcal{W} \times \mathcal{U}_i \to \mathbb{R}^6$ are defined by:

$$g_i(e_i, \overline{e}_i, u_i) \coloneqq f_i(e_i + x_{i,d}, u_i) - f_i(\overline{e}_i + x_{i,d}, u_i),$$

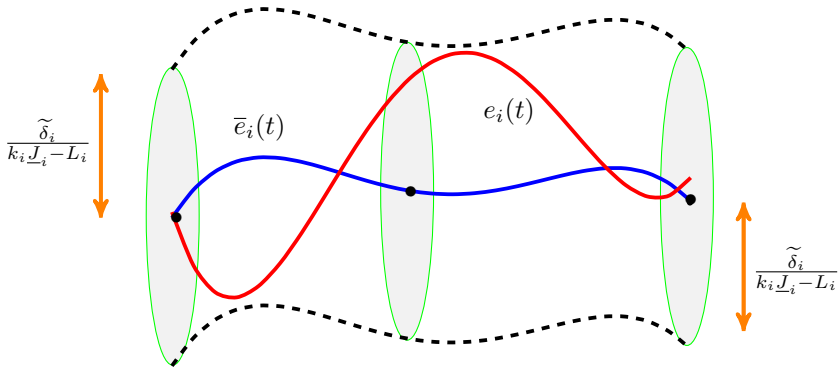

Figure 4.1: The hyper-tube of agent i centered along the trajectory $\overline{e}_i(t)$ (depicted by blue line). Under the proposed control law, the real trajectory $e_i(t)$ (depicted with red line) lies inside the hyper-tube for all times, i.e., $\mathfrak{e}_i(t) \in \Omega_i$, $\forall t \in \mathbb{R}_{\geq 0}$.

and they are upper bounded by:

$$\begin{aligned} \|g_i(e_i, \overline{e}_i, u_i)\|_2 &\leq \|f_i(e_i + x_{i,d}, u_i) - f_i(\overline{e}_i + x_{i,d}, u_i)\|_2 \\ &\leq L_i \|e_i + x_{i,d} - \overline{e}_i - x_{i,d}\|_2 \\ &= L_i \|e_i - \overline{e}_i\|_2 \\ &= L_i \|\mathfrak{e}_i\|_2, \end{aligned} \tag{4.11}$$

where $L_i > 0$ are the Lipschitz constants of the functions f_i with respect to the variables x_i.

Lemma 4.1. *The state feedback laws designed by:*

$$\kappa_i(e_i, \overline{e}_i) \coloneqq K_i(e_i - \overline{e}_i) = -k_i(e_i - \overline{e}_i) = -k_i \mathfrak{e}_i, \tag{4.12}$$

where the control gains k_i are chosen as such that the following hold:

$$k_i > \frac{L_i}{\underline{J}_i}, \tag{4.13}$$

render the sets:

$$\Omega_i \coloneqq \left\{ \mathfrak{e}_i \in \mathbb{R}^n : \|\mathfrak{e}_i\|_2 \leq \frac{\widetilde{\delta}_i}{k_i \underline{J}_i - L_i} \right\}, \tag{4.14}$$

RCI sets for the error dynamical systems (4.10)*, according to Definition 2.5.*

Proof. The proof can be found in Appendix B. □

The aforementioned result is crucial. As it will be shown hereafter, the nominal control actions $\overline{u}_i(t)$ will be the solution of a repetitive DFHOCP solved at every sample time from each agent $i \in \mathcal{V}$. The nominal trajectories $\overline{e}_i(t)$ computed by the DFHOCP define a hyper-tube centered along them with radius $\frac{\widetilde{\delta}_i}{k_i \underline{J}_i - L_i}$ (see Figure 4.1). The real trajectories $e_i(t)$ will always lie inside the above tube.
By using (4.8), (4.12), the *closed-loop system* of each agent is written as:

$$\dot{e}_i = f_i\Big(e_i, \overline{u}_i + K_i\left[e_i - \overline{e}_i\right]\Big) + \delta_i, \quad i \in \mathcal{V}. \tag{4.15}$$

Remark 4.2. The volume of the hyper-tubes depends on the bound of the disturbances $\widetilde{\delta}_i$, the Lipschitz constants L_i of functions f_i as well as the constants $\underline{J}_i$. Moreover, by tuning the parameters k_i appropriately, we can adjust the volume of the tube.

Define the sets that capture the *state constraints* of each agent and they are posed by Problem 4.1 by:

$$\mathcal{E}_i \coloneqq \Big\{x_i \in \mathbb{R}^n : \ \|e_i + x_{i,d} - e_j - x_{j,d}\|_2 < \mathfrak{d}_i, \forall\ j \in \mathcal{N}_i, \\ \text{and } e_i + x_{i,d} \in \mathcal{W}\Big\}, i \in \mathcal{V}, \tag{4.16}$$

The two constraints in the set $\mathcal{E}_i$ refer to connectivity preservation between neighboring agents and the requirement of the agents to remain in the workspace $\mathcal{W}$ for all times, respectively.

4.3.2 Decentralized Online Control Design of $\overline{u}_i(t)$

Consider a sequence of sampling time steps $\{t_k\}$, $k \in \mathbb{N}$, with a constant sampling period h, $0 < h < T$, where T is the prediction horizon, such that $t_{k+1} \coloneqq t_k + h$, $\forall k \in \mathbb{N}$. At every discrete sampling time step t_k, a DFHOCP is solved as follows:

$$\min_{\overline{u}_i(\cdot)} \left\{ \|\overline{e}_i(t_k + T)\|^2_{P_i} + \int_{t_k}^{t_k+T} \Big[\|\overline{e}_i(\mathfrak{s})\|^2_{Q_i} + \|\overline{u}_i(\mathfrak{s})\|^2_{R_i}\Big] d\mathfrak{s} \right\} \tag{4.17a}$$

subject to:

$$\dot{\overline{e}}_i(\mathfrak{s}) = f_i(\overline{e}_i(\mathfrak{s}) + x_{i,d}, \overline{u}_i(\mathfrak{s})), \tag{4.17b}$$

$$\overline{e}_i(\mathfrak{s}) \in \overline{\mathcal{E}}_i, \quad \overline{u}_i(\mathfrak{s}) \in \overline{\mathcal{U}}_i, \quad \mathfrak{s} \in [t_k, t_k + T], \tag{4.17c}$$

$$\overline{e}_i(t_k + T) \in \mathcal{F}_i. \tag{4.17d}$$

The matrices Q_i, $P_i \in \mathbb{R}^n$ and $R_i \in \mathbb{R}^n$ are positive definite weight matrices. The sets $\mathcal{F}_i$ stands for the terminal set that is used to enforce the stability of the nominal system (see [58] for more details). Hereafter, the sets $\overline{\mathcal{E}}_i$ and $\overline{\mathcal{U}}_i$ are explained. In

Algorithm 1 Implementation of feedback control laws $u_i(t)$, $i \in \mathcal{V}$

Step 0: At time $t_0 := 0$, set $e_i(0) = \overline{e}_i(0)$.
Step 1: At time t_k and current state $(e_i(t_k), \overline{e}_i(t_k))$, solve DFHOCP (4.17a)-(4.17d) to obtain the nominal control action $\overline{u}_i(t_k)$ and the actual control action

$$u_i(t_k) = \overline{u}_i(t_k) + \kappa_i(e_i(t_k), \overline{e}_i(t_k)).$$

Step 2: Apply the control $u_i(t_k)$ to the system (4.7a), during sampling interval $[t_k, t_{k+1})$, where $t_{k+1} = t_k + h$.
Step 3: Measure the state $e_i(t_{k+1})$ at the next time instant t_{k+1} of the system (4.7a) and compute the successor state $\overline{e}_i(t_{k+1})$ of the nominal system (4.7b) under the nominal control action $\overline{u}_i(t_k)$.
Step 4: Set

$$(e_i(t_k), \overline{e}_i(t_k)) \leftarrow (e_i(t_{k+1}), \overline{e}_i(t_{k+1})),$$
$$t_k \leftarrow t_{k+1}$$

Step 5 : Go to **Step 1**.

order to guarantee that while the FHOCP (4.17a)-(4.17d) is solved for the nominal dynamics (4.7b), the real states e_i and control inputs u_i satisfy the corresponding state and input constraints $\mathcal{E}_i$ and $\mathcal{U}_i$, respectively, the latter sets are appropriately modified. By recalling that:

$$\overline{e}_i = e_i - \mathfrak{e}_i,$$
$$\overline{u}_i = u_i - K_i\mathfrak{e}_i,$$

and employing the set operators of Definition 2.2, the constraint tightening is performed as:

$$\overline{\mathcal{E}}_i := \mathcal{E}_i \ominus \Omega_i, \tag{4.18a}$$
$$\overline{\mathcal{U}}_i := \mathcal{U}_i \ominus K_i\Omega_i, \tag{4.18b}$$

with Ω_i as given in (4.14). The advantage of the tube-based framework compared to other robust NMPC approaches is that the constraint tightening is performed offline and it does not depend on the length of the horizon. Algorithm 1 depicts the procedure of how the control law is calculated and applied to a real robot. This is a procedure of implementing the continuous-time tube-based NMPC in a real-time system that has been introduced in [123].

Due to the fact that the connectivity between initially connected agents, i.e, $j \in \mathcal{N}_i, \forall i \in \mathcal{V}$, needs to be preserved and the agents have communication capabilities within the sensing range $\mathfrak{d}_i$ as given in (4.6), we adopt here a decentralized procedure explained as follows. Assume that each agent knows its labeling number of the set $\mathcal{V}$. After each sampling time step t_k, $\forall k \geq 0$ that agent i solves its own DFHOCP and obtains the estimated open-loop trajectory $\overline{e}_i(\mathfrak{s})$, $\mathfrak{s} \in [t_k, t_k + T]$, it transmits it to its neighboring agents $j \in \mathcal{N}_i$. Then, agents' $j \in \mathcal{N}_i$ hard constraints $\overline{\mathcal{E}}_j$ are updated by incorporating the predicted trajectory of agent i, i.e., $\overline{e}_i(\mathfrak{s})$, $\mathfrak{s} \in [t_k, t_k + T]$. Among all agents $j \in \mathcal{N}_i$, the one with higher priority, i.e., smaller labeling number in the set $\mathcal{V}$, solves its own DFHOCP (for example, agent 2 has higher priority than agents 3, 4, ...). This *sequential procedure* is continued until all agents $i \in \mathcal{V}$ solve their own DFHOCP, and then the sampling time step is updated. It will be showed thereafter that by adopting the aforementioned sequential communication procedure, and given that at $t = 0$ the DFHOCP (4.17a) - (4.17d) of all agents are feasible, the agents are navigated to their desired configurations $x_{i,d}$, while all distance and input constraints imposed by Problem 4.1 are satisfied.

In other words, each time an agent solves its own individual optimization problem, it knows the (open-loop) state predictions that have been generated by the solution of the optimization problem of all agents within agent i's sensing range at that time, for the next T time units. These pieces of information are required, as each agent's trajectory is constrained not by constant values, but by the trajectories of its associated agents through time: at each solution time t_k and within the next T time units, an agent's predicted configuration at time $\mathfrak{s} \in [t_k, t_k + T]$ needs to be constrained by the predicted configuration of its neighboring and perceivable agents (agents within its sensing range) at the same time instant $\mathfrak{s}$, so that collisions are avoided. We assume that the above pieces of information are *always available*, *accurate* and can be exchanged without delay.

Remark 4.3. It should be noted that the constraint sets $\overline{\mathcal{E}}_i$, $i \in \mathcal{V}$ in (4.17c) depend on the estimated open-loop trajectories $\overline{e}_i(\mathfrak{s})$ and $\overline{e}_j(\mathfrak{s})$ for all $i \in \mathcal{V}$ and $j \in \mathcal{N}_i$. Moreover, they are updated when each agent has received the transmitted trajectories by its neighbors.

Under these considerations, the theorem that guarantees the stability of the system (4.1), under the proposed control laws (4.8), (4.13) for all initial conditions $e_i(0) \in \mathcal{E}_i$, $i \in \mathcal{V}$ is stated as follows.

Theorem 4.1. *Suppose that Assumption 4.1 - 4.3 hold. Suppose also that at time* $t = 0$ *the DFHOCP* (4.17a)-(4.17d) *sequentially solved by all the agents* $i \in \mathcal{V}$*, is feasible. Then, the proposed decentralized feedback control laws* (4.8), (4.13)*, render the closed-loop system* (4.15) *of each agent* $i \in \mathcal{V}$ *ISS with respect to* $\delta_i \in \Delta_i$*, for every initial conditions* $e_i(0) \in \mathcal{E}_i$*.*

Proof. The proof can be found in Appendix B. □

Remark 4.4. Assumption 4.1, Assumption 4.2 and communication capabilities among the agents are standard assumptions in order for the nominal stability of decentralized NMPC schemes to be guaranteed. We refer the reader to [58, 157] for more details.

Remark 4.5. The proposed framework can deal with any graph configuration of the initially connected agents without any further assumptions. In particular, the initial state of the agents $x_i(0)$, $i \in \mathcal{V}$ specifies the structure of the sets $\mathcal{N}_i$. With the proposed decentralized feedback control laws, the neighboring sets $\mathcal{N}_i$ has been proven to be invariant, i.e., the connectivity of all initially connected agents is preserved.

Remark 4.6. A major advantage of tube-based approach compared to other robust NMPC approaches, is that the DFHOCP is solved only for the nominal system dynamics, thus the complexity is the same with the complexity of solving a nominal sampled-data NMPC (see [123] for more details).

Remark 4.7. In this chapter, we have considered nonlinear systems that the state x_i and the control input u_i are in the dimension n. A tube-based design for second-order nonlinear systems can be found in Appendix D. Tube-based NMPC analysis for nonhonomic systems is provided in Appendix E.

Remark 4.8. It should be noted that the volume of the tubes depends on the upper bound of the disturbances $\widetilde{\delta}_i$, the Lipschitz constant L_i and the constant $\underline{J}_i$. By tuning the gains k_i as in (4.13) appropriately, the volume of the tube can be adjusted. Moreover, note that as the gain k_i grows, the set $K_i\Omega_i$ is enlarged. Thus, there is an upper bound of how big the gain k_i can be set, since it is required that the sets $\overline{\mathcal{U}}_i$ from (4.18b) is non-empty in order for the DFHOCP (4.17a)-(4.17d) to have a feasible solution.

4.4 Simulation Results

For a simulation scenario, consider $N = 3$ agents $\mathcal{V} = \{1, 2, 3\}$ with uncertain nonlinear dynamics given as follows:

$$\begin{aligned}\dot{x}_{i,1} &= \frac{0.1 - 0.1e^{-x_{i,2}}}{1 + e^{-x_{i,2}}} + 0.25x_{i,1}^2 + 2u_{i,1} + 0.3\cos(t),\\ \dot{x}_{i,2} &= 0.25x_{i,1}^2 + u_{i,2} + 0.1u_{i,2}^3 + 0.2\cos(2t),\end{aligned}$$

where:

$$\begin{aligned}x_i &= [x_{i,1}, x_{i,2}]^\top \in \mathbb{R}^2,\\ u_i &= [u_{i,1}, u_{i,2}]^\top \in \mathbb{R}^2,\\ f_i(x_i, u_i) &= \begin{bmatrix} f_{i,1}(x_i, u_i) \\ f_{i,2}(x_i, u_i) \end{bmatrix} = \begin{bmatrix} \frac{0.1 - 0.1e^{-x_{i,2}}}{1 + e^{-x_{i,2}}} + 0.25x_{i,1}^2 + 2u_{i,1} \\ 0.25x_{i,1}^2 + u_{i,2} + 0.1u_{i,2}^3 \end{bmatrix},\\ \delta_i(t) &= [0.3\cos(t), 0.2\cos(2t)]^\top,\end{aligned}$$

with $\|\delta_i(t)\|_2 \leq 0.3 = \widetilde{\delta}_i$, $\forall t \in \mathbb{R}_{\geq 0}$. From (4.4) we get:

$$\begin{aligned}J_i(x_i, u_i) &= \sum_{k=1}^{2}\sum_{j=1}^{2} \mathfrak{l}_n(k)\mathfrak{l}_n(j)^\top \frac{\partial f_{i,k}(x_i, u_i)}{\partial u_j}\\ &= \begin{bmatrix} 2 & 0 \\ 0 & 1 + 0.3u_{i,2}^2 \end{bmatrix},\end{aligned}$$

with:

$$\lambda_{\min}\left(\frac{J_i + J_i^\top}{2}\right) \geq \underline{J}_i = 1.$$

By observing the latter inequality, the fact that f_i are continuously differentiable with $f_i(0_{2\times 1}, 0_{2\times 1}) = 0_{2\times 1}$ and the fact that the Jacobian linearization of the system is stabilizable, it holds that Assumption 4.1 - 4.3 are satisfied. The agents are operating in the workspace:

$$\mathcal{W} = \{x_i \in \mathbb{R}^2 : -5 \leq x_{i,1}, x_{i,2} \leq 5\},$$

with $L_i = 2.5$. The radius and the sensing range of the agents are set to $\mathfrak{r}_i = 1$ and $\mathfrak{d}_i = 5$, respectively. The sensing radii result to the following neighboring sets: $\mathcal{N}_1 = \{2\} = \mathcal{N}_3$ and $\mathcal{N}_2 = \{1, 3\}$. The agents' initial positions are:

$$x_1(0) = [-3.0, 2.9]^\top, \quad x_2(0) = [-2.5, -0.2]^\top, \quad x_3(0) = [-2.9, -4]^\top.$$

Their corresponding desired configurations are:

$$x_{1,d} = [0.1206, 1.1155]^\top, \quad x_{2,d} = [2.0, 0.0]^\top, \quad x_{3,d} = [0.9, -2.8]^\top.$$

According to Definition 4.1, the above configurations are feasible since it holds that:

$$\|x_{i,d} - x_{j,d}\| < \mathfrak{d}_i, \quad \forall i \in \mathcal{V}, \quad j \in \mathcal{N}_i.$$

The sampling time step and the total execution time are $h = 0.1\,\text{sec}$ and $10\,\text{sec}$, respectively. The control gains are chosen as $k_i = 3.5$, which result to a tube of radius 0.3. The matrices Q_i, R_i and P_i are set to $0.5I_2$. The initial error constraints of each agent are given by:

$$\mathcal{E}_1 = \{e_1 \in \mathbb{R}^2 : -5.1206 \leq e_{1,1} \leq 4.8794, -6.1155 \leq e_{1,2} \leq 3.8845\},$$
$$\mathcal{E}_2 = \{e_2 \in \mathbb{R}^2 : -7.0 \leq e_{2,1} \leq 3.0, -5.0 \leq e_{2,2} \leq 5.0\},$$
$$\mathcal{E}_3 = \{e_3 \in \mathbb{R}^2 : -5.9 \leq e_{3,1} \leq 4.1, -2.2 \leq e_{3,2} \leq 2.2\},$$

and the corresponding modified error constraints which are used for the solution of the online NMPC as:

$$\overline{\mathcal{E}}_1 = \{e_1 \in \mathbb{R}^2 : -4.8206 \leq e_{1,1} \leq 4.5794, -5.8155 \leq e_{1,2} \leq 3.5845\},$$
$$\overline{\mathcal{E}}_2 = \{e_2 \in \mathbb{R}^2 : -6.7 \leq e_{2,1} \leq 2.7, -4.7 \leq e_{2,2} \leq 4.7\},$$
$$\overline{\mathcal{E}}_3 = \{e_3 \in \mathbb{R}^2 : -5.6 \leq e_{3,1} \leq 3.8, -1.9 \leq e_{3,2} \leq 1.9\}.$$

The input constraints of each agent are set to:

$$\mathcal{U}_i = \{u_i \in \mathbb{R}^2 : -2.125 \leq u_{i,1}, u_{i,2} \leq 2.125\}, \quad i \in \mathcal{V}.$$

The corresponding modified input constraint sets for the online NMPC are given as:

$$\overline{\mathcal{U}}_i = \{\overline{u}_i \in \mathbb{R}^2 : -1.075 \leq \overline{u}_{i,1}, \overline{u}_{i,2} \leq 1.075\}, \quad i \in \mathcal{V}.$$

Fig. 4.3, Fig. 4.5 and Fig. 4.7 depict the evolution of the real and the nominal trajectories as well as the tubes of agents 1, 2 and 3, respectively. The tubes are centered along the nominal trajectory of each agent, while the real trajectory always remain within the tubes for all times. Fig. 4.2, Fig. 4.4 and Fig. 4.6 represent the evolution of the error signals $e_1(t)$, $e_2(t)$ and $e_3(t)$, respectively. The evolution of the trajectories of all agents in the workspace is depicted in Fig. 4.8. The distance between the neighboring agents $1-2$ and $2-3$ is represented in Fig. 4.9. Finally, the control effort of each agent is shown in Fig. 4.10.

It can be observed that all agents reach their desired configurations by satisfying all the constraints imposed by Problem 1. The simulation was performed in MATLAB R2015a Environment utilizing the NMPC optimization toolbox provided in [62].

Remark 4.9. It should be noted that in this chapter heterogeneous agents are considered, i.e., functions f_i, $i \in \mathcal{V}$ in (4.1) may be different for each agent. In the aforementioned simulation example, for convenience and simplified calculations, we considered homogeneous agents, i.e., the functions f_i are the same for all agents.

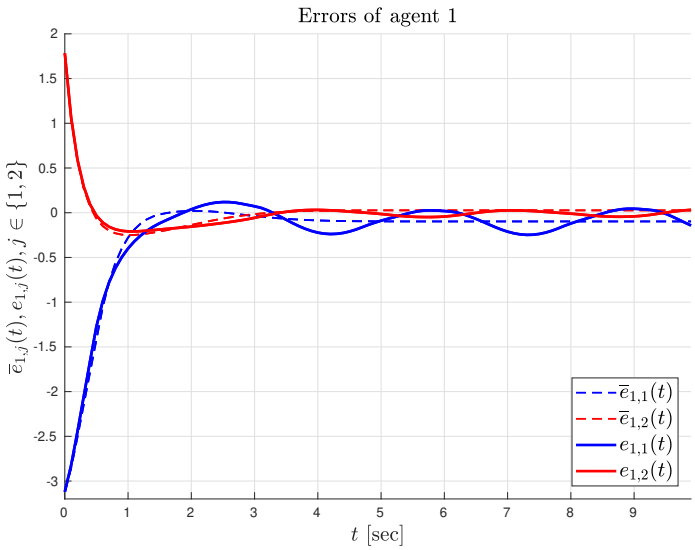

Figure 4.2: The evolution of the real error signal $e_1(t)$ and the nominal error signal $\overline{e}_1(t)$ over the time interval $[0, 10]$ sec.

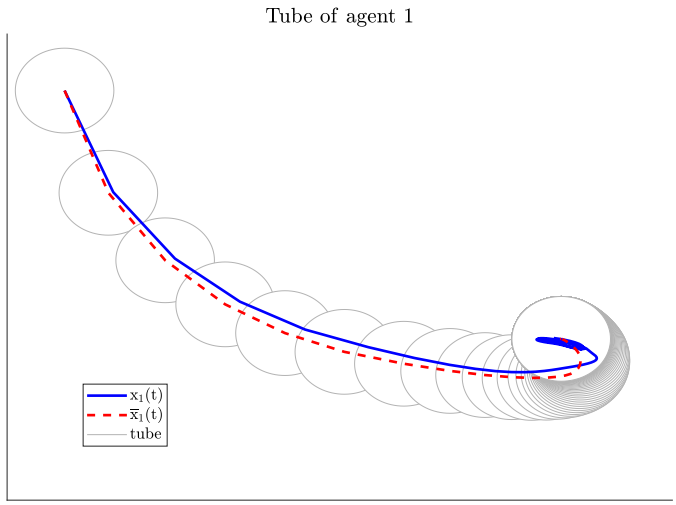

Figure 4.3: The evolution of the trajectory of agent 1 in the workspace $\mathcal{W}$ over the time interval $[0, 10]$ sec. The solid and the dashed lines represent the real and the nominal trajectory, respectively. The gray circles represent the tube.

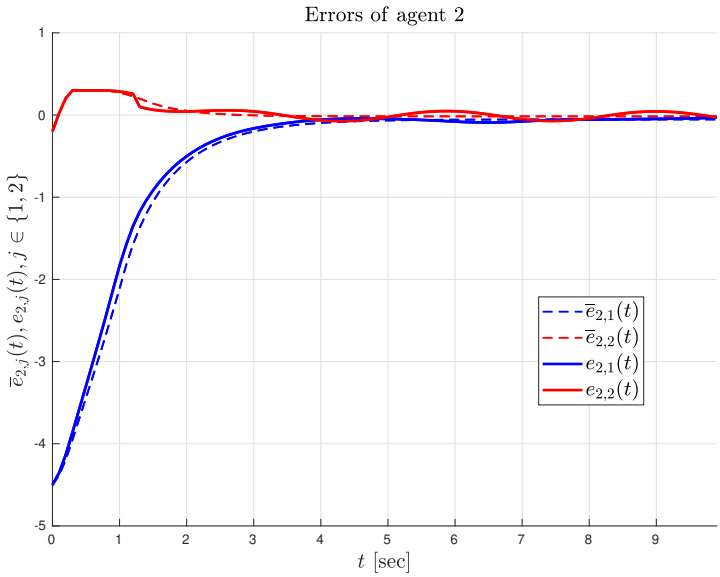

Figure 4.4: The evolution of the real error signals $e_2(t)$ and the nominal error signals $\overline{e}_2(t)$ over the time interval $[0, 10]$ sec.

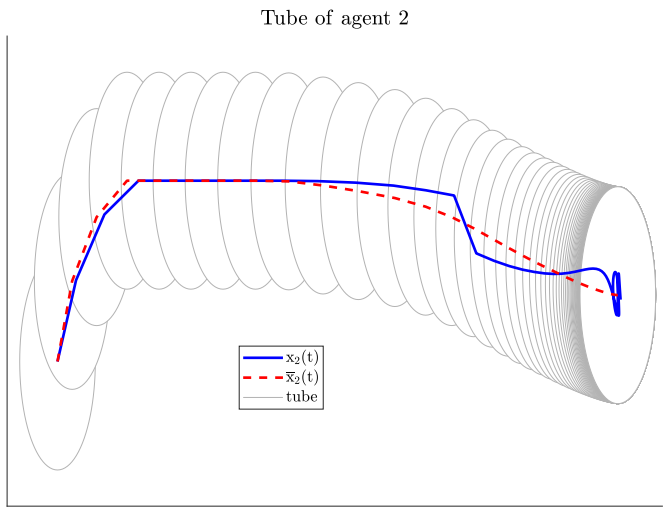

Figure 4.5: The evolution of the trajectory of agent 2 in the workspace $\mathcal{W}$ over the time interval $[0, 10]$ sec. The solid and the dashed lines represent the real and the nominal trajectory, respectively. The gray circles represent the tube.

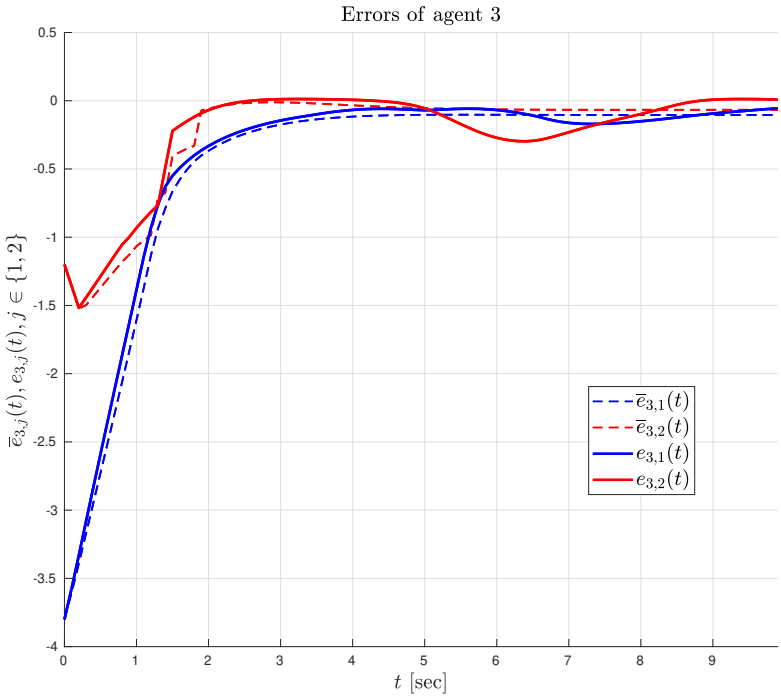

Figure 4.6: The evolution of the real error signals $e_3(t)$ and the nominal error signals $\overline{e}_3(t)$ over the time interval $[0, 10]$ sec.

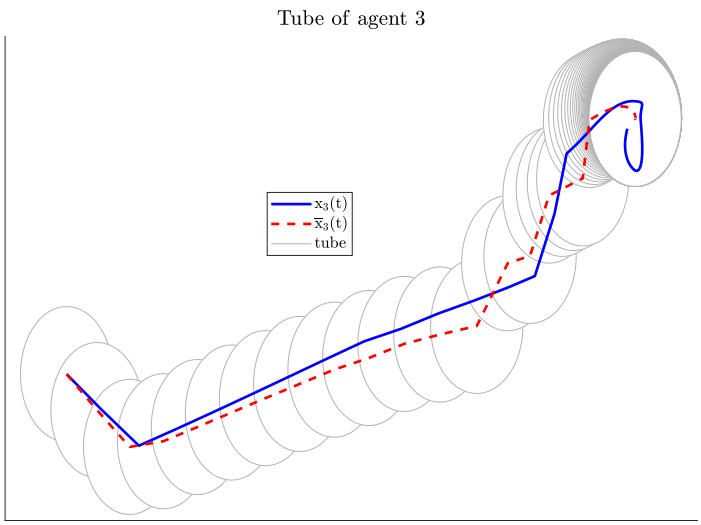

Figure 4.7: The evolution of the trajectory of agent 3 in the workspace $\mathcal{W}$ over the time interval $[0, 10]$ sec. The solid and the dashed lines represent the real and the nominal trajectory, respectively. The gray circles represent the tube.

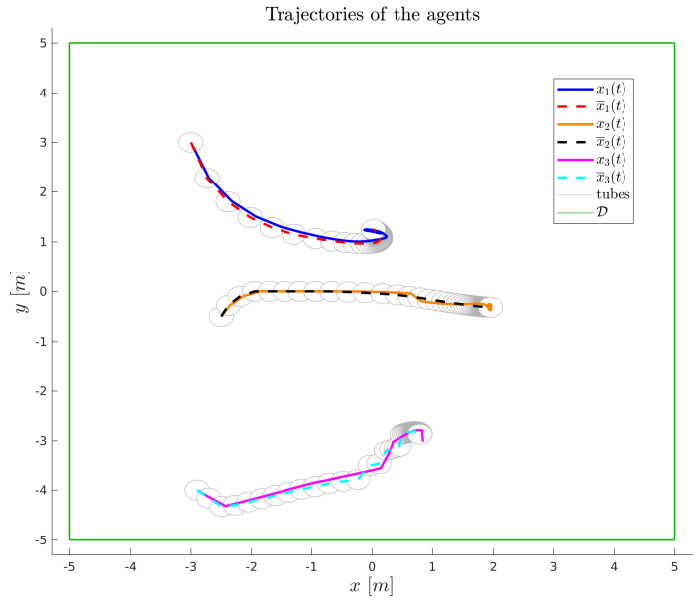

Figure 4.8: The workspace $\mathcal{W}$ along with the trajectories of all agents.

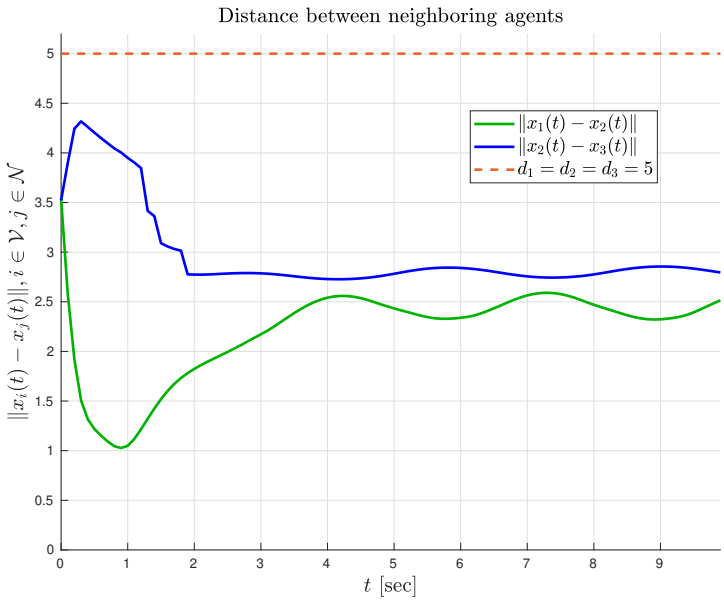

Figure 4.9: The distance between the neighboring agents $1-2$ and $2-3$. The distance remains below the threshold $d_1 = d_2 = d_3 = 5$ for all times, i.e., the connectivity of the neighboring agents is preserve.

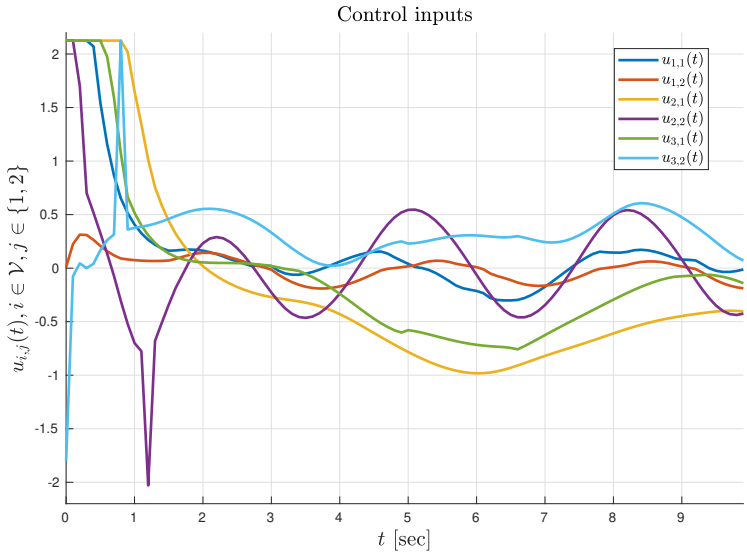

Figure 4.10: The control input signals $u_i(t)$, $i \in \mathcal{V}$ satisfying the constraints $u_i(t) \in \mathcal{U}_i$, $\forall i \in \mathcal{V}$, $t \in \mathbb{R}_{\geq 0}$.

4.5 Summary

This chapter investigates the problem of decentralized tube-based MPC for uncertain nonlinear continuous-time multi-agent systems. Each agent has a limited sensing range within which can exchange information with neighboring agents. The task involves navigation to predefined configurations with connectivity preservation of the initially connected agents. Each agent solves a nominal DFHOCP in order to calculate a potion of its control input. The other portion of the control input is calculated offline in order to guarantee that the real trajectory of the agent remains in a bounded hyper-tube for all times, due to disturbances. Simulation results verify the proposed approach.

Chapter 5

Time-constrained High-Level Planning of Multi-agent Systems

In this chapter the problem of cooperative task planning of multi-agent systems subject to imposed time constraints is investigated. We consider timed constraints given by Metric Interval Temporal Logic (MITL). We propose a method for automatic control synthesis in a two-stage systematic procedure. Under the proposed method, it is guaranteed that all the agents satisfy their own individual task specifications as well as the team satisfies a team task specification.

5.1 Introduction

The specification language that has extensively been used to express desired tasks for robots is Linear Temporal Logic (LTL) (see, e.g., [82–90]). LTL has proven a valuable tool for controller synthesis, because it provides a compact mathematical formalism for specifying desired behaviors of a system. There is a rich body of literature containing algorithms for verification and synthesis of multi-agent systems under temporal logic specifications [98]. A three-step hierarchical procedure to address the problem of multi-agent systems under LTL specifications is described as follows [91–97]: first the dynamics of each agent is abstracted into a discrete transition system using abstractions methods. Second, by invoking ideas from formal verification, a discrete plan that meets the high-level tasks is synthesized for each agent. Third, the discrete plan is translated into a sequence of continuous controllers for the original continuous dynamical system of each agent.

Explicit time constraints in the system modeling have been included e.g., in [84], where a method of automated planning of optimal paths of a group of agents satisfying a common high-level mission specification was proposed. The mission was given in LTL and the goal was the minimization of a cost function that captures the maximum time between successive satisfactions of the formula. Authors in [99, 100] used a different approach, representing the motion of each agent in the environment with a timed automaton. The composition of the team automaton

was achieved through synchronization and the UPPAAL verification tool ([101]) was utilized for specifications given in Computational Tree Logic (CTL). In the same direction, authors in [102] modeled the multi-robot framework with timed automata and weighted transition systems considering LTL specifications and then, an optimal motion of the robots satisfying instances of the optimizing proposition was proposed.

Most of the previous works on multi-agent planning consider temporal properties which essentially treat time in a qualitative manner. For real applications, a multi-agent team might be required to perform a specific task within a certain time bound, rather than at some arbitrary time in the future (quantitative manner). Controller synthesis under timed specifications has been considered in [103–107]. In [103], an optimal control problem for continuous-time stochastic systems subject to objectives specified in MITL was studied, whereas in [104], a framework that enables online planning for robotic systems in dynamic environments under Metric Temporal Logic (MTL) specifications is presented. In [105], the authors focused on motion planning based on the construction of a timed automaton from a given MITL specification. In [106], the MTL formula for a single-agent was translated into linear constraints and a Mixed Integer Linear Programming (MILP) problem was solved. Authors in [107] have studied timed motion objectives for time-abstraction robot navigation. However, all these works are restricted to single-agent planning and cannot be extended to multi-agent systems in a straightforward way. The multi-agent case has been considered in [159, 160], where the authors have addressed the Vehicle Routing Problem (VHP), which is modeled as an optimization problem, that aims at finding the optimal set of routes for a set of vehicles to traverse, in order to deliver the load to a given set of customers. The corresponding approach does not rely on automata-based verification, as it is based on a construction of linear inequalities and the solution of a Mixed-Integer Linear Programming (MILP) problem.

Motivated by the above, in this chapter, we aim at designing an automated planning procedure for a team of agents. The agents are assigned with an individual, independent timed temporal specification each and, at the same time, a single global team specification. This constitutes the first step towards including time constraints to temporal logic-based multi-agent control synthesis. We consider a quantitative logic called MITL in order to specify explicit time constraints. The proposed solution is fully automated, decentralized in handling the individual specifications and centralized only in handling the global team specification.

The remainder of the chapter is structured as follows. Section 5.2 provides the model of the multi-agent system, the task specification, several motivation examples as well as the formal problem statement. Section 5.3 discusses the technical details of the solution. Section 5.4 is devoted to an illustrative example. Finally, the conclusions are discussed in Section 5.5.

5.2 Problem Formulation

5.2.1 System Model

Consider a multi-agent team composed of N agents operating in a bounded workspace $\mathcal{W} \subseteq \mathbb{R}^n$. Let $\mathcal{V} = \{1, \ldots, N\}$ denote the index set of the agents. We assume that the workspace $\mathcal{W}$ is partitioned into a finite number (assume Z) of regions of interest $\pi_1, \ldots, \pi_Z$ where:

$$\mathcal{W} = \bigcup_{k \in \mathcal{Z}} \pi_k \text{ and } \pi_k \cap \pi_{k'} \neq \emptyset, \forall\ k \neq k' \text{ with } k, k' \in \mathcal{Z}, \tag{5.1}$$

for the index set $\mathcal{Z} = \{1, \ldots, Z\}$. We denote by π_k^i the agent i being at region π_k, where $i \in \mathcal{V}, k \in \mathcal{Z}$. In this work, we focus on interaction and high-level control strategies rather than on nonlinear models, and we assume that the dynamics of each agent are given by:

$$\dot{x}_i = u_i, \quad u_i \in \mathcal{U}_i, \quad i \in \mathcal{V}, \tag{5.2}$$

where $\mathcal{U}_i$ is a set of input constraints. The partitioned environment (5.1) is a discretization that allows us to control the agents with dynamics (5.2) using finite models such as finite transition systems (e.g., [92, 110, 161, 162]). We define a weighted transition system (see Definition 5.1) so that:

- if there exists a controller $u_i,\ i \in \mathcal{V}$ such that agent i can be driven from any point within the region π^i to a neighboring region π^j, then we allow for a transition between the respective system states; and

- the weight of each transition estimates the time each agent needs in order to move from one region to another. In particular, the travel time is here determined as the worst-case shortest time needed to travel from an arbitrary point of the current region to the boundary of the following region. This estimate is indeed conservative, however, it is sufficient for specifications that we are generally interested in within multi-agent control. Namely, it is suitable for scenarios where tasks are given deadlines and upper rather than lower bound requirements are associated with events along the agents' runs.

Definition 5.1. The motion of each agent $i \in \mathcal{V}$ in the workspace is modeled by a WTS $\mathcal{T}_i = (\Pi_i, \Pi_i^{\text{init}}, \longrightarrow_i, \mathfrak{t}_i, \Sigma_i, \mathcal{L}_i)$ where

- $\Pi_i = \left\{\pi_1^i, \pi_2^i, \ldots, \pi_Z^i\right\}$ is the set of states of agent i. Any state of an agent i can be denoted as $\pi_k^i \in \Pi_i$ for $i \in \mathcal{V}$, $k \in \mathcal{Z}$. The number of states for each agent is $|\Pi_i| = Z$.

- $\Pi_i^{\text{init}} \subseteq \Pi_i$ is the initial states of agent i, i.e. the set of regions where agent i may start.

- $\longrightarrow_i \subseteq \Pi_i \times \Pi_i$ is the transition relation. For example, by $\pi_3^3 \longrightarrow_3 \pi_5^3$ we mean that agent 3 can move from region π_3 to region π_5.

- $\mathfrak{t}_i : \longrightarrow_i \rightarrow \mathbb{Q}_{\geq 0}$ is a map that assigns a positive weight (duration) to each transition. For example, $\mathfrak{t}_2(\pi_2^2, \pi_5^2) = 0.7$, where $\pi_2^2 \longrightarrow_2 \pi_5^2$, means that agent 2 needs at most 0.7 time units to move from any point of region π_2 to the boundary of the neighboring region π_5.

- Σ_i is a finite set of atomic propositions known to agent i. We assume that $\Sigma_i \cap \Sigma_{i'} = \emptyset$ for all i, $i' \in \mathcal{V}$ with $i \neq i'$.

- $\mathcal{L}_i : \Pi_i \rightarrow 2^{\Sigma_i}$ is a labeling function that assigns to each state $\pi_k^i \in \Pi_i$, $k \in \mathcal{Z}$, a subset of atomic propositions Σ_i that are satisfied when agent i is in region π_k.

Individual Timed Runs and Words

The behaviors of the individual agents can be captured through their timed runs and timed words. The timed run

$$r_i^t = (r_i(0), \tau_i(0))(r_i(1), \tau_i(1))(r_i(2), \tau_i(2)) \ldots,$$

of each WTS $\mathcal{T}_i,\ i \in \mathcal{V}$ and the corresponding timed word

$$w_i^t = (\mathcal{L}_i(r_i(0)), \tau_i(0))(\mathcal{L}_i(r_i(1)), \tau_i(1))(\mathcal{L}_i(r_i(2))), \tau_i(2)) \ldots,$$

are defined by using the terminology of Definition 2.12.

Remark 5.1. Note that in this chapter, we omit the set of actions Act from the definition of WTS, the definition of timed runs and timed words, as they are defined in Definition 2.11 and Definition 2.12.

Collective Timed Run and Word

At the same time, the agents form a team and we are interested in their global, collective behaviors, which is formalized through the following definition.

Definition 5.2 (*Collective Run of the Agents*)**.** Let $r_1^t, \ldots, r_N^t$ be individual timed runs of the agents $1, \ldots, N$, respectively, as defined above. Then, the *collective timed run*

$$r_G = (r_G(0), \tau_G(0))(r_G(1), \tau_G(1))(r_G(2), \tau_G(2)) \ldots,$$

of the team of agents is defined inductively as follows:

1. $(r_G(0), \tau_G(0)) = ((r_1(0), \ldots, r_N(0)), \tau_G(0))$.
2. Let $(r_G(j), \tau_G(j)) = ((r_1(j_1), \ldots, r_N(j_N)), \tau_G(j))$, where $j \geq 0$ is an index for the current state and time stamp of the collective timed run. Then, the next state and time stamp $(r_G(j+1), \tau_G(j+1)) = ((r_1(z_1), \ldots, r_N(z_N)), \tau_G(j+1))$ are given by the following rules:
   - $\ell = \underset{i \in \mathcal{V}}{\operatorname{argmin}} \{\tau_i(j_i + 1)\}$.
   - $\tau_G(j+1) = \tau_\ell(j_\ell + 1)$.
   - $r_i(z_i) = \begin{cases} r_\ell(j_\ell + 1) & \text{if } i = \ell \\ r_i(j_\ell) & \text{if } i \neq \ell. \end{cases}$

Intuitively, given the current states $r_1(j_1), \ldots, r_N(j_N)$ and the next states $r_1(j_1 + 1), \ldots, r_N(j_N + 1)$ of the individual agents at time $\tau_G(j)$, ℓ is the index of the agent i who will finish its current transition from $r_\ell(j_\ell)$ to $r_\ell(j_\ell + 1)$ the soonest amongst of all the agents. The time of agent ℓ's arrival to its next state $r_\ell(j_\ell + 1)$ becomes the new time stamp $\tau_G(j+1)$ of the collective timed run. The next state of the collective timed run reflects that each agent i which cannot complete its transition from $r_i(j_k)$ to $r_i(j_k + 1)$ before $\tau_G(j+1)$ remains in $r_i(j_i)$.

In what follows, we write:

$$r_G^t = (r_G(0), \tau_G(0))(r_G(1), \tau_G(1)) \ldots,$$

where

$$r_G(j) = (r_1(j_1), \ldots, r_N(j_N)), \quad j, j_i \geq 0, i \in \mathcal{V},$$

denotes the collective timed run.

Definition 5.3. We define the global set of atomic propositions $\Sigma_G = \bigcup_{i \in \mathcal{V}} \Sigma_i$ and for every state $r_G(j) = (r_1(j_1), \ldots, r_N(j_N))$ of a collective timed run, where $j, j_i \geq 0$ and $i \in \mathcal{V}$, we define the labeling function $\mathcal{L}_G : \Pi_1 \times \cdots \times \Pi_N \to \Sigma_G$ as

$$\mathcal{L}_G(r_G(j)) = \bigcup_{i \in \mathcal{V}}^{N} \mathcal{L}_i(r_i(j_i)).$$

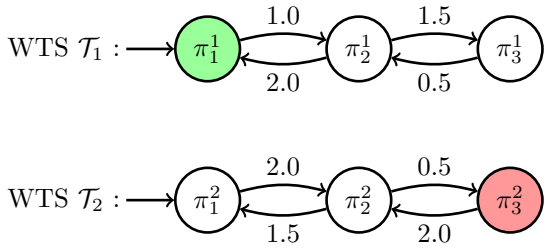

Figure 5.1: Two WTSs $\mathcal{T}_1, \mathcal{T}_2$ representing two agents with $\Pi_1 = \{\pi_1^1, \pi_2^1, \pi_3^1\}$, $\Pi_1^{\text{init}} = \{\pi_1^1\}$, $\Pi_2 = \{\pi_1^2, \pi_2^2, \pi_3^2\}, \Pi_2^{\text{init}} = \{\pi_1^2\}$. The transitions are depicted with arrows which are annotated with the corresponding weights.

Therefore, a collective timed run r_G^t naturally produces a timed word

$$w_G^t = (\mathcal{L}_G(r_G(0)), \tau_G(0))(\mathcal{L}_G(r_G(1)), \tau_G(1)) \dots,$$

over the set Σ_G.

Example 5.1. Consider $N = 2$ agents operating in a workspace with $\mathcal{W} = \pi_1 \cup \pi_2 \cup \pi_3$ and $\mathcal{V} = \{1, 2\}$ modeled by the WTSs illustrated in Figure 5.1. Let the sets of atomic propositions be $\Sigma_1 = \{\text{green}\}$, and $\Sigma_2 = \{\text{red}\}$. The labeling functions are $\mathcal{L}_1(\pi_1^1) = \{\text{green}\}$, $\mathcal{L}_1(\pi_2^1) = \mathcal{L}_1(\pi_3^1) = \emptyset$, and $\mathcal{L}_2(\pi_1^2) = \mathcal{L}_2(\pi_2^2) = \emptyset$, $\mathcal{L}_2(\pi_3^2) = \{\text{red}\}$. A timed run for each agent is given as follows:

$$\begin{aligned}
r_1^t = {} & (r_1(0) = \pi_1^1, \tau_1(0) = 0.0)(r_1(1) = \pi_2^1, \tau_1(1) = 1.0) \\
& (r_1(2) = \pi_3^1, \tau_1(2) = 2.5)(r_1(3) = \pi_2^1, \tau_1(3) = 3.0) \\
& (r_1(4) = \pi_1^1, \tau_1(4) = 5.0), \dots \\
r_2^t = {} & (r_2(0) = \pi_1^2, \tau_2(0) = 0.0)(r_2(1) = \pi_2^2, \tau_2(1) = 2.0) \\
& (r_2(2) = \pi_3^2, \tau_2(2) = 2.5)(r_2(3) = \pi_2^2, \tau_2(3) = 4.5) \\
& (r_2(4) = \pi_3^2, \tau_2(4) = 5.0) \dots
\end{aligned}$$

Given r_1^t and r_2^t the collective run r_G is given according to Definition 5.2 as follows:

$$\begin{aligned}
r_G^t = {} & (\underbrace{(\pi_1^1, \pi_1^2)}_{r_G(0)}, \tau_G(0) = 0.0)(\underbrace{(\pi_2^1, \pi_1^2)}_{r_G(1)}, \tau_G(1) = 1.0) \\
& (\underbrace{(\pi_2^1, \pi_2^2)}_{r_G(2)}, \tau_G(2) = 2.0)(\underbrace{(\pi_3^1, \pi_3^2)}_{r_G(3)}, \tau_G(3) = 2.5) \\
& (\underbrace{(\pi_2^1, \pi_3^2)}_{r_G(4)}, \tau_G(4) = 3.0)(\underbrace{(\pi_2^1, \pi_2^2)}_{r_G(5)}, \tau_G(5) = 4.5) \\
& (\underbrace{(\pi_1^1, \pi_3^2)}_{r_G(6)}, \tau_G(6) = 5.0) \dots
\end{aligned}$$

The produced collective timed word is:

$$w_G^t = (\{\text{green}\}, 0.0)(\emptyset, 1.0)(\emptyset, 2.0)(\{\text{red}\}, 2.5) \\ (\{\text{red}\}, 3.0)(\emptyset, 4.5)(\{\text{green}, \text{red}\}, 5.0)\ldots.$$

5.2.2 Specification

Local Agent's Specification

Each agent $i \in \mathcal{V}$ is given an individual, local, independent specification in the form of an MITL formula φ_i over the set of atomic propositions Σ_i. The satisfaction of φ_i is decided from the agent's own perspective, i.e., on the timed run r_i^t.

Global Team Specification

In addition, the team of agents is given a global team specification, which is a MITL formula φ_G over the set of atomic propositions Σ_G. The team specification satisfaction is decided on the collective timed run r_G^t.

Example 5.1 (Continued)**.** Recall the two agents from Example 5.1. Each of the agents is given a local, independent, specification and at the same time, the team is given an overall goal that may require collaboration or coordination. Examples of local specification formulas are:

$$\varphi_1 = \square\Diamond_{[0,10]}\{\text{green}\}, \\ \varphi_2 = \square\{\{\text{red}\} \Rightarrow \square_{[0,5]}\{\neg\text{red}\}\},$$

stating that "The green region is periodically visited with at most 10 time units between two consecutive visits" and "Whenever a red region is visited, it will not be visited for the following 5 time units again", respectively. While φ_1 is satisfied on r_1^t, φ_2 is not satisfied on r_2^t. An example of the global specification is:

$$\varphi_G = \square\Diamond_{[0,5]}\{\text{green} \wedge \text{red}\},$$

that imposes requirement on the agents' collaboration; it states that agents 1 and 2 will periodically and simultaneously visit the green and the red region, respectively, with at most 5 time units between two consecutive visits.

5.2.3 Problem Statement

Problem 5.1 (Run Synthesis)**.** Given N agents governed by dynamics as in (5.2), a task specification MITL formula φ_G for the team of agents, over a set of atomic propositions Σ_G and N local task specifications φ_i over Σ_i, $i \in \mathcal{V}$, synthesize a sequence of individual timed runs $r_1^t, \ldots, r_N^t$ such that the following hold

$$\left(r_G^t \models \varphi_G\right) \wedge \left(r_1^t \models \varphi_1 \wedge \cdots \wedge r_N^t \models \varphi_N\right). \tag{5.3}$$

Though it might seem that the satisfaction of the individual specifications $\varphi_1, \ldots, \varphi_N$ can be treated as the satisfaction of the formula $\bigwedge_{i \in \mathcal{V}} \varphi_i$ on the collective timed run r_G^t, this is generally not the case, as demonstrated through the following example.

Example 5.1 (Continued)**.** Recall the two agents from Example 5.1 and a local specification:

$$\varphi_2 = \square\{\{\text{red}\} \Rightarrow \square_{[0,2]}\{\neg\text{red}\}\}.$$

While this specification is satisfied on r_2^t since:

$$w_2^t = (\emptyset, 0.0)(\emptyset, 2.0)(\{\text{red}\}, 2.5)(\emptyset, 4.5)(\{\text{red}\}, 5.0)\ldots,$$

it can be observed that it is not satisfied on r_G^t.

Formally, we have:

$$r_G^t \models \bigwedge_{i \in \mathcal{V}} \varphi_i \nLeftrightarrow r_1^t \models \varphi_1 \wedge \cdots \wedge r_N^t \models \varphi_N.$$

Hence, Problem 5.1 may not be treated in a straightforward, fully centralized way. We propose a two-stage solution that first pre-computes all timed runs of the individual agents in a decentralized way and stores them efficiently in weighted transition systems enhanced with a Büchi acceptance condition. Second, these are combined and inspected with respect to guaranteeing the satisfaction of the team specification by the collective timed run.

5.3 Proposed Solution

In this section, we introduce a systematic solution to Problem 5.1. Our overall approach builds on the following steps:

1. We construct TBAs $\mathcal{A}_i, \quad i \in \mathcal{V}$ and $\mathcal{A}_G$ that accept all the timed words satisfying the specification formulas $\varphi_i, i \in \mathcal{V}$ and φ_G, respectively (Section 5.3.1).

2. We construct a *local Büchi WTS* $\widetilde{\mathcal{T}}_i = \mathcal{T}_i \circledast \mathcal{A}_i$, for all $i \in \mathcal{V}$. The accepting timed runs of $\widetilde{\mathcal{T}}_i$ are the timed runs of the $\mathcal{T}_i$ that satisfy the corresponding local specification formula $\varphi_i, i \in \mathcal{V}$ (Section 5.3.2).

3. We construct a *product Büchi WTS* $\mathcal{T}_G = \widetilde{\mathcal{T}}_1 \circledast \cdots \circledast \widetilde{\mathcal{T}}_N$ such that its timed runs are collective timed runs of the team and their projections onto the agents' individual timed runs are admissible by the local Büchi WTSs $\widetilde{\mathcal{T}}_1, \ldots, \widetilde{\mathcal{T}}_N$ respectively (Section 5.3.3).

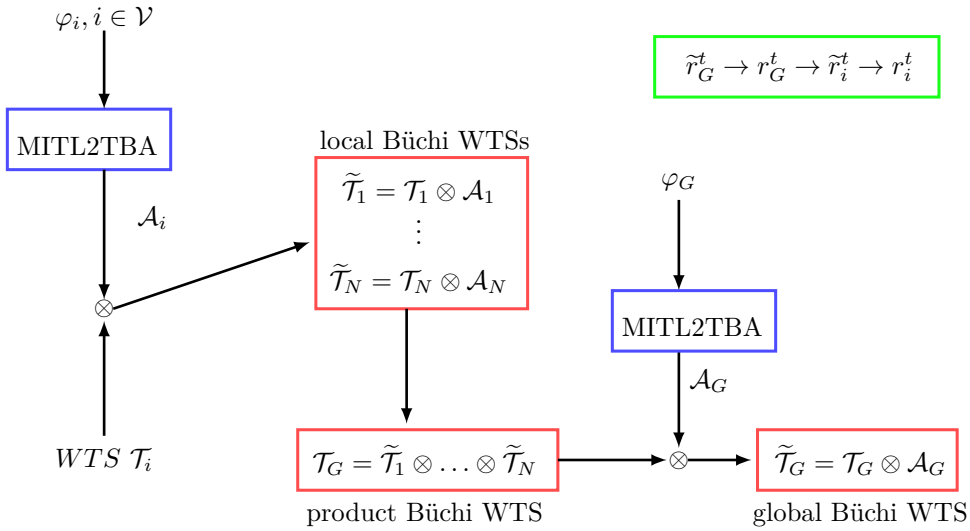

Figure 5.2: A graphic illustration of the proposed framework.

4. We construct a *global Büchi WTS* $\widetilde{\mathcal{T}}_G = \mathcal{T}_G \circledast \mathcal{A}_G$. The accepting timed runs of the $\widetilde{\mathcal{T}}_G$ are the timed runs of the $\mathcal{T}_G$ that satisfy the team formula φ_G (Section 5.3.4).

5. We find an accepting timed run $\widetilde{r}_G^t$ of the global Büchi WTS $\widetilde{\mathcal{T}_G}$ and project it onto timed runs of the product Büchi WTS $\mathcal{T}_G$, then onto timed runs of the local Büchi WTSs $\widetilde{\mathcal{T}}_1, \ldots, \widetilde{\mathcal{T}}_N$, and finally onto individual timed runs $r_1^t, \ldots, r_N^t$ of the original WTSs $\mathcal{T}_1, \ldots, \mathcal{T}_N$. By construction, $r_1^t, \ldots, r_N^t$ are guaranteed to satisfy $\varphi_1, \ldots, \varphi_N$, respectively, and furthermore r_G^t satisfies φ_G (Section 5.3.5).

The aforementioned procedure is depicted in Fig. 5.2.

5.3.1 Construction of TBAs

As stated in Section 2.7, every MITL formula φ can be translated into a language equivalent TBA. Several approaches are proposed for the translation e.g., [136–138, 163]. Here, we translate each local specification φ_i, where $i \in \mathcal{V}$ into a TBA $\mathcal{A}_i = (S_i, S_i^{\text{init}}, \text{CL}_i, \text{Inv}_i, E_i, \text{FS}_i, \Sigma_i)$, and the global specification φ_G into a TBA $\mathcal{A}_G = (S_G, S_G^{\text{init}}, \text{CL}_G, \text{Inv}_G, E_G, \text{FS}_G, \Sigma_G)$.

5.3.2 Construction of the local Büchi WTSs $\widetilde{\mathcal{T}}_1, \ldots, \widetilde{\mathcal{T}}_N$

Definition 5.4. Given a WTS $\mathcal{T}_i = (\Pi_i, \Pi_i^{\text{init}}, \longrightarrow_i, \mathfrak{t}_i, \Sigma_i, \mathcal{L}_i)$, and a TBA $\mathcal{A}_i = (S_i, S_i^{\text{init}}, \text{CL}_i, \text{Inv}_i, E_i, \text{FS}_i, \Sigma_i)$ with CL_i clocks and $\text{CL}_i^{\max}$ being the largest constant appearing in $\mathcal{A}_i$. Then, their *local Büchi WTS*:

$$\widetilde{\mathcal{T}}_i = \mathcal{T}_i \circledast \mathcal{A}_i = (Q_i, Q_i^{\text{init}}, \rightsquigarrow_i, \widetilde{\mathfrak{t}}_i, \widetilde{\text{FS}}_i, \Sigma_i, \widetilde{\mathcal{L}}_i)$$

is defined as follows:

- $Q_i \subseteq \Pi_i \times S_i \times \mathbb{Q}_{\geq 0}^{\text{CL}_i}$ is the set of states;
- $Q_i^{\text{init}} = \Pi_i^{\text{init}} \times S_i^{\text{init}} \times \{0\}^{\text{CL}_i}$ is the set of initial states;
- $(q, q') \in \rightsquigarrow_i$ iff:
  - $q = (\pi, s, \nu_1, \ldots, \nu_{\text{CL}_i}) \in Q_i$,
    $q' = (\pi', s', \nu_1', \ldots, \nu_{\text{CL}_i}') \in Q_i$,
  - $(\pi, \pi') \in \longrightarrow_i$, and
  - there exist $\mathfrak{g}$, σ, RS, such that $(s, \mathfrak{g}, \sigma, R, s') \in E_i$, $\nu_1, \ldots, \nu_{\text{CL}_i} \models \mathfrak{g}$, $\nu_1', \ldots, \nu_{\text{CL}_i}' \models \text{Inv}_i(s')$, and for all $i \in \{1, \ldots, \text{CL}_i\}$

$$\nu_i' = \begin{cases} 0, & \text{if } c_i \in \text{RS}, \\ \nu_i + \mathfrak{t}_i(\pi, \pi'), & \text{if } c_i \notin \text{RS, and} \\ & \nu_i + \mathfrak{t}_i(\pi, \pi') \leq \text{CL}_i^{\max} \\ \infty, & \text{otherwise.} \end{cases}$$

  Then, $\widetilde{\mathfrak{t}}_i(q, q') = \mathfrak{t}_i(\pi, \pi')$;
- $\widetilde{\text{FS}}_i = \{(\pi_i, s_i, \nu_1, \ldots, \nu_{\text{CL}_i}) \in Q_i : s_i \in \text{FS}_i\}$ is a set of accepting states; and
- $\widetilde{\mathcal{L}}_i(\pi_i, s_i, \nu_1, \ldots, \nu_{\text{CL}_i}) = \mathcal{L}_i(\pi_i)$ is a labeling function.

Each local Büchi WTS $\widetilde{\mathcal{T}}_i$, $i \in \mathcal{V}$, is in fact a WTS with a Büchi acceptance condition $\widetilde{F}_i$. A timed run of $\widetilde{\mathcal{T}}_i$ can be written as $\widetilde{r}_i^t = (q_i(0), \tau_i(0))(q_i(1), \tau_i(1)) \ldots$ using the terminology of Definition 2.12. It is *accepting* if $q_i(j) \in \widetilde{\text{FS}}_i$ for infinitely many $j \geq 0$. An accepting timed run of $\widetilde{\mathcal{T}}_i$ projects onto a timed run of $\mathcal{T}_i$ that satisfies the local specification formula φ_i by construction.

An alternative definition for the local Büchi WTS $\widetilde{T}_i$ has been proposed in [20, 164], and is given as follows.

Definition 5.5. Given a WTS $\mathcal{T}_i = (\Pi_i, \Pi_i^{\text{init}}, \longrightarrow_i, \mathfrak{t}_i, \Sigma_i, \mathcal{L}_i)$, and a TBA $\mathcal{A}_i = (S_i, S_i^{\text{init}}, \text{CL}_i, \text{Inv}_i, E_i, \text{FS}_i, \Sigma_i)$ with CL_i clocks. Then, their *local Büchi WTS*:

$$\widetilde{\mathcal{T}}_i = \mathcal{T}_i \circledast \mathcal{A}_i = (Q_i, Q_i^{\text{init}}, \rightsquigarrow_i, \text{CL}_i, \text{Inv}_i, \widetilde{\mathfrak{t}}_i, \widetilde{\text{FS}}_i, \Sigma_i, \widetilde{\mathcal{L}}_i),$$

is defined as follows:

- $Q_i = \Pi_i \times S_i$ is the set of states;
- $Q_i^{\text{init}} = \Pi_i^{\text{init}} \times S_i^{\text{init}}$ is the set of initial states;
- $\rightsquigarrow_i$ is the set of transitions where $(q_i, q_i') \in \rightsquigarrow_i$ iff:
  - $q_i = (\pi_i, s_i) \in Q_i$ and $q' = (\pi_i', s_i') \in Q_i$,
  - $(\pi_i, \pi_i') \in \longrightarrow_i$, and
  - there exist $\mathfrak{g}_i, \sigma_i, \text{RS}_i$ such that $(s_i, \mathfrak{g}_i, \text{RS}_i, \sigma_i, s_i') \in E_i$ where $\sigma_i = \mathcal{L}_i(\pi_i')$;
- $\text{Inv}_i(\pi_i, s_i) = \text{Inv}_i(s_i)$ a map labeling each state with clock constraints;
- $\widetilde{\mathfrak{t}}_i(q_i, q_i') = \mathfrak{t}_i(\pi_i, \pi_i')$ if $(q_i, q_i') \in \rightsquigarrow_i$, is a map that assigns a positive weight to each transition;
- $\widetilde{F}_i = \{(\pi_i, s_i) \in Q_i : s_i \in \text{FS}_i\}$ is a set of accepting states; and
- $\widetilde{\mathcal{L}}_i(\pi_i, s_i) = \mathcal{L}_i(\pi_i)$ is a labeling function.

5.3.3 Construction of the product Büchi WTS $\mathcal{T}_G$

In this section, we aim to construct a finite product WTS $\mathcal{T}_G$ whose timed runs represent the collective behaviors of the team and whose Büchi acceptance condition ensures that the accepting timed runs account for the local specifications. In other words, $\mathcal{T}_G$ is a product of all the local WTS $\widetilde{\mathcal{T}}_i$ built above. In the construction of $\mathcal{T}_G$, we need to specifically handle the cases when transitions of different agents are associated with different time duration, i.e, different transition weights. To this end, we introduce a vector $b = (b_1, \ldots, b_N) \in \mathbb{Q}_{\geq 0}^N$. Each element of the vector is a rational number $b_i \in \mathbb{Q}_{\geq 0}, i \in \mathcal{V}$ which can be either 0, when the agent i has just completed its transition, or the time elapsed from the beginning of the agent's current transition, if this transition is not completed yet. Then, the state of the team of agents is written in the form $q_G = (q_1, \ldots, q_N, b_1, \ldots, b_N, \ell)$ where q_i is a state of $\widetilde{\mathcal{T}}_i$, for all $i \in \mathcal{V}$, and $\ell \in \mathcal{V}$ has a special meaning in relation to the acceptance condition of $\mathcal{T}_G$ that will become clear shortly. Taking the above into consideration we define the global model $\mathcal{T}_G$ as follows.

Definition 5.6. Given N local Büchi WTSs $\widetilde{\mathcal{T}}_1, \ldots, \widetilde{\mathcal{T}}_N$ from Definition 5.4, their *product Büchi WTS*:

$$\begin{aligned}\mathcal{T}_G &= \widetilde{\mathcal{T}}_1 \circledast \ldots \circledast \widetilde{\mathcal{T}}_N \\ &= (Q_G, Q_G^{\text{init}}, \longrightarrow_G, \mathfrak{t}_G, \text{FS}_G, \Sigma_G, \mathcal{L}_G),\end{aligned}$$

is defined as follows:

- $Q_G \subseteq Q_1 \times \cdots \times Q_N \times \mathbb{Q}_{\geq 0}^N \times \{1, \ldots, N\}$ is the set of states;
- $Q_G^{\text{init}} = Q_1^{\text{init}} \times \ldots \times Q_N^{\text{init}} \times \{0\}^N \times \{1\}$ is the set of initial states;
- $(q_G, q'_G) \in \longrightarrow_G$ iff:
  - $q_G = (q_1, \ldots, q_N, b_1, \ldots, b_N, \ell) \in Q_G$,
    $q'_G = (q'_1, \ldots, q'_N, b'_1, \ldots, b'_N, \ell') \in Q_G$,
  - there exists $q_i'' \in Q_i$ such that $(q_i, q_i") \in \rightsquigarrow_i$, for some $i \in \mathcal{V}$,
  - $$b'_i = \begin{cases} 0, & \text{if } b_i + \mathfrak{t}_{\min} = \widetilde{\mathfrak{t}}_i(q_i, q_i'') \\ & \text{and } q'_i = q_i'', \\ b_i + \mathfrak{t}_{\min}, & \text{if } b_i + \mathfrak{t}_{\min} < \widetilde{\mathfrak{t}}_i(q_i, q_i'') \\ & \text{and } q'_i = q_i, \end{cases}$$
    where $\mathfrak{t}_{\min} = \min\limits_{k \in \{1,\ldots,N\}} \{\widetilde{\mathfrak{t}}_i(q_i, q_i'') - b_i\}$ is (loosely speaking) the smallest time step that can be applied, and
  - $$\ell' = \begin{cases} \ell, & \text{if } q_\ell \notin \widetilde{\text{FS}}_\ell \\ ((\ell + 1) \mod N), & \text{otherwise} \end{cases}$$

  Then, $\mathfrak{t}_G(q_G, q'_G) = \mathfrak{t}_{\min}$;
- $\text{FS}_G = \{(q_1, \ldots, q_N, b_1, \ldots, b_N, N) \in Q_G : q_N \in \widetilde{\text{FS}}_N\}$ is a set of accepting states;
- $\Sigma_G = \bigcup\limits_{i \in \mathcal{V}} \Sigma_i$ is a set of atomic propositions; and
- $\mathcal{L}_G((q_1, \ldots, q_N, b_1, \ldots, b_N, \ell)) = \bigcup\limits_{i \in \mathcal{V}} \widetilde{\mathcal{L}}_i(q_i)$ is a labeling function.

The index ℓ in a state $q_G = (q_1, \ldots, q_N, b_1, \ldots, b_N, \ell) \in Q_G$ allows to project an accepting timed run of $\mathcal{T}_G$ onto an accepting run of every one of the local Büchi WTS. An accepting run r_G^t of the product Büchi WTS $\mathcal{T}_G$ projects onto an accepting run r_i^t of the local Büchi WTS $\widetilde{\mathcal{T}}_i$ that produces a timed word w_i^t accepted by the corresponding TBA $\mathcal{A}_i$.

An alternative definition for the product Büchi WTS $\mathcal{T}_G$ has been proposed in [20, 164], and is given as follows.

Definition 5.7. Given N local Büchi WTSs $\widetilde{\mathcal{T}}_1, \ldots, \widetilde{\mathcal{T}}_N$ from Definition 5.5, their *product Büchi WTS*

$$\begin{aligned}\mathcal{T}_G &= \widetilde{\mathcal{T}}_1 \circledast \ldots \circledast \widetilde{\mathcal{T}}_N \\ &= (Q_G, Q_G^{\text{init}}, \longrightarrow_G, \text{CL}_G, \text{Inv}_G, \mathfrak{t}_G, \text{FS}_G, \Sigma_G, \mathcal{L}_G),\end{aligned}$$

is defined as follows:

- $Q_G = Q_1 \times \cdots \times Q_N$ is the set of states;
- $Q_G^{\text{init}} = Q_1^{\text{init}} \times \ldots \times Q_N^{\text{init}}$ is the set of initial states;
- $\longrightarrow_G$ is set of transitions where $(q_G, q'_G) \in \longrightarrow_G$ iff:
  - $q_G = (q_1, \ldots, q_N) \in Q_G, \quad q'_G = (q'_1, \ldots, q'_N) \in Q_G$; and
  - $(q_i, q'_i) \in \rightsquigarrow_i$, for every $i \in \mathcal{V}$;
- $\text{CL}_G = \{\text{CL}_1, \ldots, \text{CL}_N\}$ is a set of clocks;
- $\text{Inv}_G(q_1, \ldots, q_N) = \bigcup_{i \in \mathcal{V}} \text{Inv}(q_i)$ is a map labeling each state with clock constraints;
- $\mathfrak{t}_G = \min_{i \in \mathcal{V}} \{\mathfrak{t}_i(q_i, q'_i)\}$, if $(q_G, q'_G) \in \longrightarrow_G$ where $q_G = (q_1, \ldots, q_N)$ and $q'_G = (q'_1, \ldots, q'_N)$ is a map that assigns a positive weight to each transition;
- $\text{FS}_G = \{q_G = (q_1, \ldots, q_N) \in Q_G : q_i \in \widetilde{\text{FS}}_i, \forall i \in \mathcal{V}\}$ is a set of accepting states;
- $\Sigma_G = \bigcup_{i \in \mathcal{V}} \Sigma_i$ is a set of atomic propositions; and
- $\mathcal{L}_G((q_1, \ldots, q_N)) = \bigcup_{i \in \mathcal{V}} \widetilde{\mathcal{L}}_i(q_i)$ is a labeling function.

5.3.4 Construction of the global Büchi WTS $\widetilde{\mathcal{T}}_G$

Definition 5.8. Given the product Büchi WTS $\mathcal{T}_G = (Q_G, Q_G^{\text{init}}, \longrightarrow_G, \mathfrak{t}_G, \text{FS}_G, \Sigma_G, \mathcal{L}_G)$, and a TBA $\mathcal{A}_G = (S_G, S_G^{\text{init}}, \text{CL}_G, \text{Inv}_G, E_G, \overline{\text{FS}}_G, \Sigma_G)$ that corresponds to the team specification formula φ_G with CL_G clocks and $\text{CL}_G^{\max}$ being the largest constant appearing in $\mathcal{A}_G$, we define their product WTS:

$$\begin{aligned}\widetilde{\mathcal{T}}_G &= \mathcal{T}_G \circledast \mathcal{A}_G \\ &= (\widetilde{Q}_G, \widetilde{Q}_G^{\text{init}}, \rightsquigarrow_G, \widetilde{\mathfrak{t}}_G, \widetilde{\text{FS}}_G, \Sigma_G, \widetilde{L}_G),\end{aligned}$$

as follows:

- $\widetilde{Q}_G \subseteq Q_G \times S_G \times \mathbb{Q}_{\geq 0}^{\text{CL}_G}$ is a set of states;
- $\widetilde{Q}_G^{\text{init}} = Q_G^{\text{init}} \times S_G^{\text{init}} \times \{0\}^{\text{CL}_G}$ is a set of initial states;
- $(q, q') \in \rightsquigarrow_G$ iff:
  - $q = (r, s, \nu_1, \ldots, \nu_{\text{CL}_G}) \in Q_G$,
    $q' = (r', s', \nu_1', \ldots, \nu_{\text{CL}_G}') \in Q_G$,
  - $(r, r') \in \longrightarrow_G$, and
  - there exist $\mathfrak{g}$, σ, RS, such that $(s, \mathfrak{g}, \sigma, \text{RS}, s') \in E_G$, $\nu_1, \ldots, \nu_{\mathcal{C}_G} \models \mathfrak{g}$, $\nu_1', \ldots, \nu_{\text{CL}_G}' \models \text{Inv}_G(s')$, and for all $i \in \{1, \ldots, \text{CL}_G\}$

$$\nu_i' = \begin{cases} 0, & \text{if } c_i \in \text{RS} \\ \nu_i + \mathfrak{t}_G(r, r'), & \text{if } c_i \notin \text{RS and} \\ & \nu_i + \mathfrak{t}_G(r, r') \leq \text{CL}_G^{\max} \\ \infty, & \text{otherwise} \end{cases}$$

  Then $\widetilde{\mathfrak{t}}_G(q, q') = \mathfrak{t}_G(r, r')$.
- $\widetilde{\text{FS}}_G = \{(r, s, \nu_1, \ldots, \nu_{\text{CL}_G}) \in Q_G : r \in \text{FS}_G \text{ and } s \in \overline{\text{FS}}_G\}$ is a set of accepting states; and
- $\widetilde{\mathcal{L}}_G((r_G, s_G, \nu_1, \ldots, \nu_{\text{CL}_G})) = \mathcal{L}_G(r_G)$ is a labeling function.

An accepting timed run of $\widetilde{T}_G$ guarantees the satisfaction of the team specification formula φ_G by construction. Furthermore, the projected individual timed runs of the original $\mathcal{T}_1, \ldots, \mathcal{T}_N$ satisfy their respective local specifications. The corresponding product Büchi WTS when $\mathcal{T}_G$ is computed by using Definition 5.7 is given as follows.

Definition 5.9. Given the product Büchi WTS:

$$\mathcal{T}_G = (Q_G, Q_G^{\text{init}}, \longrightarrow_G, \text{CL}_G, \text{Inv}_G, \mathfrak{t}_G, \text{FS}_G, \Sigma_G, \mathcal{L}_G),$$

and a TBA:

$$\mathcal{A}_G = (S_G, S_G^{\text{init}}, \overline{\text{CL}}_G, \overline{\text{Inv}}_G, E_G, \overline{\text{FS}}_G, \Sigma_G),$$

that corresponds to the team specification formula φ_G with CL_G clocks, we define their product WTS:

$$\widetilde{\mathcal{T}}_G = \mathcal{T}_G \circledast \mathcal{A}_G = (\widetilde{Q}_G, \widetilde{Q}_G^{\text{init}}, \rightsquigarrow_G, \widetilde{\text{CL}}_G, \widetilde{\text{Inv}}_G, \widetilde{\mathfrak{t}}_G, \widetilde{\text{FS}}_G, \Sigma_G, \widetilde{\mathcal{L}}_G),$$

as follows:

- $\widetilde{Q}_G = Q_G \times S_G$ is the set of states;
- $Q_i^{\text{init}} = Q_G^{\text{init}} \times S_G^{\text{init}}$ is the set of initial states;
- $\rightsquigarrow_G$ is the set of transitions where $(\widetilde{q}_G, \widetilde{q}'_G) \in \rightsquigarrow_G$ iff:
  - $\widetilde{q}_G = (q_G, s_G) \in \widetilde{Q}_G$ and $\widetilde{q}'_G = (q'_G, s'_G) \in \widetilde{Q}_G$,
  - $(q_G, q'_G) \in \longrightarrow_G$, and
  - $\exists \mathfrak{g}, \sigma, \text{RS}$ such that $(s, \mathfrak{g}, \text{RS}, \sigma, s') \in E_G$ where $\sigma = \mathcal{L}_G(q'_G)$;
- $\text{CL}_{\widetilde{G}} = \{\text{CL}_G, \overline{\text{CL}}_G\}$ is a set of clocks;
- $\text{Inv}_{\widetilde{G}}(q_G, s_G) = \text{Inv}_G(q_G) \cup \overline{\text{Inv}}_G(s_G)$ a map labeling each state with clock constraints;
- $\widetilde{\mathfrak{t}}_G((q_G, s_G), (q'_G, s'_G)) = \mathfrak{t}(q_G, q'_G)$ if $((q_G, s_G), (q'_G, s_G)) \in \rightsquigarrow_G$, is a map that assigns a positive weight to each transition;
- $\widetilde{F}_G = \{(q_G, s_G) \in \widetilde{Q}_G : q_G \in \text{FS}_G \text{ and } s_G \in \overline{\text{FS}}_G\}$ is a set of accepting states; and
- $\widetilde{\mathcal{L}}_G(q_G, s_G) = \mathcal{L}_G(q_G)$ is a labeling function.

5.3.5 Projection to the desired timed runs of $\mathcal{T}_1, \ldots, \mathcal{T}_N$

An accepting run $\widetilde{r}_G^t$ of the global Büchi WTS $\widetilde{\mathcal{T}}_G$ can be found efficiently leveraging graph search techniques, such as modified Dijkstra algorithm with cost function the map $\widetilde{t}_G$ (see [20, 164] for more details). Once $\widetilde{r}_G^t$ is obtained, we have provided guidelines for projection of $\widetilde{r}_G^t$ onto the individual timed runs of $\mathcal{T}_1, \ldots, \mathcal{T}_N$. In particular, $\widetilde{r}_G^t$ is projected onto a timed run r_G^t of $\mathcal{T}_G$, which is projected onto timed runs $\widetilde{r}_1^t, \ldots, \widetilde{r}_N^t$ of $\widetilde{\mathcal{T}}_1, \ldots, \mathcal{T}_N$, which are finally projected onto timed runs $r_1^t, \ldots, r_N^t$ of $\mathcal{T}_1, \ldots, \mathcal{T}_N$, respectively. Such a projection guarantees that $r_1^t, \ldots, r_N^t$ are a solution to Problem 5.1.

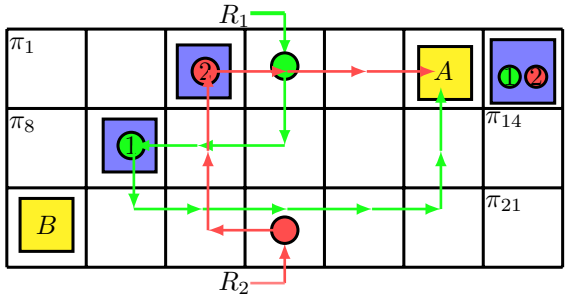

Figure 5.3: An illustrative example with 2 robots evolving in a common workspace. Let $\mathcal{W} = \pi_1 \cup \ldots \cup \pi_{21}$. The regions are enumerated starting from the left region in every row and ending in the right. The initial positions of robots R_1, R_2 are depicted by a green and a red circle, respectively, the desired meeting points in yellow and the recharging spots by the agents' respective colors inside a blue box. The accepting runs for task specifications ϕ_1, ϕ_2, ϕ_G are depicted with green and red arrows for agent 1 and agent 2, respectively.

5.4 An Illustrative Example

For an illustrative example, consider 2 robots in the shared workspace of Figure 5.3. The workspace is partitioned into $Z = 21$ cells and a robot's state is defined by the cell that the robot is currently present at. Agent 1 (R1) is depicted in green and it is two times faster than Agent 2 (R2) which is depicted in red. We assume that the environment imposes such moving constraints that the traveling right and up is faster than left and down. Let Agent 1 need 1 time unit for up and right moves and 2 time units for down and left moves. Let also Agent 2 need 2 time units for up and right moves and 4 time units for down and left moves.

We consider a scenario where the robots have to eventually meet at yellow regions (global team task), and at the same time, they have to recharge within a certain time interval in recharge locations (blue squares with the circles in the respective color). The individual specifications are:

$$\varphi_1 = \Diamond_{[0,6]}\{\text{recharge1}\},$$
$$\varphi_2 = \Diamond_{[0,12]}\{\text{recharge2}\},$$

stating that agent 1 has to recharge within 6 time units and agent 2 within 12 units, respectively, and the team task is:

$$\varphi_G = \Diamond_{[0,30]}\{\{\text{meet}_1^A \wedge \text{meet}_2^A\} \vee \{\text{meet}_1^B \wedge \text{meet}_2^B\}\},$$

stating that the agents have to meet either in the yellow region A or B within 30 time units.

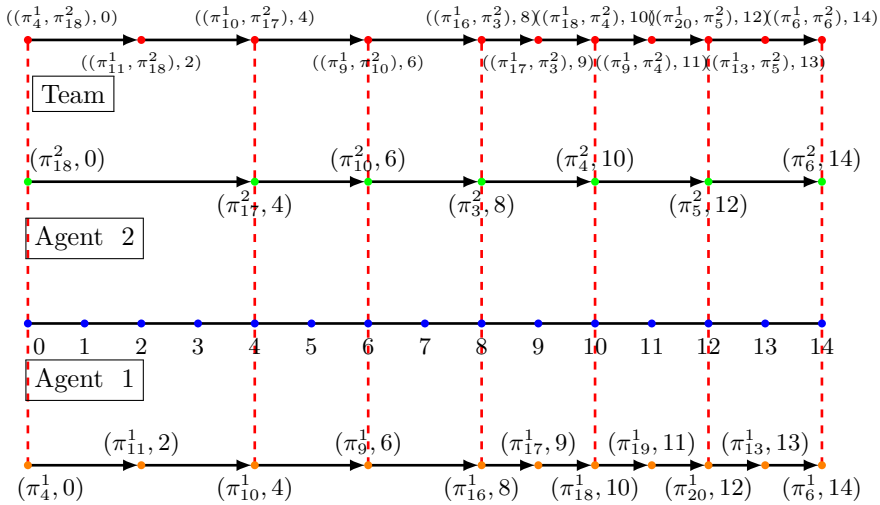

Figure 5.4: The accepting runs $\widetilde{r}^t_1, \widetilde{r}^t_2$, the collective run $\widetilde{r}^t_G$ and the corresponding timed stamps. The red dashed lines denote the times that both agents have the same time stamps.

An accepting collective timed run is:

$$\widetilde{r}^t_G = ((\pi^1_4, \pi^2_{18}), 0)((\pi^1_{11}, \pi^2_{18}), 2) \dots ((\pi^1_9, \pi^2_{10}), 6)((\pi^1_{16}, \pi^2_3), 8) \dots \\ ((\pi^1_{13}, \pi^2_5), 13)((\pi^1_6, \pi^2_6), 14) \dots,$$

with corresponding timed word:

$$\widetilde{w}^t_G = (\emptyset, 0)(\emptyset, \pi^2_{18}), 2) \dots (\{\text{recharge1}\}, 6)(\{\text{recharge2}\}, 8) \dots \\ (\emptyset, \pi^2_5), 13)((\{\text{meet}^A_1, \text{meet}^A_2\}, 14) \dots,$$

which satisfies the formula ϕ_G.
The run $\widetilde{r}^t_G$ can be projected onto the individual timed runs

$$\widetilde{r_1}^t = (\pi^1_4, 0)(\pi^1_{11}, 2)(\pi^1_{10}, 4)(\pi^1_9, 6)(\pi^1_{16}, 8)(\pi^1_{17}, 9)(\pi^1_{18}, 10) \\ (\pi^1_{19}, 11)(\pi^1_{20}, 12)(\pi^1_{13}, 13)(\pi^1_6, 14) \dots,$$
$$\widetilde{r_2}^t = (\pi^2_{18}, 0)(\pi^2_{17}, 4)(\pi^2_{10}, 6)(\pi^2_3, 8)(\pi^2_4, 10)(\pi^2_5, 12)(\pi^2_6, 14) \dots$$

(they are depicted in Figure 5.3 with green and red arrows, respectively) with corresponding timed words

$$\widetilde{w}^t_1 = (\emptyset, 0)(\emptyset, 2)(\emptyset, 4)(\{\text{recharge1}\}, 6)(\emptyset, 8) \\ (\emptyset, 9)(\emptyset, 10)(\emptyset, 11)(\emptyset, 12)(\emptyset, 13)(\{\text{meet}^A_1\}, 14) \dots,$$
$$\widetilde{w}^t_2 = (\emptyset, 0)(\emptyset, 4)(\emptyset, 6)(\{\text{recharge2}\}, 8)(\emptyset, 10)(\emptyset, 12)(\{\text{meet}^A_2\}, 14) \dots$$

which satisfy formulas ϕ_1 and ϕ_2, respectively. All conditions from (5.3) are satisfied. The runs and the words of the illustrative example are depicted in Figure 5.4.

5.5 Conclusions

We have proposed a systematic method for multi-agent controller synthesis aiming cooperative planning under high-level specifications given in MITL formulas. The solution involves a sequence of algorithmic automata constructions such that not only team specifications but also individual specifications should be fulfilled.

Chapter 6

Decentralized Abstractions for Time-constrained Planning

This chapter presents a fully automated procedure for controller synthesis for multi-agent systems under coupling constraints. Each agent is modeled with dynamics consisting of two terms: the first one models the coupling constraints and the other one is an additional bounded control input. We aim to design these inputs so that each agent meets an individual high-level specification given as an MITL. First, a decentralized abstraction that provides a space and time discretization of the multi-agent system is designed. Second, by utilizing this abstraction and techniques from formal verification, we propose an algorithm that computes the individual runs which provably satisfy the high-level tasks. The overall approach is demonstrated in a simulation example conducted in MATLAB environment.

6.1 Introduction

In Chapter 5, an automata-based solution for multi-agent systems was proposed, where MITL formulas were introduced in order to synthesize controllers such that every agent fulfills an individual specification and the team of agents fulfills a global task. Specifically, the abstraction of each agent's dynamics was considered to be given and an upper bound of the time that each agent needs to perform a transition from one region to another was assumed. Furthermore, potential coupled constraints between the agents were not taken into consideration. Motivated by this, in this chapter, we aim to address the aforementioned issues. We assume that the dynamics of each agent consists of two parts: the first part is a consensus type term representing the coupling between the agent and its neighbors, and the second one is an additional control input which will be exploited for high-level planning. Hereafter, we call it a free input. A decentralized abstraction procedure is provided, which leads to an individual WTS for each agent and provides a basis for high-level planning. Additionally, this abstraction is associated to a time quantization which

allows us to assign precise time duration to the transitions of each agent.

Abstractions for both single and multi-agent systems can be found in [110–112, 114–116]. Compositional frameworks are provided in [114] for safety specifications of discrete time systems, and [115], which is focused on feedback linearizable systems with a cascade interconnection. In addition, local invariant sets for discrete time coupled linear systems are considered in [117] and are leveraged for control synthesis. The above results are therefore not applicable to the decentralized abstraction of the multi-agent control systems we consider, which evolve in continuous time and do not require a specific network interconnection.

Motivated by the work [116], we start from the dynamics of each agent and we construct a WTS for each agent in a decentralized manner. Each agent is assigned an individual task given in MITL formulas. We aim to design the free inputs so that each agent performs the desired individual task within specific time bounds. In particular, we provide an automatic controller synthesis method for coupled multi-agent systems under high-level tasks with timed constraints. A motivation for this framework comes from applications such as the deployment of aerial robotic teams. In particular, the consensus coupling allows the robots to stay sufficiently close to each other and maintain a connected network during the evolution of the system. Additionally, individual MITL formulas are leveraged to assign area monitoring tasks to each robot individually. The MITL formalism enables us to impose time constraints on the monitoring process.

This chapter is organized as follows. In Section 6.2 a description of the necessary mathematical tools, the notations and the definitions are given. Section 6.3 provides the dynamics of the system and the formal problem statement. Section 6.5 discusses the technical details of the solution. Section 6.6 is devoted to a simulation example. Finally, conclusions and future work are discussed in Section 6.7.

6.2 Notation and Preliminaries

For a subset S of $\mathbb{R}^n$, denote by $\mathrm{cl}(S), \mathrm{int}(S)$ and $\partial S = \mathrm{cl}(S) \backslash \mathrm{int}(S)$ its closure, interior and boundary, respectively.

Definition 6.1. A *cell decomposition* $S = \{S_\ell\}_{\ell \in \mathbb{I}}$ of a set $\mathcal{D} \subseteq \mathbb{R}^n$, where $\mathbb{I} \subseteq \mathbb{N}$ is a finite or countable index set, is a family of uniformly bounded convex sets $S_\ell, \ell \in \mathbb{I}$ such that $\mathrm{int}(S_\ell) \cap \mathrm{int}(S_{\hat{\ell}}) = \emptyset$ for all $\ell, \hat{\ell} \in \mathbb{I}$ with $\ell \neq \hat{\ell}$ and $\bigcup_{\ell \in \mathbb{I}} S_\ell = \mathcal{D}$. The interiors of the cells are non-empty.

Given a vector $x_i = (x_i^1, \ldots, x_i^n) \in \mathbb{R}^n$, the component operator $c(x_i, \ell) = x_i^\ell \in \mathbb{R}, \ell = 1, \ldots, n$ gives the projection of x_i onto its ℓ-th component (see [128, Chapter 7]). Similarly, for the stack vector $x = (x_1, \ldots, x_N) \in \mathbb{R}^{Nn}$ the component operator is defined as $c(x, \ell) = (c(x_1, \ell), \ldots, c(x_N, \ell)) \in \mathbb{R}^N, \ell = 1, \ldots, n$. By using the component operator, the norm of a vector $x \in \mathbb{R}^{Nn}$ can be evaluated as:

$$\|x\|_2 = \left\{ \sum_{\ell=1}^{n} \|c(x, \ell)\|_2^2 \right\}^{\frac{1}{2}}.$$

Denote by $\widetilde{x} \in \mathbb{R}^{|\mathcal{E}|n}$ the stack column vector of the vectors $x_i - x_j, \{i, j\} \in \mathcal{E}$ with the edges ordered as in the case of the incidence matrix $\mathscr{D}(\mathcal{G})$. Then, the following holds:

$$\widetilde{x} = \left[\mathscr{D}(\mathcal{G})^\top \otimes I_n \right] x. \tag{6.1}$$

6.3 Problem Formulation

6.3.1 System Model

We focus on multi-agent systems with coupled dynamics of the form:

$$\dot{x}_i = -\sum_{j \in \mathcal{N}_i} (x_i - x_j) + u_i, \quad x_i \in \mathbb{R}^n, \quad i \in \mathcal{V}. \tag{6.2}$$

The dynamics (6.2) consists of two parts; the first part is a consensus protocol representing the coupling between the agent and its neighbors, and the second one is a control input which will be exploited for high-level planning and is called free input. In this work, it is assumed that the free inputs are bounded by a positive constant $u_{\max}$, i.e.,

$$\|u_i(t)\|_2 \leq u_{\max}, \ \forall \ i \in \mathcal{V}, \quad t \geq 0.$$

The topology of the multi-agent network is modeled through a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, and the following assumption is made.

Assumption 6.1. The communication graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ of the system is undirected, connected and static i.e., every agent preserves the same neighbors for all times.

6.4 Specification

Our goal is to control the multi-agent system (6.2) so that each agent's behavior obeys a desired individual specification φ_i given in MITL. In particular, it is required to drive each agent to a desired subset of the workspace $\mathbb{R}^n$ within certain time limits and provide certain atomic tasks there. Atomic tasks are captured through a

finite set of *services* Σ_i, $i \in \mathcal{V}$. The position x_i of each agent $i \in \mathcal{V}$ is labeled with services that are offered there. Thus, a *service labeling function*

$$\Lambda_i : \mathbb{R}^n \to 2^{\Sigma_i}, \tag{6.3}$$

is introduced for each agent $i \in \mathcal{V}$ which maps each state $x_i \in \mathbb{R}^n$ to the subset of services $\Lambda_i(x_i)$ which hold true at x_i i.e., the subset of services that the agent i *can* provide in position x_i. It is noted that although the term service labeling function it is used, these functions are not necessarily related to the labeling functions of a WTS as in Definition 2.11. Define also by $\Lambda(x) = \bigcup_{i \in \mathcal{V}} \Lambda_i(x)$ the union of all the service labeling functions. We also assume that $\Sigma_i \cap \Sigma_j = \emptyset$, for all $i, j \in \mathcal{V}, i \neq j$ which means that the agents do not share any services. Let us now introduce the following assumption which is necessary for formally defining the problem.

Assumption 6.2. There exists a decomposition $S = \{S_\ell\}_{\ell \in \mathbb{I}}$ of the workspace $\mathbb{R}^n$ which forms a cell decomposition according to Definition 6.1 and respects the labeling function Λ i.e., for all $S_\ell \in S$ it holds that $\Lambda(x) = \Lambda(x'), \forall\, x, x' \in S_\ell$. This assumption implies that the same services hold at all the points that belong to the same cell of the decomposition.

Define for each agent $i \in \mathcal{V}$ the labeling function:

$$\mathcal{L}_i : S \to 2^{\Sigma_i}, \tag{6.4}$$

which denotes the fact that when agent i visits a region $S_\ell \in S$ it can choose to *provide* a subset of the services that are being offered there i.e., it chooses to satisfy a subset of $\mathcal{L}_i(S_\ell)$.

The trajectory of each agent i is denoted by $x_i(t), t \geq 0, i \in \mathcal{V}$. The trajectory $x_i(t)$ is associated with a unique sequence:

$$r^t_{x_i} = (r_i(0), \tau_i(0))(r_i(1), \tau_i(1))(r_i(2), \tau_i(2)) \ldots,$$

of regions that the agent i crosses, where for all $j \geq 0$ it holds that: $x_i(\tau_i(j)) \in r_i(j)$ and

$$\Lambda_i(x_i(t)) = \mathcal{L}_i(r_i(j)), \quad \forall\, t \in [\tau_i(j), \tau_i(j+1)),$$

for some $r_i(j) \in S$ and $r_i(j) \neq r_i(j+1)$. The equality $\Lambda_i(\cdot) = \mathcal{L}_i(\cdot), i \in \mathcal{V}$ is feasible due to Assumption 6.2. The timed word

$$w^t_{x_i} = (\mathcal{L}_i(r_i(0)), \tau_i(0))(\mathcal{L}_i(r_i(1)), \tau_i(1))(\mathcal{L}_i(r_i(2)), \tau_i(2)) \ldots,$$

where

$$w_i(j) = \mathcal{L}_i(r_i(j)), j \geq 0, i \in \mathcal{V},$$

is associated uniquely with the trajectory $x_i(t)$, and represents the sequence of services that *can be provided* by the agent i following the trajectory $x_i(t), t \geq 0$.

Define a *timed service word* as:

$$\widetilde{w}^t_{x_i} = (\beta_i(z_0), \widetilde{\tau}_i(z_0))(\beta_i(z_1), \widetilde{\tau}_i(z_1))(\beta_i(z_2), \widetilde{\tau}_i(z_2))\ldots, \tag{6.5}$$

where $z_0 = 0 < z_1 < z_2 < \ldots$ is a sequence of integers, and for all $j \geq 0$ it holds that $\beta_i(z_j) \subseteq \mathcal{L}_i(r_i(z_j))$ and $\widetilde{\tau}_i(z_j) \in [\tau_i(z_j), \tau_i(z_j+1))$. The timed service word is a sequence of services that are actually provided by agent i and is compliant with the trajectory $x_i(t), t \geq 0$ by construction.

The specification task φ_i given as an MITL formula over the set of services Σ_i, captures requirements on the services to be provided by agent i, for each $i \in \mathcal{V}$. We say that a trajectory $x_i(t)$ satisfies a formula φ_i given in MITL over the set Σ_i, and formally write $x_i(t) \models \varphi_i, \forall t \geq 0$, if and only if there exists a *timed service word* $\widetilde{w}^t_{x_i}$ that complies with $x_i(t)$ and satisfies φ_i according to semantics of Definition 2.14.

Example 6.1. Consider $N = 2$ agents performing in the partitioned environment of Fig. 6.1. Both agents have the ability to pick up, deliver and throw two different balls. Their sets of services are $\Sigma_1 = \{\text{pickUp1}, \text{deliver1}, \text{throw1}\}$ and $\Sigma_2 = \{\text{pickUp2}, \text{deliver2}, \text{throw2}\}$, respectively, and satisfy $\Sigma_1 \cap \Sigma_2 = \emptyset$. Three points of the agents' trajectories that belong to different cells with different services are captured. Assume that $t_1 < t'_1 < t_2 < t_2 < t'_2 < t_3 < t'_3$. The trajectories $x_1(t)$, $x_2(t)$, $t \geq 0$ are depicted with the red lines. According to Assumption 6.2, the cell decomposition $S = \{S_\ell\}_{\ell \in \mathbb{I}} = \{S_1, \ldots, S_6\}$ is given where $\mathbb{I} = \{1, \ldots, 6\}$ respects the labeling functions $\Lambda_i, \mathcal{L}_i, i \in \{1, 2\}$. In particular, it holds that:

$$\begin{aligned}
&\Lambda_1(x_1(t)) = \mathcal{L}_1(r_1(0)) = \{\text{pickUp1}\}, t \in [0, t_1),\\
&\Lambda_1(x_1(t)) = \mathcal{L}_1(r_1(1)) = \{\text{throw1}\}, t \in [t_1, t_2),\\
&\Lambda_1(x_1(t)) = \mathcal{L}_1(r_1(2)) = \{\text{deliver1}\}, t \in [t_2, t_3),\\
&\Lambda_1(x_1(t)) = \mathcal{L}_1(r_1(3)) = \emptyset, t \geq t_3.\\
&\Lambda_2(x_2(t)) = \mathcal{L}_2(r_2(0)) = \{\text{pickUp2}\}, t \in [0, t'_1),\\
&\Lambda_2(x_2(t)) = \mathcal{L}_2(r_2(1)) = \{\text{deliver2}\}, t \in [t'_1, t'_2),\\
&\Lambda_2(x_2(t)) = \mathcal{L}_2(r_2(2)) = \{\text{throw2}\}, t \in [t'_2, t'_3),\\
&\Lambda_2(x_2(t)) = \mathcal{L}_2(r_2(3)) = \emptyset, t \geq t'_3.
\end{aligned}$$

By using the fact that $w_i(j) = \mathcal{L}(r_i(j)), \forall\, i \in \{1, 2\}, j \in \{1, 2, 3\}$, the corresponding individual timed words are given as:

$$\begin{aligned}
w^t_{x_1} &= (\{\text{pickUp1}\}, 0)(\{\text{throw1}\}, t_1)(\{\text{deliver1}\}, t_2)(\emptyset, t_3),\\
w^t_{x_2} &= (\{\text{pickUp2}\}, 0)(\{\text{deliver2}\}, t'_1)(\{\text{throw2}\}, t'_2)(\emptyset, t'_3).
\end{aligned}$$

According to (6.5), two time service words (depicted with in Fig. 6.1) are given as:

$$\begin{aligned}
\widetilde{w}^t_1 &= (\beta_1(z_0), \widetilde{\tau}_1(z_0))(\beta_1(z_1), \widetilde{\tau}_1(z_1)),\\
\widetilde{w}^t_2 &= (\beta_2(z'_0), \widetilde{\tau}_2(z'_0))(\beta_2(z'_1), \widetilde{\tau}_2(z'_1)),
\end{aligned}$$

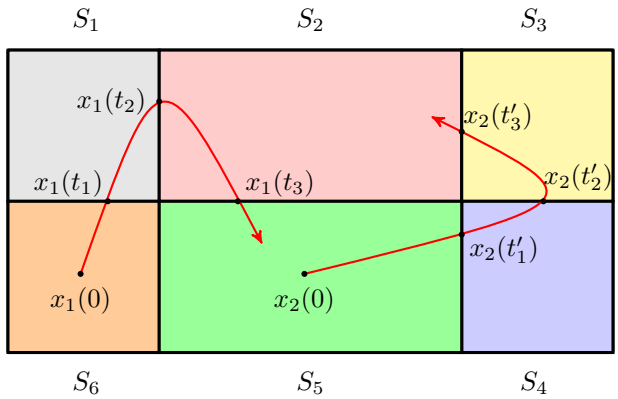

Figure 6.1: An example of two agents performing in a partitioned workspace.

where for agent 1 we have:

$$z_0 = 0, \quad z_1 = 2,$$
$$\beta_1(z_0) = \{\text{pickUp1}\} \subseteq \mathcal{L}_1(r_1(z_0)), \quad \beta_1(z_1) = \{\text{deliver1}\} \subseteq \mathcal{L}_1(r_1(z_1))$$

The corresponding elements for agent 2 are:

$$z'_0 = 0, \quad z'_1 = 2,$$
$$\beta_2(z'_0) = \{\text{pickUp2}\} \subseteq \mathcal{L}_2(r_2(z'_0)), \quad \beta_2(z'_1) = \{\text{throw2}\} \subseteq \mathcal{L}_2(r_2(z'_1))$$

The time stamps $\widetilde{\tau}_1(z_0), \widetilde{\tau}_1(z_1)$ should satisfy the following conditions:

$$\widetilde{\tau}_1(z_0) \in [\tau_1(z_0), \tau_1(z_0+1)) = [0, t_1),$$
$$\widetilde{\tau}_1(z_1) \in [\tau_1(z_1), \tau_1(z_1+1)) = [t_2, t_3),$$
$$\widetilde{\tau}_2(z'_0) \in [\tau_2(z'_0), \tau_2(z'_1)) = [0, t'_1),$$
$$\widetilde{\tau}_2(z'_1) \in [\tau_1(z'_1), \tau_1(z'_1+1)) = [t'_2, t'_3).$$

6.4.1 Problem Statement

We are now ready to define the problem treated in this chapter formally as follows.

Problem 6.1. Given N agents that are governed by dynamics as in (6.2), modeled by the undirected communication graph $\mathcal{G}$, N task specification formulas $\varphi_1, \ldots, \varphi_N$ expressed in MITL over the mutually disjoint sets of services $\Sigma_1, \ldots, \Sigma_N$, respectively, service labeling functions $\Lambda_1, \ldots, \Lambda_N$, as in (6.3), a cell decomposition $S = \{S_\ell\}_{\ell \in \mathbb{I}}$ as in Assumption 6.2 and the labeling functions $\mathcal{L}_1, \ldots, \mathcal{L}_N$, given by (6.4), assign control laws to the free inputs $u_1, \ldots, u_N$ such that each agent fulfills its individual specification i.e., $x_i(t) \models \varphi_i, \forall i \in \mathcal{V}, t \geq 0$, given the upper bound $u_{\max}$.

Remark 6.1. It should be noted that, in this work, the dependencies between the agents are induced through the coupled dynamics (6.2) and not in the discrete level, by allowing for couplings between the services (i.e., $\Sigma_i \cap \Sigma_j \neq \emptyset$, for some $i,\ j \in \mathcal{V}$). Hence, even though the agents do not share atomic propositions, the constraints on their motion due to the dynamic couplings may restrict them to fulfill the desired high-level tasks. Treating additional couplings through individual atomic propositions in the discrete level, constitutes a far from trivial problem, which is a topic of current work.

Remark 6.2. In Chapter 5, the multi-agent system was considered to have fully-actuated dynamics. The only constraints on the system were due to the presence of time-constrained MITL formulas. In the current framework, we have two types of constraints: the constraints due to the coupling dynamics of the system (6.2), which constrain the motion of each agent, and, the timed constraints that are inherently imposed from the time bounds of the MITL formulas. Thus, there exist formulas that cannot be satisfied either due to the coupling constraints or the time constraints of the MITL formulas. These constraints, make the procedure of the controller synthesis in the discrete level substantially different and more elaborate than the corresponding multi-agent LTL frameworks in the literature ([94, 95, 165]).

Remark 6.3. The motivation for introducing the cell decomposition $S = \{S_\ell\}_{\ell \in \mathbb{I}}$ in this Section, comes from the requirement to know *a priori* which services hold in each part of the workspace. As will be clarified through the problem solution, this is necessary since the abstraction of the workspace (which is part of our proposed solution) may not be compliant with the initial given cell decomposition, and thus, new cell decompositions might be required.

6.5 Proposed Solution

In this section, a systematic solution to Problem 6.1 is introduced. Our overall approach builds on abstracting system (6.2) through a WTS for each agent and exploiting the fact that the timed runs in the i-th WTS project onto the trajectories of agent i while preserving the satisfaction of the individual MITL formulas $\varphi_i, i \in \mathcal{V}$. The following analysis is performed.

1. Initially, the boundedness of the agents' relative positions is proved, in order to guarantee boundedness of the coupling terms $-\sum_{j \in \mathcal{N}(i)}(x_i - x_j)$. This property is required for the derivation of the symbolic models. (Section 6.5.1);
2. We utilize decentralized abstraction techniques for the multi-agent system, i.e., a discretization of both the workspace and time in order to model the motion capabilities of each agent by a WTS $\mathcal{T}_i,\ i \in \mathcal{V}$ (Section 6.5.2);
3. Given the WTSs, consistent runs are defined in order to take into consideration the coupling constraints among the agents. The computation of the product of the individual WTSs is also required (Section 6.5.3);

4. A five-step automated procedure for controller synthesis which serves as a solution to Problem 6.1 is provided in Section 6.5.4;

5. Finally, the computational complexity of the proposed approach is discussed in Section 7.3.4.

The next sections provide the proposed solution in detail.

6.5.1 Boundedness Analysis

Theorem 6.1. *Consider the multi-agent system* (6.2) *modeled by the undirected communication graph* $\mathcal{G}$*. Suppose that Assumption 6.1 holds (i.e.* $\lambda_2(\mathcal{G}) > 0$*) and let* $u_i, i \in \mathcal{V}$ *satisfy* $\|u_i(t)\|_2 \leq u_{\max},\ \forall\ i \in \mathcal{V}, t \geq 0$*. Furthermore, let* $\bar{R} > K_2 v_{max}$ *be a positive constant, where* $K_2 = \frac{2\sqrt{N}(N-1)\left\|\mathscr{D}(\mathcal{G})^\top\right\|}{\lambda_2^2(\mathcal{G})} > 0$ *and where* $\mathscr{D}(\mathcal{G})$ *is the network adjacency matrix. Then, for each initial condition* $x_i(0) \in \mathbb{R}^n$*, there exists a time* $T > 0$ *such that* $\widetilde{x}(t) \in \mathcal{X},\ \forall t \geq T$*, where* $\mathcal{X} = \{x \in \mathbb{R}^{Nn} : \|\widetilde{x}\|_2 \leq \bar{R}\}$*.*

Proof. The proof can be found in Appendix C. □

It should be noticed that the relative boundedness of the agents' positions guarantees a global bound on the coupling terms $-\sum_{j\in\mathcal{N}_i}(x_i - x_j)$, as defined in (6.2). This bound will be later exploited in order to capture the behavior of the system in $\mathcal{X} = \{x \in \mathbb{R}^{Nn} : \|\widetilde{x}\|_2 \leq \bar{R}\}$, by a discrete state WTS.

6.5.2 Abstraction

In this section we provide the abstraction technique that is adopted in [116] in order to capture the dynamics of each agent into WTSs. Thereafter, we work completely in discrete level, which is necessary in order to solve Problem 6.1.

Firstly, some additional notation is introduced. Given an index set $\mathbb{I}$ and an agent $i \in \mathcal{V}$ with neighbors $j_1, \ldots, j_{N_i} \in \mathcal{N}_i$, define the mapping $\mathrm{pr}_i : \mathbb{I}^N \to \mathbb{I}^{N_i+1}$ which assigns to each N-tuple $\mathbf{l} = (l_1, \ldots, l_N) \in \mathbb{I}^N$ the N_i+1 tuple $\mathbf{l}_i = (l_i, l_{j_1}, \ldots, l_{j_{N_i}}) \in \mathbb{I}^{N_i+1}$ which denotes the indices of the cells where the agent i and its neighbors belong.

Well-posed Abstractions

Loosely speaking, an abstraction is characterized by a discretization of the workspace into cells, which we denote by $\bar{S} = \{\bar{S}_l\}_{l\in\bar{\mathbb{I}}}$, a time step δt and selection of feedback laws in place of the free inputs u_i, $\forall i \in \mathcal{V}$. The time step δt models the time that an agent needs to transit from one cell to another, and u_i is the controller that

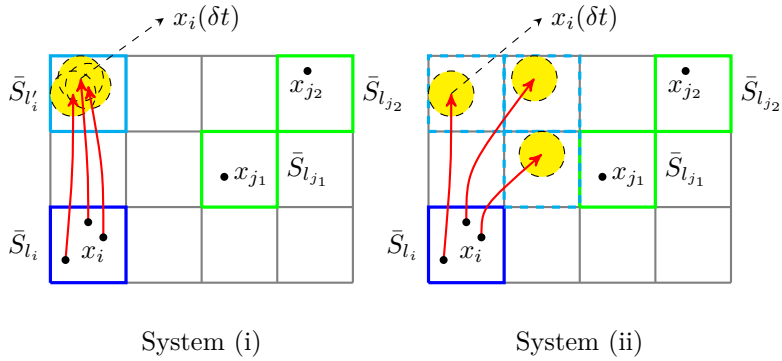

Figure 6.2: Illustration of a space-time discretization which is well-posed for system (i) but non-well posed for system (ii).

guarantees such a transition. Note that the time step δt is the same for all the agents. Let us denote by $(\bar{S}, \delta t)$ the aforementioned *space-time discretization*.

Before defining formally the concept of well-posed abstractions, an intuitive graphical representation is provided. Consider a cell decomposition $\bar{S} = \{\bar{S}_l\}_{l \in \bar{\mathbb{I}} = \{1,\dots,12\}}$ as depicted in Fig. 6.2 and a time step δt. The tails and the tips of the arrows in the figure depict the initial state and the endpoints of agent's i trajectories at time δt respectively. In both cases in the figure we focus on agent i and consider the same cell configuration for i and its neighbors. By configuration we mean the cell that the agent i and its neighbors belong at a current time. However, different dynamics are considered for Cases (i) and (ii). In Case (i), it can be observed that for the three distinct initial positions in cell $\bar{S}_{l_i}$, it is possible to drive agent i to cell $\bar{S}_{l'_i}$ at time δt. We assume that this is possible for all initial conditions in this cell and irrespectively of the initial conditions of i's neighbors in their cells and the inputs they choose. It is also assumed that this property holds for all possible cell configurations of i and for all the agents of the system. Thus, we have a *well-posed discretization* for system (i). On the other hand, for the same cell configuration and system (ii), the following can be observed. For three distinct initial conditions of i the corresponding reachable sets at δt, which are enclosed in the dashed circles, lie in different cells. Thus, it is not possible given this cell configuration of i to find a cell in the decomposition which is reachable from every point in the initial cell and we conclude that discretization is not well-posed for system (ii).

More specifically, consider a $(\bar{S}, \delta t)$-space-time discretization which is the outcome of the abstraction technique that is designed for the problem solution and will be presented in Section 6.5.2. Let $\bar{S} = \{\bar{S}_l\}_{l \in \bar{\mathbb{I}}}$ be a cell decomposition in which the agent i occupies the cell $\bar{S}_{l_i}$, δt be a time step and

$$\bar{d}_{\max} := \sup\{\|x - y\|_2 : x, y \in \bar{S}_l, l \in \mathbb{I}\},$$

be the diameter of the cell decomposition $\bar{S}$. It should be noted that this decompo-

sition is not necessarily the same cell decomposition S from Assumption 6.2 and Problem 6.1. Through the aforementioned space and time discretization $(\bar{S}, \delta t)$ we aim to capture the reachability properties of the continuous system (6.2), by creating an individual WTS for each agent. If there exists a free input for each state in $\bar{S}_{l_i}$ that navigates the agent i into the cell $\bar{S}_{l_i'}$ precisely at time δt, regardless of the locations of the agent i's neighbors within their current cells, then a transition from l_i to l_i' is enabled in the WTS. This forms the well-possessedness of transitions. A more detailed mathematical derivation as well as feedback laws $u_i(t)$, $i \in \mathcal{V}$ which guarantee a well-posed space-time discretization $(\bar{S}, \delta t)$ can be found in [116].

Sufficient Conditions

We present at this point the sufficient conditions that relate the dynamics of the multi-agent system (6.2), the time step δt and the diameter $\bar{d}_{\max}$, and guarantee the existence of the aforementioned well-posed transitions for each cell. Based on our previous work [116], in order to derive well-posed abstractions, a nonlinear system of the form:

$$\dot{x}_i = f_i(x_i, \mathbf{x}_j) + u_i, \quad i \in \mathcal{V}, \tag{6.6}$$

where $\mathbf{x}_j = (x_{j_1}, \ldots, x_{j_{N_i}}) \in \mathbb{R}^{N_i n}$, should fulfill the following sufficient conditions.

(C1) There exists $M > v_{\max} > 0$ such that $\|f_i(x_i, \mathbf{x}_j)\|_2 \leq M,\ \forall i \in \mathcal{V}, \forall\ x \in \mathbb{R}^{Nn}$: $\mathrm{pr}_i(x) = (x_i, \mathbf{x}_j)$ and $\widetilde{x} \in \mathcal{X}$, by applying the projection operator pr_i for $\mathbb{I} = \mathbb{R}^n$;

(C2) There exist a Lipschitz constant $L_1 > 0$ such that:

$$\|f_i(x_i, \mathbf{x}_j) - f_i(x_i, \mathbf{y}_j)\|_2 \leq L_1 \|(x_i, \mathbf{x}_j) - (x_i, \mathbf{y}_j)\|_2, \\ \forall\ i \in \mathcal{V}, \quad x_i, y_i \in \mathbb{R}^n, \quad \mathbf{x}_j, \mathbf{y}_j \in \mathbb{R}^{N_i n}. \tag{6.7}$$

(C3) There exist a Lipschitz constant $L_2 > 0$ such that:

$$\|f_i(x_i, \mathbf{x}_j) - f_i(y_i, \mathbf{x}_j)\|_2 \leq L_2 \|(x_i, \mathbf{x}_j) - (y_i, \mathbf{x}_j)\|_2, \\ \forall\ i \in \mathcal{V}, \quad x_i, y_i \in \mathbb{R}^n, \quad \mathbf{x}_j, \mathbf{y}_j \in \mathbb{R}^{N_i n}.$$

Using the result of Theorem 6.1, it can be shown that system (6.2) with:

$$f_i(x_i, \mathbf{x}_j) = -\sum_{j \in \mathcal{N}(i)} (x_i - x_j),$$

satisfies the the conditions **(C1)**-**(C3)**. The proof can be found in Appendix C.

Based on the sufficient conditions for well posed abstractions in [116], the diameter $\bar{d}_{\max}$ and time step δt of the discretization $(\bar{S}, \delta t)$ can be selected as:

$$\bar{d}_{\max} \in \left(0, \frac{(1-\lambda)^2 v_{\max}^2}{4ML}\right], \tag{6.8}$$

$$\delta t \in \left[\frac{(1-\lambda)v_{\max} - \sqrt{(1-\lambda)^2 v_{\max}^2 - 4ML\bar{d}_{\max}}}{2ML}, \frac{(1-\lambda)v_{\max} + \sqrt{(1-\lambda)^2 v_{\max}^2 - 4ML\bar{d}_{\max}}}{2ML}\right], \tag{6.9}$$

where:

$$L \coloneqq 3L_2 + 4L_1 \max\left\{\sqrt{N_i}, i \in \mathcal{V}\right\},$$

and with the dynamics bound M and the Lipschitz constants L_1, L_2 as previously defined. Furthermore, $\lambda \in (0,1)$ is a design parameter which quantifies the part of the free input that is additionally exploited for reachability purposes. In particular, given an agent's initial cell configuration, the agent can reach any point inside an appropriate ball at δt through a parameterized feedback law in place of the free input u_i. The center of this ball is selected as the endpoint of a reference trajectory for agent i, which is obtained by considering its neighbors fixed at certain reference points in their cells during the transition, providing thus an estimate of the agent's reachable states at δt. The radius of this ball increases proportionally to λ, and thus, also to the number of the agent's successor cells, which are the ones intersecting the ball. It is noted that an increasing choice of λ results in finer discretizations, therefore providing a quantifiable trade-off between the discrete model's accuracy and complexity. Furthermore, it follows from the acceptable values of $\bar{d}_{\max}$ that the cells can be selected coarser, when (i) the available control $u_{\max}$ is larger, and, (ii) the coupling term bound M together with the dynamics' variation, which is captured through the parameter L, are smaller. Analogous restrictions need to hold for the time step δt. In particular, the time step cannot be selected very large, because of the feedback which is used to modify the agent's couplings in accordance to the dynamics of its reference trajectory. More specifically, the corresponding control effort increases with time due to the evolution of the agent's neighbors away from their reference points during the transition interval. Finally, the time step cannot be selected very small compared to the diameter of the cells, because controlling the agent to the same point from each initial condition in its cell, will require a large control effort over a very short transition interval.

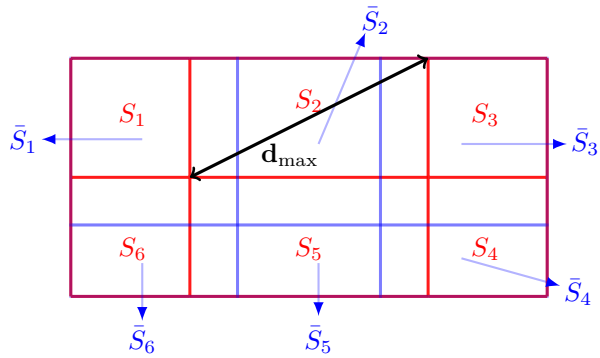

Figure 6.3: An example with a given cell decomposition $S = \{S_l\}_{l\in\{1,\dots,6\}}$ from Problem 6.1 and a non-compliant cell decomposition $\bar{S} = \{\bar{S}_l\}_{l\in\bar{\mathbb{I}}=\{1,\dots,6\}}$ which is the outcome of the proposed abstraction technique.

Remark 6.4. Assume that a cell-decomposition of diameter $\bar{d}_{\max}$ and a time step δt which guarantee well-posed transitions, namely, which satisfy (6.8) and (6.9), have been chosen. Then, it is also possible to choose any other cell-decomposition with diameter $\hat{d}_{\max} \leq \bar{d}_{\max}$ since, by (6.9), the range of acceptable δt increases.

Having shown that the dynamics of system (6.2) satisfy the sufficient conditions **(C1)-(C3)**, a well-posed space-time discretization $(\bar{S}, \delta t)$ has been obtained. Recall now Assumption 6.2. It remains to establish the compliance of the cell decomposition $S = \{S_\ell\}_{\ell\in\mathbb{I}}$, which is given in the statement of Problem 6.1, with the cell decomposition $\bar{S} = \{\bar{S}_l\}_{l\in\bar{\mathbb{I}}}$, which is the outcome of the abstraction. By the term of compliance, we mean that:

$$\bar{S}_l \cap S_\ell \in S \cup \{\emptyset\}, \quad \forall \bar{S}_l \in \bar{S}, \quad S_\ell \in S, \quad l \in \bar{\mathbb{I}}, \quad \ell \in \mathbb{I}.$$

In order to address this problem, define:

$$\hat{S} = \{\hat{S}_{\hat{l}}\}_{\hat{l}\in\hat{\mathbb{I}}} = \{\bar{S}_l \cap S_\ell : \ l \in \bar{\mathbb{I}}, \ell \in \mathbb{I}\}\backslash\{\emptyset\},$$

which forms a cell decomposition and is compliant with the cell decomposition S from Problem 6.1 with diameter

$$\hat{d}_{\max} = \sup\{\|x - y\|_2 : x, y \in \hat{S}_{\hat{l}}, \hat{l} \in \hat{\mathbb{I}}\} \leq \bar{d}_{\max},$$

and serves as the abstraction solution of this problem. By taking all the intersections $\bar{S}_l \cap S_\ell, \forall\ l \in \bar{\mathbb{I}}, \forall\ \ell \in \mathbb{I}$ and enumerating them through the index set $\hat{\mathbb{I}}$, the cells $\{\hat{S}_{\hat{l}}\}_{\hat{l}\in\hat{\mathbb{I}}}$ are constructed. For the cells $\{\hat{S}_{\hat{l}}\}_{\hat{l}\in\hat{\mathbb{I}}}$, the following holds: $\forall\ \ell \in \mathbb{I}, \exists\ l \in \bar{\mathbb{I}}$ such that $\hat{S}_{\hat{l}} = \bar{S}_l \cap S_\ell$ and $\text{int}(\hat{S}_{\hat{l}}) \cap \text{int}(\hat{S}_{\hat{l}'}) \neq \emptyset$ for all $\hat{l}' \in \hat{\mathbb{I}}\backslash\{\hat{l}\}$. After all the intersections

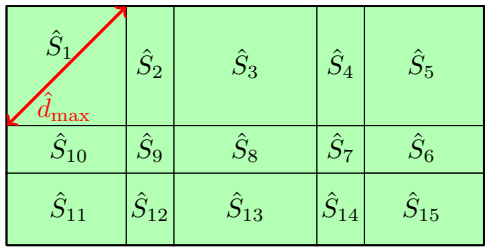

Figure 6.4: The resulting compliant cell decomposition $\hat{S} = \{\hat{S}_{\hat{l}}\}_{\hat{l}\in\hat{\mathbb{I}}=\{1,\ldots,15\}}$ of the Example 6.2 which serves as solution to Problem 6.1.

we have $\cup_{\hat{l}\in\hat{\mathbb{I}}}\hat{S}_{\hat{l}} = \mathcal{X}$. The diameter of the cell decomposition $\hat{S} = \{\hat{S}_{\hat{l}}\}_{\hat{l}\in\hat{\mathbb{I}}}$ is defined as $\hat{d}_{\max} = \sup\{\|x-y\| : x, y \in \hat{S}_{\hat{l}}, \hat{l} \in \hat{\mathbb{I}}\} \leq \bar{d}_{\max}$. Hence, according to Remark 6.4, we have a well-posed abstraction. Example 6.2 is an illustration of these derivations.

Example 6.2. Let $S = \{S_\ell\}_{\ell\in\{1,\ldots,6\}}$ be the cell decomposition of Problem 6.1, which is depicted in Fig. 6.3 by the red rectangles. In the same figure, we illustrate the cell decomposition $\bar{S} = \{\bar{S}_l\}_{l\in\bar{\mathbb{I}}=\{1,\ldots,6\}}$; $\bar{S}$ serves as potential solution of this Problem satisfying all the abstraction properties that have been mentioned in this Section. It can be observed that the two cell decompositions are not compliant. However, by using the methodology below Remark 6.4, a new cell decomposition $\hat{S} = \{\hat{S}_{\hat{l}}\}_{\hat{l}\in\hat{\mathbb{I}}=\{1,\ldots,15\}}$, which is compliant with S and has 15 regions, can be obtained; $\hat{S}$ forms the final cell decomposition solution of Problem 6.1 and is depicted in Fig. 6.4. Let also $\bar{d}_{\max}, \hat{d}_{\max}$ be the diameters of the cell decompositions $\bar{S}, \hat{S}$ respectively. Then, it holds that $\hat{d}_{\max} \leq \bar{d}_{\max}$, which is in accordance with Remark 6.4.

Discrete System Abstraction

For the solution to Problem 6.1, the cell decomposition $\hat{S}$ with diameter $\hat{d}_{\max}$ and the time step δt will be exploited. Thus, the WTS of each agent is defined as follows:

Definition 6.2. The motion of each agent $i \in \mathcal{V}$ in the workspace is modeled by the WTS $\mathcal{T}_i = (S_i, S_i^{\text{init}}, \text{Act}_i, \longrightarrow_i, \mathbf{t}_i, \Sigma_i, \mathcal{L}_i)$ where:

- $S_i = \hat{\mathbb{I}}$ is the set of states of each agent which is the set of indices of the cell decomposition;
- $S_i^{\text{init}} \subseteq S_i$ is a set of initial states defined by the agents' initial positions in the workspace;
- $\text{Act}_i = \hat{\mathbb{I}}^{N_i+1}$, the set of actions representing where agent i and its neighbors are located; For a pair $(l_i, \mathbf{l}_i, l_i')$ we have that $(l_i, \mathbf{l}_i, l_i') \in \longrightarrow_i$ iff $l_i \xrightarrow{\mathbf{l}_i}_i l_i'$ is well-posed for each $l_i, l_i' \in S_i$ and $\mathbf{l}_i = (l_i, l_{j_1}, \ldots, l_{j_{N_i}}) \in \text{Act}_i$;

- $\mathfrak{t}_i :\longrightarrow_i\rightarrow \mathbb{Q}_{\geq 0}$, is a map that assigns a positive weight (duration) to each transition. The duration of each transition is exactly equal to $\delta t > 0$;
- Σ_i, is the set of atomic propositions which are inherent properties of the workspace; and
- $\mathcal{L}_i : S_i \rightarrow 2^{\Sigma_i}$, is the labeling function that maps every state $s_i \in S_i$ into the services that can be provided in this state.

The individual WTSs of the agents will allow us to work completely in the discrete level and design sequences of controllers that solve Problem 6.1. Every WTS $\mathcal{T}_i$, $i \in \mathcal{V}$ generates timed runs and timed words of the form:

$$r_i^t = (r_i(0), \tau_i(0))(r_i(1), \tau_i(1))(r_i(2), \tau_i(2)) \ldots,$$
$$w_i^t = (\mathcal{L}_i(r_i(0)), \tau_i(0))(\mathcal{L}_i(r_i(1)), \tau_i(1))(\mathcal{L}_i(r_i(2)), \tau_i(2)) \ldots,$$

respectively, over the set 2^{Σ_i} according to Definition 2.12 and Definition 2.13 with $\tau_i(j) = j\delta t, \forall\ j \geq 0$. The relation between the timed words that are generated by the WTSs $\mathcal{T}_i$, $i \in \mathcal{V}$ with the timed service words produced by the trajectories $x_i(t)$, $i \in \mathcal{V}$, $t \geq 0$ is provided through the following remark.

Remark 6.5. By construction, each timed word produced by the WTS $\mathcal{T}_i$ is a service timed word associated with the trajectory $x_i(t)$ of the system (6.2). Hence, if we find a timed word of $\mathcal{T}_i$ satisfying a formula φ_i given in MITL, we also find for each agent i a desired timed word of the original system, and hence trajectories $x_i(t)$ that are a solution to the Problem 6.1. Therefore, the produced timed words of $\mathcal{T}_i$ are compliant with the service timed words of the trajectories $x_i(t)$.

6.5.3 Consistency of Runs

Due to the coupled dynamics between the agents, it is required that each individual agent's run is compliant with the corresponding discrete trajectories of its neighbors, which determine the actions in the agent's run. Therefore, even though we have the individual WTS of each agent, the runs that the latter generates may not be performed by the agent due to the constrained motion that is imposed by the coupling terms. Hence, we need to synchronize the agents at each time step δt and determine which of the generated runs of the individual WTS can be performed by the agent. Hereafter, they will be called *consistent runs*. In order to address the aforementioned issue, we provide a centralized product WTS which captures the behavior of the coupled multi-agent system as a team, and the generated product run (see Definition 6.4) can later be projected onto consistent individual runs. The following two definitions deal with the product WTS and consistent runs respectively.

Definition 6.3. Given the individual WTSs $\mathcal{T}_i, i \in \mathcal{V}$ from Definition 6.2, the product WTS $\mathcal{T}_p = (S_p, S_p^{\text{init}}, \longrightarrow_p, \mathfrak{t}_p, \cup_{i=1}^N \Sigma_i, \mathcal{L}_p)$ is defined as follows:

- $S_p = \hat{\mathbb{I}}^N$; $(s_1, \ldots, s_N) \in S^{\text{init}}$ if $s_i \in S_i^{\text{init}}, \forall i \in \mathcal{V}$;
- $(\mathbf{l}, \mathbf{l}') \in \longrightarrow_p$ iff $l_i' \in \text{Post}_i(l_i, \text{pr}_i(\mathbf{l})), \forall i \in \mathcal{V}, \forall \mathbf{l} = (l_1, \ldots, l_N), \mathbf{l}' = (l_1', \ldots, l_N')$;
- $\mathfrak{t}_p : \longrightarrow_p \rightarrow \mathbb{Q}_{\geq 0}$ as in the individual WTS's case, with transition weight $\mathfrak{t}_p(\cdot) = \delta t$;
- $\mathcal{L}_p : \hat{\mathbb{I}}^N \rightarrow 2^{\cup_{i=1}^N \Sigma_i}$ defined as $\mathcal{L}_p(\mathbf{l}) = \cup_{i=1}^N \mathcal{L}_i(l_i)$.

Definition 6.4. Given a timed run:

$$r_p^t = ((r_p^1(0), \ldots, r_p^N(0)), \tau_p(0))((r_p^1(1), \ldots, r_p^N(1)), \tau_p(1)) \ldots,$$

that is generated by the product WTS $\mathcal{T}_p$, the induced set of projected runs:

$$\{r_i^t = (r_p^i(0), \tau_p(0))(r_p^i(1), \tau_p(1)) \ldots : i \in \mathcal{V}\},$$

of the WTSs $\mathcal{T}_1, \ldots, \mathcal{T}_N$, respectively, will be called *consistent runs*. Since the duration of each agent's transition is δt it holds that $\tau_p(j) = j\delta t, j \geq 0$.

Therefore, through the product WTS $\mathcal{T}_p$, we can always generate individual consistent runs for each agent. It remains to provide a systematic approach of how to determine consistent runs $\widetilde{r}_1, \ldots, \widetilde{r}_N$ which are associated with the corresponding time serviced words $\widetilde{w}_1^t, \ldots, \widetilde{w}_N^t$. Note that we use the tilde accent to denote timed runs and words that correspond to the problem solution. The corresponding compliant trajectories $x_1(t), \ldots, x_N(t)$ of the timed words $\widetilde{w}_1^t, \ldots, \widetilde{w}_N^t$ satisfy the corresponding MITL formulas $\varphi_1, \ldots, \varphi_N$, and they are a solution to Problem 6.1.

Consider here an example that explains the notation that has been introduced until now.

Example 6.3. Consider $N = 3$ agents performing in the workspace with $\mathcal{N}_i = \{1, 2\}$ as depicted in Fig. 6.5. $S = \{S_\ell\}_{\ell \in \mathbb{I} = \{1, \ldots, 6\}}$ is the given cell decomposition from Problem 6.1 and $S = \{\bar{S}_l\}_{l \in \mathbb{I} = \{1, \ldots, 28\}}$ is the cell decomposition which is the outcome of the proposed abstraction technique. Let the atomic propositions be $\{\sigma_1, \ldots, \sigma_6\} = \{\text{orange}, \text{green}, \text{blue}, \text{yellow}, \text{red}, \text{grey}\}$. The red arrows represent both the transitions of the agent i and its neighbors. The dashed lines indicate the edges in the network graph. For the atomic propositions we have that $\mathcal{L}_i(14) = \{\sigma_1\}$, $\mathcal{L}_i(17) = \{\sigma_5\}$, $\mathcal{L}_i(10) = \{\sigma_2\}$, $\mathcal{L}_i(20) = \{\sigma_4\}$, $\mathcal{L}_{j_1}(28) = \{\sigma_6\} = \mathcal{L}_{j_1}(27)$, $\mathcal{L}_{j_1}(24) = \{\sigma_5\}$, $\mathcal{L}_{j_1}(22) = \{\sigma_4\}$, $\mathcal{L}_{j_2}(2) = \{\sigma_1\}$, $\mathcal{L}_{j_2}(12) = \{\sigma_2\} = \mathcal{L}_{j_2}(5)$, $\mathcal{L}_{j_2}(9) = \{\sigma_3\}$. Note also the diameter of the cells $\hat{d}_{\max} = \bar{d}_{\max}$. For the cell configurations we have:

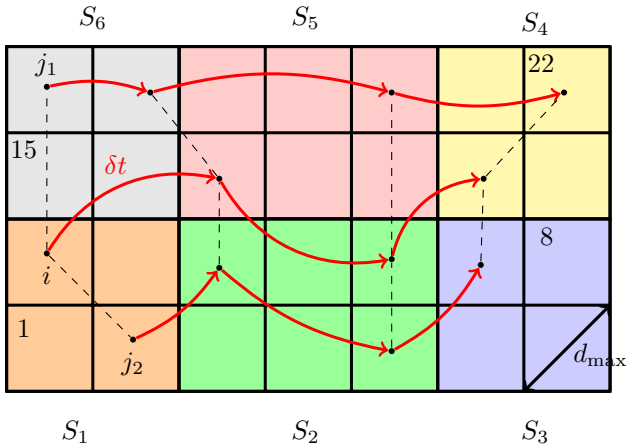

Figure 6.5: Timed runs of the agents i, j_1, j_2

$$\text{Init } (t=0): \begin{cases} \mathbf{l}_i = (14, 28, 2), \\ \mathbf{l}_{j_1} = (28, 14), \\ \mathbf{l}_{j_2} = (2, 14), \end{cases}$$

$$\text{Step 1 } (t=\delta t): \begin{cases} \mathbf{l}_i = (17, 27, 13), \\ \mathbf{l}_{j_1} = (27, 17), \\ \mathbf{l}_{j_2} = (13, 17), \end{cases}$$

$$\text{Step 2 } (t=2\delta t): \begin{cases} \mathbf{l}_i = (10, 24, 5), \\ \mathbf{l}_{j_1} = (24, 10), \\ \mathbf{l}_{j_2} = (5, 10), \end{cases}$$

$$\text{Step 3 } (t=3\delta t): \begin{cases} \mathbf{l}_i = (20, 22, 9), \\ \mathbf{l}_{j_1} = (22, 20), \\ \mathbf{l}_{j_2} = (9, 20), \end{cases}$$

which are actions to the corresponding transitions. Three consistent timed runs are given as:

$$\begin{aligned} r_i^t &= (r_i(0)=14, \tau_i(0)=0)(r_i(1)=17, \tau_i(1)=\delta t) \\ &\quad (r_i(2)=10, \tau_i(2)=2\delta t)(r_i(3)=20, \tau_i(3)=3\delta t), \\ r_{j_1}^t &= (r_{j_1}(0)=28, \tau_{j_1}(0)=0)(r_{j_1}(1)=27, \tau_{j_1}(1)=\delta t) \\ &\quad (r_{j_1}(2)=24, \tau_{j_1}(2)=2\delta t)(r_{j_1}(3)=22, \tau_{j_1}(3)=3\delta t), \\ r_{j_2}^t &= (r_{j_2}(0)=2, \tau_{j_2}(0)=0)(r_{j_2}(1)=13, \tau_{j_2}(1)=\delta t) \\ &\quad (r_{j_2}(2)=5, \tau_{j_2}(2)=2\delta t)(r_{j_2}(3)=9, \tau_{j_2}(3)=3\delta t). \end{aligned}$$

It can be observed that $r_i^t \models (\varphi_i = \Diamond_{[0,6]}\{\text{yellow}\})$ if $3\delta t \in [0,6]$, $r_{j_1}^t \models (\varphi_{j1} = \Diamond_{[3,10]}\{\text{red}\})$ if $2\delta t \in [3,10]$ and $r_{j_2}^t \models (\varphi_{j2} = \Diamond_{[3,9]}\{\text{blue}\})$ if $3\delta t \in [3,9]$.

6.5.4 Controller Synthesis

The proposed controller synthesis procedure is described with the following steps.

- **Step 1**: N TBAs $\mathcal{A}_i,\ i \in \mathcal{V}$ that accept all the timed runs satisfying the corresponding specification formulas φ_i are constructed;
- **Step 2**: A Büchi WTS $\widetilde{\mathcal{T}}_i = \mathcal{T}_i \otimes \mathcal{A}_i$ for every $i \in \mathcal{V}$ is constructed according to Definition 5.5. The accepting runs of $\widetilde{\mathcal{T}}_i$, are the individual runs of the $\mathcal{T}_i$ that satisfy the corresponding MITL formula φ_i;
- **Step 3**: We pick a set of accepting runs $\{\widetilde{r}_1^t, \ldots, \widetilde{r}_N^t\}$ from Step 2. We check if they are consistent according to Definition 6.4. If this is true then we proceed with Step 5. If this is not true then Step 3 is repeated with a different set of accepting runs. At worst case, we perform a finite predefined number of selections R_{selec}; if a consistent set of accepting runs is not found, we proceed with the less efficient centralized procedure in Step 4, which however searches through all sets of all possible accepting runs;
- **Step 4**: The product $\widetilde{\mathcal{T}}_p = \mathcal{T}_p \otimes \mathcal{A}_p$ where $\mathcal{A}_p$ is the TBA that accepts all the words that satisfy the formula $\varphi = \varphi_1 \wedge \ldots \wedge \varphi_N$ is created. An accepting run $\widetilde{r}_p$ of the product is projected into the accepting runs $\{\widetilde{r}_1, \ldots, \widetilde{r}_N\}$;
- **Step 5**: The abstraction procedure allows to find an explicit feedback law for each transition in $\mathcal{T}_i$. Therefore, an accepting run $\widetilde{r}_i^t$ in $\mathcal{T}_i$ that takes the form of a sequence of transitions is realized in the system in (6.2) via the corresponding sequence of feedback laws.

The proposed framework is depicted in Fig. 6.6. The dynamics (6.2) of each agent i is abstracted into a WTS $\mathcal{T}_i$ (orange rectangles). Then the product between each WTS $\mathcal{T}_i$ and the TBA $\mathcal{A}_i$ is computed according to Definition 5.5. The TBA $\mathcal{A}_i$ accepts all the words that satisfy the formula φ_i (blue rectangles). For every Büchi WTS $\widetilde{\mathcal{T}}_i$ the controller synthesis procedure that was described in this section (red rectangles) is performed and a sequence of accepted runs $\{\widetilde{r}_1^t, \ldots, \widetilde{r}_N^t\}$ is designed. Every accepted run $\widetilde{r}_i^t$ maps into a decentralized controller $u_i(t)$ which is a solution to Problem 6.1.

Proposition 3. A solution obtained from Steps 1-5, gives a sequence of controllers $u_1, \ldots, u_N$ that guarantees the satisfaction of the formulas $\varphi_1, \ldots, \varphi_N$ of the agents $1, \ldots, N$ respectively, governed by the dynamics as in (6.2), thus, they are a solution to Problem 6.1.

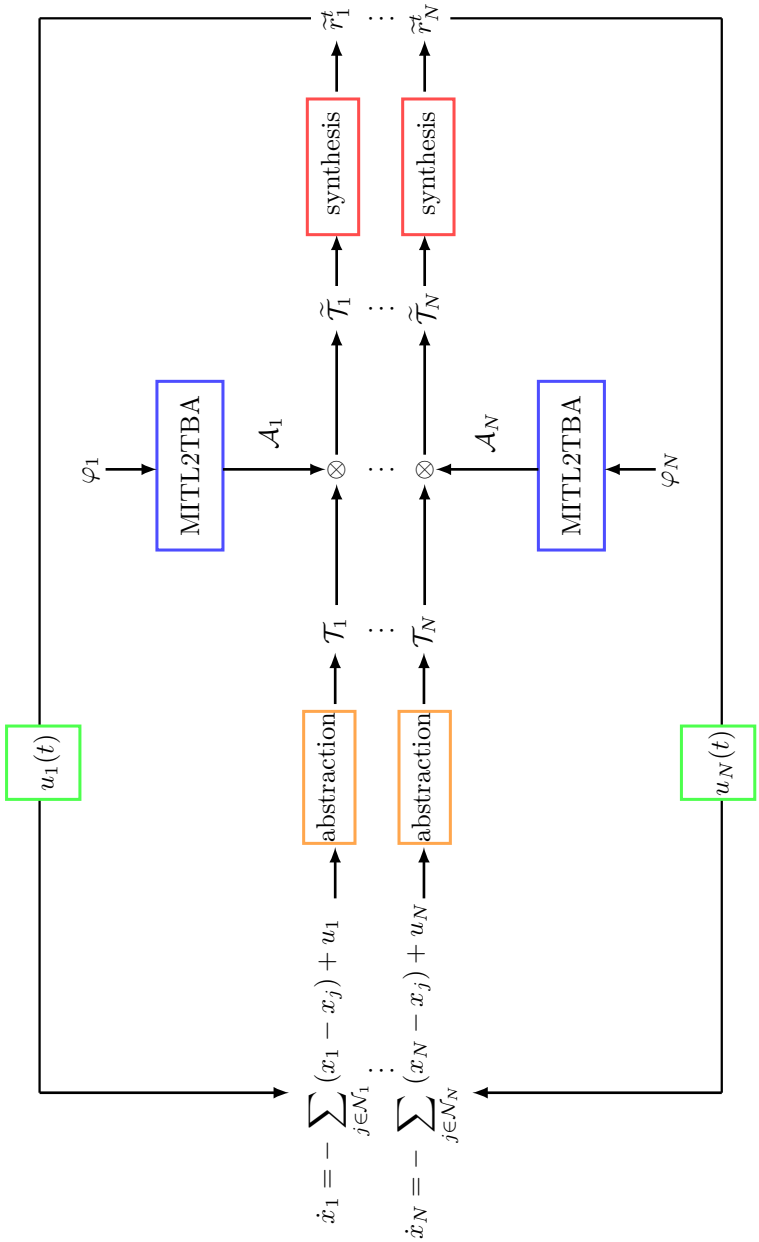

Figure 6.6: A graphic illustration of the proposed framework.

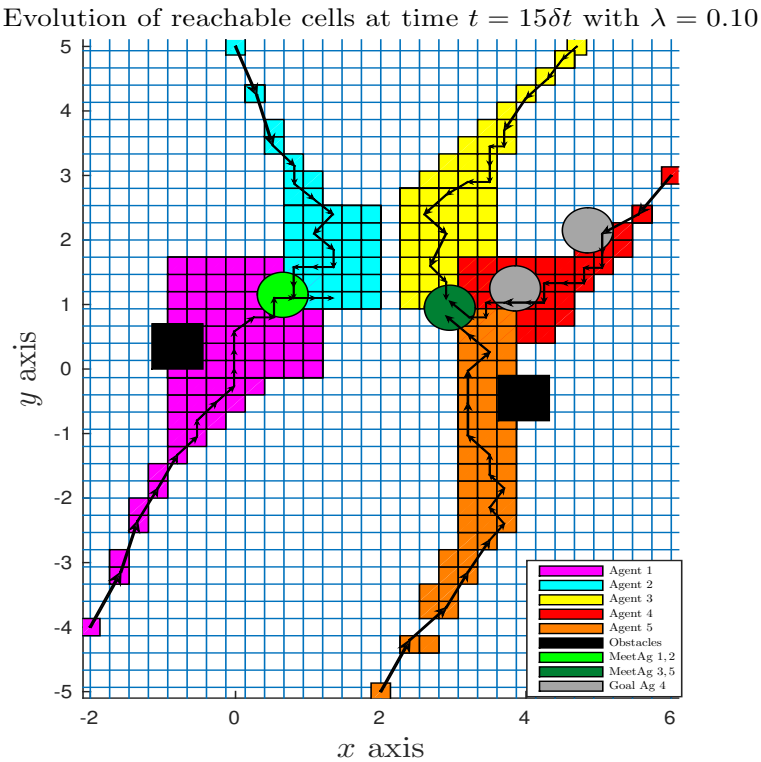

Figure 6.7: A simulation scenario with $N = 5$ agents and $\lambda = 0.10$, $\bar{d}_{\max} = 0.25$. The figure shows the evolution of the agents' reachable cells up to time $t = 15\delta t$.

6.6 Simulation Results

For the simulation example, a multi-agent system with $x_i \in \mathbb{R}^2$, $i \in \mathcal{V} = \{1, 2, 3, 4, 5\}$, $\mathcal{N}_1 = \{2\}, \mathcal{N}_2 = \{1, 3\}$, $\mathcal{N}_3 = \{2, 4\}$, $\mathcal{N}_4 = \{3, 5\}$ and $\mathcal{N}_5 = \{4\}$ is considered. According to (6.2), the dynamics are given as:

$$\begin{aligned}
\dot{x}_1 &= -(x_1 - x_2) + u_1, \\
\dot{x}_2 &= -(x_2 - x_1) - (x_2 - x_3) + u_2, \\
\dot{x}_3 &= -(x_3 - x_2) - (x_3 - x_4) + u_3, \\
\dot{x}_4 &= -(x_4 - x_3) - (x_4 - x_5) + u_4, \\
\dot{x}_5 &= -(x_5 - x_4) + u_5,
\end{aligned}$$

The simulation parameters are set to $\delta t = 0.1$, $\lambda = 0.10$, and $\bar{d}_{\max} = 0.25$, with $L_1 = \sqrt{2}$ and $L_2 = 2$ obtained from the agents' dynamics. The workspace is decomposed into square cells, which are depicted with blue color in Fig. 6.7. The initial agents' positions are set to $(-2, 4)$, $(0, 5)$, $(4.7, 5)$, $(6, 3)$ and $(4, -5)$. The

Step	Reach. States	Time	Step	Reach. States	Time
δt	4	0.03 sec	$9\delta t$	4960	27.01 sec
$2\delta t$	16	0.05 sec	$10\delta t$	10822	68.79 sec
$3\delta t$	24	0.05 sec	$11\delta t$	20706	102.46 sec
$4\delta t$	40	0.06 sec	$12\delta t$	34856	153.95 sec
$5\delta t$	126	0.09 sec	$13\delta t$	59082	194.68 sec
$6\delta t$	378	1.59 sec	$14\delta t$	69060	220.12 sec
$7\delta t$	1074	3.51 sec	$15\delta t$	74546	322.61 sec
$8\delta t$	2908	10.06 sec		Total Time: 1105 sec	

Table 6.1: This table shows the simulation statistics. Columns 1, 4 show the considered time steps. Columns 2, 5 show the corresponding number of states reachable in the WTS $\mathcal{T}_p$. Columns 3, 6 indicate the time required for their computation.

specification formulas are set to:

$$
\begin{aligned}
\varphi_1 &= \Diamond_{[1.2,1.7]} \{\text{meet}_{12}\} \wedge \Box_{[0,2]} \{\neg \text{obs}_1\}, \\
\varphi_2 &= \Diamond_{[1.2,1.7]} \{\text{meet}_{21}\}, \\
\varphi_3 &= \Diamond_{[1.5,1.8]} \{\text{meet}_{35}\}, \\
\varphi_4 &= \Diamond_{[0.1,0.5]} \{\text{grey}_1\} \wedge \Diamond_{[0.7,1.2]} \{\text{grey}_2\}, \\
\varphi_5 &= \Diamond_{[1.2,1.8]} \{\text{meet}_{53}\} \wedge \Box_{[0,2]} \{\neg \text{obs}_2\},
\end{aligned}
$$

respectively. Descriptively, agents 1, 2 as well as agents 3, 5 need to meet in the light and dark green region, respectively, within certain time bounds; agents 1 and 5 have additional safety specifications; and agent 4 has two reachability goals within certain time bounds (deadlines). The cell decomposition presented in this chapter, the reachable cells of each agent up to time $t = 15\delta t$ and the goal regions are depicted in Fig. 6.7. The reachable cells of each agent are depicted with purple, cyan, yellow, red and orange, respectively. The individual consistent runs $\widetilde{r}_1^t$, $\widetilde{r}_2^t$, $\widetilde{r}_3^t$, $\widetilde{r}_4^t$ and $\widetilde{r}_5^t$ of agents 1, 2, 3, 4 and 5, respectively, that satisfy the formulas φ_1, φ_2, φ_3, φ_4 and φ_5, respectively are depicted in Fig. 6.7 with black arrows. Each arrow represents a transition from a state to another according to Definition 6.2. Table 6.1 shows the simulation statistics of the simulation scenario. For each time step δt, $2\delta t$, ..., $15\delta t$ the number of reachable states of the product WTS $\mathcal{T}_p$ is mentioned, along with the necessary computation time.

6.7 Conclusions

A systematic method for controller synthesis of dynamically coupled multi-agent path-planning has been proposed, in which timed constraints of fulfilling a high-level specification are imposed to the system. The solution involves a boundedness analysis, the abstraction of each agent's motion into WTSs, TBAs as well as Büchi WTSs construction. The simulation example demonstrates our solution approach.

Chapter 7

Scalable Time-constrained Planning of Multi-robot Systems

This chapter presents a scalable procedure for time-constrained planning of a class of nonlinear multi-robot systems. In particular, we consider N robotic agents operating in a workspace which contains Regions of Interest (RoI), in which atomic propositions for each robot are assigned. The main goal is to design decentralized and robust control laws so that each robot meets an individual high-level specification given as a Metric Interval Temporal Logic (MITL), while using only local information based on a limited sensing radius. Furthermore, the robots need to fulfill certain desired transient constraints such as collision avoidance between them. The controllers, which guarantee the transition between RoI, are the solution of a Decentralized Finite-Horizon Optimal Control Problem (DFHOCP) and serve as actions for the individual Weighted Transition System (WTS) of each robot. The time duration required for the transition between RoI is modeled by a weight. The DFHOCP is solved at every sampling time step by each robot and then necessary information is exchanged between neighboring robots. The proposed approach is scalable, since it does not require a product computation among the WTS of the robots. The proposed framework is experimentally tested and the results show that the approach is promising for solving real-life robotic as well as industrial applications.

7.1 Introduction

As it has been presented in Chapter 5 and Chapter 6, a suitable temporal logic for dealing with tasks that are required to be completed within certain time bounds is the Metric Interval Temporal Logic (MITL). MITL has been used in control synthesis frameworks in Chapter 5 and Chapter 6. Given a robot dynamics and an MITL formula, the control design procedure is the following: first, the robot dynamics are abstracted into a Weighted Transition System (WTS), in which the time duration for navigating between states is modeled by a weight in the WTS (abstraction); second, an offline product between the WTS and an automaton that accepts the runs that satisfy the given formula is computed; and third, once an accepting run in the product is found, it maps into a sequence of feedback control laws of the robot dynamics.

Controller synthesis for multi-robot systems under MITL specifications has been investigated in [22, 24, 159]. In Chapter 6 as well as in [22, 24], the under consideration dynamics were first order and we considered actuation over each state of each agent. Additionally, a global product WTS of the individual WTSs of each robot was computed. Furthermore, no transient constraints between the agents are taken into consideration. In particular, the work [24] can only handle multi-agent timed constrained planning in $\mathbb{R}^2$ dimension, which is usually not the case in real-life robotic applications. Authors in [159] have addressed the Vehicle Routing Problem (VHP), which is modeled as an optimization problem, that aims at finding the optimal set of routes for a fleet of vehicles to traverse, in order to deliver the load to a given set of customers. However, the dynamics of the agents were not taken into consideration. Moreover, issues such as control input saturation were not considered. In the same context, none of the aforementioned works deal with real-time experimental validation of the corresponding proposed frameworks.

Motivated by the aforementioned, in this chapter, we aim to address the latter issues. In particular, we deal with second-order nonlinear dynamics in $\mathbb{R}^n$ with input constraints. Then, by assigning an MITL formula to each robot, we provide decentralized feedback control laws that guarantee transitions between RoI under transient constraints among the robots. The controllers are the solution of a Decentralized Finite-Horizon Optimal Control Problem (DFHOCP). Furthermore, an algorithm that computes the runs of each agent that in turn map into continuous control laws and provably satisfy the MITL formulas is provided. These control laws correspond to the transitions indicated above. The proposed scheme is experimentally validated in our lab facilities with a group of Nexus robots (see Fig. 7.3). Moreover, the proposed approach is scalable since does not require a product computation among the WTS of the agents.

The idea of avoiding the global product between the robots lies in the fact that we address the multi-robot coupling with the low-level continuous time control design. More specifically, we exploit the inherent advantage of NMPC with reference to other control techniques: we capture the coupling constraints through the hard constraints of each agent by assuming communication capabilities between the agents. In the same context, all the algorithmic computations are performed offline and the robots are executing online only a sequence of a control actions that are the outcome of the planner. Thus, the latter leads to a framework that is scalable with the number of robots. Our contribution is thus a fully automated framework for a general class of uncertain multi-robot systems consisting of both constructing purely decentralized abstractions and conducting timed temporal logic planning with transient constraints in a scalable way.

This chapter is structured as follows. Section 7.2 provides the modeling of the proposed framework along with the formal problem statement. Section 7.3 discusses the technical details of the solution, while Section 7.4 is devoted to a real-time experiment demonstrating the proposed approach. Finally, the summary of this chapter is discussed in Section 7.5.

7.2 Problem Formulation

7.2.1 System Model

Consider a team of N robots with labels $\mathcal{V} \coloneqq \{1, \dots, N\}$ operating in a bounded workspace $\mathcal{W} \subseteq \mathbb{R}^n$. The robots are governed by the following kinematics and dynamics:

$$\dot{x}_i = J_i(x_i)v_i, \tag{7.1a}$$
$$\dot{v}_i = f_i(x_i, v_i) + G_i(x_i)u_i, \tag{7.1b}$$

where x_i, $v_i \in \mathbb{R}^n$ stands for the position/orientation and the linear/angular velocity of the robot $i \in \mathcal{V}$, respectively; $J_i : \mathbb{R}^n \to \mathbb{R}^{n\times n}$ is the Jacobian transformation matrix; $f_i : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}^n$, $G_i : \mathbb{R}^n \to \mathbb{R}^n \times \mathbb{R}^n$ are known and continuously differentiable vector fields; and $u_i \in \mathbb{R}^n$ stands for the control input vector. Consider also velocity constraints and input constraints:

$$v_i \in \mathcal{V}_i \coloneqq \{v_i \in \mathbb{R}^n : \|v_i\|_2 \leq \widetilde{v}_i\},$$
$$u_i \in \mathcal{U}_i \coloneqq \{u_i \in \mathbb{R}^n : \|u_i\|_2 \leq \widetilde{u}_i\},$$

where the constants $\widetilde{v}_i$, $\widetilde{u}_i$ are *a priori* given.

Assumption 7.1. The linear systems $\dot{\overline{\eta}}_i = A_i\overline{\eta}_i + B_i\overline{u}_i$, where $\overline{\eta}_i \coloneqq [\overline{x}_i^\top, \overline{v}_i^\top]^\top \in \mathbb{R}^{2n}$, that are the outcome of the Jacobian linearization of the nominal dynamics (7.1a)-(7.1b) around the equilibrium states $(x_i, v_i) = (0_{n\times 1}, 0_{n\times 1})$ are stabilizable.

Remark 7.1. Assumption 7.1 is required for the NMPC nominal stability to be guaranteed [58].

In the given workspace, there exist $Z \in \mathbb{N}$ disjoint Regions of Interest (RoI) labeled by $\mathcal{Z} \coloneqq \{1, \ldots, Z\}$. We assume that the RoI are modeled by balls, i.e., $\mathcal{R}_z \coloneqq \mathcal{B}(y_z, p_z)$, $z \in \mathcal{Z}$, where y_z and $p_z > 0$ stand for the center and radius of RoI $\mathcal{R}_z$, respectively. Define also the union of RoI by:

$$\mathcal{R} \coloneqq \bigcup_{z\in\mathcal{Z}} \mathcal{R}_z.$$

Due to the fact that we are interested in imposing safety constraints, at each time $t \geq 0$, the robot i is occupying a ball $\mathcal{B}(x_i(t), \mathfrak{r}_i)$ that covers its volume, where $x_i(t)$ and $\mathfrak{r}_i > 0$ are its center and radius, respectively. Moreover, in order to be able to impose transient constraints among the robots, we assume that each robot $i \in \mathcal{V}$ has communication capabilities within a limited sensing range $\mathfrak{d}_i > 0$ such that:

$$\mathfrak{d}_i > \max_{i,j\in\mathcal{V}, i\neq j} \{\mathfrak{r}_i + \mathfrak{r}_j\}, \tag{7.2}$$

The latter implies that each agent has sufficiently large sensing radius so as to measure the agent with the biggest volume, due to the fact that the agents' radii are not the same. Define the set of robots j that are within the sensing range of agent i at time t as:

$$\mathcal{N}_i(t) \coloneqq \{j \in \mathcal{V}\backslash\{i\} : \|x_i(t) - x_j(t)\|_2 < \mathfrak{d}_i\}. \tag{7.3}$$

7.2.2 Objectives

The goal of this chapter is to design decentralized feedback control laws that steer the robots with dynamics as in (7.1a)-(7.1b) between RoI so that they obey individual high-level tasks given in MITL under transient constraints between them. Define the labeling functions:

$$\mathcal{L}_i : \mathcal{R} \to 2^{\Sigma_i}, \tag{7.4}$$

which map each RoI with a subset of atomic propositions that hold true there. Note that some of the RoI may be assigned with labels that indicate *unsafe regions*, i.e., the robot is required to avoid visiting them (*safety specifications*).

Definition 7.1. A trajectory $x_i(t)$ of robot $i \in \mathcal{V}$ is *associated with a timed run*:

$$r_i^t = (r_i(0), \tau_i(0))(r_i(1), \tau_i(1))(r_i(2), \tau_i(2)) \dots,$$

where $r_i(l) \in \mathcal{R}$, $\forall l \in \mathbb{N}$, is a sequence of RoI that the robot crosses, if the following hold:

1. $\tau_i(0) = 0$, i.e., the robot starts the motion at time $t = 0$;
2. $\mathcal{B}(x_i(\tau_i(0)), \mathfrak{r}_i) \subsetneq r_i(0)$, i.e., initially, the volume of the robot is entirely within the RoI $r_i(0) \in \mathcal{R}$;
3. $\mathcal{B}(x_i(\tau_i(l)), \mathfrak{r}_i) \subsetneq r_i(l)$, $\forall l \in \mathbb{N}$, i.e., the robot changes discrete state *only when* its entire volume is contained in the corresponding RoI;
4. $\tau_i(l+1) \coloneqq \tau_i(l) + \mathfrak{t}_i(r_i(l), r_i(l+1))$, $\forall l \in \mathbb{N}$, where:

$$\mathfrak{t}_i : \mathcal{R} \times \mathcal{R} \to \mathbb{Q}_+, \tag{7.5}$$

are functions that model the duration that the robot needs to be driven between regions $r_i(l)$ and $r_i(l+1)$.

Definition 7.2. A trajectory $x_i(t)$ *satisfies* an MITL formula φ_i over the set of atomic propositions Σ_i, formally written as $x_i(t) \models \varphi_i$, $\forall t \geq 0$, if and only if there exists a timed run r_i^t to which the trajectory $x_i(t)$ is associated, according to Definition 7.1, which satisfies φ_i. For the semantics of MITL we refer to Definition 2.14.

7.2.3 Problem Statement

The problem considered in this chapter is stated as follows.

Problem 7.1. Consider N robots governed by dynamics (7.1a)-(7.1b), covered by the balls $\mathcal{B}(x_i(t), \mathfrak{r}_i)$, operating in the workspace $\mathcal{W} \subseteq \mathbb{R}^n$ with sensing communication capabilities captured by the sets $\mathcal{N}_i$ as defined in (7.3). The workspace contains the RoI $\mathcal{R}_z$, $z \in \mathcal{Z}$ modeled also by balls. Given task specification formulas φ_i for each robot $i \in \mathcal{V}$ expressed in MITL over the set of atomic propositions Σ_i and labeling functions $\mathcal{L}_i$ as in (7.4); then, design decentralized feedback control laws $u_i = \kappa_i(x_i, v_i) \in \mathcal{U}_i$, such that the robot trajectories in the workspace fulfill the MITL specifications φ_i, i.e., $x_i(t) \models \varphi_i$, $\forall t \geq 0$, according to Definition 7.1, while collision avoidance constraints are imposed among the robots, i.e.:

$$\|x_i - x_j\|_2 > \mathfrak{r}_i + \mathfrak{r}_j, \quad \forall i \in \mathcal{V},\ j \in \mathcal{V} \backslash \{i\}.$$

Remark 7.2. Note that Problem 7.1 constitutes a general problem due to the fact that the dynamics (7.1a)-(7.1b) arise in most robotic applications and transient constraints among the robots are taken into consideration.

7.3 Problem Solution

In this section, a systematic framework for solving Problem 7.1 is provided as follows:

1. In Section 7.3.1, decentralized feedback control laws that guarantee the transition between RoI in the given environment are provided. The laws are the outcome of a DFHOCP solved at each time step.

2. Then, by using the outcome of Section 7.3.1, we abstract the dynamics (7.1a)-(7.1b) into a WTS for each robot, exploiting the fact that the timed runs in the WTS project onto associated trajectories according to Definition 7.1. (Section 7.3.2)

3. By invoking ideas from Chapter 5, a controller synthesis procedure that provides a sequence of control laws that serve as solution to Problem 7.1 is consulted. (Section 7.3.3)

4. Lastly, the computational complexity of the proposed framework is discussed in Section 7.3.4.

7.3.1 Decentralized Feedback Control Design

Consider the robot i with with dynamics (7.1a)-(7.1b) occupying a RoI $\mathcal{R}_{i,s} \in \mathcal{R}$ at time $\mathfrak{t}_{i,s} \geq 0$. The decentralized feedback control should guarantee that the robot is navigated towards a desired RoI $\mathcal{R}_{i,d} \in \mathcal{R}$, $\mathcal{R}_{i,d} \neq \mathcal{R}_{i,s}$ without intersection with any other RoI or other agents $j \in \mathcal{V}$, $j \neq i$. Denote by $y_{i,d} \in \mathcal{R}_{i,d}$ the center of the RoI $\mathcal{R}_{i,s}$. Define the error vector:

$$e_i \coloneqq x_i - x_{i,d} \in \mathbb{R}^n, \quad i \in \mathcal{V},$$

as well as the *error kinematics/dynamics* by:

$$\dot{e}_i = J_i(e_i + x_{i,d})v_i, \tag{7.6a}$$
$$\dot{v}_i = f_i(e_i + x_{i,d}, v_i) + G_i(e_i + x_{i,d})u_i, \tag{7.6b}$$

Define the sets that capture the state constraints of each robot as:

$$\mathcal{E}_i \coloneqq \Big\{ e_i(t) \in \mathbb{R}^n : \|e_i(t) + x_{i,d} - e_j(t) - x_{j,d}\|_2 > \mathfrak{r}_i + \mathfrak{r}_j,\ \forall j \in \mathcal{N}_i(t), \\ \mathcal{B}(e_i(t) + x_{i,d}, \mathfrak{r}_i) \cap \{\mathcal{R} \backslash \{\mathcal{R}_{i,s}, \mathcal{R}_{i,d}\}\} = \emptyset \Big\}.$$

The first constraint captures the fact that the robots should not collide with each other; the latter one, captures the fact that each robot needs to be navigated from RoI $\mathcal{R}_{i,s}$ to RoI $\mathcal{R}_{i,d}$ without intersecting with any other RoI of the workspace due to the fact that we are interested in imposing safety specifications.

Assumption 7.2. It is assumed that:

$$\max_{i \in \mathcal{V}} \{\mathfrak{r}_i\} < \min_{z \in \mathcal{Z}} \{p_z\}. \tag{7.7}$$

More specifically, (7.7) states that the radius of the ball that covers the robot with the biggest volume, is smaller than the radius of any of the RoI of the workspace. As it is shown later, this assumption is required in order to compute the time that a robot needs to be navigated between the RoI in the workspace.

Consider a sequence of sampling time steps $\{t_k\}$, $k \in \mathbb{N}$, with a constant sampling period $0 < h < T$, where T stands for the finite prediction horizon. It holds that $t_{k+1} = t_k + h$, $\forall k \in \mathbb{N}$. It should be noted that both t_k and T are multiples of h. At every discrete sampling time step t_k a DFHOCP is solved by each robot $i \in \mathcal{V}$ as follows:

$$\min_{\overline{u}_i(\cdot)} \left\{ \|\overline{\xi}_i(t_k + T)\|^2_{P_i} + \int_{t_k}^{t_k+T} \Big[\|\overline{\xi}_i(\mathfrak{s})\|^2_{Q_i} + \|\overline{u}_i(\mathfrak{s})\|^2_{R_i} \Big] d\mathfrak{s} \right\} \tag{7.8a}$$

subject to:

$$\dot{\overline{\xi}}_i(\mathfrak{s}) = \mathfrak{f}_i(\overline{e}_i(\mathfrak{s}), \overline{v}_i(\mathfrak{s}), \overline{u}_i(\mathfrak{s})), \tag{7.8b}$$

$$\overline{\xi}_i(\mathfrak{s}) \in \mathcal{E}_i \times \mathscr{V}_i, \quad \overline{u}_i(\mathfrak{s}) \in \mathcal{U}_i, \quad \forall \mathfrak{s} \in [t_k, t_k + T], \tag{7.8c}$$

$$\overline{\xi}_i(t_k + T) \in \mathcal{F}_i. \tag{7.8d}$$

In the aforementioned optimal control problem we defined:

$$\overline{\xi}_i \coloneqq [\overline{e}_i, \overline{v}_i]^\top \in \mathbb{R}^{2n},$$

$$\mathfrak{f}_i(\overline{\xi}_i, \overline{u}_i) \coloneqq \begin{bmatrix} J_i(\overline{e}_i + x_{i,d})\overline{v}_i \\ f_i(\overline{e}_i + x_{i,d}, \overline{v}_i) + G_i(\overline{e}_i + x_{i,d})\overline{u}_i \end{bmatrix}.$$

The matrices Q_i, $P_i \in \mathbb{R}^{2n}$ and $R_i \in \mathbb{R}^n$ are positive definite weight matrices. The sets $\mathcal{F}_i$ stand for the terminal sets that are used to enforce the stability of the

nominal system (see [58] for more details).

Due to the fact that Problem 7.1 imposes transient constraints between the agents (collision avoidance) and the agents have communication capabilities within the sensing range $\mathfrak{d}_i$ as given in (7.2)-(7.3), we adopt here a decentralized procedure explained as follows. Assume that each agent knows its labeling number in the set $\mathcal{V}$. After each sampling time step t_k, $\forall k \geq 0$ that agent i solves its own DFHOCP and obtains the estimated open-loop trajectory $\overline{\xi}_i(\mathfrak{s})$, $\mathfrak{s} \in [t_k, t_k + T]$, it transmits it to all agents $j \in \mathcal{N}_i(t_k)$, $j \neq i$, i.e., to agents that are within agent i's sensing radius at time t_k. Then, agents' $j \in \mathcal{N}_i(t_k)$, $j \neq i$ hard constraints $\overline{\mathcal{E}}_j$ are updated by incorporating the predicted trajectory of agent i, i.e., $\overline{\xi}_i(\mathfrak{s})$, $\mathfrak{s} \in [t_k, t_k + T]$. Among all agents $j \in \mathcal{N}_i(t_k)$, the one with higher priority, i.e., smaller labeling number in the set $\mathcal{V}$, solves its own DFHOCP (for example, agent 2 has higher priority than agents 3, 4, ...). This *sequential procedure* is continued until all agents $i \in \mathcal{V}$ solve their own DFHOCP, and then the sampling time step is updated.

In other words, each time an agent solves its own individual optimization problem, it knows the (open-loop) state predictions that have been generated by the solution of the optimization problem of all agents within agent i's sensing range at that time, for the next T time units. These pieces of information are required, as each agent's trajectory is constrained not by constant values, but by the trajectories of its associated agents through time: at each solution time t_k and within the next T time units, an agent's predicted configuration at time $\mathfrak{s} \in [t_k, t_k + T]$ needs to be constrained by the predicted configuration of its neighboring and perceivable agents (agents within its sensing range) at the same time instant $\mathfrak{s}$, so that collisions are avoided. We assume that the above pieces of information are *always available*, *accurate* and can be exchanged without delay. By adopting the aforementioned sequential communication procedure, and given that at $t = 0$ the DFHOCP (7.8a) - (7.8d) of all agents are feasible, the agents are navigated to the desired RoI, while all distance and input constraints imposed by Problem 7.1 are satisfied.

Remark 7.3. It should be noted that the constraint sets $\overline{\mathcal{E}}_i$, $i \in \mathcal{V}$ in (7.8c) depend on the estimated open-loop trajectories $\overline{e}_i(\mathfrak{s})$ and $\overline{e}_j(\mathfrak{s})$ for all $i \in \mathcal{V}$, $j \in \mathcal{N}(t_k)$, with $\mathfrak{s} \in [t_k, t_k + T]$. Moreover, they are updated when each robot has received the transmitted trajectories by its neighbors.

Remark 7.4. By considering a real-time scenario where the state vector $\overline{\xi}$ is comprised of 12 real numbers encoded by 4 bytes the overall downstream bandwidth required by each robot is:

$$BW_d = 12 \times 32 \text{ [bits]} \times |\mathcal{N}_i(t_k)| \times \frac{T}{h} \times f \text{ [sec}^{-1}\text{]}.$$

Given a conservative sampling time $f = 100$ Hz and a horizon of $\frac{T}{h} = 100$ time steps, the wireless protocol IEEE 802.11n-2009 (a standard for present-day devices) can accommodate up to

$$|\mathcal{N}_i(t_k)| = \frac{600\ [\text{Mbit} \cdot \text{sec}^{-1}]}{12 \times 32[\text{bit}] \times 10^4[\text{sec}^{-1}]} \approx 16 \cdot 10^2 \text{ robots},$$

within the range of one robot.

The following theorem guarantees the navigation of the agents between RoI and thereafter we will propose algorithms computing the corresponding transition times.

Theorem 7.1. *Suppose that:*

1. *Assumption 7.1 and Assumption 7.2 hold;*
2. *the robots start at time* $\mathfrak{t}_{i,s} \geq 0$ *from the RoI* $\mathcal{R}_{i,s}$ *and they need to be navigated to RoI* $\mathcal{R}_{i,d}$ *for every* $i \in \mathcal{V}$*;*
3. *at time* $\mathfrak{t}_{i,s}$ *the DFHOCP* (7.8a)-(7.8d) *sequentially solved by all the robots* $i \in \mathcal{V}$*, is feasible.*

Then, under the decentralized feedback control law, which is the outcome of the solution of DFHOCP (7.8a)-(7.8d)*, the robots are driven to the corresponding desired RoI.*

Proof. The proof of the theorem consists of two parts: firstly, recursive feasibility is established, that is, initial feasibility is shown to imply subsequent feasibility; secondly, and based on the first part, it is shown that the state of each agent x_i converges to the desired state $x_{i,d}$, while all state and control input constraints are satisfied. The feasibility analysis and the convergence analysis follow similar arguments with nominal NMPC stability of single- and multi-agent systems (see the feasibility proof of Theorem 4.1 and [31, 60, 166] for more details). □

7.3.2 Discrete System Abstraction

Theorem 7.1 implies that for each robot $i \in \mathcal{V}$ with kinematics/dynamics as in (7.1a), (7.1b), starting from the RoI $\mathcal{R}_{i,s}$ at time $\mathfrak{t}_{i,s}$, is driven by the controller which is the outcome of (7.8a)-(7.8d) towards a desired RoI $\mathcal{R}_{i,d}$, while all state, input and transient constraints are satisfied. Hereafter, we provide an algorithm for constructing the WTS of each agent. By observing Theorem 7.1 and taking into account Assumption 7.2, it holds that there exists a time instant $\mathfrak{t}_{i,d}$ such that the volume of robot i will be included strictly within the RoI $\mathcal{R}_{i,d}$. Furthermore, due

Algorithm 2 Computation of $\mathfrak{t}_{i,d} \coloneqq \mathfrak{t}_i(\mathcal{R}_{i,s}, \mathcal{R}_{i,d})$

1: **Input**: $\mathfrak{t}_{i,s}$, $\overline{x}_i(t_k)$, $k \in \mathbb{N}$;
2: **Output**: $\mathfrak{t}_{i,d}$;
3: $t_k \leftarrow \mathfrak{t}_{i,s}$;
4: $y_{i,d} \leftarrow y_d$; {center of RoI $\mathcal{R}_{i,d}$}
5: flag $\leftarrow 1$;
6: **while** flag $= 1$; **do**
7: solve DFHOCP (7.8a)-(7.8d) with $\widetilde{\mathcal{E}}_i$ as in (7.9);
8: measure $\overline{x}_i(t_k)$;
9: **if** $\left\| (\|\overline{x}_i(t_k)\| + \mathfrak{r}_i) - x_{i,d} \right\|_2 < p_d$ **then**
10: flag $\leftarrow 0$; {robot i is whithin RoI $\mathcal{R}_{i,d}$}
11: break;
12: **Go to** "line 15"
13: **end if**
14: $t_k \leftarrow t_k + h$;
15: **end while**
16: $\mathfrak{t}_{i,d} \leftarrow t_k$;

to the fact that we have knowledge of the nominal dynamics and the MITL tasks φ_i are independent for each robot, for the computation of the time $\mathfrak{t}_{i,d}$ an offline computer simulation of the DFHOCP (7.8a)-(7.8d) with state constraints as:

$$\widetilde{\mathcal{E}}_i \coloneqq \Big\{ e_i(t) \in \mathbb{R}^n : \mathcal{B}(e_i(t) + x_{i,d}, \mathfrak{r}_i) \cap \{\mathcal{R} \backslash \{\mathcal{R}_{i,s}, \mathcal{R}_{i,d}\}\} \Big\}, \tag{7.9}$$

is conducted. In particular, (7.9) captures constraints regarding the navigation of robot i from RoI $\mathcal{R}_{i,s}$ to RoI $\mathcal{R}_{i,d}$ without intersecting with any other RoI of the workspace. It should be noted that if any collision is about to occur in real-time when the robots are executing the online control actions, the transition time between RoI of the workspace $\mathcal{W}$ will be different. In order to overcome the aforementioned issue, we provide thereafter an algorithm that monitors the collision offline and updates the transition times appropriately. Then, the process of computing $\mathfrak{t}_{i,d}$ is described in Algorithm 2. The abstraction that captures the dynamics of each robot into a WTS is given through the following definition.

Definition 7.3. The motion of robot i in the workspace $\mathcal{W}$ is modeled by the WTS

$$\mathcal{T}_i = (S_i, S_i^{\text{init}}, \text{Act}_i, \longrightarrow_i, \mathfrak{t}_i, \Sigma_i, \mathcal{L}_i),$$

where:

- $S_i = \mathcal{R} = \bigcup_{z \in \mathcal{Z}} \mathcal{R}_z$ is the set of states of the robot that contains all the RoI of the workspace $\mathcal{W}$;

- $S_i^{\text{init}} \subseteq S_i$ is a set of initial states defined by the robot' s initial position $x_i(0)$ in the workspace;
- Act_i is the set of actions containing the union of all feedback controllers which can navigate the robot i between RoI;
- $\longrightarrow_i \subseteq S_i \times \text{Act}_i \times S_i$ is the transition relation. We say that $(\mathcal{R}_{i,s}, u_i, \mathcal{R}_{i,d}) \in \longrightarrow_i$, with $\mathcal{R}_{i,s}$, $\mathcal{R}_{i,d} \in \mathcal{R}$ with $\mathcal{R}_{i,s} \neq \mathcal{R}_{i,d}$ if there exist a feedback control law $u_i \in \text{Act}_i$ which can drive the robot from the region $\mathcal{R}_{i,s}$ to the region $\mathcal{R}_{i,d}$ without intersecting with any other RoI of the workspace;
- $\mathfrak{t}_i$ is the time weight as given in (7.5) and it is computed by Algorithm 2;
- Σ_i is the set of atomic propositions imposed by Problem 7.1; and
- $\mathcal{L}_i$ is the labeling function as given in (7.4);

The aforementioned WTS of each robot allows us to work directly at the discrete level and design a sequence of feedback controllers that solve Problem 1. By construction, each timed run produced by the WTS $\mathcal{T}_i$, where the notion of timed run is given in Definition 2.12, is associated with the trajectory $x_i(t)$ of the system (7.1a)-(7.1b), as given in Definition 7.1. Hence, if a timed run of $\mathcal{T}_i$ of each robot $i \in \mathcal{V}$ satisfying the given MITL formula φ_i is found, a desired timed word of the original system, and hence a trajectory $x_i(t)$ that is a solution to Problem 7.1 is found.

7.3.3 Control Synthesis

Fig. 7.1 depicts a framework under which a sequence of feedback control laws $u_i(x_i, v_i)$ of each robot that guarantees the satisfaction of the MITL formula φ_i can be computed. First, a TBA $\mathcal{A}_i$ that accepts all the timed runs satisfying the specification formula φ_i is constructed. Second, a product between the WTS $\mathcal{T}_i$ given in Definition 7.3 and the TBA $\mathcal{A}_i$ is computed which gives the product WTS $\widetilde{\mathcal{T}}_i$. By performing graph search to the product WTS $\widetilde{\mathcal{T}}_i$, a timed run that satisfies the MITL formula φ can be found.

In view of (7.9) and the offline plan computation above, it is possible that while each agent is executing online its individual actions and transits between RoI, there might be a cluster of agents that collide with each other. In such a scenario, the online feedback control law avoids the possible collisions, but the navigation time between the RoI will have been different that the one computed by Algorithm 2. In order to resolve this, we propose an offline collision detection algorithm (see Algorithm 3) which detects the cluster of agent that will avoid potential collision when the plan of each agent is executed and updates the transition times between RoI of each agent appropriately.

Algorithm 3 Offline collision detection and formula bounds relaxation

Input: $\mathfrak{t}_{i,s}$, φ_i, $i \in \mathcal{V}$;
Output: $\mathfrak{t}^{\text{real}}_{i,d}$, $\overline{\varphi}_i$, $i \in \mathcal{V}$;
$\max_i \leftarrow 0$;
for $i \in \mathcal{V}$ **do**
  cumputePlanAgent(i); {Execute the simulated plan of each agent}
  executePlanAgent(i);
end for
for $i \in$ collisionClusterMonitoring **do**
  $T^i_{\text{maneuver}} \leftarrow$ computeManueverTime;
  if $T^i_{\text{maneuver}} > \max_i$ **then**
    $\max_i \leftarrow T^i_{\text{maneuver}}$
  end if
  $\mathfrak{t}^{\text{real}}_{i,d} \leftarrow \mathfrak{t}_{i,s} + T^i_{\text{maneuver}}$
end for
for $i \in \mathcal{V}$ **do**
  $\overline{\varphi}_i \leftarrow \text{relaxBounds}(\varphi_i, \max_i)$;
end for

More specifically, the input to Algorithm 3 is the transition times of each agent and the output is the updated realistic transition times denoted by $\mathfrak{t}^{\text{real}}_{i,d}$ as well as the formula bounds relaxation. The function cumputePlanAgent(i) computes the sequence of RoI that agent i needs to follow in order to satisfy the formula. The function executePlanAgent(i) executes a simulated plan for each agent. Then, by using a monitoring function

collisionClusterMonitoring,

the cluster of the agents that are colliding can be detected. Then, we need to update the transition times of each of the colliding agents by a term which models the time duration of the maneuvering that the corresponding agent is performing in order to avoid the collision. This time is denoted by T^i_{maneuver} in Algorithm 3. After computing the maximum of the aforementioned times, the time bounds of the MITL formula of each agent are relaxed. The function relaxBounds($\varphi_i, \max_i$) updates each formula time interval of the form $[a, b]$, $a > b \geq 0$, to $[a, b + \max_i]$.

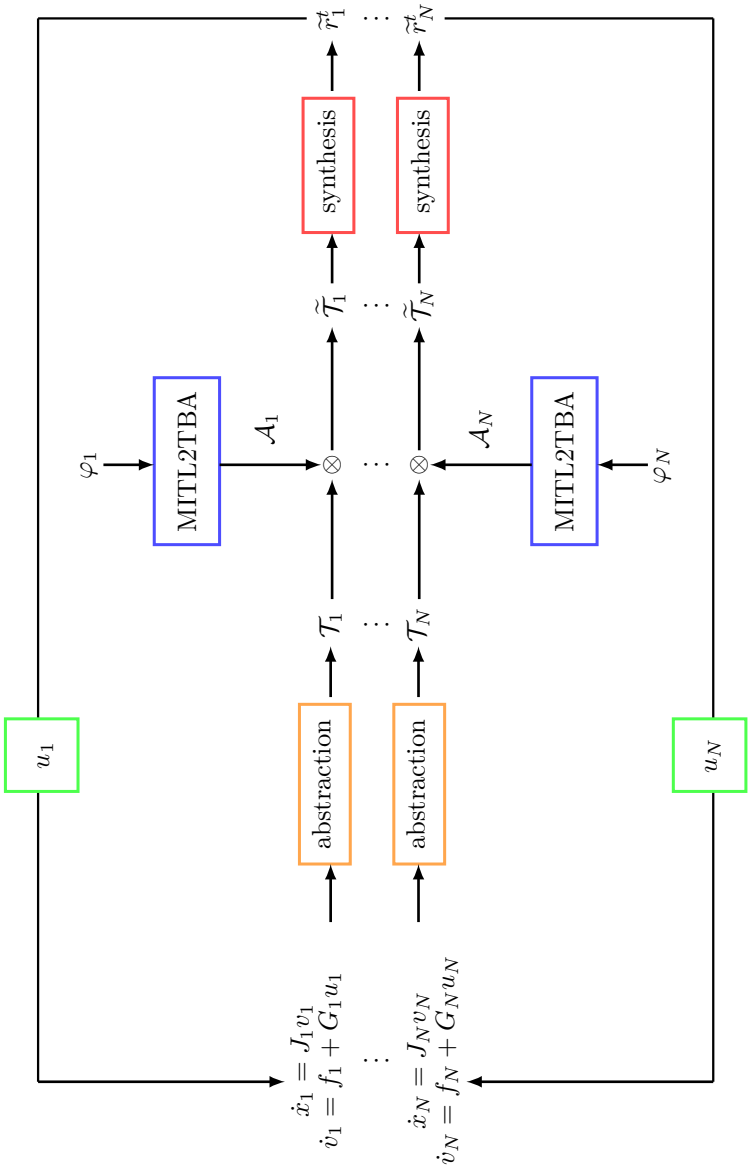

Figure 7.1: A graphic illustration of the proposed framework.

Proposition 4. The solution that it is obtained from the controller synthesis procedure of this section, provides a sequence of feedback control laws $u_i(x_i, v_i)$ that guarantees the satisfaction of the formula φ of the robot governed by dynamics as in (7.1a)-(7.1b), thus, providing a solution to Problem 7.1.

7.3.4 Complexity Analysis

The proposed framework consists of the computational complexity of the following steps:

- **C1:** the computational complexity of the offline construction of WTS $\widetilde{\mathcal{T}}_i$ and graph search which is decoupled among the agents and can be computed in parallel;
- **C2:** Algorithm 2 is an offline computer simulation and the computational complexity is the same with the complexity of a nominal NMPC algorithm;
- **C3:** Algorithm 3 is an offline computer simulation of collision detection which scales with the number of agents;
- **C4:** the DFHOCP (7.8a)-(7.8d) is the only online commutation of the proposed framework and has the same complexity with the nominal NMPC algorithm.

By taking into account that **C1** is standard in timed verification, the fact that **C2**, **C4** have the same complexity with nominal NMPC, and the fact that **C3** is a computer simulation that scales with the number of agents, the proposed approach is scalable with the number of agents.

7.4 Experimental Results

This section is devoted to an experimental validation of the problem addressed in this chapter. We demonstrate the efficacy of the proposed framework via a real-time experiment employing $N = 3$ Nexus 10011 mobile robots (Fig. 7.2). The experiment was conducted at Smart Mobility Lab (SML) (see Fig. 7.3 and [5]). The robots dimensions are $400 \times 360 \times 100$ mm and consist of 4 aluminum mecanuum wheels which provide omni-directional capabilities through the 3 degrees of freedom (moving forward/backward, left/right and rotation). By controlling the speed of each wheel, the Nexus Robot 10011 is able to move forward, backward, left, and right. The robot can also rotate clockwise and counterclockwise. By combining the three degree of freedom, the Nexus Robot is able to move towards any direction.

SML provides a motion capture system (MoCap) with 12 cameras spread across the lab. The MoCap provides the robot state vector, including pose, orientation as well as linear and angular velocities at frequency of 100Hz. The hardware used in experiments are connected using Gigabit Ethernet connections, USB connections, and wireless 5.8 Ghz connection. The Raspberry Pi on the robot is connected with a TP-link Router via wireless 5.8Ghz connections. The host computer as well as the MoCap are connected with the TP-Link router via Gigabit Ethernet connections. The host computer runs the node of the controller, which receives the measurement from MoCap, and calculates the control signal. The software implementation of the proposed control strategy was conducted in C++ under Robot Operating System (ROS) [167]. Moreover, the optimization algorithms described in this chapter are implemented by employing the NLopt Optimization library found in [168].

ROS is currently the most popular operating system for a large variety of robotic platforms. ROS is flexible framework for writing robot software. It is a collection of tools, libraries, and conventions that aim to simplify the task of creating complex and robust robot behavior across a wide variety of robotic platforms. It keeps a minimum and simple interfaces between different functional packages. More importantly, the real-time message passing and handling services of ROS allow multiple ROS packages to be running in parallel, which is extremely beneficial for real-time applications. A running ROS consists of a single ROS core and several running processes as ROS nodes. The ROS core provides a distributed computing environment allowing multiple processes to be running simultaneously. They communicate via ROS messaging services and can be running on different robots, computers and processing units even with different programming languages.

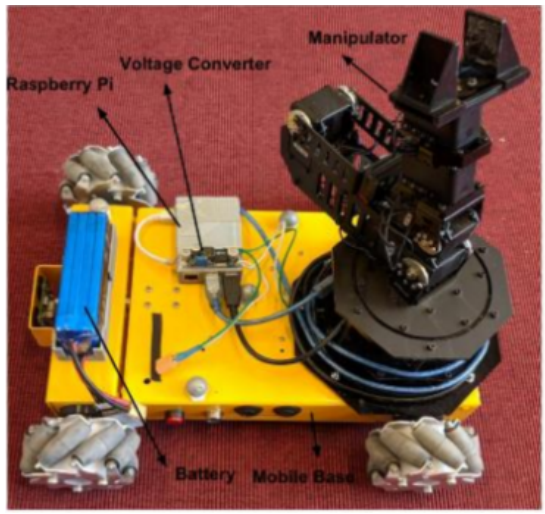

Figure 7.2: A Nexus 10011 mobile robot with an attached 3 Degrees of Freedom (DoF) manipulator.

The state of each robot is $x_i = [x_{i,1}, x_{i,2}, x_{i,3}]^\top$ where $x_{i,1}$, $x_{i,2}$ indicate the position of the robot and $x_{i,3}$ its orientation. The workspace that the robots can operate in as well as a panoramic view of it is depicted in Fig. 7.3 and Fig. 7.4, respectively. The workspace is captured by the set:

$$\mathcal{W} \coloneqq \{w \in \mathbb{R}^2 : |w_k| \leq 2.5, \quad k \in \{1, 2\}\},$$

and it contains 5 RoI which are divided as follows:

- the RoI $\mathcal{R}_z$, $z \in \{1, 2, 3, 4\}$ depicted with blue color in Fig. 7.4 which stand for the RoI that the robots are required to visit. The RoI $\mathcal{R}_z$, $z \in \{1, 2, 3, 4\}$ map into the atomic propositions that model missions for each robot;
- the RoI $\mathcal{R}_5$ depicted with red color in Fig. 7.4 stands for an unsafe region that the robots should avoid collision with. It holds that $\mathcal{L}_i(\mathcal{R}_5) = \{\text{obs}\}$ for every $i \in \mathcal{V}$.

The control input constraints of each robot are set to:

$$\mathcal{U}_i = \{u_i \in \mathbb{R}^3 : |u_{i,k}| \leq 0.15, \quad k \in \{1, 2, 3\}\}, \quad i \in \mathcal{V},$$

where $u_{i,1}$, $u_{i,2}$ stand for the linear velocities and $u_{i,3}$ stands for the angular velocity. The ball that cover the volume of each robot has radius $\mathfrak{r}_i = 0.4$m for every $i \in \mathcal{V}$. The sensing radius of each robot is $\mathfrak{d}_i = 2$m. The robots 1, 2 and 3 are initially

Figure 7.3: The experimental setup demonstrating the proposed framework. Three Nexus 10011 mobile robots, in the workspace of Smart Mobility Lab (SML) [5] that contains 5 RoI.

place in the ROI $\mathcal{R}_1$, $\mathcal{R}_2$ and $\mathcal{R}_3$, respectively. The set of atomic propositions of each robot is given by:

$$\begin{aligned}\Sigma_1 &= \{\text{obs}, \text{mission}_{11}, \text{mission}_{13}\},\\ \Sigma_2 &= \{\text{obs}, \text{mission}_{22}, \text{mission}_{24}\},\\ \Sigma_3 &= \{\text{obs}, \text{mission}_{33}, \text{mission}_{32}\},\end{aligned}$$

with the corresponding labeling functions:

$$\begin{aligned}&\mathcal{L}_1(\mathcal{R}_1) = \{\text{mission}_{11}\},\ \mathcal{L}_1(\mathcal{R}_2) = \emptyset,\\ &\mathcal{L}_1(\mathcal{R}_3) = \{\text{mission}_{13}\}, \mathcal{L}_1(\mathcal{R}_4) = \emptyset,\\ &\mathcal{L}_2(\mathcal{R}_1) = \emptyset,\ \mathcal{L}_2(\mathcal{R}_2) = \{\text{mission}_{22}\},\\ &\mathcal{L}_2(\mathcal{R}_3) = \emptyset, \mathcal{L}_2(\mathcal{R}_4) = \{\text{mission}_{24}\},\\ &\mathcal{L}_3(\mathcal{R}_1) = \emptyset,\ \mathcal{L}_3(\mathcal{R}_2) = \{\text{mission}_{32}\},\\ &\mathcal{L}_3(\mathcal{R}_3) = \{\text{mission}_{33}\},\ \mathcal{L}_3(\mathcal{R}_4) = \emptyset.\end{aligned}$$

The desired MITL tasks are set to:

$$\begin{aligned}\varphi_1 &= \Box_{[0,120]}\{\neg\text{obs}\} \wedge \Diamond_{[10,25]}\{\text{mission}_{13}\} \wedge \Diamond_{[30,45]}\{\text{mission}_{11}\},\\ \varphi_2 &= \Box_{[0,120]}\{\neg\text{obs}\} \wedge \Diamond_{[25,45]}\{\text{mission}_{22}\} \wedge \Diamond_{[50,80]}\{\text{mission}_{24}\},\\ \varphi_3 &= \Box_{[0,120]}\{\neg\text{obs}\} \wedge \Diamond_{[30,45]}\{\text{mission}_{33}\} \wedge \Diamond_{[60,75]}\{\text{mission}_{32}\},\end{aligned}$$

respectively. The prediction horizon is chosen $T = 2.0\,\text{sec}$. The NMPC weight matrices are set to:

$$Q_i = P_i = R_i = 0.5I_3, \quad i \in \mathcal{V}.$$

By using Algorithm 2 and Algorithm 3, the transition times of the navigation of the robots between the RoI of the workspace are computed as follows:

$$\begin{aligned} \mathfrak{t}_1(\mathcal{R}_1, \mathcal{R}_3) &= \mathfrak{t}_1(\mathcal{R}_3, \mathcal{R}_1) = 18, \\ \mathfrak{t}_2(\mathcal{R}_2, \mathcal{R}_4) &= \mathfrak{t}_2(\mathcal{R}_4, \mathcal{R}_2) = 20, \\ \mathfrak{t}_3(\mathcal{R}_2, \mathcal{R}_3) &= \mathfrak{t}_3(\mathcal{R}_3, \mathcal{R}_2) = 16. \end{aligned}$$

By using the proposed framework, we find a sequence of runs of each agent that fulfills the given MITL task. The sequence of runs maps into a sequence of feedback control laws that the robot execute online and fulfill the given tasks. By online executing the proposed plan, the trajectories of the robots in the workspace are depicted in Fig. 7.5- Fig. 7.7. The control inputs of each robot that guarantee the satisfaction of the given formulas are depicted in Fig. 7.8- Fig. 7.16.

Video: A video demonstrating the experiment of this chapter can be found in the following link:

https://www.youtube.com/watch?v=9ZNVlEjKZ9g

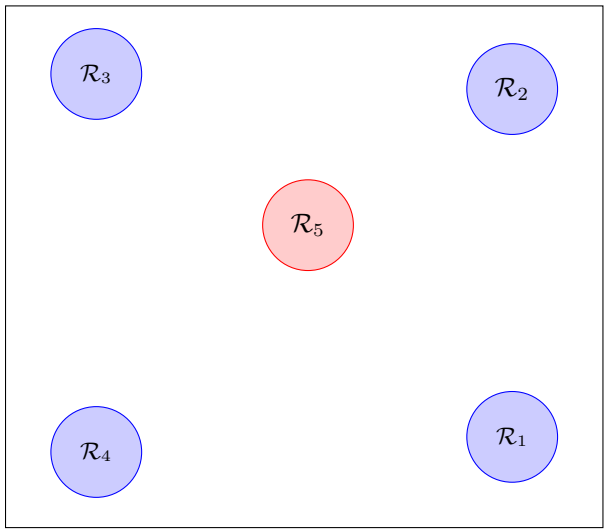

Figure 7.4: A panoramic view of the workspace with the 5 RoI.

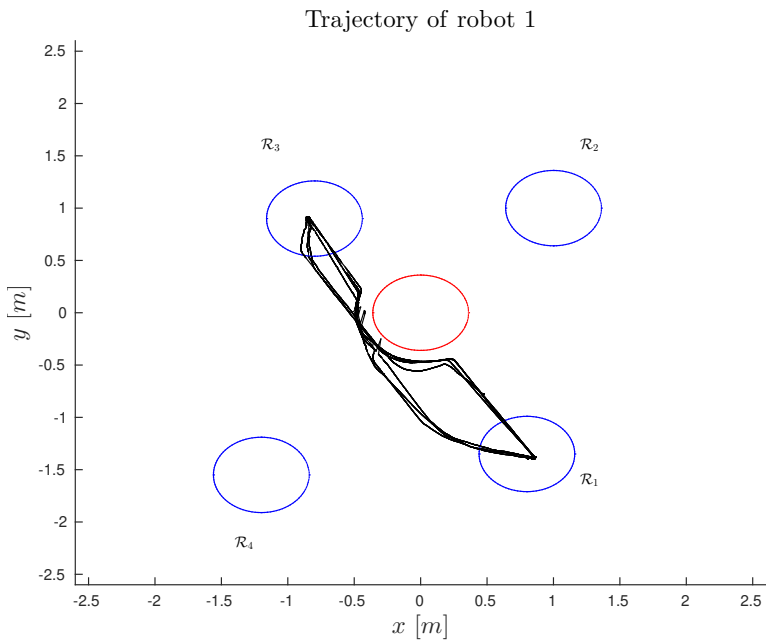

Figure 7.5: The trajectory of robot 1 in the workspace.

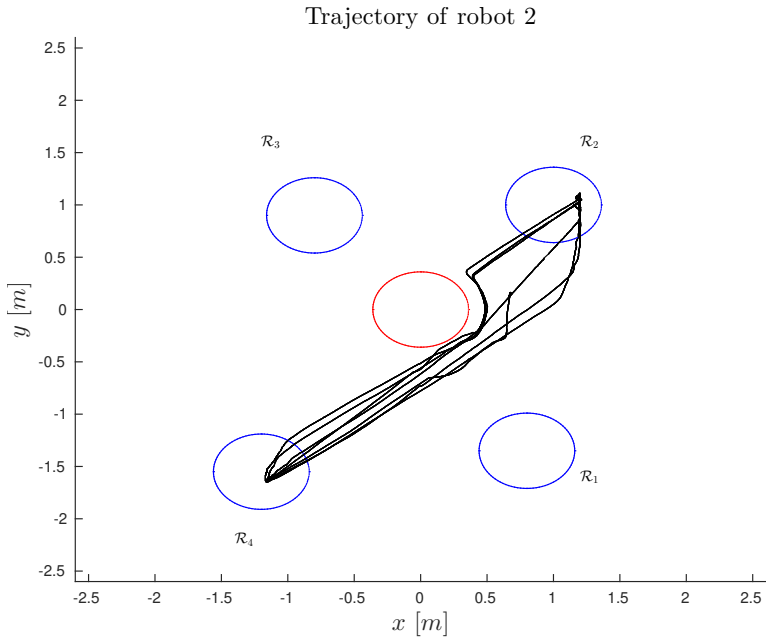

Figure 7.6: The trajectory of robot 2 in the workspace.

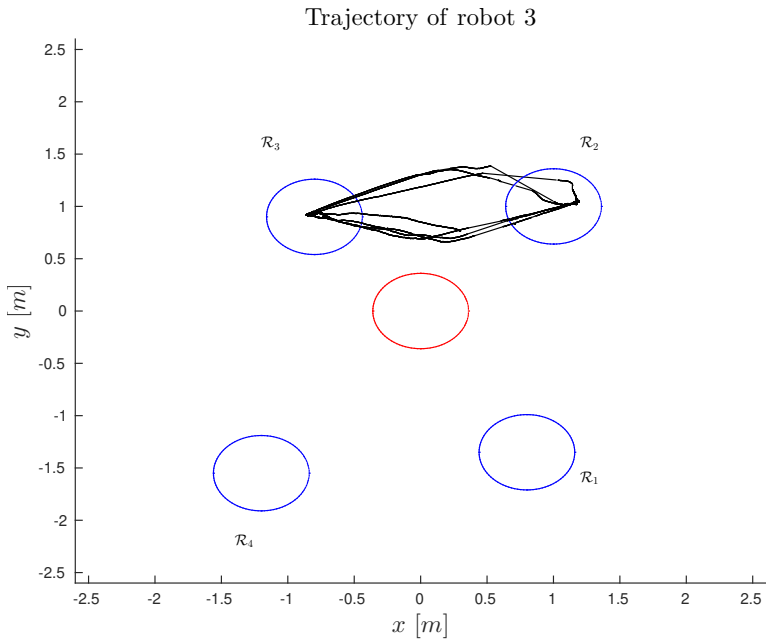

Figure 7.7: The trajectory of robot 3 in the workspace.

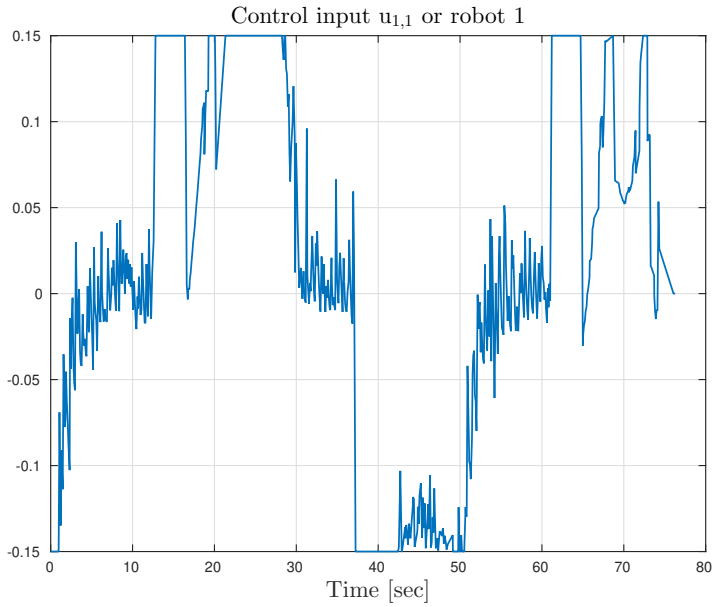

Figure 7.8: The evolution of the control signal $u_{1,1}$ or robot 1 over the time.

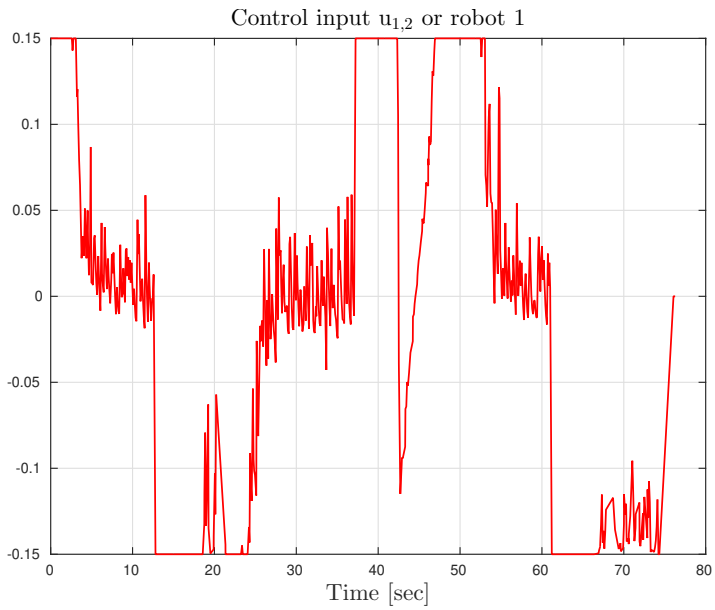

Figure 7.9: The evolution of the control signal $u_{1,2}$ or robot 1 over the time.

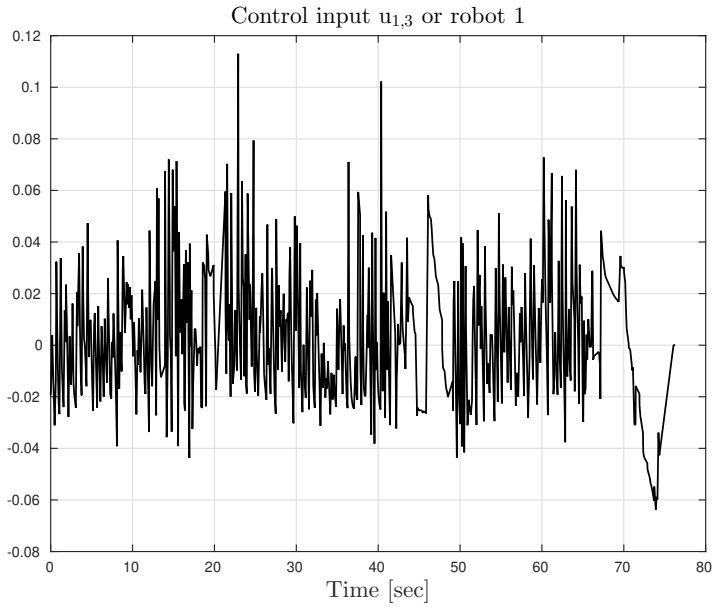

Figure 7.10: The evolution of the control signal $u_{1,3}$ or robot 1 over the time.

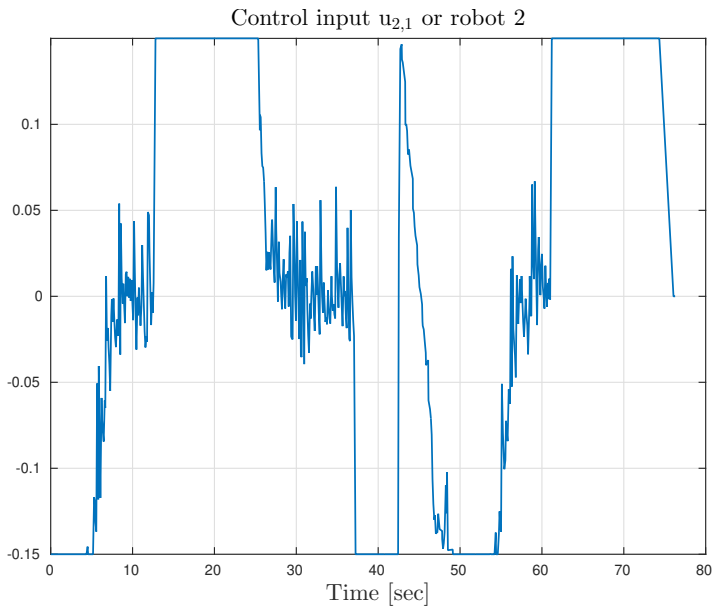

Figure 7.11: The evolution of the control signal $u_{2,1}$ or robot 2 over the time.

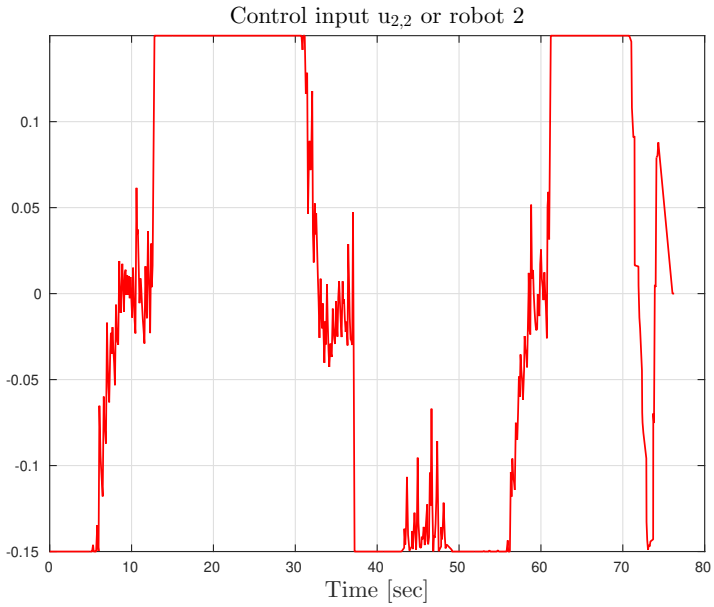

Figure 7.12: The evolution of the control signal $u_{2,2}$ or robot 2 over the time.

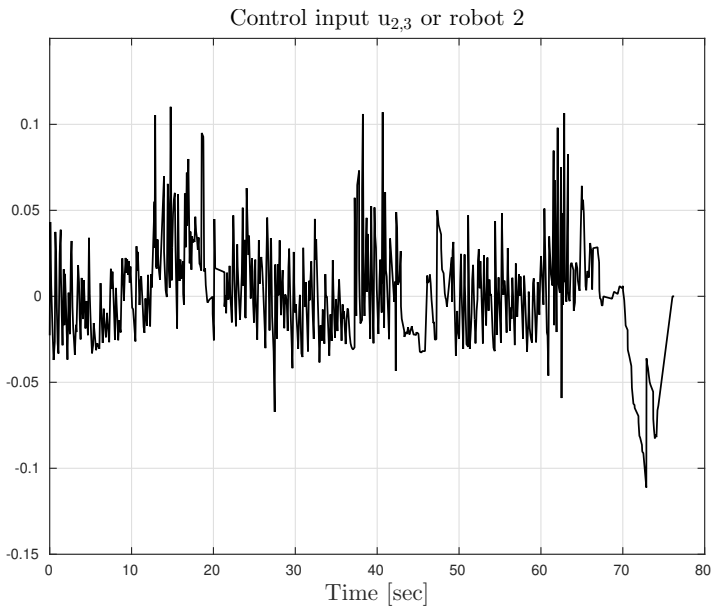

Figure 7.13: The evolution of the control signal $u_{2,3}$ or robot 2 over the time.

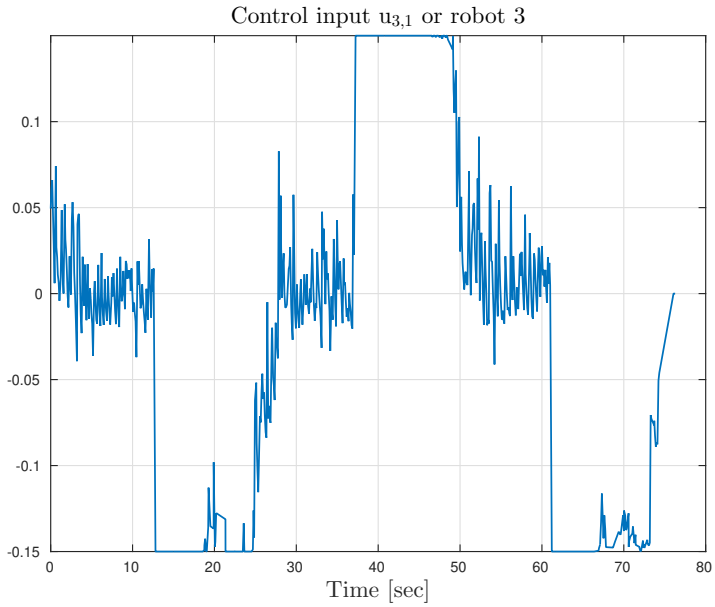

Figure 7.14: The evolution of the control signal $u_{3,1}$ or robot 3 over the time.

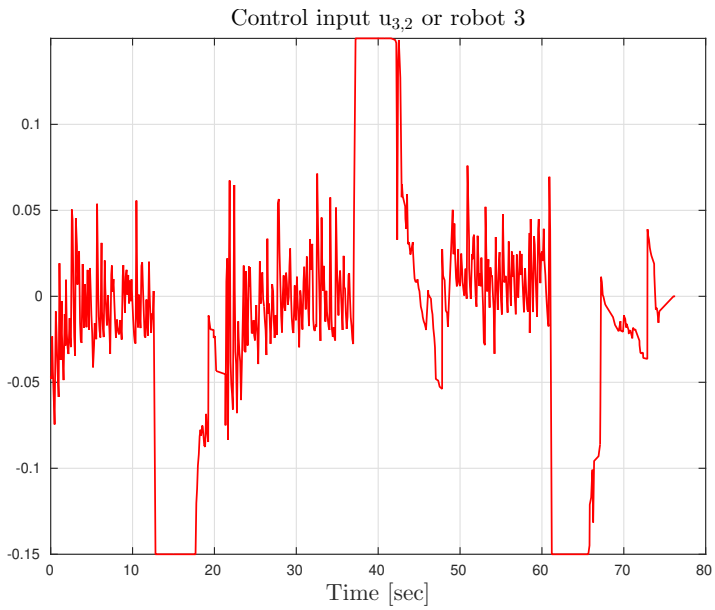

Figure 7.15: The evolution of the control signal $u_{3,2}$ or robot 3 over the time.

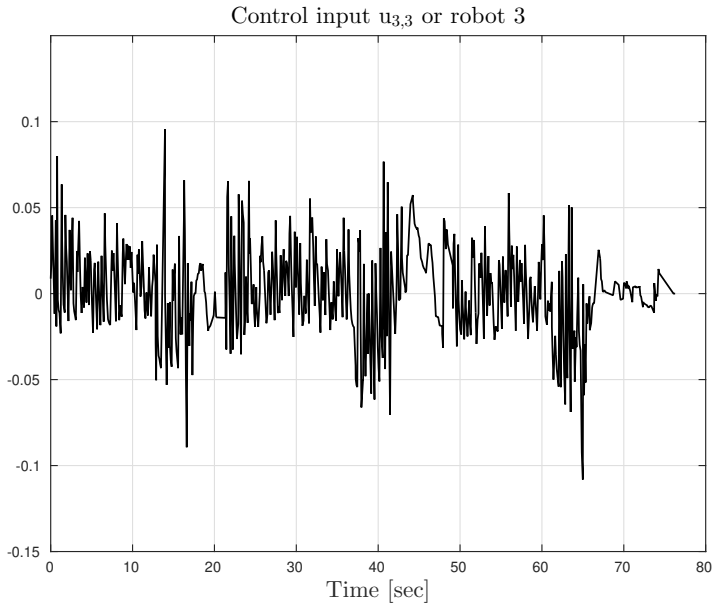

Figure 7.16: The evolution of the control signal $u_{3,3}$ or robot 3 over the time.

7.5 Summary

In this chapter, a scalable framework for time-constrained planning of multi-robot systems has been proposed. Considering N robots operating in a bounded workspace which contains RoI, assigned with tasks given in MITL, a framework for efficiently designing decentralized feedback control laws that guarantee the satisfaction of the corresponding tasks has been provided. The controllers are the outcome of DFHOCP solved by each robot at each sampling time step and form the actions of the WTS. By proposing high-level controller synthesis algorithms, a sequence of feedback laws for each robot can be designed. The approach is scalable since the local products are computed offline and only the DFHOCP of each robot is computed online which has similar complexity with the nominal NMPC framework.

Chapter 8

Summary and Future Research Directions

This chapter summarizes the content of the thesis and provides potential future research directions. This thesis is divided into three main parts, corresponding to Chapter 3 (Part 1), Chapter 4 (Part 2) and Chapters 5, 6 and 7 (Part 3).

8.1 Summary

In Chapter 3, we have proposed a class of model-free decentralized feedback control laws which guarantee distance- and orientation-based formation of multiple rigid bodies modeled by Newton-Euler dynamics. The connectivity between neighboring agents is guaranteed to be maintained for all times. Moreover, potential collisions between neighboring agents are avoided by the proposed control law. Simulation results have verified the validity of the proposed framework.

In Chapter 4, a novel robust NMPC framework has been proposed for navigation of an uncertain nonlinear multi-agent system in predefined configurations under certain distance and control input constraints. The proposed decentralized feedback control protocol consists of two terms: a nominal control input, which is computed online and is the outcome of a DFHOCP that each agent solves at every sampling time step, for its nominal system dynamics; and an additive state feedback law which is computed offline and guarantees that the real trajectories of each agent will belong to a hyper-tube centered along the nominal trajectory. The volume of the hyper-tube depends on the upper bound of the disturbances as well as the bounds of the derivatives of the dynamics. In addition, by introducing certain distance constraints, the proposed scheme guarantees that the initially connected agents remain connected. Under standard assumptions that arise in nominal NMPC schemes, controllability assumptions as well as communication capabilities between the agents, it is guaranteed that the multi-agent system is ISS with respect to the disturbances, for all initial conditions satisfying the state constraints. Simulation results verify the correctness of the proposed framework.

In Chapter 5, we have proposed a systematic method for multi-agent controller synthesis aiming cooperative planning under high-level specifications given as MITL formulas. The solution involves a sequence of algorithmic automata constructions such that not only team specifications but also individual specifications should be fulfilled.

In Chapter 6, a fully automated procedure for controller synthesis for multi-agent systems under coupling constraints has been presented. Each agent is modeled with dynamics consisting of two terms: the first one models the coupling constraints and the second one is an additional bounded control input. We have proposed a systematic method for designing these inputs so that each agent meets an individual high-level specification given using MITL. First, a decentralized abstraction that provides a space and time discretization of the multi-agent system is designed. Second, by utilizing this abstraction and techniques from Chapter 5, we have proposed an algorithm that computes the individual runs which provably satisfy the high-level tasks.

In Chapter 7, a scalable procedure for time-constrained planning of a class of nonlinear multi-robot systems has been proposed. We provide decentralized and robust control laws so that each robot meets an individual high-level specification given as a MITL, while using only local information based on a limited sensing radius. Furthermore, the robots needs to fulfill certain desired transient constraints such as collision avoidance. The controllers serve as actions for the individual WTS of each robot, and the time duration required for transition between regions is modeled by a weight. The proposed approach is scalable, since it does not require a product computation among the WTS of the robots. An experimental study in our lab facilities validates the efficiency of the proposed approach.

8.2 Future Research Directions

Regarding Chapter 3, we are currently working towards extending this framework in guaranteeing collision avoidance among all the agents of the network. This could be performed by employing potential field-based feedback control strategies. Other efforts will be devoted towards real-time experiments with a multi-agent team of quadrotors at the SML of KTH.

Future research directions for Chapter 4 will be devoted towards incorporating event-triggered communication between the robots in order to save valuable actuation and sensing resources. A drawback of the NMPC is that it requires the model knowledge of the inherent nonlinear system in order for the online FHOCP to stabilize the given system. We are currently working with data-driven NMPC approaches which, given data of the system, are able to estimate the nonlinear model by using neural networks. The idea is to use universal approximation results from the deep-learning community to learn the model of the system and perform tube-base analysis with respect to the discrepancy between the real and the estimated model.

Regarding Chapter 5 and Chapter 6, future research directions include adding robust metrics to the formula in order to quantify how far away the MITL formulas are from their satisfaction. Regarding Chapter 7, we are currently investigating how the disturbances/uncertainties might affect the satisfaction of the MITL formulas. In this context, the tube-based approach of Appendix D can be used in a decentralized manner, and the time bounds of the MITL formulas can be adjusted depending on the upper bound of the disturbances/uncertainties. Another interesting extension of the current approach discussed in Chapter 7 is to deal with dynamic and unknown environments where the agents do not have *a priori* knowledge of the obstacles. In such a scenario, a framework that deals with online replanning after each transition that the environment has changed is required (newly detected obstacles should be avoided). Another potential extension of the results of this chapter is to employ ideas of probabilistic modeling of the given system with Markov Decision Processes (MDP) and further imposing high-level tasks that include both time and probability.

Regarding the implementation of Chapter 7, the demonstrated experiment has been conducted in an indoor lab environment (SML) by employing a very accurate localization system (MOCAP). A new experiment in a realistic outdoor environment with various on-board sensor data (e.g., Global Positioning System (GPS), Inertial Measurement Unit (IMU), vision etc.) as well as suitable sensor fusion algorithms can be conducted.

Another interesting topic of research is the consideration of communication delays in the modeling of the system, which has been neglected in this thesis, and the investigation of how the delays affect the proposed frameworks.

Appendix A

Proofs of Chapter 3

A.1 Proof of Theorem 3.1

By differentiating (3.15) and (3.20) with respect to time, we obtain:

$$\dot{\xi}(\xi,t) = (\rho(t))^{-1}\left[\dot{e}(t) - \dot{\rho}(t)\xi\right],$$
$$\dot{\xi}^v(\xi,\xi^v,t) = (\rho^v(t))^{-1}\left[\dot{e}^v(\xi,t) - \dot{\rho}^v(t)\xi^v\right],$$

which, by substituting (3.10) and (3.1), become:

$$\dot{\xi}(\xi,t) = (\rho(t))^{-1}\left[\overline{\mathbb{F}}_p(x)\overline{\mathscr{D}}^\top(\mathcal{G})\underline{J}(x)\underline{v}(t) - \dot{\rho}(t)\xi\right],$$
$$\dot{\xi}^v(\xi,\xi^v,t) = (\rho^v(t))^{-1}\Big\{\overline{M}^{-1}(x)\big[u - \overline{C}(x,\dot{x})v - \overline{g}(x) - \overline{\delta}(x,\dot{x},t)\big] - \dot{v}_{\mathrm{des}}(\xi,t) - \dot{\rho}^v(t)\xi^v\Big\}.$$

By employing (3.18), (3.23) as well as the fact that $v(t) = e^v(\xi,t) + v_{\mathrm{des}}(\xi,t) = \rho^v(t)\xi^v(\xi,t) + v_{\mathrm{des}}(\xi,t)$ from (3.20), the following is obtained:

$$\begin{aligned}
\dot{\xi} &= h(\xi,t)\\
&\coloneqq -(\rho(t))^{-1}P(x)r(\xi)(\rho(t))^{-1}\varepsilon(\xi) - (\rho(t))^{-1}\dot{\rho}(t)\xi\\
&\quad + (\rho(t))^{-1}\overline{\mathbb{F}}_p(x)\overline{\mathscr{D}}^\top(\mathcal{G})\underline{J}(x)\rho^v(t)\xi^v(\xi,t), \qquad \text{(A.1a)}\\
\dot{\xi}^v &= h^v(\xi,\xi^v,t)\\
&\coloneqq -(\rho^v(t))^{-1}\overline{M}^{-1}(x)\Gamma(\rho^v(t))^{-1}r^v(\xi^v)\varepsilon^v(\xi^v)\\
&\quad - (\rho(t))^{-1}\Big\{\overline{M}^{-1}(x)\big[\overline{C}(x,\dot{x})(\rho^v(t)\xi^v(\xi,t) + v_{\mathrm{des}}(\xi,t))\\
&\quad + \overline{g}(x) + \overline{\delta}(x,\dot{x},t)\big] + \dot{v}_{\mathrm{des}}(\xi,t) + \dot{\rho}^v(t)\xi^v\Big\}, \qquad \text{(A.1b)}
\end{aligned}$$

where:

$$P(x) \coloneqq \overline{\mathbb{F}}_p(x)\overline{\mathscr{D}}^\top(\mathcal{G})\overline{\mathscr{D}}(\mathcal{G})\overline{\mathbb{F}}_p^\top(x).$$

By defining $\overline{\xi} \coloneqq [\xi^\top, (\xi^v)^\top]^\top \in \mathbb{R}^{4M+6N}$, the closed loop system of (A.1) can be written in compact form as:

$$\begin{aligned}\dot{\overline{\xi}} &= \overline{h}(t, \overline{\xi}) \\ &\coloneqq \begin{bmatrix} h(\xi, t) \\ h^v(\xi, \xi^v, t) \end{bmatrix}. \end{aligned} \tag{A.2}$$

Let us also define the open set:

$$\Omega_{\overline{\xi}} \coloneqq \Omega_{\xi^p} \times \Omega_{\xi^q} \times \Omega_{\xi^v},$$

with:

$$\begin{aligned}\Omega_{\xi^p} &\coloneqq (-C_{1,\mathrm{col}}, C_{1,\mathrm{con}}) \times \cdots \times (-C_{M,\mathrm{col}}, C_{M,\mathrm{con}}), \\ \Omega_{\xi^q} &\coloneqq (-1, 1)^{3M}, \\ \Omega_{\xi^v} &\coloneqq (-1, 1)^{6N}.\end{aligned}$$

In what follows, we proceed in two phases. First, the existence of a unique maximal solution $\overline{\xi}(t)$ of (A.2) over the set $\Omega_{\overline{\xi}}$ for a time interval $[0, \tau_{\max})$ is ensured, i.e.,

$$\overline{\xi}(t) \in \Omega_{\overline{\xi}}, \quad \forall t \in [0, \tau_{\max}).$$

Then, we prove that the proposed control scheme (3.23) guarantees, for all $t \in [0, \tau_{\max})$, the boundedness of all closed loop signals, as well as that $\overline{\xi}(t)$ remains strictly within a compact subset of $\Omega_{\overline{\xi}}$, which leads by contradiction to $\tau_{\max} = +\infty$.

Phase A:

By selecting the parameters $C_{k,\mathrm{col}}, C_{k,\mathrm{con}}, k \in \mathcal{M}$, according to (3.12), we guarantee that the set $\Omega_{\overline{\xi}}$ is nonempty and open. Moreover, as shown in (3.13), we guarantee that $\xi^p(0) \in \Omega_{\xi^p}$ and $\xi^q(0) \in \Omega_{\xi^q}$. In addition, by selecting $\rho^v_{i_m}(0) > |e^v_{i_m}(0)|, \forall i \in \mathcal{V}, m \in \{1, \ldots, 6\}$, we also guarantee that $\xi^v(0) \in \Omega_{\xi^v}$. Hence, $\overline{\xi}(0) \in \Omega_{\overline{\xi}}$. Furthermore, $\overline{h}$ is continuous on t and locally Lipschitz on $\overline{\xi}$ over the set $\Omega_{\overline{\xi}}$. Therefore, according to Theorem 2.1 in Section 2.1, there exists a maximal solution $\overline{\xi}(t)$ of (A.2) on the time interval $[0, \tau_{\max})$ such that $\overline{\xi}(t) \in \Omega_{\overline{\xi}}, \forall t \in [0, \tau_{\max})$.

Phase B:

We have proven in Phase A that $\overline{\xi}(t) \in \Omega_{\overline{\xi}}, \forall t \in [0, \tau_{\max})$ and more specifically, that:

$$\xi_k^p(t) = \frac{e_k^p(t)}{\rho_k^p(t)} \in (-C_{k,\text{col}}, C_{k,\text{con}}), \tag{A.3a}$$

$$\xi_{k_n}^q(t) = \frac{e_{k_n}^q(t)}{\rho_k^p(t)} \in (-1, 1), \tag{A.3b}$$

$$\xi_{i_m}^v(t) = \frac{e_{i_m}^v(t)}{\rho_i^v(t)} \in (-1, 1), \tag{A.3c}$$

$\forall k \in \mathcal{M}, n \in \{1,2,3\}, m \in \{1,\dots,6\}, i \in \mathcal{V}$, from which we conclude that $e_k^p(t), e_{k_n}^q(t)$ and $e_{i_m}^v(t)$ are bounded by $\max\{C_{k,\text{col}}, C_{k,\text{con}}\}, \rho_k^q(t)$ and $\rho_{i_m}^v(t)$, respectively, $\forall t \in [0, \tau_{\max})$. Furthermore, the error vector $\varepsilon(\xi)$, as given in (3.18), is well defined $\forall t \in [0, \tau_{\max})$. Therefore, consider the positive definite and radially unbounded function $V_1 : \mathbb{R}^{4M} \to \mathbb{R}_{\geq 0}$, with $V_1(\varepsilon) = \frac{1}{2}\varepsilon^\top \varepsilon$. Time differentiation of V_1 yields $\dot{V}_1 = \varepsilon^\top r(\xi)\dot{\xi}$, which, after substituting (A.1a), becomes:

$$\begin{aligned}\dot{V}_1 = &-\varepsilon^\top r(\xi)(\rho(t))^{-1} P(x) r(\xi)(\rho(t))^{-1}\varepsilon \\ &-\varepsilon^\top r(\xi)(\rho(t))^{-1}\left[\dot{\rho}(t)\xi - \overline{\mathbb{F}}_p(x)\overline{\mathscr{D}}^\top(\mathcal{G})\underline{J}(x)\rho^v(t)\xi^v\right].\end{aligned}$$

Note that:

- $\dot{\rho}(t), \rho^v(t), \overline{\mathscr{D}}(\mathcal{G})$ are bounded by construction;
- $\underline{J}$ and ξ^v, p, q are bounded $\forall t \in [0, \tau_{\max})$ owing to Assumption 3.1 and (A.3), respectively, and hence $\overline{\mathbb{F}}_p(x)$ is also bounded $\forall t \in [0, \tau_{\max})$ due to its continuity.

Therefore, by also exploiting the fact that $\rho(t), r(\xi)$ are diagonal, $\dot{V}_1$ becomes:

$$\begin{aligned}\dot{V}_1 \leq &-\left[(\rho(t))^{-1} r(\xi)\varepsilon\right]^\top P(x)\left[r(\xi)(\rho(t))^{-1}\varepsilon\right] \\ &+ \left\|(\rho(t))^{-1} r(\xi)\varepsilon\right\|_2 \overline{B}_1,\end{aligned}$$

where $\overline{B}_1$ is a positive constant, independent of $\tau_{\max}$, satisfying:

$$\left\|\dot{\rho}(t)\xi - \overline{\mathbb{F}}_p(x)\overline{\mathscr{D}}^\top(\mathcal{G})\underline{J}(x)\rho^v(t)\xi^v\right\|_2 \leq \overline{B}_1, \tag{A.4}$$

By invoking Lemma A.1 from Appendix A.2, $\dot{V}_1$ becomes:

$$\begin{aligned}\dot{V}_1 &\leq -\lambda_{\min}(P)\|(\rho(t))^{-1} r(\xi)\varepsilon\|_2^2 + \|(\rho(t))^{-1} r(\xi)\varepsilon\|_2 \overline{B}_1 \\ &\leq -\|(\rho(t))^{-1} r(\xi)\varepsilon)\|_2 \left[\lambda_{\min}(P)\|(\rho(t))^{-1} r(\xi)\varepsilon)\|_2 - \overline{B}_1\right],\end{aligned}$$

with $\lambda_{\min}(P) > 0$. Therefore, $\dot{V}_1 < 0$ when:

$$\|(\rho(t))^{-1} r(\xi)\varepsilon\|_2 > \frac{\overline{B}_1}{\lambda_{\min}(P)}.$$

By using the definitions of $r(\xi)$ and $\rho(t)$ as well as their positive definiteness $\forall t \in [0, \tau_{\max})$, the last inequality can be shown to be equivalent to:

$$\|\varepsilon\|_2 > \frac{\overline{B}_1 \tilde{r}}{\lambda_{\min}(P)},$$

where:

$$\tilde{r} \coloneqq \max\left\{ \max_{k\in\mathcal{M}}\{C_{k,\mathrm{col}} + C_{k,\mathrm{con}}\}, \max_{k\in\mathcal{M}}\{\rho^q_{k,0}\}\right\}.$$

Therefore, we conclude that:

$$\|\varepsilon(\xi(t))\|_2 \leq \bar{\varepsilon} \coloneqq \max\left\{\varepsilon(\xi(0)), \frac{\overline{B}_1 \tilde{r}}{\lambda_{\min}(P)}\right\}, \tag{A.5}$$

$\forall t \in [0, \tau_{\max})$. Furthermore, from (3.16), by taking the inverse logarithm function, we obtain:

$$\begin{aligned} -C_{k,\mathrm{col}} < \frac{e^{-\bar{\varepsilon}} - 1}{e^{-\bar{\varepsilon}} + 1} C_{k,\mathrm{col}} = \xi^p_{k,\min} \leq \xi^p_k(t) \leq \xi^p_{k,\max} \\ = \frac{e^{\bar{\varepsilon}} - 1}{e^{\bar{\varepsilon}} + 1} C_{k,\mathrm{con}} < C_{k,\mathrm{con}}, \end{aligned} \tag{A.6a}$$

$$-1 < \frac{e^{-\bar{\varepsilon}} - 1}{e^{-\bar{\varepsilon}} + 1} = \xi^q_{\min} \leq \xi^q_{k_n}(t) \leq \xi^q_{\max} = \frac{e^{\bar{\varepsilon}} - 1}{e^{\bar{\varepsilon}} + 1} < 1, \tag{A.6b}$$

$\forall t \in [0, \tau_{\max}), k \in \mathcal{M}, n \in \{1, 2, 3\}$. Thus, the reference velocity vector $\underline{v}_{\mathrm{des}}(\xi, t)$, as designed in (3.18), remains bounded $\forall t \in [0, \tau_{\max})$. Moreover, since $v(t) = \rho^v(t)\xi^v(\xi, t) + v_{\mathrm{des}}(\xi, t)$, we also conclude the boundedness of $v(t), \forall t \in [0, \tau_{\max})$. Finally, differentiating v_{des} with respect to time, substituting (A.1a) and using (A.6), the boundedness of $\dot{v}_{\mathrm{des}}, \forall t \in [0, \tau_{\max})$, is deduced as well.

Applying the aforementioned line of proof, we consider the positive definite and radially unbounded function $V_2 : \mathbb{R}^{6N} \to \mathbb{R}_{\geq 0}$, with:

$$V_2(\varepsilon^v) = \tfrac{1}{2}(\varepsilon^v)^\top \Gamma \varepsilon^v,$$

since the error vector $\varepsilon^v(\xi^v)$ is well defined $\forall t \in [0, \tau_{\max})$, due to (A.3c). Time differentiation of V_2 yields:

$$\dot{V}_2 = (\varepsilon^v)^\top \Gamma r^v(\xi^v)\dot{\xi}^v,$$

which, after substituting (A.1b), becomes:

$$\begin{aligned} \dot{V}_2 = &- (\varepsilon^v)^\top \Gamma r^v(\xi^v)(\rho^v(t))^{-1} \overline{M}^{-1}(x) \Gamma (\rho^v(t))^{-1} r^v(\xi^v)\varepsilon^v \\ &- (\varepsilon^v)^\top r^v(\xi^v)(\rho^v(t))^{-1} \Big\{ \overline{M}^{-1}(x) \left[\overline{g}(x) + \overline{\delta}(x, \dot{x}, t) \right. \\ &+ \left. \overline{C}(x, \dot{x})(\rho^v(t)\xi^v(\xi, t) + v_{\mathrm{des}}(\xi, t))\right] + \dot{v}_{\mathrm{des}}(\xi, t) + \dot{\rho}^v(t)\xi^v \Big\}. \end{aligned}$$

By exploiting the boundedness of ξ^v and the positive definiteness and diagonality of $\Gamma, \rho^v(t), r^v(\xi^v), \forall t \in [0, \tau_{\max})$ due to (A.3c), the boundedness of $\rho^v, \dot{\rho^v}$, v_{des}, $\dot{v}_{\text{des}}$, $\overline{\delta}(x, \dot{x}, t)$, the continuity of $\overline{M}^{-1}, \overline{C}$, $\overline{g}$ and the positive definiteness of $\overline{M}^{-1}$, $\dot{V}_2$ becomes:

$$\dot{V}_2 \leq -\lambda_{\min}\left(\Gamma\overline{M}^{-1}\Gamma\right)\left\|(\rho^v(t))^{-1}r^v(\xi^v)\varepsilon^v(\xi^v)\right\|_2^2 + \left\|(\rho^v(t))^{-1}r^v(\xi^v)\varepsilon^v(\xi^v)\right\|_2\overline{B}_2,$$

where $\lambda_{\min}\left(\Gamma\overline{M}^{-1}\Gamma\right) > 0$ and $\overline{B}_2$ is a positive constant, independent of $\tau_{\max}$, that satisfies:

$$\left\|\overline{M}^{-1}(x)\Big[\overline{g}(x) + \overline{\delta}(x, \dot{x}, t) + \overline{C}(x, \dot{x})(\rho^v(t)\xi^v(\xi, t) + v_{\text{des}}(\xi, t))\Big] + \dot{v}_{\text{des}}(\xi, t) + \dot{\rho}^v(t)\xi^v\right\|_2 \leq \overline{B}_2.$$

Therefore, we conclude that $\dot{V}_2 < 0$ when:

$$\left\|(\rho^v(t))^{-1}r^v(\xi^v)\varepsilon^v(\xi^v)\right\|_2 > \frac{\overline{B}_2}{\lambda_{\min}(\Gamma\overline{M}^{-1}\Gamma)},$$

which is equivalent to:

$$\|\varepsilon^v\|_2 > \frac{\overline{B}_2\tilde{r}_v}{\lambda_{\min}(\Gamma\overline{M}^{-1}\Gamma)},$$

with:

$$\tilde{r}_v \coloneqq \max\left\{\rho^v_{i_m,0}, i \in \mathcal{V}, m \in \{1, \ldots, 6\}\right\}.$$

Hence, we conclude that:

$$\|\varepsilon^v(\xi^v(\xi(t), t))\|_2 \leq \overline{\varepsilon}^v \coloneqq \max\left\{\varepsilon^v(\xi^v(\xi(0), 0)), \frac{\overline{B}_2\tilde{r}_v}{\lambda_{\min}(\Gamma\overline{M}^{-1}\Gamma)}\right\},$$

$\forall t \in [0, \tau_{\max})$. Furthermore, from (3.21a), we obtain:

$$-1 < \frac{e^{-\overline{\varepsilon}^v} - 1}{e^{-\overline{\varepsilon}^v} + 1} = \xi^v_{\min} \leq \xi^v_{i_m}(t) \leq \xi^v_{\max} = \frac{e^{\overline{\varepsilon}^v} - 1}{e^{\overline{\varepsilon}^v} + 1} < 1, \tag{A.7}$$

$\forall t \in [0, \tau_{\max}), m \in \{1, \ldots, 6\}, i \in \mathcal{V}$, which leads to the boundedness of the decentralized control protocol (3.23).

Up to this point, what remains to be shown is that $\tau_{\max}$ can be extended to ∞. In this direction, notice by (A.6) and (A.7) that:

$$\overline{\xi}(t) \in \Omega'_{\overline{\xi}} = \Omega'_{\xi^p} \times \Omega'_{\xi^q} \times \Omega'_{\xi^v}, \quad \forall t \in [0, \tau_{\max}),$$

where:

$$\Omega'_{\xi^p} = [\xi^p_{1,\min}, \xi^p_{1,\max}] \times \cdots \times [\xi^p_{M,\min}, \xi^p_{M,\max}],$$
$$\Omega'_{\xi^q} = [\xi^q_{\min}, \xi^q_{\max}]^{3M},$$
$$\Omega'_{\xi^v} = [\xi^v_{\min}, \xi^v_{\max}]^{6N},$$

are nonempty and compact subsets of $\Omega_{\xi^p}, \Omega_{\xi^q}$ and Ω_{ξ^v}, respectively. Hence, assuming that $\tau_{\max} < \infty$ and since $\Omega'_{\overline{\xi}} \subseteq \Omega_{\overline{\xi}}$, Proposition 2 in Section 2.1 dictates the existence of a time instant $t' \in [0, \tau_{\max})$ such that $\overline{\xi}(t') \notin \Omega'_{\overline{\xi}}$, which is a contradiction. Therefore, $\tau_{\max} = \infty$. Thus, all closed loop signals remain bounded and moreover $\overline{\xi}(t) \in \Omega'_{\overline{\xi}} \subseteq \Omega_{\overline{\xi}}, \forall t \in \mathbb{R}_{\geq 0}$. Finally, multiplying (A.6a) and (A.6b) by $\rho^p_k(t)$ and $\rho^q_k(t)$, respectively, we also conclude:

$$-C_{k,\mathrm{col}}\rho^p_k(t) < e^p_k(t) < C_{k,\mathrm{con}}\rho^p_k(t),$$
$$-\rho^q_k(t) < e^q_{k_n}(t) < \rho^q_k(t),$$

$\forall k \in \mathcal{M}, n \in \{1, 2, 3\}, t \in \mathbb{R}_{\geq 0}$, which leads to the completion of the proof.

A.2 Proof of Auxiliary Lemma

Lemma A.1. *The matrix:*

$$P(x) \coloneqq \overline{\mathbb{F}}_p(x)\overline{\mathscr{D}}^\top(\mathcal{G})\overline{\mathscr{D}}(\mathcal{G})\overline{\mathbb{F}}_p^\top(x),$$

is positive definite for every $t \in [0, \tau_{\max})$.

Proof. Firstly, note that Assumption 3.2 implies that $\mathcal{G}$ is connected at $t = 0$. Hence, in view of (A.3a), $\mathcal{G}$ will stay connected for all $t \in [0, \tau_{\max})$. Moreover, since we do not consider adding edges to the graph, $\mathcal{G}$ will also be a tree for all $t \in [0, \tau_{\max})$, and thus, the matrix $\mathscr{D}^\top(\mathcal{G})\mathscr{D}(\mathcal{G})$ is positive definite for all $t \in [0, \tau_{\max})$, according to Lemma 2.2. Therefore, the matrix:

$$\overline{\mathscr{D}}^\top(\mathcal{G})\overline{\mathscr{D}}(\mathcal{G}) = \begin{bmatrix} \mathscr{D}^\top(\mathcal{G})\mathscr{D}(\mathcal{G}) \otimes I_3 & 0_{3M\times 3M} \\ 0_{3M\times 3M} & \mathscr{D}^\top(\mathcal{G})\mathscr{D}(\mathcal{G}) \otimes I_3 \end{bmatrix},$$

is also positive definite. Moreover, (A.3a) implies that:

$$\|p_{\ell_k}(t) - p_{\ell_m}(t)\|_2 > d_{k,\text{col}}, \forall t \in [0, \tau_{\max}).$$

Hence, there exists at least one $w \in \{x, y, z\}$ such that:

$$(p_{\ell_k})_w(t) \neq (p_{\ell_m})_w(t), \forall t \in [0, \tau_{\max}),$$

where $p_{\ell_a} = [(p_{\ell_a})_x, (p_{\ell_a})_y, (p_{\ell_a})_z]^\top, a \in \{k, m\}$. Therefore, $\text{rank}(\mathbb{F}_p(x)) = M$ and $\text{rank}(\overline{\mathbb{F}}_p(x)) = 4M$, which implies the positive definiteness of:

$$P(x) = \overline{\mathbb{F}}_p(x)\overline{\mathscr{D}}^\top(\mathcal{G})\overline{\mathscr{D}}(\mathcal{G})\overline{\mathbb{F}}_p^\top(x).$$

□

Appendix B

Proofs of Chapter 4

B.1 Proof of Lemma 4.1

Consider the candidate Lyapunov function:

$$\mathfrak{L}(\mathfrak{e}_i) \coloneqq \frac{1}{2}\|\mathfrak{e}_i\|_2^2 > 0,$$

with $\mathfrak{L}(0_{n\times 1}) = 0$. The time derivative of $\mathfrak{L}$ along the trajectories of the system (4.10), is given by:

$$\begin{aligned}
\dot{\mathfrak{L}}(\mathfrak{e}_i) &= \mathfrak{e}_i^\top \dot{\mathfrak{e}}_i \\
&= \mathfrak{e}_i^\top \Big[g_i(e_i, \overline{e}_i, u_i) + f_i(\overline{e}_i + x_{i,d}, u_i) - f_i(\overline{e}_i + x_{i,d}, \overline{u}_i) + \delta_i \Big] \\
&= \mathfrak{e}_i^\top g_i(e_i, \overline{e}_i, u_i) + \mathfrak{e}_i^\top \Big[f_i(\overline{e}_i + x_{i,d}, u_i) - f_i(\overline{e}_i + x_{i,d}, \overline{u}_i) \Big] + \mathfrak{e}_i^\top \delta_i \\
&\leq \|\mathfrak{e}_i\|_2 \|g_i(e_i, \overline{e}_i, u_i)\|_2 + \|\mathfrak{e}_i\|_2 \|\delta_i\|_2 \\
&\qquad + \mathfrak{e}_i^\top \big[f_i(\overline{e}_i + x_{i,d}, u_i) - f_i(\overline{e}_i + x_{i,d}, \overline{u}_i) \big].
\end{aligned}$$

By using the upper bound of g_i as in (4.11), the latter becomes:

$$\dot{\mathfrak{L}}(\mathfrak{e}_i) \leq L_i \|\mathfrak{e}_i\|_2^2 + \widetilde{\delta}_i \|\mathfrak{e}_i\|_2 + \mathfrak{e}_i^\top \big[f_i(\overline{e}_i + x_{i,d}, u_i) - f_i(\overline{e}_i + x_{i,d}, \overline{u}_i) \big]. \tag{B.1}$$

According to Proposition 1, and due to the fact that the sets $\mathcal{U}_i$ are convex, i.e.,

$$\mathrm{Co}(u_i, \overline{u}_i) \subseteq \mathcal{U}_i, \quad \forall u_i, \overline{u}_i \in \mathcal{U}_i, \quad i \in \mathcal{V},$$

there exist vectors $\xi_{i,1}, \ldots, \xi_{i,n} \in \mathrm{Co}(u_i, \overline{u}_i)$, with $\xi_{i,q} \neq u_i$ and $\xi_{i,q} \neq \overline{u}_i$, $\forall q = 1, \ldots, n$, such that:

$$\begin{aligned}
f_i(\overline{e}_i + x_{i,d}, u_i) &- f_i(\overline{e}_i + x_{i,d}, \overline{u}_i) \\
&= \left[\sum_{k=1}^{n} \sum_{j=1}^{n} l_n^k \left(l_n^j \right)^\top \frac{\partial f_{i,k}(\overline{e}_i + x_{i,d}, \xi_{i,k})}{\partial u_j} \right] (u_i - \overline{u}_i) \\
&= J_i(\overline{e}_i + x_{i,d}, \cdot)(u_i - \overline{u}_i).
\end{aligned}$$

By using the latter result, (B.1) becomes:

$$\dot{\mathfrak{L}}(\mathfrak{e}_i) \leq L_i\|\mathfrak{e}_i\|_2^2 + \widetilde{\delta}_i\|\mathfrak{e}_i\|_2 + \mathfrak{e}_i^\top J_i(\overline{e}_i + x_{i,d}, \cdot)(u_i - \overline{u}_i). \tag{B.2}$$

By designing the control laws $u_i - \overline{u}_i$ as in (4.8), (4.12) we get:

$$\dot{\mathfrak{L}}(\mathfrak{e}_i) \leq L_i\|\mathfrak{e}_i\|_2^2 + \rho_i\widetilde{\delta}_i^2 - k_i\mathfrak{e}_i^\top J_i(\overline{e}_i + x_{i,d}, \cdot)\mathfrak{e}_i.$$

Writing the matrices $J_i(\cdot)$ as:

$$J_i = \frac{J_i + J_i^\top}{2} + \frac{J_i - J_i^\top}{2},$$

and taking into account that:

$$y^\top \left(\frac{J_i - J_i^\top}{2}\right) y = 0, \quad \forall y \in \mathbb{R}^n,$$

we get:

$$\dot{\mathfrak{L}}(\mathfrak{e}_i) \leq L_i\|\mathfrak{e}_i\|_2^2 + \widetilde{\delta}_i\|\mathfrak{e}_i\|_2 - k_i\mathfrak{e}_i^\top \left[\frac{J_i(\cdot) + J_i^\top(\cdot)}{2}\right] \mathfrak{e}_i.$$

By using (4.5) from Assumption 4.3 and the fact that:

$$y^\top P y \geq \lambda_{\min}(P)\|y\|_2^2, \quad \forall y \in \mathbb{R}^n, \quad P \in \mathbb{R}^{n\times n}, \quad P > 0,$$

we obtain:

$$\begin{aligned}
\dot{\mathfrak{L}}(\mathfrak{e}_i) &\leq L_i\|\mathfrak{e}_i\|_2^2 + \widetilde{\delta}_i\|\mathfrak{e}_i\|_2 - k_i\lambda_{\min}\left[\frac{J_i(\cdot) + J_i^\top(\cdot)}{2}\right]\|\mathfrak{e}_i\|_2^2 \\
&\leq L_i\|\mathfrak{e}_i\|_2^2 + \widetilde{\delta}_i\|\mathfrak{e}_i\|_2 - k_i\underline{J}_i\|\mathfrak{e}_i\|_2^2 \\
&= -\left(k_i\underline{J}_i - L_i\right)\|\mathfrak{e}_i\|_2^2 + \widetilde{\delta}_i\|\mathfrak{e}\|_2 \\
&= -\|\mathfrak{e}_i\|\Big[\left(k_i\underline{J}_i - L_i\right)\|\mathfrak{e}\|_2 - \widetilde{\delta}_i\Big].
\end{aligned}$$

Thus, it is guaranteed that $\dot{\mathfrak{L}}(\mathfrak{e}_i) < 0$ when:

$$\|\mathfrak{e}_i\|_2 > \frac{\widetilde{\delta}_i}{k_i\underline{J}_i - L_i}.$$

Moreover, due to the fact that $\mathfrak{e}_i(0) = 0_{n\times 1}$, $\forall i \in \mathcal{V}$ we have that:

$$\|\mathfrak{e}_i(t)\|_2 \leq \frac{\widetilde{\delta}_i}{k_i \underline{J}_i - L_i}, \quad \forall t \in \mathbb{R}_{\geq 0}, \quad i \in \mathcal{V},$$

which leads to the conclusion of the proof.

B.2 Proof of Theorem 4.1

The proof of the theorem consists of two parts:

(i) Recursive Feasibility

It will be shown that recursive feasibility is established, and it implies subsequent feasibility. Before proceeding to the proof, consider the following definition.

Definition B.1. A control input $\overline{u}_i : [t_k, t_k + T] \to \mathbb{R}^n$ for a state $\overline{e}_i(t_k)$ of agent $i \in \mathcal{V}$ is called *admissible* for the DFHOCP (4.17a)-(4.17d) if the following hold:

1. $\overline{u}_i(\cdot)$ is piecewise continuous;
2. $\overline{u}_i(\mathfrak{s}) \in \overline{\mathcal{U}}_i, \ \forall \mathfrak{s} \in [t_k, t_k + T]$;
3. $\overline{e}_i\big(t_k + \mathfrak{s};\ \overline{u}_i(\cdot), \overline{e}_i(t_k)\big) \in \overline{\mathcal{E}}_i, \ \forall \mathfrak{s} \in [0, T]$; and
4. $\overline{e}_i\big(t_k + T;\ \overline{u}_i(\cdot), \overline{e}_i(t_k)\big) \in \mathcal{F}_i$.

Consider a sampling instant t_k for which a solution $\overline{u}_i^\star\big(\cdot;\ \overline{e}_i(t_k)\big)$ to DFHOCP (4.17a)-(4.17d) of agent $i \in \mathcal{V}$ exists. Suppose now a time instant t_{k+1} such that $t_{k+1} = t_k + h$, and consider that the optimal control signal calculated at t_k is comprised of the following two portions:

$$\overline{u}_i^\star\big(\cdot;\ \overline{e}_i(t_k)\big) = \begin{cases} \overline{u}_i^\star\big(\mathfrak{s};\ \overline{e}_i(t_k)\big), & \mathfrak{s} \in [t_k, t_{k+1}] \\ \overline{u}_i^\star\big(\mathfrak{s};\ \overline{e}_i(t_k)\big), & \mathfrak{s} \in [t_{k+1}, t_k + T_p] \end{cases}, \tag{B.3}$$

Both portions are admissible since the calculated optimal control input is admissible, and hence they both conform to the input constraints. Furthermore, the predicted states $\overline{e}_i(s;\ \overline{u}_i^\star(\cdot), \overline{e}_i(t_k))$ will satisfy the state constraints for every $\mathfrak{s} \in [t_k, t_k + T]$ and it also holds that:

$$\overline{e}_i\big(t_k + T;\ \overline{u}_i^\star(\cdot), \overline{e}_i(t_k)\big) \in \mathcal{F}_i. \tag{B.4}$$

According to Assumption 4.2, there exists an admissible control input $u_{i,\mathrm{loc}}(\overline{e}_i)$ that renders $\mathcal{F}_i$ invariant over $[t_k + T, t_{k+1} + T]$. Given the above facts, we can

construct an admissible input $\overline{u}_i(\cdot)$ starting at time t_{k+1} by sewing together the second portion of (B.3) and the input $u_{i,\mathrm{loc}}(\overline{e}_i)$ as:

$$\widetilde{u}_i(\mathfrak{s}) = \begin{cases} \overline{u}_i^\star\big(s;\ \overline{e}_i(t_k)\big), & \mathfrak{s} \in [t_{k+1}, t_k + T] \\ u_{i,\mathrm{loc}}\big(\overline{e}_i(\mathfrak{s})\big), & s \in (t_k + T, t_{k+1} + T] \end{cases},$$

Applied at time t_{k+1}, $\widetilde{u}_i(\mathfrak{s})$ is an admissible control input with respect to the input constraints as a composition of admissible control inputs, for all $\mathfrak{s} \in [t_{k+1}, t_{k+1} + T]$. What remains to prove is the following statement.

Statement : $e_i\big(t_{k+1} + \mathfrak{s};\ \overline{u}_i^\star(\cdot), \overline{e}_i(t_{k+1})\big) \in \mathcal{E}_i, \quad \forall \mathfrak{s} \in [0, T].$

Initially, at time t_{k+1}, $\widetilde{u}_i$ is an admissible control input according to Definition B.1. The, according to 3) of Definition B.1 we have that:

$$\overline{e}_i\big(t_{k+1} + \mathfrak{s};\ \overline{u}_i^\star(\cdot), \overline{e}_i(t_{k+1})\big) \in \overline{\mathcal{E}_i} = \mathcal{E}_i \ominus \Omega_i, \quad \forall \mathfrak{s} \in [0, T],$$

By invoking (4.9) and the fact that Ω_i are RCI sets it is guaranteed that:

$$e_i\big(t_{k+1} + \mathfrak{s};\ \overline{u}_i^\star(\cdot), \overline{e}_i(t_{k+1})\big) - \overline{e}_i\big(t_{k+1} + \mathfrak{s};\ \overline{u}_i^\star(\cdot), \overline{e}_i(t_{k+1})\big) \in \Omega_i, \quad \forall \mathfrak{s} \in [0, T].$$

Adding the latter to the former yields:

$$e_i\big(t_{k+1} + \mathfrak{s};\ \overline{u}_i^\star(\cdot), e_i(t_{k+1})\big) \in \big(\mathcal{E}_i \ominus \Omega_i\big) \oplus \Omega_i, \quad \forall \mathfrak{s} \in [0, T].$$

By using Definition (2.2), we have that:

$$\big(\mathcal{E}_i \ominus \Omega_i\big) \oplus \Omega_i \subseteq \mathcal{E}_i,$$

from which it holds that:

$$e_i\big(t_{k+1} + \mathfrak{s};\ \overline{u}_i^\star(\cdot), e_i(t_{k+1})\big) \in \mathcal{E}_i, \quad \forall \mathfrak{s} \in [0, T],$$

which concludes the proof of the statement.

By taking the aforementioned into consideration, the feasibility of a solution to the optimization problem at time t_k implies feasibility at all times t_{n+1}, with $n > k$. Thus, since at time $t = 0$ a solution is assumed to be feasible, a solution to the optimal control problem is feasible for all $t \in \mathbb{R}_{\geq 0}$, and for all agents $i \in \mathcal{V}$.

(ii) Convergence Analysis

Recall that:

$$e_i = x_i - x_{i,d}, \quad \mathfrak{e}_i = e_i - \overline{e}_i.$$

Then, we get:

$$\|x_i(t) - x_{i,d}\|_2 \leq \|\overline{e}_i(t)\|_2 + \|\mathfrak{e}_i(t)\|_2. \tag{B.5a}$$

By using the bound from (4.14), the inequality (B.5a) is written as:

$$\|x_i(t) - x_{i,d}\|_2 \leq \|\overline{e}_i(t)\|_2 + \frac{\widetilde{\delta}_i}{k_i \underline{J}_i - L_i}. \tag{B.6a}$$

Since only the nominal system dynamics (4.7b) are used for the online computation of the control actions $\overline{u}_i(\mathfrak{s}) \in \overline{\mathcal{U}}_i$, $\mathfrak{s} \in [t_k, t_k + T]$ through the DFHOCP (4.17a)-(4.17d), by invoking nominal NMPC stability results from Chapter 2 (see also [58] for more details), it can be shown that there exist class $\mathcal{KL}$ functions $\boldsymbol{\beta}_i$, such that:

$$\|\overline{e}_i(t)\| \leq \boldsymbol{\beta}_i(\|\overline{e}_i(0)\|_2, t), \quad \forall t \in \mathbb{R}_{\geq 0}. \tag{B.7}$$

By combining (B.6a) with (B.7) we get:

$$\|x_i(t) - x_{i,d}\|_2 \leq \boldsymbol{\beta}_i(\|\overline{e}_i(0)\|_2, t) + \frac{\widetilde{\delta}_i}{k_i \underline{J}_i - L_i}, \tag{B.8a}$$

for every $t \in \mathbb{R}_{\geq 0}$. The latter inequality leads to the conclusion of the proof.

Appendix C

Proofs of Chapter 6

C.1 Proof of Theorem 6.1

Consider the following candidate Lyapunov function $\mathfrak{L} : \mathbb{R}^{Nn} \to \mathbb{R}_{\geq 0}$ as:

$$\mathfrak{L}(x) = \frac{1}{2} \sum_{i=1}^{N} \sum_{j \in \mathcal{N}_i} \|x_i - x_j\|_2^2 = \|\widetilde{x}\|_2^2 > 0.$$

The time derivative of $\mathfrak{L}$ along the trajectories of (6.2), can be computed as:

$$\begin{aligned} \dot{\mathfrak{L}} &= [\nabla \mathfrak{L}(x)]^\top \ \dot{x} \\ &= \sum_{k=1}^{n} \left\{ c\left(\frac{\partial \mathfrak{L}}{\partial x}, k\right)^\top \Big[-\mathscr{L}(\mathcal{G}) \ c(x,k) + c(v,k) \Big] \right\}, \end{aligned} \tag{C.1}$$

where:

$$c\left(\frac{\partial \mathfrak{L}}{\partial x}, k\right) = \begin{bmatrix} \frac{\partial \mathfrak{L}}{\partial x_1^k} & \cdots & \frac{\partial \mathfrak{L}}{\partial x_N^k} \end{bmatrix}^\top.$$

By computing the partial derivative of the Lyapunov function with respect to vector $x_i, \ i \in \mathcal{V}$ we get:

$$\frac{\partial \mathfrak{L}}{\partial x_i} = \sum_{j \in \mathcal{N}_i} (x_i - x_j), \quad i \in \mathcal{V},$$

from which we have that:

$$c\left(\frac{\partial \mathfrak{L}}{\partial x}, k\right)^\top = c(x,k)^\top \ \mathscr{L}(\mathcal{G}), \quad k = 1, ..., n.$$

Thus, by substituting the last in (C.1) we get:

$$\begin{aligned}\dot{\mathfrak{L}} &= -\sum_{k=1}^{n}\Big\{c(x,k)^{\top}\ [\mathscr{L}(\mathcal{G})]^2\ c(x,k)\Big\} \\ &\qquad + \sum_{k=1}^{n}\Big\{c(x,k)^{\top}\ [\mathscr{L}(\mathcal{G})]^2\, c(v,k)\Big\} \\ &\leq -\sum_{k=1}^{n}\Big\{c(x,k)^{\top}\ [\mathscr{L}(\mathcal{G})]^2\ c(x,k))\Big\} \\ &\qquad + \left|\sum_{k=1}^{n}\big\{c(x,k)^{\top}\ \mathscr{L}(\mathcal{G})\ c(v,k)\big\}\right|. \end{aligned} \tag{C.2}$$

For the first term of (C.2) we have that:

$$\sum_{k=1}^{n}\big\{c(x,k)^{\top}\ \mathscr{L}(\mathcal{G})^2\ c(x,k))\big\} = \sum_{k=1}^{n}\|\mathscr{L}(\mathcal{G})\ c(x,k)\|_2^2\,.$$

For the second term of (C.2) we get:

$$\begin{aligned}\left|\sum_{k=1}^{n}\big\{c(x,k)^{\top}\ \mathscr{L}(\mathcal{G})\ c(\nu,k)\big\}\right| &= \left|\sum_{k=1}^{n}\big\{c(x,k)^{\top}\ \mathscr{D}(\mathcal{G})\ \mathscr{D}(\mathcal{G})^{\top}\ c(\nu,k)\big\}\right| \\ &\leq \sum_{k=1}^{n}\big\{\big\|\mathscr{D}(\mathcal{G})^{\top}\ c(x,k)\big\|_2\ \ \big\|\mathscr{D}(\mathcal{G})^{\top}\big\|\ \ \|c(\nu,k)\|_2\big\} \\ &= \big\|\mathscr{D}(\mathcal{G})^{\top}\big\|\ \sum_{k=1}^{n}\{\|c(\widetilde{x},k)\|_2\ \ \|c(\nu,k)\|_2\}\,. \end{aligned} \tag{C.3}$$

By using the Cauchy-Schwarz inequality in (C.3) we get:

$$\begin{aligned}&\left|\sum_{k=1}^{n}\big\{c(x,k)^{\top}\ \mathscr{L}(\mathcal{G})\ c(v,k)\big\}\right| \\ &\quad\leq \big\|\mathscr{D}(\mathcal{G})^{\top}\big\|\ \left(\sum_{k=1}^{n}\|c(\widetilde{x},k)\|_2^2\right)^{\frac{1}{2}}\ \left(\sum_{k=1}^{n}\|c(v,k)\|_2^2\right)^{\frac{1}{2}} \\ &\quad= \big\|\mathscr{D}(\mathcal{G})^{\top}\big\|\,\|\widetilde{x}\|_2\|v\|_2 \leq \big\|\mathscr{D}(\mathcal{G})^{\top}\big\|\,\|\widetilde{x}\|_2\sqrt{N}\|v\|_{\infty}, \end{aligned}$$

where:

$$\|v\|_{\infty} = \max\{\|v_i\|_2 : i \in \mathcal{V}\} \leq v_{\max}.$$

Thus, by combining the previous inequalities, (C.2) is written

$$\dot{\mathfrak{L}} \leq -\sum_{k=1}^{n} \left\{ \|\mathscr{L}(\mathcal{G})c(x,k)\|_2^2 \right\} + \sqrt{N} \left\| \mathscr{D}(\mathcal{G})^\top \right\| \|\widetilde{x}\|_2 v_{\max}. \tag{C.4}$$

By exploiting Lemma C.1, (C.4) is written as:

$$\begin{aligned}
\dot{\mathfrak{L}} &\leq -\lambda_2^2(\mathcal{G}) \sum_{k=1}^{n} \left\{ \left\| c(x^\perp,k) \right\|_2^2 \right\} + \sqrt{N} \left\| \mathscr{D}(\mathcal{G})^\top \right\| \|\widetilde{x}\|_2 v_{\max} \\
&= -\lambda_2^2(\mathcal{G}) \ \|x^\perp\|_2^2 + \sqrt{N} \left\| \mathscr{D}(\mathcal{G})^\top \right\| \|\widetilde{x}\|_2 v_{\max} \\
&\leq -\frac{\lambda_2^2(\mathcal{G})}{2(N-1)} \|\widetilde{x}\|_2^2 + \sqrt{N} \left\| \mathscr{D}(\mathcal{G})^\top \right\| \|\widetilde{x}\|_2 v_{\max} \\
&\leq -K_1 \|\widetilde{x}\|_2 \left(\|\widetilde{x}\|_2 - K_2 v_{\max} \right).
\end{aligned} \tag{C.5}$$

where:

$$K_1 = \frac{\lambda_2^2(\mathcal{G})}{2(N-1)} > 0.$$

By using the following implication:

$$\widetilde{x} = D^\top(\mathcal{G})x \Rightarrow \|\widetilde{x}\|_2 = \|\mathscr{D}(\mathcal{G})^\top x\|_2 \leq \|\mathscr{D}(\mathcal{G})^\top\| \|x\|_2,$$

we have that:

$$0 < \mathfrak{L}(x) = \|\widetilde{x}\|_2^2 \leq \|\mathscr{D}(\mathcal{G})^\top\|^2 \|x\|_2^2,$$

and $\dot{\mathfrak{L}}(x) < 0$, when $\|\widetilde{x}\|_2 \geq \overline{R} > K_2 v_{\max}$. Thus, there exists a finite time $T > 0$ such that the trajectory will enter the compact set:

$$\mathcal{X} = \{x \in \mathbb{R}^{Nn} : \|\widetilde{x}\|_2 \leq \overline{R}\},$$

and remain there for all $t \geq T$ with $\overline{R} > K_2 v_{\max}$. This can be extracted from the following. Let us define the compact set:

$$\Omega = \left\{ x \in \mathbb{R}^{Nn} : K_2 v_{\max} < \overline{R} \leq \|\widetilde{x}\|_2 \leq \overline{M} \right\},$$

where:

$$\overline{M} = \mathfrak{L}(x(0)) = \|\widetilde{x}(0)\|_2^2.$$

Without loss of generality it is assumed that it holds $\overline{M} > \overline{R}$. Let us define the compact sets:

$$\begin{aligned}
\mathcal{S}_1 &= \left\{ x \in \mathbb{R}^{Nn} : \|\widetilde{x}\|_2 \leq \overline{M} \right\}, \\
\mathcal{S}_2 &= \left\{ x \in \mathbb{R}^{Nn} : \|\widetilde{x}\|_2 \leq K_2 v_{\max} \right\}.
\end{aligned}$$

From the equivalences:

$$\begin{aligned}
\forall\ x \in \mathcal{S}_1 \Leftrightarrow \mathfrak{L}(x) = \|\widetilde{x}\|_2^2 \leq \overline{M}^2, \forall\ x \in \mathcal{S}_2 \\
\Leftrightarrow \mathfrak{L}(x) = \|x\|_2^2 \leq K_2^2 v_{\max}^2,
\end{aligned}$$

we have that the boundaries $\partial\mathcal{S}_1$, $\partial\mathcal{S}_2$ of sets $\mathcal{S}_1$, $\mathcal{S}_2$ respectively, are two level sets of the Lyapunov function $\mathfrak{L}$. By taking the above into consideration we have that $\partial\mathcal{S}_2 \subsetneq \partial\mathcal{S}_1$. Hence, we get from (C.5) that:

$$\dot{\mathfrak{L}}(x) < 0, \quad \forall\, x \in \Omega = \mathcal{S}_1 \backslash \mathcal{S}_2, \tag{C.6}$$

In view of (C.6) and the fact that the sets $\mathcal{S}_1, \mathcal{S}_2$ are defined in terms of level sets of $\mathfrak{L}$, we conclude that both $\mathcal{S}_1$ and $\mathcal{S}_2$ are invariant with respect to the system (6.2). Consequently, according to [169, Lemma 5.1] the trajectory that starts inside the set $\mathcal{S}_1$ has to enter the interior of the set of $\mathcal{S}_2$ in finite time $T > 0$ and remain there for all time $t \geq T$.

C.2 Auxiliary Lemma

Lemma C.1. *Let $x^{\perp}$ be the projection of the vector $x \in \mathbb{R}^{Nn}$ to the orthogonal complement of the subspace:*

$$H = \Big\{ x \in \mathbb{R}^{Nn} : x_1 = \ldots = x_N \Big\}.$$

Then, the following properties hold:

$$\|\mathscr{L}(\mathcal{G})\ c(x,k)\|_2 \geq \lambda_2(\mathcal{G})\ \|c(x^{\perp},k)\|_2,\ \forall\ k \in \mathcal{V},$$
$$\|x^{\perp}\|_2 \geq \frac{1}{\sqrt{2(N-1)}} \|\widetilde{x}\|_2.$$

Proof. The proof can be found in [170, Appendix A]. □

C.3 Sufficient Conditions for Well-possessedness of the Abstraction

We investigate here if the system (6.2) satisfies the sufficient conditions **(C1)**-**(C3)** for well-posed abstractions.

(C1) For every $i \in \mathcal{V}, \forall\ x \in \mathbb{R}^{Nn} : \widetilde{x} \in \mathcal{X}$ and $\mathrm{pr}_i(x) = (x_i, \mathbf{x}_j)$ it holds that:

$$\begin{aligned}\|f_i(x_i, \mathbf{x}_j)\|_2 &= \left\| -\sum_{j \in \mathcal{N}_i} (x_i - x_j) \right\|_2 \\ &\leq \sum_{j \in \mathcal{N}_i} \|x_i - x_j\|_2 \\ &\leq \sum_{(i,j) \in \mathcal{E}} \|x_i - x_j\|_2 = \Delta x \leq \overline{R}.\end{aligned}$$

Thus, $M = \overline{R}$. We have also that:

$$\|\mathscr{D}(\mathcal{G})^\top\| = \sqrt{\lambda_{\max}(\mathscr{D}(\mathcal{G})\mathscr{D}(\mathcal{G})^\top)} = \sqrt{\lambda_{\max}(\mathcal{G})},$$

and:

$$\lambda_2(\mathcal{G}) \leq \frac{N}{N-1} \min\Big\{N_i : i \in \mathcal{V}\Big\},$$

from [171]. For $N > 2$ it holds that $\lambda_2(\mathcal{G}) < N$. From Theorem (6.1) we have that $\overline{R} > K_2 v_{\max} \Leftrightarrow M > K_2 v_{\max}$. It holds that $M > v_{\max}$ since:

$$\begin{aligned}K_2 &= \frac{2\sqrt{N}(N-1)\left\|\mathscr{D}(\mathcal{G})^\top\right\|}{\lambda_2^2(\mathcal{G})} \\ &= \frac{2\sqrt{N}(N-1)\sqrt{\lambda_{\max}(\mathcal{G})}}{\sqrt{\lambda_2^3(\mathcal{G})}\sqrt{\lambda_2(\mathcal{G})}} \\ &\geq \frac{2\sqrt{N}(N-1)}{\sqrt{N^3}}\sqrt{\frac{\lambda_{\max}(\mathcal{G})}{\lambda_2(\mathcal{G})}} \\ &\geq \frac{2\sqrt{N}(N-1)}{\sqrt{N^3}} \\ &> 1.\end{aligned}$$

(C2) Starting from the left hand side of (6.7) we get:

$$\begin{aligned}\|f_i(x_i,\mathbf{x}_j)-f_i(x_i,\mathbf{y}_j)\|_2 &= \Big\| -\sum_{j\in\mathcal{N}_i}(x_i-x_j)+\sum_{j\in\mathcal{N}_i}(x_i-y_j)\Big\|_2 \\ &\leq \max\Big\{\sqrt{N_i} : i\in\mathcal{V}\Big\}\ \|(x_i,\mathbf{x}_j)-(x_i,\mathbf{y}_j)\|_2.\end{aligned}$$

Thus, the condition **(C2)** holds and the Lipschitz constant is:

$$L_1=\max\{\sqrt{N_i} : i\in\mathcal{V}\}>0,$$

where the inequality:

$$\left(\sum_{i=1}^{\rho}\alpha_i\right)^2 \leq \rho\ \left(\sum_{i=1}^{\rho}\alpha_i^2\right),$$

is used.

(C3) By using the same methodology as in **(C2)**, we conclude that:

$$L_2=\max\{N_i : i\in\mathcal{V}\}>0.$$

Appendix D

Tube-based NMPC for Second-order Nonlinear Dynamics

Consider a robot operating in a bounded workspace $\mathcal{W} \subseteq \mathbb{R}^n$. The robot is governed by the following kinematics and dynamics:

$$\dot{x} = v, \tag{D.1a}$$
$$\dot{v} = f(x, v) + Gu + \delta, \tag{D.1b}$$

where x, $v \in \mathbb{R}^n$ stand for the position/orientation and the linear/angular velocity of the robot, respectively; $f : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}^n$ is a known and continuously differentiable vector field with $f(0_{n\times 1}, 0_{n\times 1}) = 0_{n\times 1}$ and $G \in \mathbb{R}^{n\times n}$; $u \in \mathbb{R}^n$ stands for the control input vector; and $\delta \in \mathbb{R}^n$ models the external disturbances and uncertainties. Consider also state constraints $x \in \mathcal{X}$, velocity constraints, input constraints as well as bounded disturbances:

$$v \in \mathscr{V} \coloneqq \{v \in \mathbb{R}^n : \|v\|_2 \leq \widetilde{v}\},$$
$$u \in \mathcal{U} \coloneqq \{u \in \mathbb{R}^n : \|u\|_2 \leq \widetilde{u}\},$$
$$\delta \in \Delta \coloneqq \{\delta \in \mathbb{R}^n : \|\delta\|_2 \leq \widetilde{\delta}\},$$

where the constants $\widetilde{v}$, $\widetilde{u}$, $\widetilde{\delta} > 0$ are *a priori* given; $\mathcal{X}$ is assumed to be a connected set with the origin as an interior point. Define the corresponding *nominal kinematics/dynamics* by:

$$\dot{\overline{x}} = v, \tag{D.2a}$$
$$\dot{\overline{v}} = f(\overline{x}, \overline{v}) + G\overline{u}, \tag{D.2b}$$

which are the real kinematics/dynamics for the case of $\delta = 0_{n\times 1}$.

Assumption D.1. The linear system $\dot{\overline{\eta}} = A\overline{\eta} + B\overline{u}$, where $\overline{\eta} \coloneqq [\overline{x}^\top, \overline{v}^\top]^\top \in \mathbb{R}^{2n}$, that is the outcome of the Jacobian linearization of the nominal dynamics (D.2a)-(D.2b) around the equilibrium states $(x, v) = (0_{n\times 1}, 0_{n\times 1})$, is stabilizable.

Assumption D.2. There exists a strictly positive constant $\underline{G}$ such that:

$$\lambda_{\min}\left(\frac{G + G^\top}{2}\right) \geq \underline{G} > 0. \tag{D.3}$$

Remark D.1. Assumption D.1 is required for the NMPC nominal stability to be guaranteed [58]. Note also that in real-time robotic systems, the matrix G usually represents the mass matrix of the robot which is always positive-definite. Thus, Assumption D.2 is satisfied.

The objective is to design a feedback control law that navigates the robot to a neighborhood of a desired configuration $x_d \in \mathbb{R}^n$, while all the imposed constraints are satisfied. Define the error vector:

$$e \coloneqq x - x_d \in \mathbb{R}^n.$$

Then, the *uncertain error kinematics/dynamics* are given by:

$$\dot{e} = v, \tag{D.4a}$$
$$\dot{v} = f(e + x_d, v) + Gu + \delta, \tag{D.4b}$$

The corresponding *nominal error kinematics/dynamics* are given by:

$$\dot{\overline{e}} = \overline{v}, \tag{D.5a}$$
$$\dot{\overline{v}} = f(\overline{e} + x_d, \overline{v}) + G\overline{u}. \tag{D.5b}$$

Consider the feedback control law:

$$u \coloneqq \overline{u}(\overline{e}, \overline{v}) + \kappa(e, v, \overline{e}, \overline{v}), \tag{D.6}$$

which consists of a nominal control action $\overline{u}(\overline{e}, \overline{v})$ and a state feedback law $\kappa(e, v, \overline{e}, \overline{v})$. The control action $\overline{u}(\overline{e}, \overline{v})$ is the outcome of a FHOCP solved online at each sampling time step; the state-feedback law is tuned offline according to a procedure that will be presented thereafter. Define by:

$$\mathfrak{e} \coloneqq e - \overline{e},$$
$$\mathfrak{v} \coloneqq v - \overline{v},$$

the deviations between the real states e, v of the uncertain system (D.4a)-(D.4b) and the states of the nominal system (D.5a)-(D.5b) with $\mathfrak{e}(0) = \mathfrak{v}(0) = 0_{n\times 1}$. It is

proven hereafter that the states $\mathfrak{e}$, $\mathfrak{v}$ remain invariant in certain compact sets. The dynamics of the states $\mathfrak{e}$ and $\mathfrak{v}$ are written as:

$$\begin{aligned}\dot{\mathfrak{e}} &= \dot{e} - \dot{\overline{e}} \\ &= v - \overline{v} \\ &= \mathfrak{v}, \end{aligned} \tag{D.7a}$$

$$\begin{aligned}\dot{\mathfrak{v}} &= \dot{v} - \dot{\overline{v}} \\ &= f(e + x_d, v) - f(\overline{e} + x_d, \overline{v}) \\ &\qquad\qquad + Gu - G\overline{u} + \delta \\ &= g(e, \overline{e}, v, \overline{v}) + G\ (u - \overline{u}) + \delta, \end{aligned} \tag{D.7b}$$

where the function g is defined by:

$$g(e, \overline{e}, v, \overline{v}) \coloneqq f(e + x_d, v) - f(\overline{e} + x_d, \overline{v}),$$

and it is upper bounded by:

$$\begin{aligned}\|g(e, \overline{e}, v, \overline{v})\|_2 &\leq \|f(e + x_d, v) - f(\overline{e} + x_d, \overline{v})\|_2 \\ &= \|f(e + x_d, v) - f(e + x_d, \overline{v}) \\ &\qquad\qquad + f(e + x_d, \overline{v}) - f(\overline{e} + x_d, \overline{v})\|_2 \\ &\leq \|f(e + x_d, v) - f(e + x_d, \overline{v})\|_2 \\ &\qquad\qquad + \|f(e + x_d, \overline{v}) - f(\overline{e} + x_d, \overline{v})\|_2 \\ &\leq L_v \|v - \overline{v}\|_2 + L_e \|e + x_d - \overline{e} - x_d\|_2 \\ &= L_v \|v - \overline{v}\|_2 + L_e \|e - \overline{e}\|_2 \\ &= L_v \|\mathfrak{v}\|_2 + L_e \|\mathfrak{e}\|_2 \\ &\leq L \left(\|\mathfrak{e}\|_2 + \|\mathfrak{v}\|_2\right). \end{aligned}$$

The constants L_e, L_v stand for the Lipschitz constants of functions f with respect to the variables e and v, respectively, and

$$L \coloneqq \max\{L_e, L_v\}.$$

Lemma D.1. *The state feedback law designed by:*

$$\begin{aligned}\kappa(e, \overline{e}, v, \overline{v}) &\coloneqq K\left[(e - \overline{e}) + (v - \overline{v})\right] \\ &\coloneqq -k(e - \overline{e}) - k(v - \overline{v}), \end{aligned} \tag{D.8}$$

where k, $\rho > 0$ are chosen such that the following hold:

$$k > \frac{1}{\underline{G}}\left[1 + (1 + 2\rho)L\right], \quad \rho > \frac{L}{2}, \tag{D.9}$$

renders the sets:

$$\Omega_1 \coloneqq \left\{ \mathfrak{e} \in \mathbb{R}^n : \|\mathfrak{e}\|_2 \leq \frac{\widetilde{\delta}}{\min\{\alpha_1, \alpha_2\}} \right\}, \tag{D.10a}$$

$$\Omega_2 \coloneqq \left\{ \mathfrak{v} \in \mathbb{R}^n : \|\mathfrak{v}\|_2 \leq \frac{2\ \widetilde{\delta}}{\min\{\alpha_1, \alpha_2\}} \right\}, \tag{D.10b}$$

RCI sets for the error dynamics (D.7a), (D.7b)*, according to Definition 2.5, where the constants* α_1, $\alpha_2 > 0$ *are defined by:*

$$\alpha_1 \coloneqq 1 - \frac{L}{2\rho}, \quad \alpha_2 \coloneqq k\ \underline{G} - 1 - (1 + 2\rho)L. \tag{D.11}$$

Proof. A backstepping control methodology will be used [172]. The state $\mathfrak{v}$ in (D.7a) can be seen as a virtual input to be designed such that the candidate Lyapunov function:

$$\mathfrak{L}_1(\mathfrak{e}) \coloneqq \frac{1}{2}\|\mathfrak{e}\|_2^2,$$

for the dynamical system (D.7a) is always decreasing. The time derivative of $\mathfrak{L}_1$ along the trajectories of system (D.7a) is given by:

$$\dot{\mathfrak{L}}_1(\mathfrak{e}) = \mathfrak{e}^\top \dot{\mathfrak{e}} = \mathfrak{e}^\top \mathfrak{v}.$$

Thus, by designing $\mathfrak{v} \equiv -\mathfrak{e}$, it yields that:

$$\dot{\mathfrak{L}}_1(\mathfrak{e}) = -\|\mathfrak{e}\|_2^2.$$

Define the backstepping auxiliary errors ζ_1, $\zeta_2 \in \mathbb{R}^n$ by:

$$\begin{aligned} \zeta_1 &\coloneqq \mathfrak{e}, \\ \zeta_2 &\coloneqq \mathfrak{v} + \mathfrak{e}. \end{aligned}$$

Then, the auxiliary error dynamics are written as:

$$\begin{aligned} \dot{\zeta}_1 &= \dot{\mathfrak{e}} \\ &= \mathfrak{v} \\ &= \zeta_2 - \mathfrak{e} \\ &= -\zeta_1 + \zeta_2, \end{aligned} \tag{D.12a}$$

$$\dot{\zeta}_2 = -\zeta_1 + \zeta_2 + g(\cdot) + G\ (u - \overline{u}) + \delta, \tag{D.12b}$$

with:

$$\begin{aligned} \|g(\cdot)\|_2 &\leq L\ (\|\mathfrak{e}\|_2 + \|\mathfrak{v}\|_2) \\ &= L\ (\|\zeta_1\|_2 + \|\zeta_1 - \zeta_2\|_2) \\ &\leq 2L\ \|\zeta_1\|_2 + L\|\zeta_2\|_2, \end{aligned} \tag{D.12c}$$

and $\zeta_1(0) = \zeta_2(0) = 0_{n\times 1}$.

Define the stack vector $\zeta \coloneqq [\zeta_1^\top, \zeta_2^\top]^\top \in \mathbb{R}^{2n}$ and consider the candidate Lyapunov function $\mathfrak{L}(\zeta) = \frac{1}{2}\|\zeta\|_2^2$ with $\mathfrak{L}(0_{2n\times 1}) = 0$. The time derivative of $\mathfrak{L}$ along the trajectories of system (D.7a)-(D.7b) is given by:

$$\begin{aligned}\dot{\mathfrak{L}}(\zeta) &= \zeta^\top \dot{\zeta} \\ &= \zeta_1^\top \dot{\zeta}_1 + \zeta_2^\top \dot{\zeta}_2 \\ &= -\|\zeta_1\|_2^2 + \|\zeta_2\|_2^2 + \zeta_2^\top\ g(\cdot) \\ &\qquad\qquad + \zeta_2^\top\ G\ (u-\overline{u}) + \zeta_2^\top\ \delta \\ &\leq -\|\zeta_1\|_2^2 + \|\zeta_2\|_2^2 + \|\zeta_2\|_2\ \|g(\cdot)\|_2 \\ &\qquad\qquad + \zeta_2^\top\ G\ (u-\overline{u}) + \zeta_2^\top\ \delta.\end{aligned}$$

By using (D.12c), the latter becomes:

$$\begin{aligned}\dot{\mathfrak{L}}(\zeta) = &- \|\zeta_1\|_2^2 + (L+1)\|\zeta_2\|_2^2 + 2L\|\zeta_1\|_2\|\zeta_2\|_2 \\ &+ \zeta_2^\top\ G\ (u-\overline{u}) + \zeta_2^\top \delta.\end{aligned}$$

By using Lemma 2.1 we have:

$$\begin{aligned}\|\zeta_1\|_2\|\zeta_2\|_2 &\leq \frac{\|\zeta_1\|_2^2}{4\rho} + \rho\|\zeta_2\|_2^2 \\ \Rightarrow 2L\|\zeta_1\|_2\|\zeta_2\|_2 &\leq \frac{L\|\zeta_1\|_2^2}{2\rho} + 2\rho L\|\zeta_2\|_2^2.\end{aligned}$$

with ρ satisfying (D.9). Then, it holds that:

$$\begin{aligned}\dot{\mathfrak{L}}(\zeta) \leq &- \left(1 - \frac{L}{2\rho}\right)\|\zeta_1\|_2^2 + [1 + (1+2\rho)L]\ \|\zeta_2\|_2^2 \\ &+ \zeta_2^\top\ G\ (u-\overline{u}) + \|\zeta_2\|_2\widetilde{\delta}.\end{aligned}$$

By designing:

$$\begin{aligned}u - \overline{u} &= -k\zeta_2 \\ &= -k\mathfrak{e} - k\mathfrak{v} \\ &= -k(e-\overline{e}) - k(v-\overline{v}),\end{aligned}$$

which is the same with (D.8) and compatible with (D.6) we get:

$$\begin{aligned}\dot{\mathfrak{L}}(\zeta) \leq &- \left(1 - \frac{L}{2\rho}\right)\|\zeta_1\|_2^2 + [1 + (1+2\rho)L]\ \|\zeta_2\|_2^2 \\ &- k\ \zeta_2^\top\ G\ \zeta_2 + \|\zeta_2\|_2\widetilde{\delta}.\end{aligned}$$

Writing the matrix G as:

$$G = \frac{G + G^\top}{2} + \frac{G - G^\top}{2},$$

and taking into account that:

$$y^\top \left(\frac{G - G^\top}{2} \right) y = 0, \quad \forall y \in \mathbb{R}^n,$$
$$y^\top P y \geq \lambda_{\min}(P) \|y\|_2^2, \quad \forall y \in \mathbb{R}^n, \quad P \in \mathbb{R}^{n \times n}, \quad P > 0,$$

we obtain:

$$\dot{\mathfrak{L}}(\zeta) \leq -\left(1 - \frac{L}{2\rho}\right) \|\zeta_1\|_2^2 + [1 + (1 + 2\rho)L] \|\zeta_2\|_2^2$$
$$- k\lambda_{\min}\left(\frac{G + G^\top}{2}\right) \|\zeta_2\|_2^2 + \|\zeta_2\|_2 \widetilde{\delta}.$$

By using Assumption D.2 and (D.11), the latter becomes:

$$\begin{aligned}
\dot{\mathfrak{L}}(\zeta) &\leq -\alpha_1 \|\zeta_1\|_2^2 - \alpha_2 \|\zeta_2\|_2^2 + \|\zeta_2\|_2 \widetilde{\delta} \\
&\leq -\min\{\alpha_1, \alpha_2\} \left(\|\zeta_1\|_2^2 + \|\zeta_2\|_2^2 \right) + \|\zeta_2\|_2 \widetilde{\delta} \\
&= -\min\{\alpha_1, \alpha_2\} \|\zeta\|_2^2 + \|\zeta_2\|_2 \widetilde{\delta} \\
&\leq -\min\{\alpha_1, \alpha_2\} \|\zeta\|_2^2 + \|\zeta\|_2 \widetilde{\delta} \\
&= \|\zeta\|_2 \left[-\min\{\alpha_1, \alpha_2\} \|\zeta\|_2 + \widetilde{\delta} \right].
\end{aligned}$$

Thus,

$$\dot{\mathfrak{L}}(\zeta) < 0 \quad \text{when} \quad \|\zeta\|_2 > \frac{\widetilde{\delta}}{\min\{\alpha_1, \alpha_2\}}.$$

Taking the latter into account and the fact that $\zeta(0) = 0_{2n \times 1}$ we have that:

$$\|\zeta(t)\|_2 \leq \frac{\widetilde{\delta}}{\min\{\alpha_1, \alpha_2\}}, \quad \forall t \geq 0.$$

Moreover, the following inequalities hold:

$$\|\mathfrak{e}\|_2 = \|\zeta_1\|_2 \leq \|\zeta\|_2$$
$$\Rightarrow \|\mathfrak{e}\|_2 \leq \frac{\widetilde{\delta}}{\min\{\alpha_1, \alpha_2\}}, \quad \forall t \geq 0,$$

and:

$$\left| \|\mathfrak{e}\|_2 - \|\mathfrak{v}\|_2 \right| \leq \|\mathfrak{e} + \mathfrak{v}\|_2 = \|\zeta_2\|_2 \leq \|\zeta\|_2$$
$$\Rightarrow \|\mathfrak{v}\|_2 \leq \frac{2\ \widetilde{\delta}}{\min\{\alpha_1, \alpha_2\}}, \quad \forall t \geq 0,$$

which leads to the conclusion of the proof. □

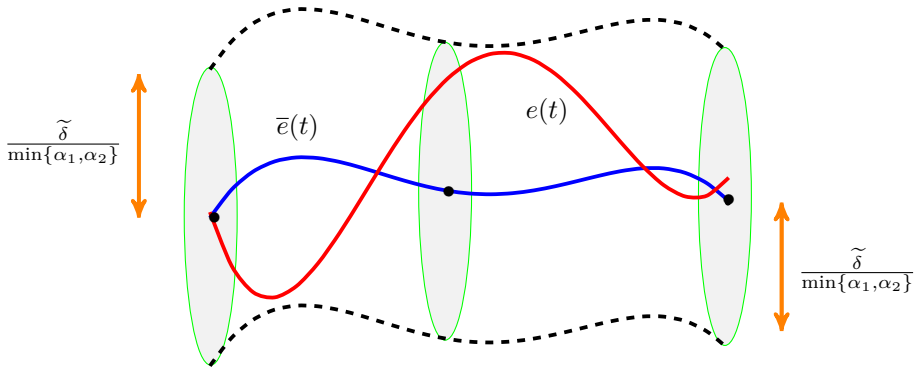

Figure D.1: The hyper-tube centered along the trajectory $\overline{e}(t)$ (depicted by blue line) with radius $\frac{\widetilde{\delta}}{\min\{\alpha_1,\alpha_2\}}$. Under the proposed control law, the real trajectory $e(t)$ (depicted with red line) lies inside the hyper-tube for all times, i.e., $\|\mathfrak{e}(t)\|_2 \leq \frac{\widetilde{\delta}}{\min\{\alpha_1,\alpha_2\}}$, $\forall t \in \mathbb{R}_{\geq 0}$.

The aforementioned result states that real trajectories $e(t)$, $v(t)$ will belong to hyper-tubes which are centered along the nominal trajectories $\overline{e}(t)$, $\overline{v}(t)$, respectively. The tubes' radii are $\frac{\widetilde{\delta}}{\min\{\alpha_1,\alpha_2\}}$ and $\frac{2\widetilde{\delta}}{\min\{\alpha_1,\alpha_2\}}$, respectively, as it is depicted in Fig. D.1.

Remark D.2. It should be noted that the volume of the hyper-tubes depends on the upper bound of the disturbances $\widetilde{\delta}$, the Lipschitz constants L and the constant $\underline{G}$. By tuning the gains k and ρ as in (D.9), (D.11) appropriately, the volume of the tubes can be adjusted.

By using (D.6), the closed-loop system is written as:

$$\dot{e} = v, \tag{D.13a}$$

$$\dot{v} = f(e + x_d, v) + \overline{u}(\overline{e}, \overline{v}) - k(e - \overline{e}) - k(v - \overline{v}) + \delta. \tag{D.13b}$$

Consider a sequence of sampling time steps $\{t_k\}$, $k \in \mathbb{N}$, with a constant sampling period $0 < h < T$, where T stands for the finite prediction horizon. It holds that $t_{k+1} = t_k + h$, $\forall k \in \mathbb{N}$. It should be noted that both t_k and T are multiples of h. At

every discrete sampling time step t_k, a FHOCP is solved as follows:

$$\min_{\overline{u}(\cdot)} \left\{ \|\overline{\xi}(t_k+T)\|_P^2 + \int_{t_k}^{t_k+T} \left[\|\overline{\xi}(\mathfrak{s})\|_Q^2 + \|\overline{u}(\mathfrak{s})\|_R^2 \right] d\mathfrak{s} \right\} \tag{D.14a}$$

subject to:

$$\dot{\overline{\xi}}(\mathfrak{s}) = \mathfrak{f}(\overline{e}(\mathfrak{s}), \overline{v}(\mathfrak{s}), \overline{u}(\mathfrak{s})), \tag{D.14b}$$

$$\overline{\xi}(\mathfrak{s}) \in \overline{\mathcal{E}} \times \overline{\mathscr{V}}, \quad \overline{u}(\mathfrak{s}) \in \overline{\mathcal{U}}, \quad \forall \mathfrak{s} \in [t_k, t_k+T], \tag{D.14c}$$

$$\overline{\xi}(t_k+T) \in \mathcal{F}. \tag{D.14d}$$

In the aforementioned optimal control problem we defined:

$$\begin{aligned} \overline{\xi} &\coloneqq [\overline{e}, \overline{v}]^\top \in \mathbb{R}^{2n}, \\ \mathfrak{f}(\overline{\xi}, \overline{u}) &\coloneqq \begin{bmatrix} \overline{v} \\ f(\overline{e} + x_d, \overline{v}) + \overline{u} \end{bmatrix}, \\ \mathcal{E} &\coloneqq \mathcal{X} \ominus \{x_d\}. \end{aligned}$$

The matrices Q, $P \in \mathbb{R}^{2n}$ and $R \in \mathbb{R}^n$ are positive definite weight matrices. The set $\mathcal{F}$ stands for the terminal set that is used to enforce the stability of the nominal system (see [58] for more details).

Hereafter, the sets $\overline{\mathcal{E}}$, $\overline{\mathscr{V}}$ and $\overline{\mathcal{U}}$ will be explained. In order to guarantee that while the FHOCP (D.14a)-(D.14d) is solved for the nominal dynamics (D.5a)-(D.5b), the real states e, v and control input u satisfy the corresponding state and input constraints $\mathcal{E}$, $\mathscr{V}$ and $\mathcal{U}$, respectively, the latter sets are appropriately modified. By recalling that:

$$\begin{aligned} \overline{e} &= e - \mathfrak{e}, \\ \overline{v} &= v - \mathfrak{v}, \\ \overline{u} &= u - K(\mathfrak{e} + \mathfrak{v}), \end{aligned}$$

and employing the set operators of Definition 2.2, the constraint tightening is performed as:

$$\overline{\mathcal{E}} \coloneqq \mathcal{E} \ominus \Omega_1, \tag{D.15a}$$

$$\overline{\mathscr{V}} \coloneqq \mathscr{V} \ominus \Omega_2, \tag{D.15b}$$

$$\overline{\mathcal{U}} \coloneqq \mathcal{U} \ominus K\overline{\Omega}, \tag{D.15c}$$

with $\overline{\Omega} \coloneqq \Omega_1 \oplus \Omega_2$, and Ω_1, Ω_2 as given in (D.10a), (D.10b), respectively. This constitutes a standard constraints set modification technique adopted in tube-based NMPC frameworks (for more details see [123]). The advantage of the tube-based framework compared to other robust NMPC approaches is that the constraint tightening is performed offline and it does not depend on the length of the horizon.

Algorithm 4 Implementation of feedback control law $u(t)$

Step 0 : At time $t_0 \coloneqq 0$, set $e(0) = \overline{e}(0)$, $v(0) = \overline{v}(0)$.
Step 1 : At time t_k and current state $(e(t_k), v(t_k), \overline{e}(t_k), \overline{v}(t_k))$, solve the FHOCP (D.14a)-(D.14d) to obtain the nominal control action $\overline{u}(t_k)$ and the actual control action

$$u(t_k) = \overline{u}(t_k) + \kappa(e(t_k), \overline{e}(t_k), v(t_k), \overline{v}(t)).$$

Step 2 : Apply the control $u(t_k)$ to the system (D.4a)-(D.4b), during sampling interval $[t_k, t_{k+1})$, where $t_{k+1} = t_k + h$.
Step 3 : Measure the state $(e(t_{k+1}),\ v(t_{k+1}))$ at the next time instant t_{k+1} of the system (D.4a)-(D.4b) and compute the successor state $(\overline{e}(t_{k+1}), \overline{v}(t_{k+1}))$ of the nominal system (D.5a)-(D.5b) under the nominal control action $\overline{u}(t_k)$.
Step 4 : Set

$$(e(t_k), \overline{e}(t_k), v(t_k), \overline{v}(t_k)) \leftarrow (e(t_{k+1}), \overline{e}(t_{k+1}), v(t_{k+1}), \overline{v}(t_{k+1})),$$
$$t_k \leftarrow t_{k+1};$$

Step 5 : Go to **Step 1**.

Algorithm 4 depicts the procedure of how the control law is calculated and applied to a real robot.

The following theorem guarantees the navigation of the robot to a neighborhood of the desired configuration x_d, while all the imposed constraints are satisfied.

Theorem D.1. *Suppose that:*

1. *assumptions D.1, D.2 hold;*
2. *the FHOCP* (D.14a)-(D.14d) *has a feasible solution at time* $t = 0$.

Then, the proposed feedback control law (D.6), (D.8), *renders the closed-loop system* (D.13a)-(D.13b) *ISS with respect to* $\delta \in \Delta$.

Proof. The proof of the theorem consists of two parts.

(i) Feasibility Analysis

It can be shown that recursive feasibility is established and it implies subsequent feasibility. The proof of this part is similar to the feasibility proof of Theorem 4.1 and [31, 60].

(ii) Convergence Analysis

Recall that:

$$e = x - x_d, \quad \mathfrak{e} = e - \overline{e}, \quad \mathfrak{v} = v - \overline{v}.$$

Then, we get:

$$\|x(t) - x_d\|_2 \leq \|\overline{e}(t)\|_2 + \|\mathfrak{e}(t)\|_2, \tag{D.16a}$$
$$\|v(t)\|_2 \leq \|\overline{v}(t)\|_2 + \|\mathfrak{v}(t)\|_2, \tag{D.16b}$$

By using the fact that:

$$\|\overline{e}\|_2 \leq \|\overline{\xi}\|_2, \quad \|\overline{v}\|_2 \leq \|\overline{\xi}\|_2,$$

as well as the bounds from (D.10a), (D.10b), the inequalities (D.16a) and (D.16b) are written as:

$$\|x(t) - x_d\|_2 \leq \|\overline{\xi}(t)\|_2 + \tfrac{\widetilde{\delta}}{\min\{\alpha_1,\alpha_2\}}, \tag{D.17a}$$
$$\|v(t)\|_2 \leq \|\overline{\xi}(t)\|_2 + \tfrac{2\widetilde{\delta}}{\min\{\alpha_1,\alpha_2\}}, \quad \forall t \geq 0. \tag{D.17b}$$

Since only the nominal system dynamics (D.5a)-(D.5b) are used for the online computation of the control action $\overline{u}(\mathfrak{s}) \in \overline{\mathcal{U}}$, $\mathfrak{s} \in [t_k, t_k + T]$ through the FHOCP (D.14a)-(D.14d), by invoking nominal NMPC stability results from Chapter 2 (see also [58] for more details), it can be shown that there exists a class $\mathcal{KL}$ function $\boldsymbol{\beta}$, such that:

$$\|\overline{\xi}(t)\| \leq \boldsymbol{\beta}(\|\overline{\xi}(0)\|_2, t), \quad \forall t \in \mathbb{R}_{\geq 0}. \tag{D.18}$$

By combining (D.17a)-(D.17b) with (D.18) we get:

$$\|x(t) - x_d\|_2 \leq \boldsymbol{\beta}(\|\overline{\xi}(0)\|_2, t) + \tfrac{\widetilde{\delta}}{\min\{\alpha_1,\alpha_2\}}, \tag{D.19a}$$
$$\|v(t)\|_2 \leq \boldsymbol{\beta}(\|\overline{\xi}(0)\|_2, t) + \tfrac{2\widetilde{\delta}}{\min\{\alpha_1,\alpha_2\}}. \tag{D.19b}$$

for every $t \in \mathbb{R}_{\geq 0}$. The latter inequalities lead to the conclusion of the proof. □

Remark D.3. It should be noted that the volume of the tube depends on the upper bound of the disturbance $\widetilde{\delta}$, the Lipschitz constant L and the constant $\underline{G}$. By tuning the parameters k, ρ as in (D.9) appropriately, the volume of the tube can be adjusted. Moreover, note that as the gain k grows, the set $K\Omega$ is enlarged. Thus, there is an upper bound of how big the gain k can be set, since it is required that the set $\overline{\mathcal{U}}$ from (D.15c) is non-empty in order for the FHOCP (D.14a)-(D.14d) to have a feasible solution.

Appendix E

Tube-based NMPC for Nonholonomic Systems

We propose here a robust tube-based analysis for the stabilization of a nonlinear nonholonomic underwater vehicle. The considered vehicle is assumed to be equipped with a set of thrusters which are effective only in surge (u_1), heave (u_2) and yaw (u_3) motion (see Fig. E.1), meaning that the vehicle is not actuated along the sway axis. Consider the kinematics of the aforementioned nonholonomic Autonomous Unmanned Vehicle (AUV) as:

$$\dot{\eta} = J(\eta)u,$$

or in a system of differential equations as:

$$\begin{aligned}
\dot{x} &= u_1 \ \cos(\psi), \\
\dot{y} &= u_1 \ \sin(\psi), \\
\dot{z} &= u_2, \\
\dot{\psi} &= u_3,
\end{aligned}$$

where $\eta \coloneqq [x, y, z, \psi]^\top \in \mathbb{R}^4$ stands for the configuration vector of the vehicle which includes the position $[x, y, z]^\top \in \mathbb{R}^3$ and the orientation $\psi \in \mathbb{R}$ of the vehicle with respect to an inertial frame F_I; $u \coloneqq [u_1, u_2, u_3] \in \mathbb{R}^3$ stands for the control input vector of the vehicle expressed in the body-fixed frame F_B. The Jacobian matrix of the system is defined by:

$$J(\eta) \coloneqq \begin{bmatrix} \cos(\psi) & 0 & 0 \\ \sin(\psi) & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}.$$

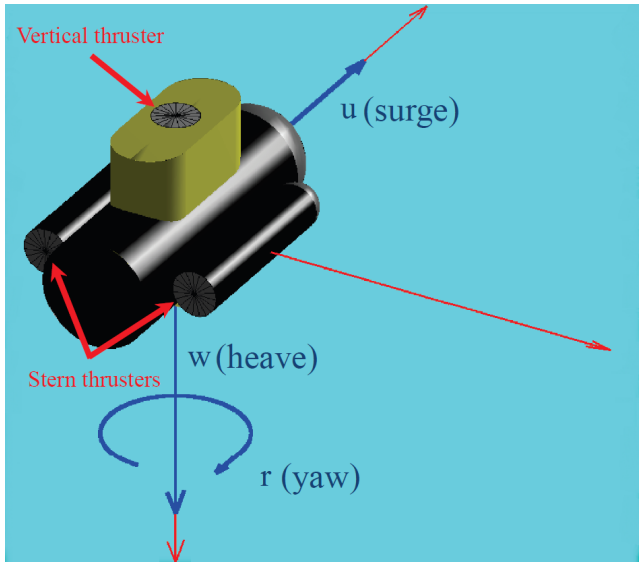

Figure E.1: The nonholonomic underwater vehicle. Blue color indicates the actuated degrees of freedom.

The body-fixed frame F_B has orthonormal axes n, o, t relative to F_I. The vehicle is under the presence of input constraints captured by the set:

$$u \in \mathcal{U} \subseteq \mathbb{R}^3.$$

Assume that $\mathcal{U}$ is compact with $0_{3\times 1} \in \mathcal{U}$. The objective is to design a feedback control law that navigates the vehicle to a neighborhood of a desired configuration:

$$p_d \coloneqq [x_d, y_d, z_d] \in \mathbb{R}^3.$$

Define the following error states:

$$e_x \coloneqq x - x_d, \tag{E.1a}$$
$$e_y \coloneqq y - y_d, \tag{E.1b}$$
$$e_d \coloneqq \sqrt{e_x^2 + e_y^2}, \tag{E.1c}$$
$$e_z \coloneqq z - z_d, \tag{E.1d}$$
$$e_o \coloneqq \sin(\psi_e) \coloneqq \frac{e_y \cos(\psi) - e_x \sin(\psi)}{e_d}, \tag{E.1e}$$

where e_d is the projected to the horizontal plane distance error; ψ_e stands for the heading angle measured from the normalized vector:

$$\mathbf{e}_d = \left[\frac{e_x}{e_d}, \frac{e_y}{e_d}, 0\right]^\top,$$

on the horizontal plane to the normalized projection of the longitudinal axis of the vehicle on the horizontal plane, defined by the vector $[\cos(\psi), \sin(\psi), 0]^\top$ (see [173] for more details). It also holds that:

$$\cos(\psi_e) \coloneqq \frac{e_x \cos(\psi) + e_y \sin(\psi)}{e_d}.$$

Then, the time derivatives of the error signals e_z and e_d and e_o are respectively given by:

$$\begin{aligned}
\dot{e}_d &= \frac{2e_x\dot{e}_x + 2e_y\dot{e}_y}{2e_d} \\
&= \frac{e_x \cos(\psi) + e_y \sin(\psi)}{e_d} u_1 \\
&= \cos(\psi_e)\ u_1, \\
\dot{e}_z &= \dot{z} - \dot{z}_d \\
&= u_2, \\
\dot{e}_o &= \frac{\frac{d}{dt}\big[e_y \cos(\psi) - e_x \sin(\psi)\big] e_d}{e_d^2} - \frac{\dot{e}_d \big[e_y \cos(\psi) - e_x \sin(\psi)\big]}{e_d^2} \\
&= \frac{\big[\dot{e}_y \cos(\psi) - e_y \sin(\psi)\dot{\psi} - \dot{e}_x \sin(\psi) - e_x \cos(\psi)\dot{\psi}\big]}{e_d} - \frac{\sin(\psi_e)\cos(\psi_e)}{e_d} u_1 \\
&= -\frac{e_y \sin(\psi) + e_x \cos(\psi)}{e_d}\dot{\psi} - \frac{\sin(\psi_e)\cos(\psi_e)}{e_d} u_1 \\
&= -\frac{\sin(\psi_e)\cos(\psi_e)}{e_d} u_1 - \cos(\psi_e)\ u_3.
\end{aligned}$$

By introducing an uncertainty vector $\delta = [\delta_d, \delta_z, \delta_o]^\top \in \mathbb{R}^3$, the aforementioned differential equations can be written in matrix form as:

$$\begin{bmatrix} \dot{e}_d \\ \dot{e}_z \\ \dot{e}_o \end{bmatrix} = \begin{bmatrix} \cos(\psi_e) & 0 & 0 \\ 0 & 1 & 0 \\ -\frac{\sin(\psi_e)\cos(\psi_e)}{e_d} & 0 & -\cos(\psi_e) \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \\ u_3 \end{bmatrix} + \begin{bmatrix} \delta_d \\ \delta_z \\ \delta_o \end{bmatrix}, \tag{E.2}$$

which stands for the *uncertain kinematics* of the vehicle in the error coordinate systems e_z, e_d and e_o. Suppose that the uncertainty vector is bounded in a known compact set:

$$\delta \in \Delta \coloneqq \big\{\delta \in \mathbb{R}^3 : \|\delta\|_2 \leq \widetilde{\delta}\big\},$$

with $\widetilde{\delta} > 0$. The error e_o and the transformed Jacobian matrix:

$$\mathbf{J} \coloneqq \begin{bmatrix} \cos(\psi_e) & 0 & 0 \\ 0 & 1 & 0 \\ -\frac{\sin(\psi_e)\cos(\psi_e)}{e_d} & 0 & -\cos(\psi_e) \end{bmatrix},$$

are well-defined when the following holds:

$$e_d(t) > 0 \;\text{ and }\; \left|\psi_e(t)\right| < \frac{\pi}{2},\; \forall t \geq 0.$$

Thus, the state constraints imposed to the system are captured by the set:

$$\mathcal{E} \coloneqq \left\{ (x, y, z, \psi) \in \mathbb{R}^4 : \sqrt{e_x^2 + e_y^2} \geq \varepsilon > 0, \;\; \left|\psi_e\right| < \frac{\pi}{2} - \varepsilon \right\}, \tag{E.3}$$

for an arbitrarily small $\varepsilon > 0$ to be chosen.

By defining the error vector $e \coloneqq [e_d, e_z, e_o]^\top \in \mathbb{R}^3$, (E.2) can be written in matrix form as:

$$\dot{e} = \mathbf{J}(e)u + \delta,\; e(0) \in \mathcal{E}. \tag{E.4}$$

By transforming the original dynamical system into the form (E.4), the original navigation problem is solved if the distance error e_d, the vertical error e_z and the orientation error e_o reduce to a neighborhood of zero, by satisfying the state and control input constraints captured by the sets $\mathcal{E}$ and $\mathcal{U}$, respectively.

Remark E.1. It should be noted that the constraint set $\mathcal{E}$ in (E.3) guarantees that $\mathbf{J}$ is non-singular since it holds that:

$$\det(\mathbf{J}) = -\left|\cos(\psi_e)\right|^2 \neq 0, \quad \forall e \in \mathcal{E}.$$

Thus, there exists a strictly positive constant $\underline{J}$ such that:

$$\lambda_{\min}\left[\frac{\mathbf{J}(e) + \mathbf{J}^\top(e)}{2}\right] \geq \underline{J} > 0, \quad \forall e \in \mathcal{E}.$$

Define the corresponding *nominal kinematics* of (E.4) by:

$$\dot{\overline{e}} = \mathbf{J}(\overline{e})\overline{u}, \quad \overline{e}(0) = e(0) \in \mathcal{E}, \tag{E.5}$$

which are the real kinematics for the case of $\delta = 0_{3\times 1}$.

Consider the feedback control law:

$$u = \overline{u}(\overline{e}) + \kappa(e, \overline{e}), \tag{E.6}$$

which consists of a nominal control action $\overline{u}(\overline{e})$ and a state-feedback law $\kappa(e, \overline{e})$. The control action $\overline{u}(\overline{e})$ is the outcome of a FHOCP solved online at each sampling time step; and the state-feedback law is designed offline according to a procedure that will be presented hereafter. Let us define by:

$$\mathfrak{e} \coloneqq e - \overline{e} \in \mathbb{R}^3,$$

the deviation between the real and the nominal error state e, $\overline{e}$, respectively. The time derivative of the deviation error state $\mathfrak{e}$ is given by:

$$\begin{aligned}\dot{\mathfrak{e}} &= \dot{e} - \dot{\overline{e}} \\ &= \mathbf{J}(e)u - \mathbf{J}(\overline{e})\overline{u} + \delta. \end{aligned} \tag{E.7}$$

Lemma E.1. *The state-feedback law designed by:*

$$\kappa(e, \overline{e}) \coloneqq K\ (e - \overline{e}) = -k\ (e - \overline{e}), \tag{E.8}$$

where k is chosen such that the following holds:

$$k > \frac{L}{\underline{J}}, \tag{E.9}$$

renders the set:

$$\Omega \coloneqq \left\{ \mathfrak{e} \in \mathbb{R}^3 : \|\mathfrak{e}\|_2 \leq \frac{\widetilde{\delta}}{k\underline{J} - L} \right\}, \tag{E.10}$$

an RCI set for the deviation error dynamics (E.7)*, according to Definition 2.5.*

Proof. Consider the candidate Lyapunov function:

$$\mathfrak{L}(\mathfrak{e}) = \frac{1}{2}\|\mathfrak{e}\|_2^2.$$

By following similar analysis and arguments with Lemma 4.1 and Lemma D.1, we get:

$$\dot{\mathfrak{L}}(\mathfrak{e}) \leq \|\mathfrak{e}\|_2 \left[(-k\underline{J} + L)\, \|\mathfrak{e}\|_2 + \widetilde{\delta} \right],$$

from which we conclude that $\dot{\mathfrak{L}}(\mathfrak{e}) < 0$ when:

$$\|\mathfrak{e}\|_2 > \frac{\widetilde{\delta}}{k\underline{J} - L}.$$

Moreover, owing to the fact that $\mathfrak{e}(0) = 0_{3\times 1}$, we obtain:

$$\|\mathfrak{e}(t)\|_2 \leq \frac{\widetilde{\delta}}{k\underline{J} - L}, \quad \forall t \geq 0,$$

which concludes the proof. □

The aforementioned result states that the real trajectory $e(t)$ always lies inside a tube with radius $\frac{\widetilde{\delta}}{k\underline{J}-L}$ centered along the nominal trajectory $\overline{e}(t)$, as depicted in Fig. 4.1 and Fig. D.1. By substituting (E.6), (E.8) into (E.4), the closed-loop error kinematics are written as:

$$\dot{e} = \mathbf{J}(e)\Big[\overline{u}(\overline{e}) + K\ (e - \overline{e})\Big] + \delta,\ e(0) \in \mathcal{E}. \tag{E.11}$$

Consider a sequence of sampling time steps $\{t_k\}$, $k \in \mathbb{N}$, with a constant sampling period $h > 0$ and a finite prediction horizon $T > 0$. It holds that $t_{k+1} = t_k + h$, $\forall k \in \mathbb{N}$. At every discrete sampling time step t_k, a FHOCP is solved as follows:

$$\min_{\overline{u}(\cdot)} \left\{ \|\overline{e}(t_k + T)\|_P^2 + \int_{t_k}^{t_k+T} \Big[\|\overline{e}(\mathfrak{s})\|_Q^2 + \|\overline{u}(\mathfrak{s})\|_R^2\Big] d\mathfrak{s} \right\} \tag{E.12a}$$

subject to:

$$\dot{\overline{e}}(\mathfrak{s}) = \mathbf{J}(\overline{e}(\mathfrak{s}))\overline{u}(\mathfrak{s}), \tag{E.12b}$$

$$\overline{e}(\mathfrak{s}) \in \overline{\mathcal{E}},\quad \overline{u}(\mathfrak{s}) \in \overline{\mathcal{U}},\quad \forall \mathfrak{s} \in [t_k, t_k + T], \tag{E.12c}$$

$$\overline{e}(t_k + T) \in \mathcal{F}. \tag{E.12d}$$

The matrices P, $Q \in \mathbb{R}^3$ and $R \in \mathbb{R}^3$ are positive definite weight matrices. The set $\mathcal{F}$ stands for the terminal set that is used in oder to enforce the stability of the nominal system. We use the notation $\overline{u}(\cdot; \overline{e}(t_k), t_k)$ to emphasize that the nominal control input is determined with the state $\overline{e}(t_k)$ at time instant t_k; $\overline{e}(\cdot; \overline{e}(t_k), t_k)$ is the trajectory of the nominal system (E.5) starting from the state $\overline{e}(t_k)$ at time t_k and driven by the control input $\overline{u}(\cdot; \overline{e}(t_k), t_k)$.

Hereafter, the sets $\overline{\mathcal{E}}$ and $\overline{\mathcal{U}}$ will be explained. In order to guarantee that while the FHOCP (E.12a)-(E.12d) is solved for the nominal dynamics (E.5), the real state e and control input u satisfy the corresponding state and input constraints $\mathcal{E}$, $\mathcal{U}$, the latter sets are appropriately modified. By recalling that:

$$\begin{aligned} \overline{e} &= e - \mathfrak{e}, \\ \overline{u} &= u - K\ \mathfrak{e}, \end{aligned}$$

and employing the set operators of Definition 2.2, the constraint tightening is performed as:

$$\overline{\mathcal{E}} \coloneqq \mathcal{E} \ominus \Omega, \tag{E.13a}$$

$$\overline{\mathcal{U}} \coloneqq \mathcal{U} \ominus K\ \Omega, \tag{E.13b}$$

with Ω as defined in (E.10). Intuitively, the sets $\mathcal{E}$ and $\mathcal{U}$ are appropriately tightened, in order to guarantee that while the nominal state $\overline{e}$ and the nominal control inputs $\overline{u}$ are calculated, the corresponding real state e and real control input u satisfy the state and input constraints $\mathcal{E}$ and $\mathcal{U}$, respectively. For the implementation of the feedback control law (E.6) to the real robot, we refer to Algorithm 5.

Algorithm 5 Implementation of the feedback control law u as given in (E.6).

Step 0 : At time $t_0 := 0$, set $e(0) = \overline{e}(0)$.
Step 1 : At time t_k and current state $(\overline{e}(t_k), e(t_k))$, solve the FHOCP (E.12a)-(E.12d) to obtain the nominal control action $\overline{u}(t_k)$ and the the actual control action:

$$u(t_k) = \overline{u}(t_k) + \kappa(e(t_k), \overline{e}(t_k)).$$

Step 2 : Apply the control $u(t_k)$ to the actual system being controlled (E.4), during the sampling interval $[t_k, t_{k+1})$, where $t_{k+1} = t_k + h$.
Step 3 : Measure the state $e(t_{k+1})$ at the next time instant t_{k+1} of the system (E.4) and compute the successor state $\overline{e}(t_{k+1})$ of the nominal system (E.5) under the nominal control action $\overline{u}(t_k)$.
Step 4 : Set:

$$(\overline{e}(t_k), e(t_k)) \leftarrow (\overline{e}(t_{k+1}), e(t_{k+1})),$$
$$t_k \leftarrow t_{k+1}.$$

Step 5 : Go to **Step 1**.

Theorem E.1. *Suppose that at time $t = 0$ the FHOCP* (E.12a)-(E.12d) *is feasible. Then the feedback control law* (E.6) *renders the closed loop system* (E.11) *ISS with respect to the disturbance $\delta \in \Delta$.*

Proof. The proof follows similar arguments with Theorem 4.1 and Theorem D.1. □

Remark E.2. It should be noted that the volume of the tube depends on the upper bound of the disturbance $\widetilde{\delta}$, the Lipschitz constant L and the constant $\underline{J}$. By tuning the gain k as in (E.9) appropriately, the volume of the tube can be adjusted. Moreover, note that as the gain k grows, the set $K\Omega$ is enlarged. Thus, there is an upper bound of how big the gain k can be set, since it is required that the set $\overline{\mathcal{U}}$ from (E.13b) is non-empty in order for the FHOCP (E.12a)-(E.12d) to have a feasible solution.

Bibliography

[1] "Driverless Platoons," *http://news.mit.edu/2016/driverless-truck-platoons-save-time-fuel-1221.*

[2] "Autonomous Driving," *http://theconversation.com/what-if-autonomous-vehicles-actually-make-us-more-dependent-on-cars-98498.*

[3] "Heterogeneous Robots Operating in a Warehouse," *https://www.123rf.com/photo _90411216 _modern-warehouse-equipped-with-robotic-arm-drone-and-robot-carriers-modern-delivery-center-concept-3.html.*

[4] R. Findeisen and F. Allgöwer, "An Introduction to Nonlinear Model Predictive Control," *21st Benelux Meeting on Systems and Control*, vol. 11, pp. 119–141, March 2002.

[5] "Smart Mobility Lab (SML)," *https://www.kth.se/dcs/research/control-of-transport/smart-mobility-lab/smart-mobility-lab-1.441539.*

[6] T. Arai, E. Pagello, and L. Parker, "Advances in Multi-robot Systems," *IEEE Transactions on Robotics and Automation*, vol. 18, no. 5, pp. 655–661, October 2002.

[7] "Google self-driving car," *https://waymo.com/.*

[8] "Bottom-Up Hybrid Control and Planning Synthesis with Application to Multi-Robot Multi-Human Coordination (BUCOPHSYS)," *http://bucophsys.eu/*, Accessed: February 2015.

[9] "Achieving Complex Collaborative Missions via Decentralized Control and Coordination of Interacting Robots (CO4ROBOTS)," *http://www.co4robots.eu/*, Accessed: February 2017.

[10] A. Nikou, C. K. Verginis, and D. V. Dimarogonas, "Robust Distance-Based Formation Control of Multiple Rigid Bodies with Orientation Alignment," *20th World Congress of the International Federation of Automatic Control (IFAC WC)*, pp. 15458–15463, Toulouse, France, July 2017.

[11] C. K. Verginis, A. Nikou, and D. V. Dimarogonas, "On the Position and Orientation Based Formation Control of Multiple Rigid Bodies with Collision Avoidance and Connectivity Maintenance," *56th IEEE Conference on Decision and Control (CDC)*, pp. 411–416, Melbourne, Australia, December 2017.

[12] C. K. Verginis, A. Nikou, and D. V. Dimarogonas, "Robust Formation Control in SE(3) for Tree-Graph Structures with Prescribed Transient and Steady State Performance," *Automatica*, vol. 103, pp. 538–548, May 2019.

[13] A. Nikou and D. V. Dimarogonas, "Decentralized Tube-based Model Predictive Control of Uncertain Nonlinear Multi-Agent Systems," *International Journal of Robust and Nonlinear Control (IJRNC)*, vol. 29, pp. 2799–2818, June 2019.

[14] A. Nikou and D. V. Dimarogonas, "Robust Tube-based Model Predictive Control for Timed-constrained Robot Navigation," *American Control Conference (ACC)*, pp. 1152–1157, Philadelphia, US, July 2019.

[15] A. Nikou, C. K. Verginis, and D. V. Dimarogonas, "A Tube-based Nonlinear MPC Scheme for Interaction Control of Underwater Vehicle Manipulator Systems," *IEEE OES Autonomous Underwater Vehicle Symposium (AUVS)*, pp. 1–6, Porto, Portugal, November 2018.

[16] A. Nikou, S. Heshmati-alamdari, and D. V. Dimarogonas, "Design and Experimental Validation of Tube-based MPC for Timed-constrained Robot Planning," *IEEE International Conference on Automation Science and Engineering (CASE)*, pp. 1181–1186, Vancouver, Canada, August 2019.

[17] S. Heshmati-alamdari, A. Nikou, and D. V. Dimarogonas, "Robust Trajectory Tracking Control for Underactuated Autonomous Underwater Vehicles," *58th IEEE Conference on Decision and Control (CDC)*, Nice, France, December 2019 (to appear).

[18] S. Heshmati-alamdari, A. Nikou, and D. V. Dimarogonas, "Robust Trajectory Tracking Control for Underactuated Autonomous Underwater Vehicles in Uncertain Environments," *IEEE Transactions on Automation Science and Engineering (TASE)*, 2019 (under review).

[19] A. Nikou, J. Tumova, and D. V. Dimarogonas, "Cooperative Task Planning Synthesis for Multi-Agent Systems Under Timed Temporal Specifications," *American Control Conference (ACC)*, pp. 13–19, Boston, MA, USA, July 2016.

[20] S. Andersson, A. Nikou, and D. V. Dimarogonas, "Control Synthesis for Multi-Agent Systems under Metric Interval Temporal Logic Specifications," *20th World Congress of the International Federation of Automatic Control (IFAC WC)*, vol. 50, pp. 2397–2402, Toulouse, France, July 2017.

[21] A. Nikou, "Cooperative Planning Control and Formation Contr of Multi-agent Systems," *Litentiate Thesis, KTH Royal Institute of Technology, Link: http://kth.diva-portal.org/smash/get/diva2:1094985/FULLTEXT01.pdf*, June 2017.

[22] A. Nikou, D. Boskos, J. Tumova, and D. V. Dimarogonas, "On the Timed Temporal Logic Planning of Coupled Multi-Agent Systems," *Automatica*, vol. 97, pp. 339–345, November 2018.

[23] A. Nikou, D. Boskos, J. Tumova, and D. V. Dimarogonas, "Cooperative Planning Synthesis for Coupled Multi-Agent Systems Under Timed Temporal Specifications," *American Control Conference (ACC)*, pp. 1847–1852, Seattle, USA, July 2017.

[24] A. Nikou, S. Heshmati-alamdari, C. K. Verginis, and D. V. Dimarogonas, "Decentralized Abstractions and Timed Constrained Planning of a General Class of Coupled Multi-Agent Systems," *56th IEEE Conference on Decision and Control (CDC)*, pp. 990–995, Melbourne, Australia, December 2017.

[25] A. Nikou, S. Heshmati-alamdari, and D. V. Dimarogonas, "Scalable Time-constrained Planning of Multi-robot Systems," *Autonomous Robots*, 2019 (under review).

[26] A. Nikou, C. K. Verginis, and D. V. Dimarogonas, "A Robust Nonlinear MPC Framework for Control of Underwater Vehicle Manipulator Systems under High-level Tasks," *IET Control Theorey and Applications*, 2019 (under review).

[27] A. Filotheou, A. Nikou, and D. V. Dimarogonas, "Robust Decentralized Navigation of Multi-Agent Systems with Collision Avoidance and Connectivity Maintenance Using Model Predictive Controllers," *International Journal of Control (IJC), Available Online: https://www.tandfonline.com/doi/full/10.1080/00207179.2018.1514129*, pp. 1–15, August 2018.

[28] A. Filotheou, A. Nikou, and D. V. Dimarogonas, "Decentralized Control of Uncertain Multi-Agent Systems with Connectivity Maintenance and Collision Avoidance," *European Control Conference (ECC)*, pp. 8–13, Limassol, Cyprus, June 2018.

[29] A. Nikou, J. Tumova, and D. V. Dimarogonas, "Probabilistic Plan Synthesis for Coupled Multi-Agent Systems," *20th World Congress of the International Federation of Automatic Control (IFAC WC)*, vol. 50, pp. 10766–10771, Toulouse, France, July 2017.

[30] S. Heshmati-alamdari, C. Bechlioulis, G. Karras, A. Nikou, D. V. Dimarogonas, and K. J. Kyriakopoulos, "A Robust Interaction Control Approach for

Underwater Vehicle Manipulator Systems," *Annual Reviews in Control*, vol. 46, pp. 315–325, January 2018.

[31] A. Nikou, C. K. Verginis, S. Heshmati-alamdari, and D. V. Dimarogonas, "A Nonlinear Model Predictive Control Scheme for Cooperative Manipulation with Singularity and Collision Avoidance," *25th IEEE Mediterranean Conference on Control and Automation (MED)*, pp. 707–712, Valletta, Malta, June 2017.

[32] C. K. Verginis, A. Nikou, and D. V. Dimarogonas, "Communication-based Decentralized Cooperative Object Transportation Using Nonlinear Model Predictive Control," *European Control Conference (ECC)*, pp. 733–738, Limassol, Cyprus, June 2018.

[33] S. Heshmati-alamdari, A. Nikou, and D. V. Dimarogonas, "A Robust Control Approach for Underwater Vehicle Manipulator Systems in Interaction with Compliant Environments," *20th World Congress of the International Federation of Automatic Control (IFAC WC)*, vol. 50, pp. 11197–11202, Toulouse, France, July 2017.

[34] A. Nikou, G. Gavridis, and K. J. Kyriakopoulos, "Mechanical Design, Modeling and Control of a Novel Aerial Manipulator," *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 4698–4703, Seattle, USA, May 2015.

[35] R. Olfati-Saber and R. Murray, "Distributed Cooperative Control of Multiple Vehicle Formations using Structural Potential Functions," *15th World Congress of the International Federation of Automatic Control (IFAC WC)*, vol. 15, no. 1, pp. 242–248, Barcelona, Spain, July 2002.

[36] S. Smith, M. E. Broucke, and B. A. Francis, "Stabilizing a Multi-Agent System to an Equilateral Polygon Formation," *17th International Symposium on Mathematical Theory of Networks and Systems*, pp. 2415–2424, July 2006.

[37] J. Hendrickx, B. Anderson, J. Delvenne, and V. Blondel, "Directed Graphs for the Analysis of Rigidity and Persistence in Autonomous Agent Systems," *International Journal of Robust and Nonlinear Control (IJRNC)*, vol. 17, no. 10-11, pp. 960–981, November 2006.

[38] B. Anderson, C. Yu, S. Dasgupta, and S. Morse, "Control of a Three-Coleader Formation in the Plane," *Systems and Control Letters*, vol. 56, no. 9, pp. 573–578, September 2007.

[39] B. Anderson, C. Yu, B. Fidan, and J. Hendrickx, "Rigid Graph Control Architectures for Autonomous Formations," *IEEE Control Systems*, vol. 28, pp. 48–63, December 2008.

[40] D. V. Dimarogonas and K. Johansson, "On the Stability of Distance-based Formation Control," *47th IEEE Conference on Decision and Control (CDC)*, pp. 1200–1205, Cancun, Mexico, December 2008.

[41] M. Cao, B. Anderson, S. Morse, and C. Yu, "Control of Acyclic Formations of Mobile Autonomous Agents," *47th IEEE Conference on Decision and Control (CDC)*, pp. 1187–1192, Cancun, Mexico, December 2008.

[42] C. Yu, B. Anderson, S. Dasgupta, and B. Fidan, "Control of Minimally Persistent Formations in the Plane," *SIAM Journal on Control and Optimization*, vol. 48, no. 1, pp. 206–233, February 2009.

[43] L. Krick, M. Broucke, and B. Francis, "Stabilization of Infinitesimally Rigid Formations of Multi-Robot Networks," *International Journal of Control (IJC)*, vol. 82, no. 3, pp. 423–439, February 2009.

[44] F. Dorfler and B. Francis, "Geometric Analysis of the Formation Problem for Autonomous Robots," *IEEE Transactions on Automatic Control (TAC)*, vol. 55, no. 10, pp. 2379–2384, October 2010.

[45] K. Oh and H. Ahn, "Formation Control of Mobile Agents Based on Inter-Agent Distance Dynamics," *Automatica*, vol. 47, no. 10, pp. 2306–2312, October 2011.

[46] M. Cao, S. Morse, C. Yu, B. Anderson, and S. Dasgupta, "Maintaining a Directed, Triangular Formation of Mobile Autonomous Agents," *Communications in Information and Systems*, vol. 11, no. 1, p. 1, November 2011.

[47] T. Summers, C. Yu, S. Dasgupta, and B. Anderson, "Control of Minimally Persistent Leader-Remote-Follower and Coleader Formations in the Plane," *IEEE Transactions on Automatic Control (TAC)*, vol. 56, no. 12, pp. 2778–2792, April 2011.

[48] M. Park, K. Oh, and H. Ahn, "Modified Gradient Control for Acyclic Minimally Persistent Formations to Escape from Collinear Position," *51st IEEE Conference on Decision and Control (CDC)*, pp. 1423–1427, Maui, Hawaii, December 2012.

[49] A. Belabbas, S. Mou, S. Morse, and B. Anderson, "Robustness Issues with Undirected Formations," *51st IEEE Conference on Decision and Control (CDC)*, pp. 1445–1450, December 2012.

[50] K. Oh and H. Ahn, "Distance-Based Undirected Formations of Single-Integrator and Double-Integrator Modeled Agents in n-Dimensional Space," *International Journal of Robust and Nonlinear Control (IJRNC)*, vol. 24, no. 12, pp. 1809–1820, January 2014.

[51] M. Basiri, A. Bishop, and P. Jensfelt, "Distributed Control of Triangular Formations with Angle-Only Constraints," *Systems and Control Letters*, vol. 59, no. 2, pp. 147–154, February 2010.

[52] T. Eren, "Formation Shape Control Based on Bearing Rigidity," *International Journal of Control (IJC)*, vol. 85, no. 9, pp. 1361–1379, May 2012.

[53] M. Trinh, K. Oh, and H. Ahn, "Angle-based Control of Directed Acyclic Formations with Three-Leaders," *2014 International Conference on Mechatronics and Control (ICMC)*, pp. 2268–2271, July 2014.

[54] S. Zhao and D. Zelazo, "Bearing Rigidity and Almost Global Bearing-only Formation Stabilization," *IEEE Transactions on Automatic Control (TAC)*, vol. 61, no. 5, pp. 1255–1268, July 2015.

[55] A. Bishop, M. Deghat, B. D. O. Anderson, and Y. Hong, "Distributed Formation Control with Relaxed Motion Requirements," *International Journal of Robust and Nonlinear Control (IJRNC)*, vol. 25, no. 17, pp. 3210–3230, October 2015.

[56] K. Fathian, D. Rachinskii, M. Spong, and N. Gans, "Globally Asymptotically Stable Distributed Control for Distance and Bearing Based Multi-Agent Formations," *American Control Conference (ACC)*, pp. 4642–4648, Boston, MA, USA, July 2016.

[57] H. Michalska and D. Mayne, "Robust Receding Horizon Control of Constrained Nonlinear Systems," *IEEE Transactions on Automatic Control (TAC)*, vol. 38, no. 11, pp. 1623–1633, December 1993.

[58] H. Chen and F. Allgöwer, "A Quasi-Infinite Horizon Nonlinear Model Predictive Control Scheme with Guaranteed Stability," *Automatica*, vol. 34, no. 10, pp. 1205–1217, October 1998.

[59] D. Mayne, J. Rawlings, C. Rao, and P. Scokaert, "Constrained Model Predictive Control: Stability and Optimality," *Automatica*, vol. 36, no. 6, pp. 789–814, June 2000.

[60] R. Findeisen, L. Imsland, F. Allgöwer, and B. Foss, "State and Output Feedback Nonlinear Model Predictive Control: An Overview," *European Journal of Control (EJC)*, vol. 9, no. 2-3, pp. 190–206, January 2003.

[61] K. Oliveira and M. Morari, "Contractive Model Predictive Control for Constrained Nonlinear Systems," *IEEE Transactions on Automatic Control*, vol. 45, no. 6, pp. 1053–1071, June 2000.

[62] L. Grüne and J. Pannek, "Nonlinear Model Predictive Control," *Springer London*, June 2017.

[63] J. Frasch, A. Gray, M. Zanon, H. Ferreau, S. Sager, F. Borrelli, and M. Diehl, "An Auto-Generated Nonlinear MPC Algorithm for Real-Time Obstacle Cvoidance of Ground Vehicles," *European Control Conference (ECC)*, pp. 4136–4141, Zurich, Switzerland, July 2013.

[64] A. Grancharova and T. Johansen, "Computation, Approximation and Stability of Explicit Feedback min–max Nonlinear Model Predictive Control," *Automatica*, vol. 45, no. 5, pp. 1134–1143, 2009.

[65] G. Pannocchia, J. Rawlings, and S. Wright, "Conditions under which Suboptimal Nonlinear MPC is Inherently Robust," *Systems and Control Letters*, vol. 60, no. 9, pp. 747–755, 2011.

[66] D. Marruedo, T. Alamo, and E. Camacho, "Input-to-State Stable MPC for Constrained Discrete-Time Nonlinear Systems with Bounded Additive Uncertainties," *IEEE Conference on Decision and Control (CDC)*, vol. 4, pp. 4619–4624, Las Vegas, Nevada, USA, December 2002.

[67] G. Pin, M. Davide, L. Magni, and T. Parisini, "Robust Model Predictive Control of Nonlinear Systems with Bounded and State-Dependent Uncertainties," *IEEE Transactions on Automatic Control (TAC)*, vol. 54, no. 7, pp. 1681–1687, June 2009.

[68] E. Kerrigan and J. Maciejowski, "Feedback Min-max Model Predictive Control Using a Single Linear Program: Robust Stability and the Explicit Solution," *International Journal of Robust and Nonlinear Control (IJRNC)*, vol. 14, no. 4, pp. 395–413, January 2004.

[69] D. Limón, T. Alamo, F. Salas, and E. Camacho, "Input to State Stability of Min-Max MPC Controllers for Nonlinear Systems with Bounded Uncertainties," *Automatica*, vol. 42, no. 5, pp. 797–803, February 2006.

[70] D. Raimondo, D. Limon, M. Lazar, L. Magni, and E. Camacho, "Min-max Model Predictive Control of Nonlinear Systems: A Unifying Overview on Stability," *European Journal of Control (EJC)*, vol. 15, no. 1, pp. 5–21, February 2009.

[71] W. Langson, I. Chryssochoos, S. Rakovic, and D. Mayne, "Robust Model Predictive Control Using Tubes," *Automatica*, vol. 40, no. 1, pp. 125–133, January 2004.

[72] D. Mayne, M. Seron, and S. Rakovic, "Robust Model Predictive Control of Constrained Linear Systems with Bounded Disturbances," *Automatica*, vol. 41, no. 2, pp. 219–224, February 2005.

[73] R. Gonzalez, M. Fiacchini, T. Alamo, J. L. Guzmán, and F. Rodríguez, "Online Robust Tube-based MPC for Time-Varying Systems: A Practical Approach," *International Journal of Control (IJC)*, vol. 84, no. 6, pp. 1157–1170, July 2011.

[74] D. Mayne and E. Kerrigan, "Tube-Based Robust Nonlinear Model Predictive Control," *7th IFAC Symposium on Nonlinear Control Systems*, vol. 40, no. 12, pp. 36–41, Pretoria, South Africa, July 2007.

[75] L. Magni, G. Nicolao, R. Scattolini, and F. Allgöwer, "Robust Model Predictive Control for Nonlinear Discrete-time Systems," *International Journal of Robust and Nonlinear Control (IJRNC)*, vol. 13, no. 3-4, pp. 229–246, February 2003.

[76] M. Cannon, J. Buerger, B. Kouvaritakis, and S. Rakovic, "Robust Tubes in Nonlinear Model Predictive Control," *IEEE Transactions on Automatic Control (TAC)*, vol. 56, no. 8, pp. 1942–1947, April 2011.

[77] D. Mayne, E. Kerrigan, E. Wyk, and P. Falugi, "Tube-Based Robust Nonlinear Model Predictive Control," *International Journal of Robust and Nonlinear Control (IJRNC)*, vol. 21, no. 11, pp. 1341–1353, May 2011.

[78] F. Bayer, M. Bürger, and F. Allgöwer, "Discrete-Time Incremental ISS: A Framework for Robust NMPC," *European Control Conference (ECC)*, pp. 2068–2073, Zurich, Switzerland, June 2013.

[79] M. Farina and R. Scattolini, "Tube-based Robust Sampled-Data MPC for Linear Continuous-time Systems," *Automatica*, vol. 48, no. 7, pp. 1473–1476, July 2012.

[80] L. Wang, X. Zhang, and Y. Shu, "Tube-based Robust Model Predictive Control for Constrained Continuous-time Nonlinear Systems," *IEEE Chinese Control and Decision Conference (CCDC)*, pp. 554–559, Yinchuan, China, August 2016.

[81] T. Sun, Y. Pan, J. Zhang, and H. Yu, "Robust Model Predictive Control for Constrained Continuous-time Nonlinear Systems," *International Journal of Control (IJC)*, vol. 91, no. 2, pp. 359–368, February 2018.

[82] S. Loizou and K. J. Kyriakopoulos, "Automatic Synthesis of Multi-Agent Motion Tasks Based on LTL Specifications," *43rd IEEE Conference on Decision and Control (CDC)*, vol. 1, pp. 153–158, Los Angeles, California, USA, December 2004.

[83] T. Wongpiromsarn, U. Topcu, and R. Murray, "Receding Horizon Control for Temporal Logic Specifications," *13th ACM International Conference on Hybrid Systems: Computation and Control (HSCC)*, pp. 101–110, Stockholm, Sweden, April 2010.

[84] A. Ulusoy, S. Smith, X. Ding, C. Belta, and D. Rus, "Optimality and Robustness in Multi-robot Path Planning with Temporal Logic Constraints," *The International Journal of Robotics Research (IJRR)*, vol. 32, no. 8, pp. 889–911, July 2013.

[85] R. Cowlagi and Z. Zhang, "Route Guidance for Satisfying Temporal Logic Specifications on Aircraft Motion," *Journal of Guidance, Control, and Dynamics*, pp. 1–12, October 2016.

[86] J. Liu, N. Ozay, U. Topcu, and R. Murray, "Synthesis of Reactive Switching Protocols from Temporal Logic Specifications," *IEEE Transactions on Automatic Control (TAC)*, vol. 58, no. 7, pp. 1771–1785, February 2013.

[87] C. Belta, B. Yordanov, and E. Gol, "Formal Methods for Discrete-time Dynamical Systems," vol. 89, 2017.

[88] Z. Xu, M. Ornik, A. Julius, and U. Topcu, "Information-guided Temporal Logic Inference with Prior Knowledge," *2019 American Control Conference (ACC)*, pp. 1891–1897, Philadelphia, USA, July 2019.

[89] M. Guo, "Hybrid Control of Multi-robot Systems under Complex Tremporal Tasks," *PhD Thesis, KTH Royal Institute of Technology, Link: http://kth.diva-portal.org/smash/get/diva2:873856/FULLTEXT01.pdf*, December 2015.

[90] Y. Kantaros and M. Zavlanos, "Sampling-based Optimal Control Synthesis for Multi-robot Systems under Global Temporal Tasks," *IEEE Transactions on Automatic Control*, vol. 64, no. 5, pp. 1916–1931, 2018.

[91] A. Bhatia, M. Maly, L. Kavraki, and M. Vardi, "Motion Planning with Complex Goals," *IEEE Robotics and Automation Magazine*, vol. 18, no. 3, pp. 55–64, September 2011.

[92] H. Kress-Gazit, G. Fainekos, and G. Pappas, "Temporal-Logic-Based Reactive Mission and Motion Planning," *IEEE Transactions on Robotics (TRO)*, vol. 25, no. 6, pp. 1370–1381, September 2009.

[93] G. Fainekos, A. Girard, H. Kress-Gazit, and G. Pappas, "Temporal Logic Motion Planning for Dynamic Robots," *Automatica*, vol. 45, no. 2, pp. 343–352, February 2009.

[94] M. Kloetzer and C. Belta, "Automatic Deployment of Distributed Teams of Robots From Temporal Motion Specifications," *IEEE Transactions on Robotics (TRO)*, vol. 26, no. 1, pp. 48–61, December 2009.

[95] M. Kloetzer, X. C. Ding, and C. Belta, "Multi-Robot Deployment from LTL Specifications with Reduced Communication," *50th IEEE Conference on Decision and Control (CDC)*, pp. 4867–4872, Orlando, Florida, December 2011.

[96] Y. Chen, X. Ding, A. Stefanescu, and C. Belta, "A Formal Approach to Deployment of Robotic Teams in an Urban-Like Environment," *Distributed Autonomous Robotic Systems*, pp. 313–327, May 2013.

[97] Y. Chen, X. Ding, A. Stefanescu, and C. Belta, "Formal Approach to the Deployment of Distributed Robotic Teams," *IEEE Transactions on Robotics*, vol. 28, no. 1, pp. 158–171, September 2011.

[98] Y. Kantaros and M. Zavlanos, "A Distributed LTL-Based Approach for Intermittent Communication in Mobile Robot Networks," *American Control Conference*, pp. 5557–5562, Boston, MA, USA, July 2016.

[99] M. Quottrup, T. Bak, and R. Zamanabadi, "Multi-robot Planning: A Timed Automata Approach," *IEEE International Conference on Robotics and Automation (ICRA)*, vol. 5, pp. 4417–4422, New Orleans, USA, May 2004.

[100] M. Andersen, T. Bak, and M. Quottrup, "Motion Planning in Multi-robot Systems Using Timed Automata," *Proceedings of the 5th IFAC/EURON Symposium on Intelligent Autonomous Vehicles*, 2004.

[101] K. Larsen, P. Pettersson, and W. Yi, "UPPAAL in a Nutshell," *International Journal on Software Tools for Technology Transfer*, vol. 1, no. 1, pp. 134–152, February 1997.

[102] A. Ulusoy, S. Smith, X. Ding, C. Belta, and D. Rus, "Optimal Multi-Robot Path Planning with Temporal Logic Constraints," *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 3087–3092, San Francisco, California, USA, October 2011.

[103] J. Fu and U. Topcu, "Computational Methods for Stochastic Control with Metric Interval Temporal Logic Specifications," *54th IEEE Conference on Decision and Control (CDC)*, pp. 7440–7447, Osaka, Japan, December 2015.

[104] B. Hoxha and G. Fainekos, "Planning in Dynamic Environments Through Temporal Logic Monitoring," *13th AAAI Conference on Artificial Intelligence*, Phoenic, Arizona, USA, September 2016.

[105] Y. Zhou, D. Maity, and J. S. Baras, "Timed Automata Approach for Motion Planning Using Metric Interval Temporal Logic," *European Control Conference (ECC)*, pp. 690–695, Aalborg, Denmark, June 2016.

[106] Y. Zhou, D. Maity, and J. S. Baras, "Optimal Mission Planner with Timed Temporal Logic Constraints," *European Control Conference (ECC)*, pp. 759–764, Linz, Austria, June 2015.

[107] C. Constantinou and S. Loizou, "Automatic Controller Synthesis of Motion-tasks with Real-time Objectives," *2018 IEEE Conference on Decision and Control (CDC)*, pp. 403–408, Miami, Florida, USA, December 2018.

[108] C. Vrohidis, P. Vlantis, C. Bechlioulis, and K. J. Kyriakopoulos, "Prescribed Time Scale Robot Navigation in Dynamic Environments," *European Control Conference (ECC)*, pp. 1803–1808, Limassol, Cyprus, June 2018.

[109] C. Vrohidis, P. Vlantis, C. Bechlioulis, and K. J. Kyriakopoulos, "Prescribed Time Scale Robot Navigation," *IEEE Robotics and Automation Letters*, vol. 3, no. 2, pp. 1191–1198, 2018.

[110] R. Alur, T. Henzinger, G. Lafferriere, and G. Pappas, "Discrete Abstractions of Hybrid Systems," *Proceedings of the IEEE*, vol. 88, no. 7, pp. 971–984, July 2000.

[111] M. Zamani, G. Pola, M. Mazo, and P. Tabuada, "Symbolic Models for Nonlinear Control Systems without Stability Assumptions," *IEEE Transactions on Automatic Control (TAC)*, vol. 57, no. 7, 2012.

[112] M. Zamani, M. Mazo, and A. Abate, "Finite Abstractions of Networked Control Systems," *53rd IEEE Conference on Decision and Control (CDC)*, pp. 95–100, Los Angeles, California, USA, 2014.

[113] J. Liu and P. Prabhakar, "Switching Control of Dynamical Systems from Metric Temporal Logic Specifications," *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 5333–5338, Hong Kong, May 2014.

[114] P. Meyer, A. Girard, and E. Witrant, "Compositional Abstraction and Safety Synthesis Using Overlapping Symbolic Models," *IEEE Transaction on Automatic Control (TAC)*, 2017.

[115] O. Hussein, A. Ames, and P. Tabuada, "Abstracting Partially Feedback Linearizable Systems Compositionally," *IEEE Control Systems Letters, vol. 1, no. 2, pp. 227-232*, October 2017.

[116] D. Boskos and D. V. Dimarogonas, "Decentralized Abstractions For Multi-Agent Systems Under Coupled Constraints," *54th IEEE Conference on Decision and Control (CDC)*, pp. 282–287, Osaka, Japan, December 2015.

[117] P. Nilsson and N. Ozay, "Synthesis of Separable Controlled Invariant Sets for Modular Local Control Design," *Boston, MA, USA, American Control Conference (ACC)*, July 6-8, 2016.

[118] R. Horn and C. Johnson, "Topics in Matrix Analysis," *Cambridge University Press*, April 1994.

[119] I. Kolmanovsky and E. Gilbert, "Theory and Computation of Disturbance Invariant Sets for Discrete-time Linear Systems," *Mathematical Problems in Engineering*, vol. 4, no. 4, pp. 317–367, 1998.

[120] A. Zemouche, N. Boutayeb, and G. Bara, "Observers for a Class of Lipschitz Systems with Extension to H-inifinity Performance Analysis," *Systems and Control Letters*, vol. 57, no. 1, pp. 18–27, January 2008.

[121] A. Zemouche, M. Boutayeb, and I. Bara, "Observer Design for Nonlinear Systems: An Approach Based on the Differential Mean Value Theorem.," *44th IEEE Conference on Decision and Control (CDC)*, pp. 6353–6358, Seville, Spain, December 2005.

[122] H. K. Khalil, "Nonlinear Systems," *Prentice Hall*, 2002.

[123] S. Yu, C. Maier, H. Chen, and F. Allgöwer, "Tube MPC Scheme Based on Robust Control Invariant Set with Application to Lipschitz Nonlinear Systems," *Systems and Control Letters*, vol. 62, no. 2, pp. 194–200, February 2013.

[124] E. Sontag, "Input to State Stability: Basic Concepts and Results," *Nonlinear and Optimal Control Theory*, pp. 163–220, June 2008.

[125] E. Sontag, "Mathematical Control Theory: Deterministic Finite Dimensional Systems," *Springer Science and Business Media*, vol. 6, November 2013.

[126] C. Bechlioulis and G. Rovithakis, "Robust Adaptive Control of Feedback Linearizable MIMO Nonlinear Systems with Prescribed Performance," *IEEE Transactions on Automatic Control (TAC)*, vol. 53, no. 9, pp. 2090–2099, October 2008.

[127] C. Bechlioulis and G. Rovithakis, "A Low-complexity Global Approximation-free Control Scheme with Prescribed Performance for Unknown Pure Feedback Systems," *Automatica*, vol. 50, no. 4, pp. 1217 – 1226, 2014.

[128] M. Mesbahi and M. Egerstedt, "Graph Theoretic Methods in Multiagent Networks," *Princeton University Press*, July 2010.

[129] R. Alur and D. Dill, "A Theory of Timed Automata," *Theoretical Computer Science*, vol. 126, no. 2, pp. 183–235, April 1994.

[130] C. Baier, J. Katoen, and K. Larsen, "Principles of model checking," *MIT Press*, April 2008.

[131] D. Souza and P. Prabhakar, "On the Expressiveness of MTL in the Pointwise and Continuous Semantics," *International Journal on Software Tools for Technology Transfer*, vol. 9, no. 1, pp. 1–4, February 2007.

[132] J. Ouaknine and J. Worrell, "On the Decidability of Metric Temporal Logic," *20th Annual IEEE Symposium on Logic in Computer Science (LICS)*, pp. 188–197, September 2005.

[133] M. Reynolds, "Metric Temporal Logics and Deterministic Timed Automata," 2010.

[134] P. Bouyer, "From Qualitative to Quantitative Analysis of Timed Systems," *Mémoire D ' habilitation, Université Paris*, vol. 7, pp. 135–175, January 2009.

[135] S. Tripakis, "Checking Timed Büchi Automata Emptiness on Simulation Graphs," *ACM Transactions on Computational Logic (TOCL)*, vol. 10, no. 3, April 2009.

[136] R. Alur, T. Feder, and T. Henzinger, "The Benefits of Relaxing Punctuality," *Journal of the ACM (JACM)*, vol. 43, no. 1, pp. 116–146, September 1996.

[137] O. Maler, D. Nickovic, and A. Pnueli, "From MITL to Timed Automata," *International Conference on Formal Modeling and Analysis of Timed Systems (FORMATS)*, pp. 274–289, September 2006.

[138] D. Nickovic and N. Piterman, "From MTL to Deterministic Timed Automata," *International Conference on Formal Modeling and Analysis of Timed Systems (FORMATS)*, September 2010.

[139] W. Ren and R. Beard, "Consensus Seeking in Multi-agent Systems Under Dynamically Changing Interaction Topologies," *IEEE Transactions on Automatic Control (TAC)*, vol. 50, no. 5, pp. 655–661, May 2005.

[140] R. Olfati-Saber and R. Murray, "Consensus Problems in Networks of Agents with Switching Topology and Time-Delays," *IEEE Transactions on Automatic Control (TAC)*, vol. 49, no. 9, pp. 1520–1533, September 2004.

[141] A. Jadbabaie, J. Lin, and S. Morse, "Coordination of Groups of Mobile Autonomous Agents Using Nearest Neighbor Rules," *IEEE Transactions on Automatic Control (TAC)*, vol. 48, no. 6, pp. 988–1001, June 2003.

[142] H. Tanner, A. Jadbabaie, and G. Pappas, "Flocking in Fixed and Switching Networks," *IEEE Transactions on Automatic Control (TAC)*, vol. 52, no. 5, pp. 863–868, May 2007.

[143] M. Egerstedt and X. Hu, "Formation Constrained Multi-agent Control," *IEEE Transactions on Robotics and Automation (TRA)*, vol. 17, no. 6, pp. 947–951, December 2001.

[144] M. Zavlanos and G. Pappas, "Distributed Connectivity Control of Mobile Networks," *IEEE Transactions on Robotics (TRO)*, vol. 24, no. 6, pp. 1416–1428, December 2008.

[145] K. Oh, M. Park, and H. Ahn, "A Survey of Multi-agent Formation Control," *Automatica*, vol. 53, pp. 424–440, March 2015.

[146] C. Bechlioulis and K. J. Kyriakopoulos, "Robust Model-Free Formation Control with Prescribed Performance and Connectivity Maintenance for Nonlinear Multi-Agent Systems," *53rd IEEE Conference on Decision and Control (CDC)*, pp. 4509–4514, Los Angeles, California, USA, December 2014.

[147] D. Panagou, "A Distributed Feedback Motion Planning Protocol for Multiple Unicycle Agents of Different Classes," *IEEE Transactions on Automatic Control (TAC)*, vol. 62, no. 3, pp. 1178–1193, June 2016.

[148] S. Loizou, "The Multi-agent Navigation Transformation: Tuning-free Multi-robot Navigation," *Robotics: Science and Systems*, vol. 6, pp. 1516–1523, June 2014.

[149] S. Loizou, "The Navigation Transformation," *IEEE Transactions on Robotics (TRO)*, vol. 33, no. 6, pp. 1516–1523, July 2017.

[150] D. Shim, H. J. Kim, and S. Sastry, "Decentralized Nonlinear Model Predictive Control of Multiple Flying Robots," *IEEE Conference on Decision and Control (CDC)*, pp. 3621–3626, Hawaii, USA, December 2003.

[151] H. Fukushima, K. Kon, and F. Matsuno, "Distributed Model Predictive Control for Multi-vehicle Formation with Collision Avoidance Constraints," *IEEE Conference on Decision and Control (CDC)*, pp. 5480–5485, Seville, Spain, December 2005.

[152] A. Richards and J. How, "Robust Distributed Model Predictive Control," *International Journal of control (IJC)*, vol. 80, no. 9, pp. 1517–1531, September 2007.

[153] W. Dunbara and R. Murray, "Distributed Receding Horizon Control for Multi-Vehicle Formation Stabilization," *Automatica*, vol. 42, no. 4, pp. 549–558, April 2006.

[154] A. Rucco, P. Aguiar, F. Fontes, F. Pereira, and J. B. Sousa, "A Model Predictive Control-based Architecture for Cooperative Path-following of Multiple Unmanned Aerial Vehicles," *Developments in Model-Based Optimization and Control, Springer Series*, vol. 464, pp. 141–160, December 2015.

[155] B. Hernandez and P. Trodden, "Distributed Model Predictive Control Using a Chain of Tubes," *11th International Conference on Control (ICC)*, Belfast, UK, November 2016.

[156] A. Richards and P. Trodden, "Robust Distributed Model Predictive Control Using Tubes," *American Control Conference (ACC)*, pp. 1619–1620, Minneapolis, USA, July 2006.

[157] M. Müller, M. Reble, and F. Allgöwer, "Cooperative Control of Dynamically Decoupled Systems via Distributed Model Predictive Control," *International Journal of Robust and Nonlinear Control (IJRNC)*, vol. 22, no. 12, pp. 1376–1397, May 2012.

[158] J. Park, S. Huh, S. Kimn, S. Seo, and G. Park, "Direct Adaptive Controller for Nonaffine Nonlinear Systems Using Self-Structuring Neural Networks," *IEEE Transactions on Neural Networks*, vol. 16, no. 2, pp. 414–422, 2005.

[159] S. Karaman and E. Frazzoli, "Linear Temporal Logic Vehicle Routing with Applications to Multi-UAV Mission Planning," *International Journal of Robust and Nonlinear Control*, vol. 21, no. 12, pp. 1372–1395, May 2011.

[160] S. Karaman and E. Frazzoli, "Vehicle Routing Problem with Metric Temporal Logic Specifications," *47th IEEE Conference on Decision and Control (CDC)*, pp. 3953–3958, Cancun, Mexico, December 2008.

[161] P. Tabuada, "Verification and Control of Hybrid Systems: a Symbolic Approach," *Springer Science and Business Media*, 2009.

[162] A. Girard and G. J. Pappas, "Approximation Metrics for Discrete and Continuous Systems," *IEEE Transactions on Automatic Control (TAC)*, vol. 52, no. 5, pp. 782–798, May 2007.

[163] T. Brihaye, M. Estiévenart, and G. Geeraerts, "On MITL and Alternating Timed Automata," *International Conference on Formal Modeling and Analysis of Timed Systems (FORMATS)*, vol. 8053, pp. 47–61, June 2013.

[164] S. Ahlberg, "Human-in-the-loop Control Synthesis for Multi-agent Systems under Metric Interval Temporal Logic Specifications," *Litentiate Thesis, KTH Royal Institute of Technology, Link: http://www.diva-portal.org/smash/get/diva2:1316374/FULLTEXT01.pdf*, June 2019.

[165] M. Guo and D. V. Dimarogonas, "Multi-agent Plan Reconfiguration Under Local LTL Specifications," *The International Journal of Robotics Research (IJRR)*, vol. 34, no. 2, pp. 218–235, November 2015.

[166] A. Filotheou, "Robust Decentralized Control of Cooperative Multi-robot Systems: An Inter-constraint Receding Horizon Approach," *Master Thesis, KTH Royal Institute of Technology, Link: http://kth.diva-portal.org/smash/get/diva2:1102597/FULLTEXT02.pdf*, May 2017.

[167] M. Quigley, K. Conley, B. P. Gerkey, J. Faust, T. Foote, J. Leibs, R. Wheeler, and A. Y. Ng, "ROS: An Open-Source Robot Operating System," *International Conference on Robotics and Automation (ICRA) Workshop*, 2009.

[168] S. G. Johnson, "The NLopt Nonlinear-Optimization Package," *http://ab-initio.mit.edu/nlopt*, 2009.

[169] D. Liberzon, "Switching in Systems and Sontrol," *Springer Science and Business Media*, 2012.

[170] D. Boskos and D. V. Dimarogonas, "Robust Connectivity Analysis for Multi-Agent Systems," *54th IEEE Conference on Decision and Control (CDC)*, pp. 6767–6772, Osaka, Japan, December 2015.

[171] M. M. Fiedler, "Algebraic Connectivity of Graphs," *Czechoslovak Mathematical Journal*, vol. 23, no. 2, pp. 298–305, 1973.

[172] M. Krstic, I. Kanellakopoulos, and P. Kokotovic, "Nonlinear and Adaptive Control Design," *Publisher: Wiley New York*, 1995.

[173] C. Bechlioulis, G. Karras, S. Heshmati-alamdari, and K. J. Kyriakopoulos, "Trajectory Tracking with Prescribed Performance for Underactuated Underwater Vehicles Under Model Uncertainties and External Disturbances," *IEEE Transactions on Control Systems Technology (TCST)*, vol. 25, no. 2, pp. 429–440, March 2017.