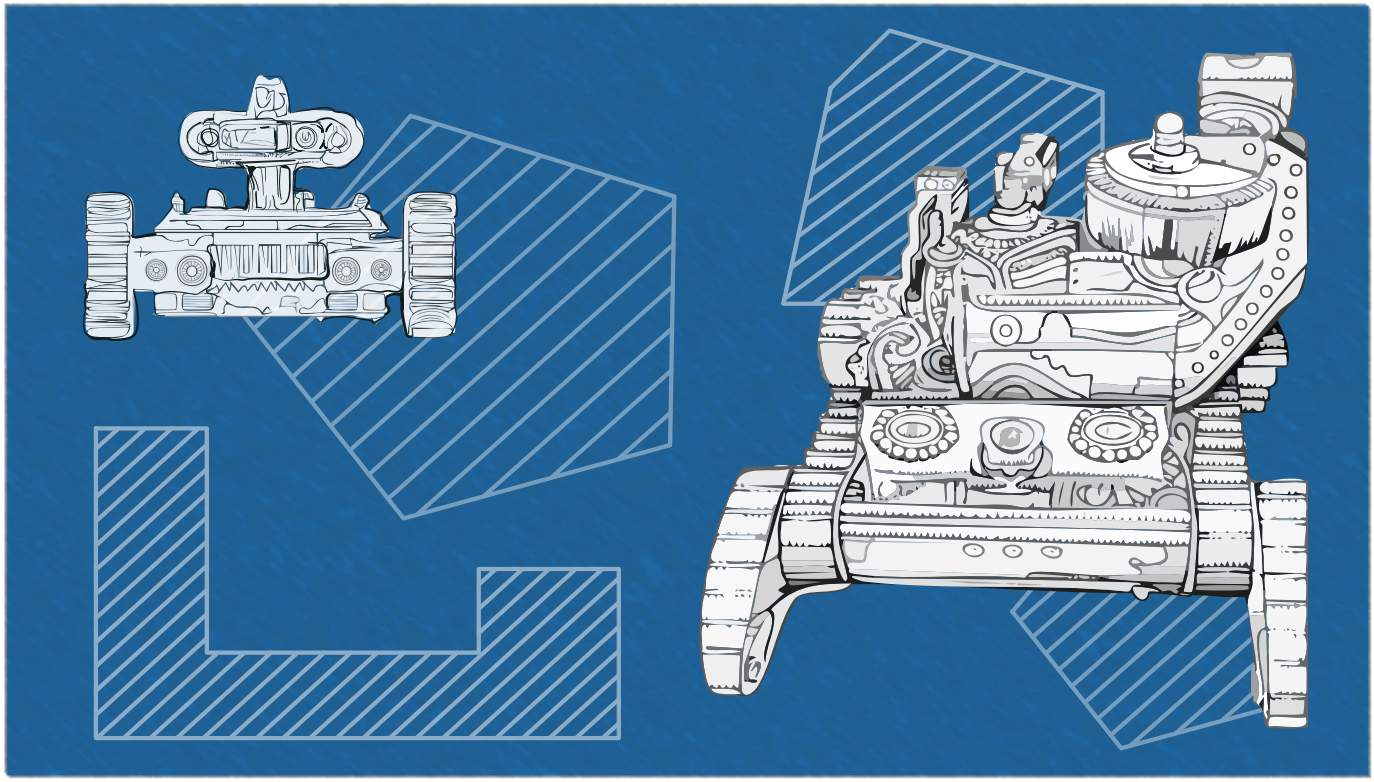
Scuola di Dottorato

Dottorato di Ricerca in Ingegneria dell'Informazione

Alessia Ferraro

SET-THEORETIC MOTION PLANNING FOR SKID-STEEREDTRACKED ROBOTS

From Single to Coordinated Multi-Robot Systems under Constraint, Uncertainties, and Disturbances in Complex Scenarios



Scuola di Dottorato

SCUOLA di DOTTORATO

Dottorato in Architettura
Dottorato in Diritto ed Economia
Dottorato in Ingegneria Civile, Ambientale e Industriale
Dottorato in Ingegneria dell'Informazione
Dottorato in Scienze Agrarie, Alimentari e Forestali

Direttore della Scuola di Dottorato
prof. Felice Arena

Collegio dei docenti
Dottorato di Ricerca in Ingegneria dell'Informazione
XXXVIII ciclo

Coordinatore ISERNIA Tommaso –

ARANITI Giuseppe
BEVACQUA Martina Teresa
BUCCAFURRI Francesco Antonio
CAMPOLO Claudia
CAROTENUTO Riccardo
COPPOLA Giuseppe
COTRONEI Mariantonia
CROCCO Lorenzo
DALLET Dominique
DE CAPUA Claudio
DELLA CORTE Francesco Giuseppe
DI DONATO Loreto
FAGGIO Giuliana
FAILLA Gioia
FISICHELLA Marco
GIUFFRE' Sofia
GRADITI Giorgio
LAX Gianluca
LAY-EKUAKILLE Aimè
MAMMONE Nadia
MANZONI Pietro
MERCURI Marco
MERENDA Massimo
MESSINA Giacomo
MISTRETTA Marina
MOLINARO Antonella
MORABITO Andrea Francesco
MORABITO Francesco Carlo
MORELLO Rosario
MUNTEAN Gabriel-Miro
MUSOLINO Giuseppe
PALMERI Roberta
PRATICO' Filippo Gianmaria
ROSACI Domenico
RUGGERI Giuseppe
RUSSO Francesco
RUSSO Mariateresa
VINEL Alexey

In copertina
Illustrazione di Skid-steered Tracked Robots.

Alessia Ferraro è una dottoranda di ricerca in Ingegneria dell'Informazione presso l'Università degli Studi Mediterranea di Reggio Calabria. La sua attività di ricerca si focalizza sulla pianificazione del movimento tramite approcci innovativi set-theoretic per skid-steered tracked mobile robots. Nell'ambito della tesi si è occupata dello sviluppo di algoritmi per il motion planning per skid-steered mobile robot in caso di singole pattiaforme e nel caso di sistemi multi-robot coordinati, operanti in scenari dinamici e complessi caratterizzati da forti vincoli, incertezze e disturbi ambientali.

SCUOLA DI DOTTORATO
Università Mediterranea di Reggio Calabria

DIPARTIMENTO
Ingegneria dell'Informazione, delle Infrastrutture
e dell'Energia Sostenibile (DIIES)

DOTTORATO DI RICERCA
Dottorato in Ingegneria dell'Informazione

S.S.D. ING-INF/04
XXXVIII CICLO

SET-THEORETIC MOTION PLANNING FOR SKID-STEERED TRACKED ROBOTS

From Single to Coordinated Multi-Robot Systems under Constraints, Uncertainties, and Disturbances in Complex Scenarios

Dottoranda
Alessia Ferraro
[redacted]

Supervisori
prof. Claudio De Capua
prof. Valerio Scordamaglia
[redacted]

Coordinatore del Dottorato
prof. Tommaso Isernia
[redacted]

Acknowledgments

This study was carried out under the financial support of the project *CHEMSYS: Cooperative Heterogeneous Multi-drone SYStem for disaster prevention and first response* (CUP C53D23008330001, project code P2022XER7W), funded under the PRIN 2022 PNRR program and developed at the *Mediterranean University of Reggio Calabria*.

First of all, I would like to express my deepest gratitude to my doctoral supervisors, Prof. Claudio De Capua and Prof. Valerio Scordamaglia, for their constant guidance, encouragement, and insightful advice during my doctoral studies. Their expertise, professionalism, and support have been instrumental in shaping my research path and helping me grow both academically and personally.

During my research period abroad, I had the opportunity to collaborate with Prof. Francesca Boem, whose supervision and scientific insight were invaluable during my stay. I am sincerely grateful to her for her availability and constructive feedback.

I would also like to extend my appreciation to all the co-authors who have contributed to my scientific works during this Ph.D. program. In particular, Prof. Giuseppe Franzè and Francesco Tedesco from the University of Calabria; Prof. Egidio D'Amato and Salvatore Rosario Bassolillo from the University Parthenope of Naples; Immacolata Notaro from the University of Campania "Luigi Vanvitelli"; and Prof. Michele Buonsanti, Vito Antonio Nardi, Luana Gurnari, Filippo Ruffa, and Prof. Fabio Filianoti from the University "Mediterranea" of Reggio Calabria. Their collaboration, insights, and professionalism have greatly enriched my research experience and contributed to the outcomes of this work.

Finally, my deepest gratitude goes to my family and my partner, for their unconditional love, patience, and continuous encouragement throughout these years. Their support has been the foundation that made every achievement possible.

Alessia Ferraro. Reggio Calabria, Italy.

Abstract

The increasingly widespread use of autonomous mobile robots in scenarios such as surveillance, search and rescue, and logistics requires navigation strategies that guarantee safety, robustness, and feasibility even in the presence of stringent constraints, external disturbances, model uncertainties, and communication delays.

This research focuses on the development of set-theoretic methodologies for trajectory and motion planning of remotely controlled skid-steered tracked mobile robots, with the aim of ensuring the feasibility and safety of navigation in the presence of model uncertainties, external disturbances, and actuation constraints in complex environments. Two methods are proposed that solve the problem of safe navigation in the presence of model uncertainties, external disturbances, and actuation constraints for the single-robot case, and two methods that solve the problem of safe navigation in the presence of model uncertainties, external disturbances, and actuation constraints for multi-robot systems, to which are added the crucial challenges of coordination management and collision prevention to maintain safe and cooperative operations.

The main contributions of the work are:

(i) A **Set-Theoretic Feasible Trajectory Planning (ST-FTP)** Method for skid-steered tracked mobile robots subject to sliding phenomena and controlled through a communication network (NCS) based on graphs and feasibility conditions formulated as semidefinite programming (SDP) problems with Linear Matrix Inequalities (LMIs) constrains; the objective of the method is to generate trajectories composed of a sequence of straight segments, each to be traveled at a desired speed, which ensure that the robot maintains a predetermined safety distance from any obstacle in the operating environment, regardless of the model uncertainties, the external disturbances, and the actuation constraints, guaranteeing that these trajectories are dynamically achievable by the system.

(ii) A **Set-Theoretic Model Predictive Control (ST-MPC)** Method that addresses the problem by ensuring safe navigation from the initial state to the target state, while managing uncertainties, external disturbances, and network-induced delays using set-theoretic arguments. Model predictive control (MPC) based on set-theoretic arguments is capable of handling unexpected obstacles, network delays, and dynamic uncertainties, ensuring compliance with constraints and safety. The contribution exploits the ideas of receding horizon set-based control and customizes them according to the proposed framework, in which at every moment in time, information about the obstacle-free region is subject to inevitable latency phenomena.

(iii) A **Set-Theoretic Feasible Coordinated Trajectory Planning (ST-FCTP)** Method that aims to plan optimal trajectories that robustly guarantee the absence of colli-

sions during the movement of the robots along assigned feasible trajectories, explicitly considering the presence of errors in the trajectory tracking. The coordinated planning method for multi-robot systems integrates disturbances and uncertainties in the planning phase through a graph search problem and LMI constraints; The key aspect of the contribution leads in the fact that the collision avoidance aspects are addressed by resorting to set-theoretic arguments, while the trajectory planning aspects are reformulated in terms of solving a mixed graph search problem and solving SDP problems with constraints expressed by LMIs constrains.

(iv) A **Set-theoretic Time-based Trajectory Synchronization (ST-TBTS)** methodology for temporal synchronization of trajectories, formulated as an LMI optimization problem, which minimizes the occupancy of the shared common spaces while ensuring feasibility and robustness. Collision avoidance and coordination are achieved through synchronization of robot trajectories by imposing delays on platform departures. The procedure exploits the trajectory's feasibility property by solving optimization problems involving LMIs constraints. The algorithm operates in separate offline and online phases. In the offline phase, the time delay intervals useful for synchronization are calculated based on the planned feasible trajectories. Then, in the online phase, the actual delay is calculated by solving a linear programming (LP) optimization problem that minimizes the occupancy time of the shared operational space.

All the proposed methodologies have been validated through extensive numerical simulations and real-world experiments using skid-steered tracked mobile robots available at the Automatic Control Laboratory of the Mediterranean University of Reggio Calabria.

Keywords: Safe navigation, Trajectory planning, Motion Planning, Predictive control, Multi-robot coordination, Set-theoretic methods, Skid-steered Tracked Mobile Robot.

Sommario

L'uso sempre più diffuso di robot autonomi in scenari quali sorveglianza, ricerca e soccorso e logistica richiede strategie di navigazione che garantiscano sicurezza, robustezza e fattibilità anche in presenza di vincoli di attuazione rigorosi, disturbi esterni, incertezze di modello e ritardi di comunicazione.

Questa ricerca si concentra sullo sviluppo di metodologie basate sulla teoria degli insiemi (set-theoretic) per la pianificazione della traiettoria e del movimento di robot mobili cingolati a guida differenziale controllati da remoto con l'obiettivo di garantire la feasibility e la sicurezza della navigazione in presenza di incertezze del modello, disturbi esterni e vincoli di attuazione in ambienti complessi. Vengono proposti due metodi che risolvono il problema della navigazione sicura in presenza di incertezze del modello, disturbi esterni e vincoli di attuazione per il caso di un singolo robot, e due metodi che risolvono il problema della navigazione sicura in presenza di incertezze del modello, disturbi esterni e vincoli di attuazione per sistemi multi-robot, ai quali si aggiungono le sfide cruciali della gestione del coordinamento e della prevenzione delle collisioni.

I principali contributi del lavoro possono essere riassunti come segue:

(i) Un metodo **Set-Theoretic Feasible Trajectory Planning (ST-FTP)** per la pianificazione di traiettorie feasible per robot cingolati a guida differenziale soggetti a fenomeni di slittamento, e controllati attraverso una rete di comunicazione, basato su algoritmi di ricerca su grafo e condizioni di feasibility formulate come problemi di programmazione semidefinita (SDP)con vincoli formulati sottoforma di disequazioni lineari di matrici (LMI); l'obiettivo del metodo è quello di generare traiettorie composte da una sequenza di segmenti rettilinei, ciascuno da percorrere alla velocità desiderata assegnata, che assicurano che il robot mantenga una distanza di sicurezza predeterminata da qualsiasi ostacolo nell'ambiente operativo, indipendentemente dalle incertezze del modello, dai disturbi esterni e dai vincoli di attuazione, garantendo che tali traiettorie siano dinamicamente realizzabili dal sistema.

(ii) Un metodo **Set-Theoretic Model Predictive Control (ST-MPC)** che affronta il problema garantendo una navigazione sicura dallo stato iniziale allo stato finale, gestendo al contempo le incertezze di modello, i disturbi esterni e i ritardi indotti dalla rete di comunicazione, utilizzando argomenti set-theoretic. Il controllo predittivo set-theoretic è capace di gestire ostacoli imprevisti, ritardi di rete e incertezze dinamiche, garantendo il rispetto dei vincoli e la sicurezza. Il contributo sfrutta le idee del controllo basato su insiemi a orizzonte recedente e le personalizza in base al quadro proposto, in cui in ogni momento le informazioni sulla regione priva di ostacoli sono soggette a inevitabili fenomeni di latenza.

*(iii)*Un metodo **Set-Theoretic Feasible Coordinated Trajectory Planning (ST-FCTP)** per sistemi multirobot che mira a pianificare traiettorie ottimali coordinate che

garantiscano in modo robusto l'assenza di collisioni durante il movimento dei robot lungo traiettorie fattibili assegnate, considerando esplicitamente la presenza di errori nel tracciamento della traiettoria. Il metodo di pianificazione coordinata per sistemi multi-robot integra disturbi e incertezze nella fase di pianificazione attraverso un problema di ricerca su grafo e vincoli LMI; l'aspetto chiave del contributo risiede nel fatto che gli aspetti relativi alla prevenzione delle collisioni vengono affrontati ricorrendo ad argomenti set-theoretic, mentre gli aspetti relativi alla pianificazione della traiettoria vengono riformulati in termini di risoluzione di un problema di ricerca su grafo e di risoluzione di problemi SDP con vincoli espressi da LMI.

(iv) Un metodo **Set-theoretic Time-based Trajectory Synchronization (ST-TBTS)** per la sincronizzazione temporale delle traiettorie, formulata come un problema di ottimizzazione LMI, che riduce al minimo l'occupazione degli spazi comuni condivisi garantendo al contempo fattibilità e robustezza. L'evitamento delle collisioni e il coordinamento delle piattaforme sono ottenuti attraverso la sincronizzazione delle traiettorie dei robot schedulandone opportunamente le partenze. La procedura sfrutta la proprietà di feasibility della traiettoria risolvendo problemi di ottimizzazione che coinvolgono vincoli LMI. L'algoritmo opera in fasi separate offline e online. Nella fase offline, gli intervalli di ritardo utili per la sincronizzazione vengono calcolati sulla base delle traiettorie feasible pre-pianificate. Successivamente, nella fase online, il ritardo effettivoimposto per ogni piattaforma viene calcolato risolvendo un problema di ottimizzazione che riduce al minimo il tempo di occupazione dello spazio operativo condiviso.

Tutte le metodologie proposte sono state validate tramite simulazioni numeriche e sperimentali con piattaforme reali disponibili presso il Laboratorio di Controlli Automatici dell'Università Mediterranea di Reggio Calabria.

Parole chiave: Navigazione sicura, Pianificazione di traiettoria, Pianificazione di movimento, Controllo predittivo, Coordinamento multi-robot, Metodi set-theoretic, Robot cingolati a guida differenziale.

Contents

1 Introduction . . . 1

1.1 Motivation . . . 1

1.2 Literature Review . . . 2

1.2.1 Path planning . . . 2

1.2.2 Trajectory Planning . . . 3

1.2.3 Set-theoretic trajectory planning methods . . . 4

1.2.4 Trajectory tracking . . . 5

1.2.5 Multi-Robot Systems . . . 6

1.2.6 Path planning for MRS . . . 7

1.2.7 Reactive control strategies for collision avoidance in MRS . . . 9

1.2.8 Time-base synchronization methods for MRS . . . 9

1.3 Objectives and Methodologies . . . 10

1.3.1 Research Objectives . . . 10

1.3.2 Methodologies . . . 12

1.3.3 Publications . . . 13

1.4 Thesis Outline . . . 14

2 Mathematical Modeling of the Robotic Platform . . . 15

2.1 Skid-steered tracked mobile robots . . . 15

2.2 Robot mathematical model . . . 16

2.2.1 Kinematics Model of Mobile Robot . . . 16

2.2.2 Sliding phenomena . . . 17

2.3 Remote Control Architecture for Mobile Robots . . . 19

3 Set-theoretic Navigation Methods for Single Robot . . . 21

3.1 Problem formulation . . . 22

3.2 Set-Theoretic Feasible Trajectory Planning Method . . . 27

3.2.1 Trajectory Tracking Error . . . 27

3.2.2 Modeling of networked control system . . . 30

3.2.3 Trajectory tracking control design . . . 31

3.2.4 Definition of the positive robust D-invariant region 32
3.2.5 Trajectory feasibility condition 34
3.2.6 Algorithm for feasible trajectory planning 37
3.2.7 Results — Experimental Validation 38
3.3 Set-Theoretic Model Predictive Control Method 45
3.3.1 Linear embedding of the nonlinear kinematics 48
3.3.2 Management of delays introduced by the communication network . . 49
3.3.3 The MPC controller 50
3.3.4 Robust positively invariant (RPI) sets 50
3.3.5 Robust one-step state ahead controllable (ROSAC) sets 52
3.3.6 Overlapped RPIs 54
3.3.7 Controller design 55
3.3.8 Path planning in $\mathcal{O}^{free}$ 56
3.3.9 A set-theoretic model predictive control scheme 58
3.3.10 Results — Experimental Validation 60
3.3.11 Energy consumption analysis 64
3.3.12 Computational complexity analysis 67

4 Set-theoretic Navigation Methods for Multi-Robot Systems 69
4.1 Problem Formulation 70
4.2 A set-theoretic Feasible Coordinated Trajectory Planning Method 75
4.2.1 Norm-bounded LDI of the trajectory tracking error dynamics 76
4.2.2 Closed-loop trajectory tracking error dynamics 76
4.2.3 Definition of the positive robust D-invariant region 77
4.2.4 Trajectory feasibility 78
4.2.5 Coordinated feasible trajectories planning algorithm 79
4.2.6 Results—Experimental Validation 81
4.3 Set-theoretic Time-based Trajectory Synchronization Method 91
4.3.1 Closed loop trajectory tracking error dynamics 92
4.3.2 Definition of the positive robust D-invariant region 92
4.3.3 Set-theoretic time-based synchronization method 93
4.3.4 Algorithm for trajectory synchronization 94
4.3.5 Results—Numerical Validation 99

5 Conclusions 107
5.1 Summary 107
5.2 Discussion 108
5.3 Future Research and Challenges 109

References 119

List of Figures

2.1 Robot Kinematics. 18
2.2 Outline of a networked control system (NCS) scheme. 20

3.1 Aisle/Corridor polyhedral representation 23
3.2 Reference frames. Robot pose in the **E** frame is denoted as $\{x, y, \theta\}$, the **L** reference frame for trajectory tracking has its origin in M with the x-axis oriented in accordance with the assigned trajectory segment $\overline{MN}$. 28
3.3 Temporal diagram of actuation update 30
3.4 Trajectory switch between segments $\overline{AB}$ and $\overline{BC}$. $x^D(t_{N_{AB}})(y^D(t_{N_{AB}}))$ represents the nominal robot position in **E** at time $t_{N_{AB}}$ 36
3.5 Flowchart of the feasible trajectory planning algorithm 39
3.6 Skid-Steered Tracked Mobile Robot Jaguar V4 by Dr.Robot 40
3.7 Control framework details: trajectory tracking control action is performed on a remote computer connected to the mobile robot through a Wi-Fi communication network using TCP-IP protocol stack. 40
3.8 The planned feasible trajectory (green line) connecting the starting point (red square) to the ending point (red triangle) is overlaid the original floor plan. The red line represents the boundary of the operational scenario measured using the SLAM algorithm. Black solid lines represent the trajectories performed. 42
3.9 Red dashed lines represent bounds. Black solid lines represent trajectory tracking errors. 42
3.10 Red dashed lines represent bounds. Black solid lines represent control speeds. 43
3.11 Red dashed line represents the bound condition. Black solid lines represent the values of $\|\xi\|^2_{P_0}$ related to D-invariant region (3.44). 43
3.12 Control architecture 51
3.13 Obstacle scenario $\mathcal{O}^{t_k}$ and operating conditions: the red star and green hexagon denote the initial and target poses, respectively. 57
3.14 State trajectory tube (3.129). 57

3.15 Indoor operating environment 61
3.16 Networked control architecture: SLAM, Controller and WIFI units. 62
3.17 Cumulative time-delay realizations. 63
3.18 The green hexagon denotes the target q_E^f, while the red star indicates the initial condition $q_E(0)$. 63
3.19 The continuous black line accounts for the SSTMR evolution under the action of (3.130)-(3.131) moving towards overlapped RPIs. 64
3.20 The continuous black line accounts for the SSTMR evolution under the action of (3.133)-(3.134) moving towards the path Γ. 65
3.21 Command inputs. The dashed red lines represent the prescribed constraints. 65
3.22 Anti-collision distances. The dashed red line accounts for d_{shell}. 66
3.23 State trajectory tubes: ROSACs families (blue ellipsoids); Overlapped RPIs (red ellipsoids) switching control rule: (3.130) − (3.131) → (3.133) − (3.134); Overlapped RPIs (green ellipsoids) switching control rule: (3.133) − (3.134) → (3.130) − (3.131). 66
3.24 Robot trajectory (solid orange line). 66

4.1 The inertial reference system is denoted by $\mathbf{E}$, $\mathcal{W}$ represents the operating environment, $\mathcal{O}$ denotes the occupied space, $\mathcal{O}_{free}^E$ denotes the free space, q^E denotes the pose of the generic $SSTMR$, $\mathcal{R}(q^E)$ denotes the region occupied by $SSTRM$ whose pose is q^E. 71
4.2 Path segment switch from $\overline{W_0W_1}$ to $\overline{W_1W_2}$. The pose of the robot $q^E(t_{\mathcal{N}_1})$ is shown in black. The green robot represents the pose at time $\mathbf{T}_{\mathcal{P}_1^E}(t_{\mathcal{N}_1}^-)$. The red robot represents the pose $\mathbf{T}_{\mathcal{P}_2^E}(t_{\mathcal{N}_1})$. 79
4.3 The red dots represent the position of the robots. The red circles represent the space occupied by the robots. The green dots represent the points along the trajectories. The green rectangle represents the regions where the robots are guaranteed to be. 81
4.4 Experimental setup architecture. 82
4.5 Experimental Operating Scenario. 82
4.6 The red and green solid lines show the sequence of segments composing the coordinated planned trajectories. Squares and triangles show the starting and ending points respectively. 86
4.7 Black (**TP-MRS**) and green (**TP-F**) solid lines represent the worst-case separation distances between robots. Red (**TP-D**) solid line represents the distance between the robots under nominal conditions (i.e. zero trajectory tracking error). The red dashed-dot line indicates the collision threshold. . . . 86
4.8 Experimental example. The solid orange (blue) lines represent the paths executed by Robot #1 (#2) in different numerical runs at different starting poses. 87

4.9 Experimental runs. Experienced distance between the geometric centers of the robots. The red dashed-dot line denotes the collision distance γ. 87
4.10 Experimental runs. Trajectories tracking errors. Dashed lines represent allowable bounds. 88
4.11 Experimental runs. Control speeds. The dashed lines represent allowable bounds. The green solid lines represent the prescribed forward and rotational speeds along the trajectories 88
4.12 Details of an experimental run. Snapshots at different times. The dashed orange (blue) line represents the path taken by Robot #1 (#2). The squares and triangles show the starting and ending points respectively. Solid green rectangles represent the regions $\mathcal{Z}(\mathbf{T}_{\mathcal{P}^E_{(1)}})$ and $\mathcal{Z}(\mathbf{T}_{\mathcal{P}^E_{(2)}})$, while dashed red circles represent the space occupied by the robots. Green arrows represent $\mathbf{T}_{\mathcal{P}^E_{(1)}}$ and $\mathbf{T}_{\mathcal{P}^E_{(2)}}$. Red arrows indicate the pose of the robots. 89
4.13 2D Graphical representation. $\overline{A_iB_i}$ and $\overline{A_jB_j}$ represent two segments of two different trajectories. The green dots represent $w(\alpha)$ and $w(\beta)$. Dark green ellipsoids are a 2D representation of the $\tilde{P}_0$ sets to which the state of the system is guaranteed to belong. The solid red line represents γ, the radius of the circle $\mathcal{R}\left(q^E_{(i)}(t)\right)$ accounting for the physical dimensions of the robotic platforms. 97
4.14 Virtual operating environment. The points represent the grid that discretizes the environment. Black points are utilized for path planning, while red points have been excluded to ensure collision avoidance with the boundaries and obstacles. Three areas with different sliding coefficients are assumed. 100
4.15 Actual distances between the robots. The solid red line highlights the threshold value 2γ. 103
4.16 Trajectory tracking errors. From left to right the values recorded for robot #1, #2, and #3 respectively. The solid red lines represent allowable bounds. . 103
4.17 Values of the forward and rotational velocities of the robots. The plots in the top-left, top-right, and bottom-center show the values for robots # 1, # 2, and # 3, respectively. The solid red lines represent allowable bounds. 104
4.18 Run #8. Snapshot at the time instant $t = 190\,s$. Dashed line represents assigned trajectories for the robot #1, #2, and #3. Solid lines represents the executed trajectories. Squares and triangles represent the starting and target points. The blue, red, and green ellipsoids represent the D-invariant regions for robot #1, #2, and #3, respectively. Black dashed circles account for the physical footprints of the robot and have radius γ. The black line between the ellipsoids represents the distance recorded, which turns out to be equal to the threshold for collision avoidance 2γ. 105

1 Graphical representation of the two statements in a generic two-dimensional state example. $x(k)$ represents the current state whereas $x(k+1)$ the successive. x_1 and x_2 are the two components of vector x. 115

List of Tables

1.1 Classification of path planning methods for mobile robots 4

3.1 The path Γ 62
3.2 Battery consumption [Wh] 67

4.1 Experimental Scenario: relevant values. 83
4.2 Comparison summary table. 85
4.3 Numerical scenario: sliding coefficients values. 100
4.4 Simulation relevant values. 101
4.5 Relevant simulations timing values. 102

1

Introduction

This Chapter introduces the motivation of this research and its aims, defines central research objectives, and summarizes the contributions. The Chapter also covers the structure of the thesis.

1.1 Motivation

Recent years have witnessed an increasing interest in unmanned vehicles, aerial (UAVs), surface (USVs), and ground (UGVs), in virtue of their capability to accomplish many tasks autonomously in different application fields (industrial, agricultural automation, search and rescue, surveillance and inspection), see the comprehensive review [1,2]. In environments such as airports, warehouses, or production facilities, there is a need to use automated surveillance systems for two main reasons. On the one hand, it is important to implement these systems because, among other things, they are able to distinguish an intruder from an authorized person, or are capable of automatically creating and updating a map of the environment, taking into account the presence of human operators and moving elements. On the other hand, these systems must be kept as economical as possible in order to remain competitive with traditional fixed-device solutions (sensor networks, cameras, etc.). These requirements have encouraged the development of intelligent systems based on mobile sensors mounted on autonomous or semi-autonomous robotic platforms. In this context, robots are therefore playing a key role in expanding the potential of surveillance systems, enabling more intelligent interaction with the environment, with humans, and possibly with other robots involved in performing complex missions. In particular, the agricultural sector faces a growing need for automation because of labor shortages, increasing global food demand, and the necessity for precision navigation across varying field conditions. Autonomous agricultural vehicles (AAVs) offer a promising solution to improve productivity, minimize resource wastage, and ensure sustainable agricultural practices. Traditional control strategies, however, often struggle in unpredictable agricultural environments, motivating the development of more robust and adaptive methodologies for enhancing operational efficiency [3]. A more complex scenario emerges in planetary exploration, where robots face challenges such as strict weight and size

constraints, harsh extraterrestrial conditions, and significant communication delays. Efficient operation in such environments requires high levels of autonomy [4]. To achieve these capabilities, a mobile robot must be able to perceive obstacles and barriers within its operating environment in order to develop a movement control strategy compatible with environmental constraints and ultimately execute the planned maneuver. In other words, a mobile robot closes the loop between perception of the environment through its sensors and execution of the movement using its locomotion systems. A key issue across all these applications is collision avoidance. Anti-collision capabilities are essential for safe navigation in complex environments, especially under realistic assumptions like limited knowledge about obstacles, model uncertainties, and communication errors. Consequently, trajectory planning and navigation must address unknown and time-varying environments while satisfying inherent physical constraints [5, 6]. In natural disaster scenarios (e.g., tsunamis, earthquakes, tornadoes, avalanches), autonomous ground robots can reduce human intervention in hazardous rescue operations [7]. Tasks such as rescuing victims from collapsed structures involve instability, dust, toxic chemicals, or radiation [8, 9]. For these cases, a control strategy should provide: low computational needs to maximize the chance of success, guaranteed collision avoidance, and navigation laws designed in a discrete-time setting.

1.2 Literature Review

1.2.1 Path planning

Global path planning is a well-studied research area where a measure of the algorithm quality is the completeness, i.e., the capability of ensuring the computation of an admissible path whenever one exists. Path planning algorithms generate a geometric path from a starting point to an end point, passing through predefined waypoints, both in the joint space and in the robot's operating space.
Path planning algorithms are usually divided according to the methodologies used to generate the geometric path, namely:

- road-map techniques;
- cell decomposition algorithms;
- artificial potential methods.

Path planning approaches can be broadly divided into three categories [10]:

a) **Search-based methods**, such as A* and its variants (D*, Theta*), which operate on a discretized representation of the environment and compute the shortest path by graph search [11–13]. Other examples include *Tangent Graph-Based Planning*, which constructs a graph by connecting tangent points around polygonal obstacles and then searches for the shortest path. This approach is particularly suitable for 3D urban environments where obstacles have well-defined boundaries [14].

b) **Sampling-based methods**, such as the Probabilistic Roadmap (PRM) and Rapidly-Exploring Random Tree (RRT), which incrementally build graphs or trees in the configuration space to connect feasible collision-free configurations [15, 16].

c) **Optimization-based methods**, which iteratively refine a candidate path by minimizing a cost function under geometric constraints. Examples include *Artificial Potential Field Methods*, where virtual attractive/repulsive forces guide the robot towards smooth collision-free paths [17]; *Optimization of Predefined Paths*, which refine an initial path to improve smoothness, path length, or energy consumption under kinematic/dynamic constraints [18]; and *Evolutionary algorithms, simulated annealing, particle swarm optimization*, which are population-based methods that explore the solution space through heuristic search strategies [19–21].

Other methods include *Partially Observable Markov Decision Processes (POMDPs)*, which provide a probabilistic framework for decision making under uncertainty. Here, the robot maintains a belief over possible states and selects actions to maximize the expected reward over a planning horizon, which is particularly useful in long-horizon navigation tasks under uncertain sensing [22]. See the table 1.1 for more details about the classification.

1.2.2 Trajectory Planning

Trajectory planning algorithms differ from path planning methods because they take a given geometric path and associate it with a temporal travel law. Trajectory planning algorithms are fundamental in robotics because defining the passage times at transition points influences not only the kinematic properties of the movement, but also its dynamic properties. From a methodological point of view, the main categories of trajectory planning algorithms essentially correspond to those of path planning methods, with the fundamental difference that here each approach directly generates a timed and dynamically admissible trajectory [23]:

- **Search-based methods**, which extend graph search to state-time space. Typical examples are lattice-based planners or variants of A* with kinematic primitives, which respect speed and acceleration constraints.
- **Sampling-based methods**, such as kinodynamic RRT or kinodynamic PRM, which generate feasible trajectories by sampling both states and control commands.
- **Optimization-based methods**, which formulate a continuous optimization problem to produce a timed trajectory, including, for example, polynomial or clothoid splines, direct collocation methods, or MPC-based predictive controllers.

All these approaches share the main objectives of trajectory planning: ensuring the absence of collisions along the entire path, correctly defining the temporal laws of motion and, in some cases, directly integrating planning with online control.

Table 1.1: Classification of path planning methods for mobile robots

Method	**Category**	**Description**
A*, D*, Theta*	Search-based	Graph search on a discretized grid; finds shortest path [11–13].
Tangent Graph-Based Planning	Search-based	Graph connecting tangent points around obstacles; shortest path [14].
PRM	Sampling-based	Probabilistic roadmap; connects random points in free space [15].
RRT	Sampling-based	Randomly growing tree; probabilistically covers free space [16].
Artificial Potential Field Methods	Optimization-based	Attractive/repulsive forces to generate smooth collision-free paths [17].
Optimization of Predefined Paths	Optimization-based	Refines initial path to minimize cost under constraints [18].
Evolutionary Algorithms / SA / PSO	Optimization-based	Population-based heuristic search for optimal paths [19, 20].
POMDPs	Other	Probabilistic planning under uncertainty; maximizes expected reward [22].

1.2.3 Set-theoretic trajectory planning methods

Roughly speaking, motion planning for autonomous vehicles consists of finding the shortest or optimal collision-free path between two points. In general, all of the methods discussed above address the trajectory generation problem independently of the tracking problem. This raises the issue of the calculated trajectory's feasibility. Indeed, solving a motion planning problem without explicitly considering the performance of the trajectory tracking control may result in the definition of trajectories that are incompatible with the capabilities of the robot or necessitate an unacceptable level of control effort. Most literature solutions do not formally take into account a crucial aspect: the need to rigorously guarantee the **feasibility** of the computed motion sequence with respect to the robot capabilities. As it is well-known, feasibility constraints significantly increase the complexity of the underlying

control problem [24, 25] but on the other hand they provide a formal way to address control actions in realistic scenarios.

To address such a problem, ***set-theoretic trajectory planning algorithms*** [26–29] have gained popularity, being able to explicitly account for the closed-loop dynamics of the robot, uncertainties, and the presence of external disturbances.
Namely, to find a collision-free trajectory through an obstacle-filled environment, such approaches use the invariance property of some safe sets. The trajectory planning problem is formulated in terms of a graph search, where the nodes represent equilibrium conditions and related positively invariant sets, defined in the space of states in which the robot can move without colliding, ensuring that all prescribed constraints are satisfied [23]. Arcs indicate how these safe sets are connected: the presence of an arc connecting two nodes indicates that it is possible to reach one of the two safe sets without leaving the other. Arcs indicate that another safe set can be reached without leaving the current safe set. Set-based motion planners have the advantage of explicitly considering control performance for trajectory tracking during motion planning, ensuring a collision-free trajectory that is compatible with the dynamics of the vehicle, with the control loop performances, and with all the prescribed constraints.

1.2.4 Trajectory tracking

Trajectory tracking is one of the most important operational tasks: a mobile robotic system must be able to follow an assigned trajectory accurately in an environment with obstacles, even in the presence of external disturbances or uncertainties, always ensuring that the maximum deviation from the assigned trajectory is limited to ensure collision-free movement. The possibility of the robot also being remotely controlled via a data communication channel can exacerbate the problems of accurately tracking a trajectory, as transmission delays and/or packet loss can significantly affect the tracking capabilities of a trajectory.

From the existing literature, the trajectory tracking control algorithms include *proportional-integral-derivative (PID) control* [30] *back-stepping control* [31], *sliding mode control* (SMC) [32], *output-feedback control* [33], *fuzzy logic control* [34] and *model predictive controller* (MPC) [35]. Among them, it is commonly accepted that MPC-based approaches are increasingly applied to vehicle navigation problems, see the recent contributions [36, 37]. The reasons behind such a success relies mainly on their natural flexibility to combine path planning with on-line stability and convergence guarantees [38].

The in-depth analysis of these contributions has highlight that the use of AVs in complex and hazardous missions requires that the underlying control architecture enjoys some key properties. First of all, the capability to guarantee constraints satisfaction at each time instant when moving in unknown environments. This clearly leads to conceive a control strategy in charge to keep a sufficiently high level of the flexibility when unpredictable events take place, for example time-varying obstacle scenarios that could prevent the use of pre-

computed control actions. A second important property consists in ensuring energy savings in order to increase the chance of success. Given that the autonomous vehicles considered are battery-powered, this results in defining a control architecture capable to mitigate as much as possible the on-board battery consumption.

It is quite evident that these two aspects are conflicting each other: the on-line updating of the control law gives rise to increasing energy demanding, while minimizing the energy consumption could lead to excessive poor control performance that could compromise the mission accomplishment. Existing approaches often point the attention to one of these issues by addressing compatible operating scenarios, e.g. in indoor environments, although possibly unknown, the energy waste assumes a less relevant role [39], conversely in planetary explorations the energy savings have a primary importance with respect to the optimization of regulation/tracking specs [40]. As a matter of fact, the satisfaction of both these requirements naturally leads to networked control configuration where time-delays must be formally managed [41] and in the worst case data losses could compromise the chance to accomplish the mission. Starting from this reasoning, the literature items show that the navigation problem for autonomous vehicles still deserves further studies at least to improve the trade-off between regulation/tracking performance and energy savings when complex operating scenarios are of interest.

1.2.5 Multi-Robot Systems

The ability to use autonomous ***multi-robot systems*** (MRS) to perform complex tasks is a key requirement in various application scenarios such as reconnaissance [42], surveillance and patrolling of large areas [43], search and rescue operations after disasters [44]. These applications leverage the collective power and versatility of MRSs to manage tasks that require coordination and efficiency. In these scenarios, it is crucial for effectiveness that several robot units can operate in a joint deployment scenario. Task coordination requires the development of algorithms that enable both the synchronization of robotic actions and the efficient distribution of operational tasks according to the individual capabilities of the robots. This requirement necessitates the planning of coordinated movement strategies that allow each robot unit to reach its goal while avoiding collisions.

Coordinated control addresses the problem of optimizing the overall performance of the entire multi-robot system by developing strategies that maximize the efficiency of task execution. Coordinated control of robotic movements presents certain algorithmic challenges that do not exist for a single mobile robot. Individual units must be able to coordinate their movements to avoid collisions and optimize overall system performance. Despite the price of increased management and control complexity, the use of a MRS offers non-negligible benefits [45–48]. In general, a multi-robot coordination algorithm must satisfy four basic conditions to be effective:

(a) must be distributed, so that individual robots can act only on the information available to them, e.g., through recognition or active communication;
(b) must be decentralized so that the algorithms executed by an individual robot do not depend on the size of the entire team;
(c) must ensure safety, as robots must be able to avoid collisions with each other and with the environment;
(d) must have emergent properties, i.e., the global properties of the system must emerge from the local interaction rules;

Numerous algorithms satisfying these conditions have been proposed in the literature and have been successfully used to form and maintain platoons [49, 50], to monitor areas [51, 52] and to patrol border areas [53, 54].

When it comes to motion planning for the MRS, the challenge is that each robot not only has to consider static obstacles in the environment, but also the possible interference from other robots. When robots in a team perform independent tasks in a common workspace, each of them becomes a moving obstacle for the others. The motion planning of each robot in the team must therefore take into account the movements of the others. The collision avoidance task is usually solved by implementing reactive control techniques, i.e., equipping each robot with the ability to react appropriately to the presence of other robots that impede its movement. However, motion planning for the units of an MRS system is inherently difficult, and research in this area has been going on for more than two decades [55]. The most common solution used so far in various applications involves a decoupled approach, where path planning methods [56, 57] are used to plan paths that allow each robot unit to reach its assigned goal, without taking into account the presence of other units, while reactive control algorithms [58] allow robots to coordinate with other units in real time to avoid collisions, maintain a formation and follow the given path.

1.2.6 Path planning for MRS

It involves determining a path defined by a sequence of waypoints, that makes it possible to reach the final destination while avoiding obstacles and optimizing certain criteria such as distance, time or energy consumption. ***Path planning*** for multi-robot systems is an important area of research in autonomous robotics. The goal is to ensure that a group of robots can move in a coordinated, safe, and efficient manner in a complex operational scenario. This non-trivial challenge is accompanied by several problems that require advanced technical solutions. Generally speaking, the principal challenges in path planning for multi-robot systems are three.

(i) **Coordination** - One of the major obstacles in path planning for multi-robot systems is the need to coordinate the movements of each robot. It is critical to avoid collisions

between the robots themselves and obstacles in the environment. Planning must take into account the dynamic interactions between the robots and the specific motion constraints of each robot.

(ii) **Computational complexity** - Trajectory planning for MRS requires computationally efficient algorithms and methods to provide real-time solutions. As the number of robots involved increases, the computational complexity increases exponentially and requires heuristic solutions or optimization-based approaches.

(iii) **Scalability** - Another important challenge is the scalability of the system. As the number of robots increases, it becomes increasingly difficult to plan paths that satisfy all the constraints and limitations of the environment. Trajectory planning must be able to effectively manage large fleets of robots without compromising safety or overall efficiency.

The importance of finding a solution to the problem of autonomous navigation and path planning have led to that the developments reached in this area have grown exponentially in the last years, and cover a wide range of methods based on different techniques and characteristics [59]. The expansion of autonomous vehicles and their frequent use within MRS has led to the adaptation of classic methods of path planning, initially designed for a single vehicle. Along with this adaptation, new techniques have appeared that consider multiple vehicles in the planning, and allow planning in real time, as the vehicles navigate through the environment.

Regarding *Graph-based algorithms* such as Dijkstra's algorithm and the A* algorithm [11]. In [60], a planning algorithm has been proposed for a heterogeneous multi-vehicle system, in which, a task assignment phase is established, along with a Dijkstra-based path planning algorithm to minimize the travel time between two locations. The work [61] implemented an A*-based algorithm with modifications to apply it to the resolution of the path planning problem in MRS.

As for *Sampling-based algorithms*, including the PRM and RRT [62], in recent years, these classical methods have been significantly improved to increase their performance and applicability. In [16] an asymptotically optimal version of RRT was proposed, which guarantees convergence to the optimal path when the number of samples approaches infinity. In [63] it is presented a multidimensional RRT* to carry out path planning within a system with multiple autonomous underwater vehicles. The idea of this work is to develop a method to improve the efficiency in the tasks of search, surveillance, and monitoring of the seabed. The purpose of the RRT* is to create a set of optimal, reachable and collision-free paths through which the different AUVs travel from an initial state to a final one.

Regarding *Artificial Potential Field methods*, which model the robot and obstacles as charged particles, are often used to plan motions that take into account the presence of obstacles due to their simplicity [64]. Works such as [65] studied the use of APF for free movement

of obstacles from Formations Leader–Followers of mobile robots. Two potential fields are generated, one attractive to the destination, and one repulsive of the obstacles working with both jointly to form a composite potential field, and making the MRS to reach the goal avoiding the obstacles, moving relative formations in order to perform the task safely and reliably.

However, these methods often encounter problems with local minima. To solve this problem, hybrid approaches have been explored that combine potential fields with heuristic solutions [66].

1.2.7 Reactive control strategies for collision avoidance in MRS

In the literature they are generally implemented by decentralized strategies that allow each robot to make decisions based on local information, which improves scalability and reduces computational overhead [67]. These algorithms allow robots to react quickly to dynamic environments and unexpected obstacles and ensure flexible deployment in complex spaces. Among the solutions proposed in the literature, artificial potential fields model robots and obstacles using attractive and repulsive forces that guide the robots to their goals while avoiding collisions [64]. In behavior-based control, robot actions are decomposed into behaviors such as obstacle avoidance and target search, which are then combined to produce the desired motion [68]. *Control barrier functions* provide a mathematical framework to ensure that safety conditions are met and control goals are achieved [69].

Other reactive control strategies include: *the dynamic window approach* [70], which takes into account the kinematic constraints of the mobile robot; the *elastic band method* [71], which adjusts the path of the robot based on the shape of obstacles; the *null-space based behavioral approach* [72, 73], where collision avoidance is formulated by coordinated control using a priority logic; the *potential field method* [64], where obstacles are modeled as repulsive forces and the goal as an attractive force; the *vector field histogram method* [74], which builds a polar histogram of obstacle density to select feasible motion directions; and the *curvature-velocity method* [75], which searches for admissible trajectories by combining curvature and velocity constraints. Finally, collision avoidance can also be addressed by implementing algorithms that regulate the speed of each robot along its assigned path. The decomposition of the autonomous navigation problem into separate subproblems of path planning and robot velocity profile assignment is discussed e.g. in [76, 77].

1.2.8 Time-base synchronization methods for MRS

Other methods address the problem of movement coordination and collision avoidance through platform prioritization and allocation of delays. In [78], the authors develop a method that uses k-means clustering to balance tasks between robots and a collision model with time

priority constraints. This model defines interference zones between the robots. To avoid collisions, a directed acyclic graph (DAG) is constructed to represent the priority constraints, and the final scheduling minimizes the overall completion time while ensuring collision-free execution. In [79], the collision avoidance problem is addressed for a multi-robot system through the imposition of delays. Specifically, the trajectories consist of sequences of regions to be followed, with an associated travel time for each region, and collisions are avoided by properly allocating initial delays along the trajectories.

Beyond scheduling and delay-based approaches, other contributions propose coordination strategies that act directly during the trajectory planning phase. These approaches are effective since they do not require the installation of dedicated hardware for collision management, and often exploit optimization or graph-based formulations to guarantee safety. For instance, some works integrate conflict resolution within centralized planners, while others rely on decentralized formulations where each agent adapts its trajectory in real time according to shared constraints or consensus rules. In this context, coordination is not only achieved by delaying motion, but also by reshaping trajectories or dynamically reallocating waypoints, thus improving scalability and robustness in large-scale multi-robot systems.

1.3 Objectives and Methodologies

1.3.1 Research Objectives

In the following, the **research objectives** (ROx) are formulated to address the identified challenges and research gaps in the field of safe and feasible navigation for mobile robots. We divide the challenges and research gaps into two main categories: *(i)* single-robot navigation under constraints and uncertainties in cluttered and unknown environments, and *(ii)* multi-robot coordination and collision avoidance, under constraints and uncertainties in cluttered and unknown environments.

The increasing deployment of autonomous ground robots in applications such as surveillance, search-and-rescue, and automated logistics requires navigation strategies that guarantee safety, feasibility, and robustness even under limited sensing, uncertain dynamics, external disturbances, and actuation/state constraints. To meet these requirements, novel set-theoretic methodologies are needed for the systematic design of controllers and planners capable of handling uncertainty while ensuring performance guarantees.

Past research has largely focused either on reactive local strategies, which adapt to obstacles in real time but often lack formal guarantees, or on centralized coordination schemes, which may ensure safety but are not scalable or robust to uncertainties. Moreover, communication delays and parameter varying models, especially in the case of remotely controlled robots, further challenge the reliability of navigation strategies. These limitations highlight the need for methodologies that can: *(i)* provide rigorous feasibility and safety guarantees in uncertain and dynamic environments; *(ii)* extend naturally from single-robot to multi-robot

systems without requiring additional hardware for collision management; and *(iii)* balance between robustness, scalability, and computational efficiency.

To bridge these gaps and advance the development of safe and feasible navigation strategies in unknown cluttered environments, we:

- **RO1. Propose a graph-based trajectory planning framework for single skid-steered mobile robots under constraints.** The operating environment is discretized into a finite grid, and feasible trajectories are obtained by solving a shortest-path problem with feasibility guaranteed via semidefinite programming (SDP).
- **RO2. Develop a set-theoretic model predictive control (MPC) framework for single robots navigating in unknown and cluttered environments.** The method ensures collision avoidance and mission completion despite communication delays, unpredictable obstacles, and model uncertainties, while guaranteeing boundedness and constraint satisfaction.
- **RO3. Formulate a set-theoretic planning approach for multi-robot systems subject to uncertainties and disturbances.** Coordinated trajectories are generated by solving a mixed graph search problem and semidefinite programs with linear matrix inequality (LMI) constraints, guaranteeing collision-free navigation even in the presence of tracking errors.
- **RO4. Introduce a time-based trajectory synchronization strategy for multi-robot systems under slippage and actuation constraints.** Coordination and collision avoidance are achieved by imposing optimized departure delays, computed through LMI-based optimization, thus minimizing shared space occupancy while ensuring trajectory feasibility.

To address the challenges and research gaps formulated above, a general research methodology was adopted in this work. It consists of several stages defined as follows.

- **Study of the state of the art.** A comprehensive analysis of existing navigation and collision avoidance strategies for single and multi-robot systems was carried out, with particular attention to reactive methods, scheduling-based approaches, and set-theoretic techniques. This stage provided the necessary background to identify open issues and limitations.
- **Problem formulation and use case analysis.** Based on the literature review and the targeted applications (e.g., cluttered unknown environments, networked control, multi-robot coordination), the main research gaps were translated into well-defined problem statements addressing feasibility, safety, and robustness.
- **Solution design.** The identified problems were addressed through the design of four set-theoretic methodologies: two for single-robot navigation and two for multi-robot coordination. Each solution was conceived to guarantee feasibility, robustness to uncertainties, and satisfaction of state and actuation constraints.

- **Modeling and formal validation.** For each proposed methodology, mathematical models were defined and feasibility conditions were formulated in terms of optimization problems involving semidefinite programming (SDP) or linear matrix inequalities (LMIs). These models enabled formal validation of safety and performance guarantees.
- **Simulation and experimental evaluation.** The effectiveness of the proposed solutions was assessed through extensive simulations and, where applicable, experimental validation on skid-steered mobile robots, highlighting advantages and trade-offs of each method.

1.3.2 Methodologies

The detailed results of this thesis are organized in four methodologies (A–D), each corresponding to the research objectives formulated in Section 1.3. Methodology A and Methodology B address the single-robot navigation problem, while Methodology C and Methodology D extend the analysis to multi-robot systems.

Methodology A — Set-theoretic trajectory planning algorithm for single robot under uncertainties, external disturbances, and state/actuation constrains. The main technical outcomes can be summarized as follows:

- The discretization of the environment into a finite-dimensional grid and the formulation of a weighted graph structure;
- The definition of feasible trajectories as shortest paths on the graph, subject to actuation and state constraints;
- The guarantee of trajectory feasibility through sufficient conditions formulated as semidefinite programming (SDP) problems;
- The validation of the proposed approach through experimental simulations.

Methodology B — Set-theoretic model predictive control for navigation in dynamical unknown environments. The main technical outcomes can be summarized as follows:

- The formulation of a networked control framework to handle communication-induced delays;
- The design of a predictive control scheme within a set-theoretic framework to guarantee feasibility and robustness;
- Anti-collision and safety guarantees even under unpredictable obstacle occurrences and uncertain dynamics;
- Proof of ultimate boundedness and constraint satisfaction of the regulated trajectory;
- Validation through experiments on skid-steered tracked mobile robots.

Methodology C — Set-theoretic coordinated trajectory planning method for multi-robot systems. The main technical outcomes can be summarized as follows:

- The reformulation of coordinated trajectory planning as a mixed graph search problem;

- The incorporation of uncertainties and external disturbances into the planning stage;
- The formulation of collision-avoidance constraints through semidefinite programming with linear matrix inequalities (LMIs);
- Validation of the proposed framework through experiments with two remote-controlled skid-steered mobile robots.

Methodology D — Set-theoretic Time-based trajectory synchronization method for multi-robot systems. The main technical outcomes can be summarized as follows:

- The formulation of the trajectory tracking error dynamics as an uncertain system with disturbances and actuation constraints;
- The synchronization of coordinated trajectories by imposing optimized departure delays;
- The computation of feasible delay intervals via LMI-based optimization, with separate offline and online phases;
- The minimization of shared-space occupancy time while guaranteeing feasibility;
- Validation of the proposed method through extensive numerical simulations.

1.3.3 Publications

The methods described above have been included in the following publications:

1. **Methodology A** – *Scordamaglia Valerio, Nardi Vito Antonio , and* ***Ferraro Alessia****. "A feasible trajectory planning algorithm for a network controlled robot subject to skid and slip phenomena." 2019 24th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA). IEEE, 2019.*
2. **Methodology B** – *Scordamaglia Valerio,* ***Ferraro Alessia****, and Franzè Giuseppe. "Autonomous tracked vehicles operating in cluttered and unknown environments: a networked set-theoretic receding horizon control strategy". IEEE Transactions on Cybernetics (2025).*
3. **Methodology C** – *Scordamaglia Valerio, and* ***Ferraro Alessia****. "A set-based method for planning coordinated trajectories for skid-steered robotic units subject to constraints, uncertainties and external disturbances." IEEE Access (2025).*
4. **Methodology D** – ***Ferraro Alessia****, De Capua Claudio and Scordamaglia Valerio. "Set-theoretic Time-based Trajectory Synchronization Approach for Skid-Steered Robotic units subject to Constraints, Uncertainties, and External Disturbances." 2025 11th International Conference on Control, Decision and Information Technologies (CoDIT). IEEE, 2025.*

1.4 Thesis Outline

The thesis is organized into 5 Chapters, their content is briefly described below.

- **Chapter 1** contains the motivation, objectives, contributions, and structure of this work.
- **Chapter 2** presents the dynamic model of the skid-steered robotic platform used throughout the thesis. This model provides the foundation for both single-robot and multi-robot navigation methodologies.
- **Chapter 3** focuses on navigation for a single robot. It introduces two set-theoretic methodologies for single robot (Methodology A and Methodology B) for safe and feasible trajectory planning in cluttered and unknown environments, explicitly considering control and state constraints, network-induced delays, external disturbances, and trajectory tracking performance.
- **Chapter 4** addresses navigation for multi-robot systems (MRS). It presents two set-theoretic methodologies (Methodology C and Methodology D) for coordinated trajectory planning and time-based trajectory synchronization, focusing on collision avoidance, inter-robot coordination, uncertainties, external disturbances, and actuation and state constraints.
- **Chapter 5** includes the summary of research outcomes and the discussion of future research.

The final part of the thesis includes the Appendix and the Bibliography.

2

Mathematical Modeling of the Robotic Platform

The effectiveness of any control strategy for autonomous robotic systems depends largely on the accuracy of the mathematical model adopted. To ensure the robustness and feasibility of control in real environments, it is essential that the model not only describes the ideal kinematics of the robot, but also captures the non-ideal phenomena and uncertainties that inevitably affect its performance. This chapter introduces the mathematical description of the skid-steered tracked mobile robot, starting from its ideal kinematic model. The model is progressively extended to include elements such as interactions between tracks and terrain (sliding phenomena) and communication delays in networked control schemes (NCS).

In particular, the chapter is structured as follows.
Section 2.1 introduces skid-steered tracked robots (SSTMRs), describing their main characteristics and applications. Section 2.2 introduces the kinematic model of the skid-steered tracked mobile robot, describing the equations that link the speeds of the tracks with the robot's position in the plane, and analyses sliding phenomena, modeled using variable coefficients that represent the interaction between the tracks and the ground. Finally, Section 2.3 presents the remote control architecture based on networked control systems (NCS) theory, discussing its advantages, critical issues and the modeling of communication delays.

2.1 Skid-steered tracked mobile robots

This thesis considers a skid-steered tracked mobile robot (SSTMR) [80], belonging to a class of ground vehicles widely used in unstructured environments thanks to their mechanical simplicity, robustness, and high maneuverability. Tracked mobile robot is a type of robot that is used in specific situations related to the nature of the environment in which it works, especially when the ground is uneven or constantly changing [81]. Tracked robots offer numerous advantages: they provide great traction on slippery surfaces and can carry heavy loads, while also ensuring high performance and power [82]. In particular, SSTMRs play a key role in several applications fields such as rescue missions [83], and surveillance [84]. However, they require more power when turning or cornering due to the slip steering method they use. In a skid-steered mobile robot, the steering action is performed by controlling the differential

velocity of left and right side of the vehicle [85], this generates a torque to counteract the resistance to rotation caused by the slippage of the tracks on the ground and the rotational inertia of the robot. In other words, the slip steering technique is based on changing the relative speeds of the two tracks, which will cause slippage and cutting of the ground to achieve steering. Skid-steering drive mechanism offers significant advantages compared to alternative steering mechanisms: it is mechanically robust and simple, providing at the same time a good maneuverability (i.e., zero radius turning capability). Such kind of locomotion scheme poses some problems from the point of view of motion control due to the presence of non negligible sliding effects.

Modeling such systems is very essential especially for controller's derivation. The model of such systems can be developed based on Newton's dynamics where force and moment equations are involved [85–87]. Others developed such models depending on empirical formulae which make them of no use for online predictions [88]. Classical mathematical representations of a mobile robot involve nominal nonholonomic kinematic equations for the mobility description [89]. Nonholonomic constraints are violated by the presence of sliding phenomena, thus requiring different mathematical models to be considered. In [90] it is introduced a classification of mobile robots based upon the influence of sliding effects on the kinematic model. In literature two main alternative approaches are proposed to properly take into account sliding effects. As shown in [91–94] reliable representations involve dynamic model of sliding phenomena providing the possibility to reach best accuracy at the price of significant growth of complexity of mathematical models [95]. An alternative approach lies in kinematic description of effects of sliding phenomena. Such an approach is attractive from a practical point of view, seeming reasonable since sliding effects can be viewed at the kinematic level in terms of disturbances of control velocities [96, 97]. The presence of sliding phenomena poses some difficulties when the prediction of motion of vehicle is required [98]. Steering is achieved by adjusting the relative speeds of the two tracks, a feature that differentiates the kinematic model of these systems from that of wheeled robots and requires specific attention in control design. This robot belongs to the class of nonholonomic wheeled vehicles and, as a consequence, the wheels can perform only pure rolling motions [99].

2.2 Robot mathematical model

2.2.1 Kinematics Model of Mobile Robot

Let **E** be the inertial reference frame with respect to which the map of the operating environment is defined. To describe the mathematical model of the robot, it is assumed to be a rigid body moving on a horizontal plane. Referring to Fig. 2.1, the first-order kinematic model of the system without slippage can be written as

$$\begin{cases} \dot{x} = cos(\theta(t))\, V(t), \\ \dot{y} = sin(\theta(t))\, V(t), \\ \dot{\theta} = \omega(t) \end{cases} \tag{2.1}$$

where θ represents the direction of the robot, x and y represent the robot's center position in **E**, and V is the robot forward speed. The speeds of the right track V_r and left track V_l of the mobile robot are

$$\begin{cases} V_r = R\rho_r \\ V_l = R\rho_l \end{cases} \tag{2.2}$$

where R is the track or sprocket rolling radius, and ρ_r (ρ_l) is the angular velocity of the driving motors of right (left) track. Hence, the robot forward translational velocity can be obtained as follows

$$V = \frac{V_r + V_l}{2}. \tag{2.3}$$

Using the last equation, the mobile rotational velocity or yaw rate ω can be given as

$$\omega = \frac{V}{r} = \frac{V_l + V_r}{2r} = \frac{V_r - V_l}{D} \tag{2.4}$$

where r is the radius of curvature of the robot and D is the tread of the robot.
Considering the inertial reference frame **NWU** the pose of the robot at time t in **E** is

$$q_E(t) = \begin{bmatrix} x(t) & y(t) & \theta(t) \end{bmatrix}^T. \tag{2.5}$$

Let

$$u(t) = \begin{bmatrix} V(t) & \omega(t) \end{bmatrix}^T \tag{2.6}$$

be the control velocity vector where the forward and rotational speeds are V and ω, respectively. By controlling these two quantities,it is possible to achieve both trajectory tracking and maneuvering without directly handling the individual track velocities, which simplifies the control design while still capturing the essential dynamics of the platform. Therefore, the robot kinematics is described, in the continuous-time domain, by the following nonlinear differential equations:

$$\dot{q}_E(t) = G(t) \cdot u(t) \tag{2.7}$$

with

$$G(t) = \begin{bmatrix} cos(\theta(t)) & 0 \\ sin(\theta(t)) & 0 \\ 0 & 1 \end{bmatrix} \tag{2.8}$$

2.2.2 Sliding phenomena

Accurate modeling of the interaction between the tracks and the ground is challenging [100–102], as it involves complex contact mechanics and variable friction effects. Such complexity

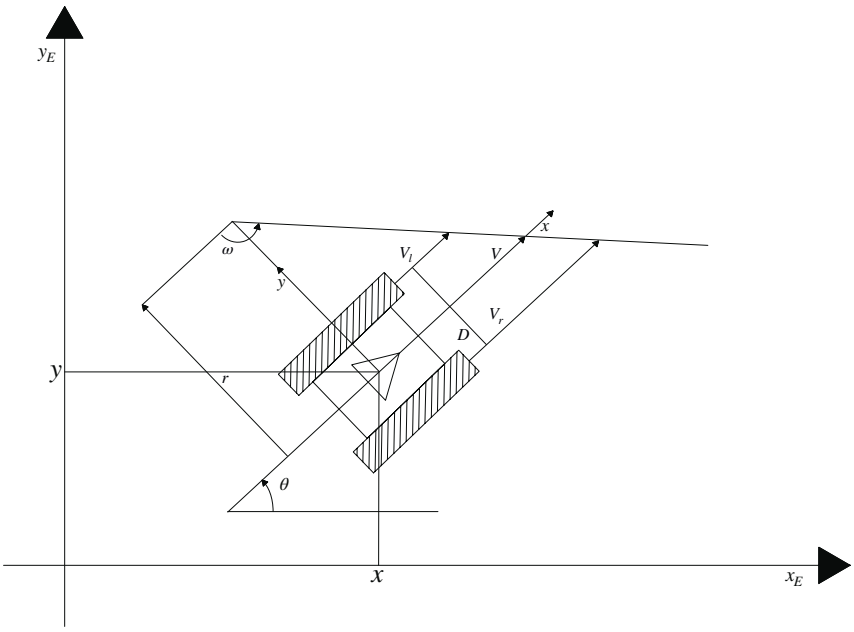


Fig. 2.1: Robot Kinematics.

makes it impractical to derive a fully detailed mathematical model suitable for control design. To account for this in a control-oriented framework, in this thesis it is adopted a macroscopic approach: the track-ground interaction is modeled through two dimensionless, positive, time-varying coefficients, $\mu_r(t)$ and $\mu_l(t)$, here in after referred as sliding coefficients, representing the sliding behavior of the right and left tracks, respectively. The forward and rotational speeds of the robot are related to the $\rho_r(t)$ and $\rho_l(t)$ motor angular speeds through the following relation:

$$u(t) = J \cdot H(t) \cdot \rho(t) \tag{2.9}$$

being

$$\rho(t) = \begin{bmatrix} \rho_r(t) & \rho_l(t) \end{bmatrix}^T \tag{2.10}$$

$$J = \begin{bmatrix} R/2 & R/2 \\ R/D & -R/D \end{bmatrix} \tag{2.11}$$

and

$$H(t) = \begin{bmatrix} \mu_r(t) & 0 \\ 0 & \mu_l(t) \end{bmatrix} \tag{2.12}$$

where R is the radius of the gears connecting the tracks to the motors, and D is the distance between tracks.

Let us assume that the robot is controlled by two commands the forward speed $\hat{V}$ and the rotational speed $\hat{\omega}$

$$\hat{u}(t) = \left[\hat{V}(t)\ \hat{\omega}(t)\right]^T \tag{2.13}$$

Nominal angular speeds of the two electric motors required to generate (2.13) are calculated assuming the two sliding coefficients are both taken as unit values

$$\hat{\rho}(t) = J^{-1} \cdot \hat{u}(t) \tag{2.14}$$

However, since the sliding coefficients $\mu_r(t)$ and $\mu_l(t)$ depend on the interaction between the track and the ground and may vary over time, the effective forward and rotational speeds of the robot result

$$u(t) = J \cdot H(t) \cdot \hat{\rho}(t) \tag{2.15}$$

In view of eqs. (2.14)-(2.15),eq. (2.7) can be rewritten in the following form

$$\dot{q}_E(t) = G(t) \cdot J \cdot H(t) \cdot J^{-1} \cdot \hat{u}(t) \tag{2.16}$$

Remark - It is important to note that the validity of eq. (2.16) is guaranteed under the assumption that the response time of the track speed controllers is negligible compared to the dynamics of the robot.

2.3 Remote Control Architecture for Mobile Robots

In this thesis, the operational setup shown Fig. 2.2 will be used, it represents a network implementation of control systems where feedback control loop is closed through a communication network. This kind of architecture is widely adopted for applications involving mobile robots [25,98,103]. Networked control system offers significant advantages, for example, higher system resource employment, testability and maintainability, flexibility in control architecture design, cost reduction and simplified deployment [104].
Despite these benefits, designing and implementing an NCS is not a trivial task. The introduction of a shared communication channel within the feedback loop gives rise to phenomena that are absent in classical control schemes. In particular, networks induce stochastic effects such as variable delays, jitter, and packet dropouts [103–105], which can degrade stability and performance if not properly addressed.

In the framework of this thesis, the adoption of an NCS architecture is motivated by the limited on-board computational resources of the tracked mobile robot (SSTMR). Executing complex algorithms on embedded hardware would significantly reduce efficiency and autonomy. Offloading the majority of computations to a remote control station mitigates these limitations, preserves battery energy, and extends the operational lifetime of the robot. Furthermore, the choice of this type of architecture guarantees the scalability of the method,

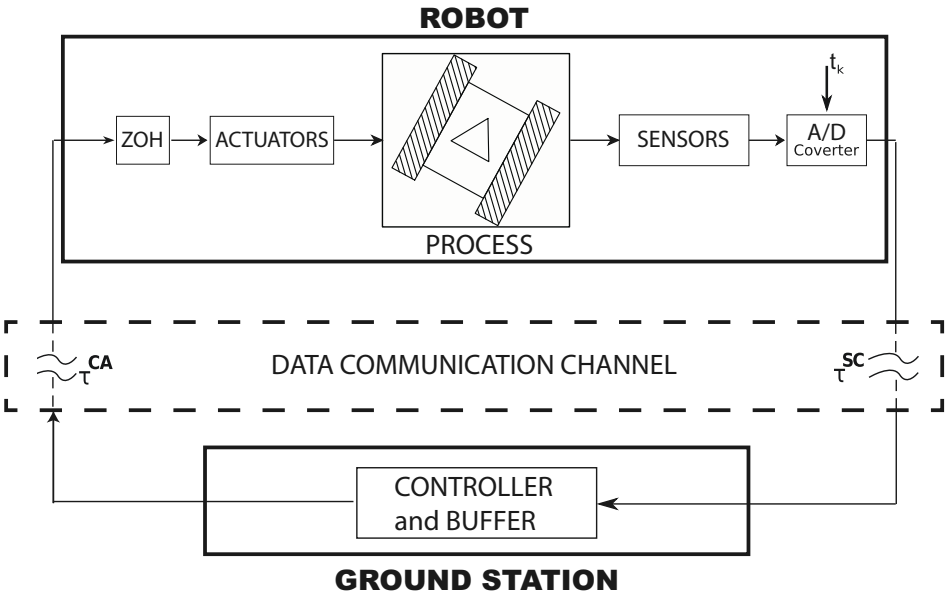


Fig. 2.2: Outline of a networked control system (NCS) scheme.

allowing it to be applied even to smaller platforms. At the same time, it enhances maintainability and allows the use of advanced control strategies that would otherwise exceed the processing capabilities of the vehicle. The drawback of this approach is that the presence of the communication channel inevitably introduces additional delays. With the theoretical tools provided by NCS theory [104], it is possible to model the impact of these delays and incorporate them into the control design. The main effects are summarized in Fig. 2.2. Specifically, let τ^{SC} denote the *sensors-to-control* delay, which represents the time required to transmit information from the sensor side to the controller. Let τ^{CA} denote the *control-to-actuator* delay, i.e., the time required for the control signal generated by the controller to reach the actuator. Both τ^{SC} and τ^{CA} are variable but assumed to be bounded. In addition, let τ^{C} be the computation delay, accounting for the time needed by the controller to process a new control action.

3

Set-theoretic Navigation Methods for Single Robot

In this chapter, the objective is to address the problem of safe and feasible navigation for a skid-steered tracked mobile robot in complex environments crowded with obstacles, taking into account uncertainties, external disturbances, implementation constraints, and delays introduced by the communication network.

The chapter is structured as follows.

Section 3.1 formulates the problem of trajectory planning in environments crowded with obstacles, modeling the robot based on the kinematic equations in Chapter 2 and including constraints, slippage phenomena, and communication delays.

Section 3.2 presents the the ***Set-Theoretic Feasible Trajectory Planning (ST-FTP)*** *Method. In 3.2.1, the tracking error is defined, while in 3.2.2 and 3.2.3, the system is described as a networked control system subject to delays, for which a robust controller is designed. In 3.2.4, robustly invariant ellipsoidal sets are introduced, used in 3.2.5 to establish trajectory feasibility conditions. Section 3.2.6 proposes a planning algorithm based on graph-search A*-like algorithm, and 3.2.7 reports the experimental results on a real robot.*

Section 3.3 introduces the ***Set-Theoretic Model Predictive Control (ST-MPC)*** *Method. In 3.3.1, the nonlinear model of the robot is embedded into a LPV form. Sections 3.3.3–3.3.6 describe the predictive controller based on robust sets and overlapping RPIs, followed in 3.3.7–3.3.9 by the path planning application and the complete control scheme. Finally, in 3.3.10, the experimental results are presented, confirming the robustness and effectiveness of the method even in scenarios with unexpected obstacles.*

3.1 Problem formulation

In this section, the trajectory planning problem is formulated for the skid-steered tracked mobile robot (SSTMR) presented in Section 2.2. The robot kinematics are expressed in continuous time by a set of nonlinear differential equations, see eqs. (2.5)–(2.16), which include the sliding coefficients $\mu_r(t)$ and $\mu_l(t)$. These coefficients are assumed to vary within closed and bounded intervals centered around their normalized nominal values, as follows:

$$\begin{aligned} &\mu_r(t) \in [1-\overline{\mu}_r,\, 1+\overline{\mu}_r] \\ &\mu_l(t) \in [1-\overline{\mu}_l,\, 1+\overline{\mu}_l], \forall t \geq 0 \end{aligned} \tag{3.1}$$

thus accounting, at each time t, for possible mismatches between the computed control input $u(t)$ and the *effectively* applied $u_e(t)$, i.e.,

$$u_e(t) \neq u(t), \forall t \geq 0 \tag{3.2}$$

In addition, the SSTMR operates under the following ellipsoidal constraints on input and state variables:

$$\begin{aligned} &u(t) \in \mathcal{U} := \{u \in \mathbb{R}^m : u^T u \leq u_{max}^2\}, \\ &x(t) \in \mathcal{X} := \{x \in \mathbb{R}^n : x^T x \leq x_{max}^2\} \quad \forall t \geq 0, \end{aligned} \tag{3.3}$$

It is assumed that the vehicle operates within an indoor cluttered planar environment populated by static but unknown obstacles. Specifically:

- *Environment planimetry*: without loss of generality, it is assumed that the overall layout is exactly known in terms of the accessible region. Let n_c be the number of corridors characterizing the planar structure. Each aisle is formally defined by resorting to polyhedral arguments, i.e.

$$CR^l : \bigcap_{s=1}^{2} \left\{ (W^l)_s^T p > (c^l)_s \right\}, \tag{3.4}$$

  where $p := [x \;\; y]^T \in \mathbb{R}^2$ are the planar components of the state space, $(W^l)_s^T$ characterizes the wall directions in the l-th corridor/aisle and, in a planar representation, are defined by 2-elements row vectors, $(c^l)_s$ denotes the shift vs. origin term of the l-th corridor walls (scalar quantity in a planar representation). The p-plane point locus $(W^l)_s^T p = (c^l)_s$ denotes a linear manifold characterizing the walls boundary whereas $(W^l)_s^T p > (c^l)_s$ are half-spaces whose intersection generates the corridor, see Fig. 3.1. The accessible region is

$$\mathcal{O}_{free}^E := \bigcup_{l=1}^{n_c} \left\{ p \in \mathbb{R}^2 : \, p \in CR^l \right\} \tag{3.5}$$

  Notice that the integer n_c takes care of same drive-through corridors in opposite directions.

- *Obstacles*:

  Definition 3.1. *Let Ob^j be an object with a polyhedral convex structure described as the intersection of l_j half-spaces:*

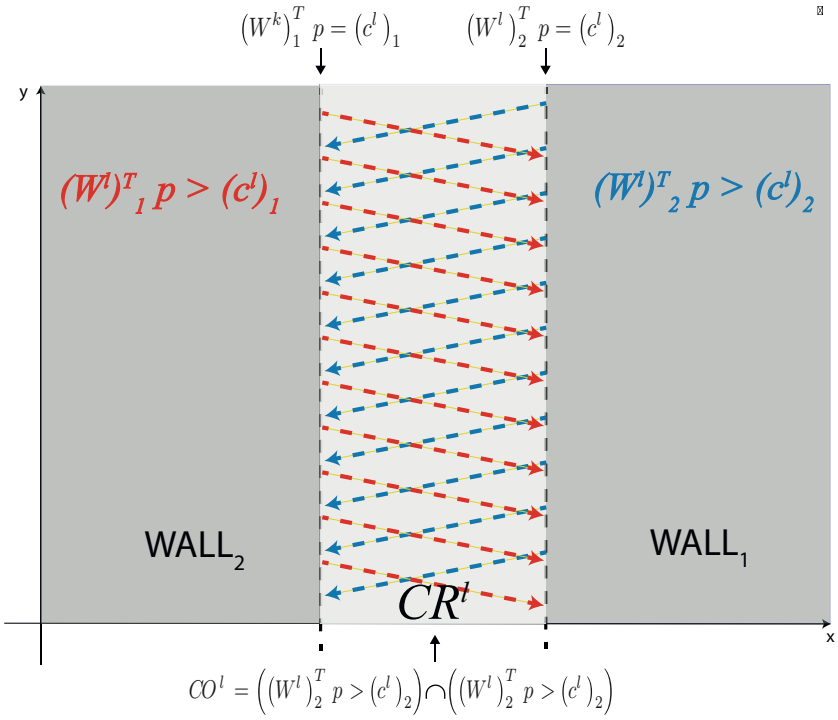


Fig. 3.1: Aisle/Corridor polyhedral representation

$$Ob^j : \left[H_1^j, \ldots, H_{l_j}^j \right]^T p \leq \left[g_1^j, \ldots, g_{l_j}^j \right]^T \tag{3.6}$$

An obstacle scenario $\mathcal{O}$ is defined as

$$\mathcal{O} := \{Ob^1, \ldots, Ob^{n_o}\} \tag{3.7}$$

where n_o denotes the number of involved objects. □

Definition 3.2. *Let $\mathcal{O}$ be an obstacle scenario. Then, the non-convex obstacle-free region pertaining to $\mathcal{O}$ is identified as follows*

$$\mathcal{O}_{free}^O := \{p \in \mathbb{R}^2 : p \in f(p)\}, \tag{3.8}$$

where

$$f(p) := \bigcap_{j=1}^{n_o} f^j(p), f^j(p) := \left\{ p \in \mathbb{R}^2 : \bigcup_{k=1}^{l_j} (H_k^j)^T p > g_k^j \right\}$$

□

Then, at each time instant t the obstacle free-region is given by

$$\mathcal{O}^t := \{p \in \mathbb{R}^2 : p \in f(p,t)\} \cap \mathcal{O}_{free}^E \tag{3.9}$$

with $f(p,t)$ having the same structure of $f(p)$.

Obstacle avoidance and motion planning in unknown cluttered environments (OAMP-UCE)

Given a cluttered environment with free space $\mathcal{O}_{free}^{E}$, an initial state $q_E(0) \in \mathbb{R}^3$ and a target state $q_E^f \in \mathbb{R}^3$, the problem consists ensuring safe navigation from $q_E(0)$ to q_E^f. The strategy must always respect state and input constraints, guarantee collision avoidance with obstacles in the environment (3.9), and cope with uncertainties introduced by the presence of communication delays, and sliding phenomena modeled as external disturbances.

*The **OAMP-UCE** problem thus requires addressing two key aspects:*

- *the management of time delays to guarantee the existence of admissible commands;*
- *the design of a robust strategy capable of satisfying constraints and avoiding collisions despite uncertainties and latency effects;*

□

To address the **OAMP-UCE** problem, in this thesis are proposed two set-theoretic methodologies for a single SSTMR:

- **Methodology A**— proposes a feasible trajectory planning approach, using ellipsoidal positively invariant sets to guarantee the trajectory feasibility, for a constrained SSTMR connected through a communication network. Proposed search algorithm takes explicitly into account closed loop trajectory tracking control, non-deterministic network variable delays, non-negligible skid and slip phenomena and a set of prescribed constraints involving robot motion and actuation. Operating scenario is firstly discretized with a finite dimensional grid. Then, an undirected weighted graph is built using grid points as nodes and trajectory segments connecting points as arcs. Given starting and ending nodes,$A*$-like algorithm is used to select an optimal feasible trajectory compliant with prescribed constraints. Trajectory is expressed in terms of succession of segments to cross at prescribed velocities connecting starting and ending points. Feasibility is guaranteed by recurring to set-theoretic arguments involving the solution of Semi-definite Programming (SDP) minimization problems with Linear Matrix Inequalities (LMI) constraints.

- **Methodology B**— addresses the constrained navigation problem for autonomous robots moving in unknown cluttered environments. In particular, a novel control architecture is conceived whose the main features can be summarized as follows: anti-collision capabilities despite time-induced time-delay occurrences along the communication medium; mission accomplishment despite unpredictable obstacle occurrences along the nominal path. These properties are formally proven together with ultimate uniformly boundedness and constraints fulfillment of the regulated trajectory regardless of the vehicle uncertainties. The path to follow is expressed in terms of succession of waypoints connecting starting and ending vehicle poses whose feasibility is guaranteed by recurring to set-theoretic arguments involving the solution of Semi-definite Programming (SDP) minimization problems with Linear Matrix Inequalities (LMI) constraints. Final exper-

iments are provided to show the effectiveness and to highlight the main advantages of the proposed control architecture.

Both approaches exploit set-theoretic arguments to systematically ensure feasibility, safety, and robustness in real-time motion planning for a single SSTMR.

Methodology A

3.2 Set-Theoretic Feasible Trajectory Planning Method

This section presents the **Set-Theoretic Feasible Trajectory Planning (ST-FTP)** Method for SSTMRs subject to sliding phenomena and controlled through a communication network (NCS), originally proposed in [25, 106]. In order to solve the **OAMP-UCE** problem, the objective of the method is to generate trajectories composed of a sequence of straight segments, each to be traveled at a desired speed, which do not intersect obstacles in the operating environment, ensuring that these trajectories are dynamically achievable by the system. Starting from the tracking error dynamics of the SSTMR, which already accounts for sliding phenomena and network-induced delays (see Chapter 2), the proposed method is based on:

- the design of a robust constrained controller that ensures compliance with state and actuation constrains;
- the definition of a robust positive D-invariant ellipsoidal set representing the permissible evolutions of the tracking error in the presence of disturbances;
- the integration of these constraints into a graph-search planning algorithm which, using semi-definite optimization (SDP) problems with linear matrix inequality (LMI) constraints, finds trajectories compatible with the robot's dynamics and the control performances.

In summary, the purpose of the method is to provide a feasible trajectory planning solution capable of jointly taking into account constraints, uncertainties, and disturbances, going beyond geometric approach.

3.2.1 Trajectory Tracking Error

In the context of this thesis, the problem of safe navigation for tracked robots is addressed through the application of different control methodologies, each of which requires a specific representation of the system model. To address the issue of safe navigation, it may be useful to introduce a state vector representing the error of pose with respect to a desired pose both

expressed in a local reference system **L**. We define $q_D(t) = \begin{bmatrix} v_D \cdot t & 0 & 0 \end{bmatrix}^T$ as the desired pose of the robot along the trajectory's segments at a time t, in a local reference frame **L** centered in M with coordinates (x_0, y_0) which is aligned with the axis x according to the trajectory segment $\overline{MN}$, see Fig. 3.2 for details.

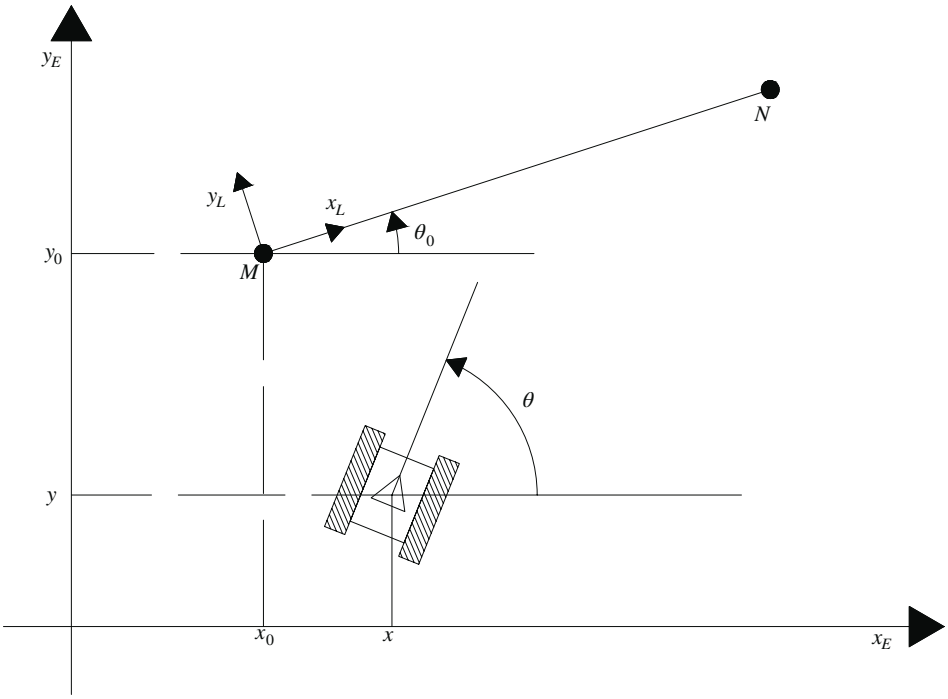


Fig. 3.2: Reference frames. Robot pose in the **E** frame is denoted as $\{x, y, \theta\}$, the **L** reference frame for trajectory tracking has its origin in M with the x-axis oriented in accordance with the assigned trajectory segment $\overline{MN}$.

The roto-translation to go from **E** to **L** is

$$q_L(t) = R_E^L(\theta_0)(q_E(t) - q_0) \tag{3.10}$$

where $q_0 = \begin{bmatrix} x_0 & y_0 & \theta_0 \end{bmatrix}^T$ with θ_0 represented in Fig. 3.2 and being

$$R_E^L(\theta_0) = \begin{bmatrix} cos(\theta_0) & sin(\theta_0) & 0 \\ -sin(\theta_0) & cos(\theta_0) & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{3.11}$$

Be

$$T_L^D(\cdot) = \begin{bmatrix} {q_L}^D(\cdot)^T & {u^D}^T \end{bmatrix}^T \tag{3.12}$$

the desired trajectory expressed in the reference frame **L**, defined in terms of permissible pairs of poses and control actions consistent with the kinematic relation (2.7) being $q_L^D(0) = \begin{bmatrix} 0 & 0 & 0 \end{bmatrix}^T$ and $u^D = \begin{bmatrix} V^D & 0 \end{bmatrix}^T$ where V^D is the desired nominal feedforward speed of the

robot in the trajectory segment $\overline{MN}$. A null nominal speed is assumed as rotational speed. At the time instant t, the desired pose expressed in the reference system $\mathbf{L}$ is calculated as follows

$$q_L^D(t) = \begin{bmatrix} V^D \cdot t & 0 & 0 \end{bmatrix}^T \tag{3.13}$$

Be

$$e(t) = q_L(t) - q_L^D(t) = \begin{bmatrix} e_x(t) & e_y(t) & e_\theta(t) \end{bmatrix}^T \tag{3.14}$$

the trajectory tracking error, $\delta u(t) = \hat{u}(t) - u^D = \begin{bmatrix} \delta V(t) & \omega(t) \end{bmatrix}^T$ the difference between the effective and nominal control velocity vectors and $d(t) = \begin{bmatrix} \delta\mu_r(t) & \delta\mu_l(t) \end{bmatrix}^T$ the vector of the variations of the sliding coefficients from the nominal values $\mu_r^N = 1$ and $\mu_l^N = 1$. By recombining eqs. (2.16), (3.10) and (3.13), the following equation is obtained

$$\dot{e}(t) = R_E^L(\theta_0) \cdot G(t) \cdot J \cdot H(t) \cdot J^{-1} \cdot \hat{u}(t) - \begin{bmatrix} V^D \\ 0 \\ 0 \end{bmatrix} \tag{3.15}$$

which can be rewritten in the scalar form

$$\begin{aligned} \dot{e}_x(t) &= \hat{V}(t) \cdot \overline{\mu}(t) \cdot \cos e_\theta(t) - \hat{\omega}(t) \cdot \frac{D}{2} \cdot \Delta_\mu(t) \cdot \cos e_\theta(t) - V^D \\ \dot{e}_y(t) &= \hat{V}(t) \cdot \overline{\mu}(t) \cdot \sin e_\theta(t) - \hat{\omega}(t) \cdot \frac{D}{2} \cdot \Delta_\mu(t) \sin e_\theta(t) \\ \dot{e}_\theta(t) &= \frac{-2}{D} \cdot \hat{V}(t) \cdot \Delta_\mu(t) + \hat{\omega}(t) \cdot \overline{\mu}(t) \end{aligned} \tag{3.16}$$

being $\overline{\mu}(t) = \frac{\mu_l(t)}{2} + \frac{\mu_r(t)}{2}$ and $\Delta_\mu(t) = \frac{\mu_l(t)}{2} - \frac{\mu_r(t)}{2}$. Finally, by applying the classical linearization procedure, it is possible to define a linear time-invariant representation around the nominal condition

$$\nabla(t) = \{q_L^D(t),\ u^D,\ \mu_r^N,\ \mu_l^N\} \tag{3.17}$$

$$\dot{e}(t) = Ae(t) + B\delta u(t) + B_D d(t) \tag{3.18}$$

$$A = \left.\frac{\partial \dot{e}(t)}{\partial e(t)}\right|_{\nabla(t)} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & V^D \\ 0 & 0 & 0 \end{bmatrix} \tag{3.19}$$

$$B = \left.\frac{\partial \dot{e}(t)}{\partial \hat{u}(t)}\right|_{\nabla(t)} = \begin{bmatrix} 1 & 0 \\ 0 & 0 \\ 0 & 1 \end{bmatrix} \tag{3.20}$$

$$B_D = \left.\frac{\partial \dot{e}(t)}{\partial \mu(t)}\right|_{\nabla(t)} = \begin{bmatrix} \frac{V^D}{2} & \frac{V^D}{2} \\ 0 & 0 \\ \frac{V^D}{D} & -\frac{V^D}{D} \end{bmatrix} \tag{3.21}$$

3.2.2 Modeling of networked control system

This thesis assumes a reliable connection-oriented and ordered communication protocol, where the number of packets lost during communication is assumed to be zero. Let us assume that the sensors operate synchronously: sampling occurs at times $t_k = k \cdot T_s$ with $k \in N$ and T_s is the sampling time. Assume that both the controller and the actuators are event-driven, i.e., they respond immediately to new data. Under these assumptions, the three delays above can be conveniently combined into a single delay term

$$\tau = \tau^{SC} + \tau^{CA} + \tau^{C} \tag{3.22}$$

hereafter referred to as network delay. Let τ_{max} (τ_{min}) be the maximum (minimum) value of τ. Let $\bar{d}$ be the smallest positive integer satisfying the relation $\bar{d} \geq \tau_{max}/T_s$. Let $\underline{d}$ be defined as the largest positive integer satisfying the relation $\underline{d} \leq \tau_{min}/T_s$. During a generic sampling period T_s, the control action can change at most $\bar{d} - \underline{d}$ times [107]. For simplicity, we will assume that $0 \leq \tau_{min} \leq \tau_{max} \leq T_s$ such that $\bar{d} = 1$ and $\underline{d} = 0$. According to Fig. 3.3, be t^1 the generic time instant between t_k and t_{k+1} when the control action changes from $\delta u(t_{k-1})$ to $\delta u(t_k)$. With recourse to the classical arguments of the theory of Networked

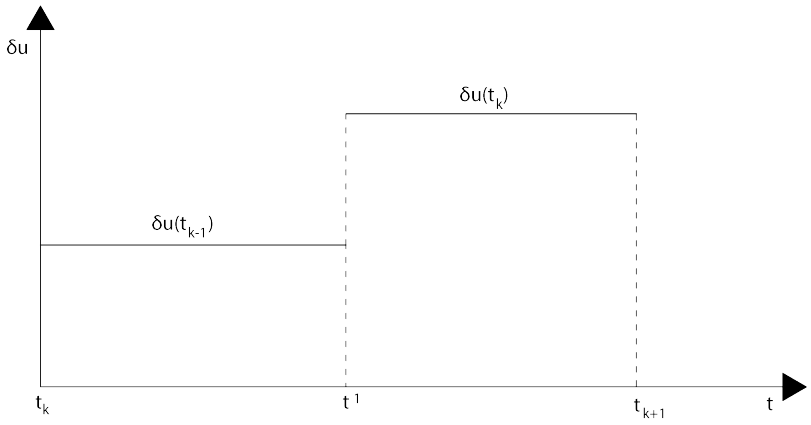


Fig. 3.3: Temporal diagram of actuation update

Control Systems [108], continuous time system (3.18) can be rewritten in the discrete form (3.26). The state of the system (3.18) at time t_{k+1} can be rewritten as:

$$e(t_{k+1}) = \mathrm{e}^{A(t_{k+1}-t^1)} e(t^1) + \int_0^{t_{k+1}-t^1} \mathrm{e}^{A\sigma} d\sigma B \delta u(t_k) + \int_0^{t_{k+1}-t^1} \mathrm{e}^{A\sigma} d\sigma B_D d(t_k) \tag{3.23}$$

In the same manner

$$e(t^1) = \mathrm{e}^{A(t^1-t_k)}e(t_k) + \int_0^{t^1-t_k} \mathrm{e}^{A\sigma}d\sigma B\delta u(t_{k-1}) + \int_0^{t^1-t_k} \mathrm{e}^{A\sigma}d\sigma B_D d(t_k) \tag{3.24}$$

By appropriately recombining (3.23) and (3.24), the following form can be obtained

$$e(t_{k+1}) = \mathrm{e}^{AT_s}e(t_k) + \int_0^{t_{k+1}-t^1} \mathrm{e}^{A\sigma}d\sigma B\delta u(t_k) + \int_{t_{k+1}-t^1}^{Ts} \mathrm{e}^{A\sigma}d\sigma B\delta u(t_{k-1}) + \tag{3.25}$$
$$+ \int_0^{Ts} \mathrm{e}^{A\sigma}d\sigma B_D d(t_k)$$

For sake of clarity, the time-dependent functions sampled at time instants t_k will henceforth be denoted by the notation $f_k = f(t_k)$. Eq. (3.25) can be rewritten in the following form

$$\tilde{\xi}_{k+1} = \underbrace{\begin{bmatrix} \mathrm{e}^{AT_s} & \int_{t_{k+1}-t^1}^{Ts} \mathrm{e}^{A\sigma}d\sigma B \\ \mathbf{0} & \mathbf{0} \end{bmatrix}}_{\tilde{A}(t_{k+1}-t^1)} \tilde{\xi}_k + \underbrace{\begin{bmatrix} \int_0^{t_{k+1}-t^1} \mathrm{e}^{A\sigma}d\sigma B \\ I \end{bmatrix}}_{\tilde{B}(t_{k+1}-t^1)} \delta u_(t_k) + \underbrace{\begin{bmatrix} \int_0^{Ts} e^{A\sigma}d\sigma B_D \\ \mathbf{0} \end{bmatrix}}_{\tilde{B}d} d_k \tag{3.26}$$

where

$$\tilde{\xi}_k = \begin{bmatrix} e_k^T & \delta u_{k-1}^T \end{bmatrix}^T \in \mathcal{R}^{n_s} \tag{3.27}$$

represents the vector of states, hereinafter named *lifted trajectory tracking error*, being $n_s = n_e + (\bar{d} - \underline{d}) \cdot n_u$ with n_e and n_u the number of states and inputs of (3.18) respectively (in this case $n_e = 3$ and $n_u = 2$).

For the purposes of this thesis, it is useful to conservatively approximate the representation (3.26) as uncertain system with norm-bounded uncertainty. Thus, resorting to the procedure proposed in [109], the system (3.26) was embedded in the following classical representation

$$\tilde{\xi}_{k+1} = \tilde{A}\tilde{\xi}_k + \tilde{B}\delta u_k + \tilde{B}_D d_k + \tilde{B}_p p_k \tag{3.28}$$

$$p_k = \Delta_k q_k \tag{3.29}$$

$$q_k = \tilde{C}_q\tilde{\xi}_k + \tilde{D}_q\delta u_k \tag{3.30}$$

where $\|\Delta_k\| < 1 \; \forall k \geq 0$, and $\tilde{B}_p$, $\tilde{C}_q$ and $\tilde{D}_q$ are matrices of proper dimensions, the vector p_k is the way by which the uncertainty of (3.26) is considered. More details can be found in [109].

3.2.3 Trajectory tracking control design

This section presents the trajectory tracking control design. Such a step is required to properly introduce the constraints for feasible trajectory planning task. A generic discrete-time control system with a linear time-invariant structure is considered

$$z_{k+1} = A_C z_k + B_C\tilde{\xi}_k \tag{3.31}$$

$$\delta u_k = C_C z_k + D_C \tilde{\xi}_k \tag{3.32}$$

being $z \in \mathbb{R}^{n_c}$ the vector of control system state variables.

Let

$$\xi_k = \begin{bmatrix} \tilde{\xi}_k^T & z_k^T \end{bmatrix}^T \in \mathbb{R}^{n_\xi} \tag{3.33}$$

be the *closed-loop trajectory tracking error* vector, (with $n_\xi = n_c + n_s$) the vector of closed-loop system state variables. The dynamics of the closed-loop trajectory tracking error can be rewritten in the form of an uncertain system with norm-bounded uncertainty subject to external disturbance by combining the eqs. (3.28)-(3.30) and (3.31)-(3.32):

$$\xi_{k+1} = \phi \xi_k + H_D d_k + B_p p_k \tag{3.34}$$

$$p_k = \Delta_k q_k \tag{3.35}$$

$$q_k = \Sigma_q \xi_k \tag{3.36}$$

with

$$\phi = \begin{bmatrix} \tilde{A} + \tilde{B} D_C & \tilde{B} C_C \\ B_C & A_C \end{bmatrix} \tag{3.37}$$

$$H_D = \begin{bmatrix} \tilde{B}_D \\ 0 \end{bmatrix} \tag{3.38}$$

$$B_p = \begin{bmatrix} \tilde{B}_p \\ 0 \end{bmatrix} \tag{3.39}$$

$$\Sigma_q = \begin{bmatrix} (\tilde{C}_q + \tilde{D}_q D_C)^T \\ (\tilde{D}_q C_C)^T \end{bmatrix}^T \tag{3.40}$$

3.2.4 Definition of the positive robust D-invariant region

Consider the following ellipsoidal constraints on the state of the system ξ and the input δ_u:

$$\xi \in \Omega_\xi,\ \Omega_\xi = \{\xi \in \mathcal{R}^{n_\xi} : \xi^T S_\xi \xi \leq 1,\ S_\xi > 0\}, \tag{3.41}$$

$$\delta_u \in \Omega_u,\ \Omega_u = \{\delta_u \in \mathcal{R}^{n_u} : \delta_u^T S_u \delta_u \leq 1,\ S_u > 0\}. \tag{3.42}$$

Furthermore, assume that the n_d external perturbations acting on (3.34) can be bounded according to the following ellipsoidal condition

$$d \in \Omega_d,\ \Omega_d = \{d \in \mathcal{R}^{n_d} : d^T M_d d \leq 1,\ M_d > 0\} \tag{3.43}$$

Let

$$\Gamma_0 = \{\xi \in \mathcal{R}^{n_\xi} : \xi^T P_0 \xi \leq 1 \;\; P_0 \geq 0\} \tag{3.44}$$

be an ellipsoidal D-invariant robust region [110, 111] for the uncertain system (3.34)-(3.36) compatible with the constraints (3.41)-(3.43).

Lemma 1 *The region (3.44) is robust positive invariant concerning the disturbance (3.43) if the two following statements are fulfilled*

Statement 1 *There exist two positive scalars τ_1 and τ_2 such that*

$$\tilde{P} \geq \tau_2 P_0 \tag{3.45}$$

with

$$\begin{aligned}\tilde{P} =& P_0 - \phi^T P_0 \phi - \tau_1 \Sigma_q^T \Sigma_q - \\ &- \begin{bmatrix} H_D^T P_0 \phi \\ B_p^T P_0 \phi \end{bmatrix}^T \begin{bmatrix} -H_D^T P_0 H_D + \tau_2 M_d & -H_D^T P_0 B_p \\ * & -B_p^T P_0 B_p + \tau_1 I \end{bmatrix}^{-1} \begin{bmatrix} H_D^T P_0 \phi \\ B_p^T P_0 \phi \end{bmatrix}\end{aligned} \tag{3.46}$$

being scalars τ_1 and τ_2 solution of the following GEVP problem

$$\min_{\tau_2} \tau_2 \tag{3.47}$$

$$\tau_1 \geq 0 \tag{3.48}$$

$$\tau_2 \geq 0 \tag{3.49}$$

s.t.

$$\begin{bmatrix} P_0 - \phi^T P_0 \phi - \tau_1 \Sigma_q^T \Sigma_q & \phi^T P_0 H_D & \phi^T P_0 B_p \\ * & -H_D^T P_0 H_D + \tau_2 M_d & -H_D^T P_0 B_p \\ * & * & -B_p^T P_0 B_p + \tau_1 I \end{bmatrix} \geq 0 \tag{3.50}$$

$$\begin{bmatrix} -H_D^T P_0 H_D + \tau_2 M_d & -H_D^T P_0 B_p \\ * & -B_p^T P_0 B_p + \tau_1 I \end{bmatrix} > 0 \tag{3.51}$$

Statement 2 *There are three positive scalars σ_1, σ_2 and σ_3 solution of the following feasibility problem*

$$\sigma_1 \geq 0 \tag{3.52}$$

$$\sigma_2 \geq 0 \tag{3.53}$$

$$\sigma_3 \geq 0 \tag{3.54}$$

$$1 - \sigma_1\tau_2 - \sigma_2 \geq 0 \tag{3.55}$$

$$\begin{bmatrix} -\phi^T P_0 \phi + \sigma_1 \tilde{P} - \sigma_3 \Sigma_q^T \Sigma_q & -\phi^T P_0 H_D & -\phi^T P_0 B_p \\ * & -H_D^T P_0 H_D + \sigma_2 M_d & -H_D^T P_0 B_p \\ * & * & -B_p^T P_0 B_p + \sigma_3 I \end{bmatrix} \geq 0 \tag{3.56}$$

Proof. See Appendix.

3.2.5 Trajectory feasibility condition

Consider the operational scenario $\Psi \subseteq \mathcal{R}^2$, representing the $\mathcal{O}_{free}^E$ environment. Let $\mathcal{C}$ be the grid of n points discretizing Ψ. Be c_i, where $i = 1 \cdots n$, the point of $\mathcal{C}$ with coordinates in the **E** inertial reference frame

$$\begin{bmatrix} x_{c_i} & y_{c_i} \end{bmatrix}^T$$

Finally, define the undirected weighted graph $\mathcal{G}$, whose points are the elements of $\mathcal{C}$, and consider two generic elements $\{c_1\} \in \mathcal{C}$ and $\{c_2\} \in \mathcal{C}$. Let $\overline{c_1 c_2}$ be the linear segments connecting c_1 with c_2. If the following conditions are met, the pair $\{c_1, c_2\}$ is connected by an arc denoted by g_{c_1,c_2}:

1. the segment $\overline{c_1 c_2}$ is completely within the operational scenario Ψ and guarantee a collision-free movement with known obstacles
2. $c_1 \neq c_2$
3. in order to limit the connectivity degree of $\mathcal{G}$, the segments $\overline{c_1 c_2}$ has a maximum length smaller than $\overline{L} \geq 0$

The nodes of $\mathcal{G}$ represent possible robot positions within the operational scenario, while the arcs represent some possible maneuvers where the vehicle moves on a collision-free trajectory segment.

Let $S_1 \in \mathcal{C}$ and $F_1 \in \mathcal{C}$ be the departure and arrival points of the robot.

The problem of feasible trajectory planning is to determine, whether there is, an admissible sequence of $\mathcal{G}$ arcs connecting the elements $\{S_1\}$ and $\{F_1\}$ while accounting for the robot's uncertain dynamics (3.34)-(3.36), the unknown but bounded external disturbance (3.43) and the presence of ellipsoidal (3.41) and (3.42) constraints.

Consider Fig. 3.4. Let A and B be two points of $\mathcal{C}$. The segment of the trajectory connecting them is denoted by $\overline{AB}$ and has length

$$l_{AB} = \sqrt{(x_A - x_B)^2 + (y_A - y_B)^2} \leq \overline{L}. \tag{3.57}$$

Assume that the trajectory segment $\overline{AB}$ is to be traversed at constant nominal speed V^{AB}. Let N_{AB} be the maximum positive integer for which the following inequality holds

$$N_{AB} \leq \frac{l_{AB}}{V^{AB} \cdot T_s} \tag{3.58}$$

where T_s is the sampling time. Let $\left[x^D(t) \;\; y^D(t)\right]^T$ be the desired position expressed in the inertial frame $\mathbf{E}$ at time instant t. Moreover be $\left[x(t) \;\; y(t)\right]^T$ the robot position expressed in the inertial frame $\mathbf{E}$ at time instant t. Consider the point $C \in \mathcal{C}$ and assume a trajectory consisting of the sequence of the two adjacent segments $\overline{AB}$ and $\overline{BC}$. Let t_A be the time when the robot is in A. Let $[t_A, t_A + N_{AB} \cdot T_s]$ be the time interval in which the robot moves along the segment $\overline{AB}$. At time $t_{N_{AB}} = t_A + N_{AB} \cdot T_s$ the transition between the two trajectory segments $\overline{AB}$ and $\overline{BC}$ occurs. Thus, at $t_{N_{AB}}$ the closed-loop trajectory tracking error ξ undergoes the instantaneous change denoted by

$$\Pi = \left[x^D(t_{N_{AB}}) - x_B \; y^D(t_{N_{AB}}) - y_B \; \delta_\theta \; 0 \cdots 0\right]^T \tag{3.59}$$

See Fig. 3.4 for details. The transition is considered admissible if the following condition is met

$$(\xi_{N_{AB}} - \Pi) \in \Gamma_0 \tag{3.60}$$

where $\xi_{N_{AB}}$ is the value of ξ before the transition from segment $\overline{AB}$ to segment $\overline{BC}$ occurs. To define $\xi_{N_{AB}}$, let Γ_A be the set to which ξ belongs at time t_A

$$\Gamma_A = \{\xi \in \Gamma_0 : \xi^T P_A \xi \leq 1, \, P_A \geq 0\} \tag{3.61}$$

The ellipsoidal set $\Gamma_{N_{AB}} \subseteq \Gamma_0$ containing all admissible tracking errors $\xi_{N_{AB}}$ can be obtained by exploiting the following result.

Lemma 2 *Consider the uncertain system (3.34)-(3.36) subject to the bounded disturbance (3.43). Let*

$$\Gamma_k = \{\xi_k \in \Gamma_0 : \xi_k^T P_k \xi_k \leq 1, P_k > 0\} \subseteq \Gamma_0 \tag{3.62}$$

be the ellipsoidal region containing the state ξ at time t_k. The minimum volume ellipsoidal region [109] containing the state ξ at time t_{k+1}

$$\Gamma_{k+1} = \{\xi_{k+1} \in \Gamma_0 : \xi_{k+1}^T P_{k+1} \xi_{k+1} \leq 1, \, P_{k+1} > 0\} \subseteq \Gamma_0 \tag{3.63}$$

can be obtained by solving the following minimization problem:

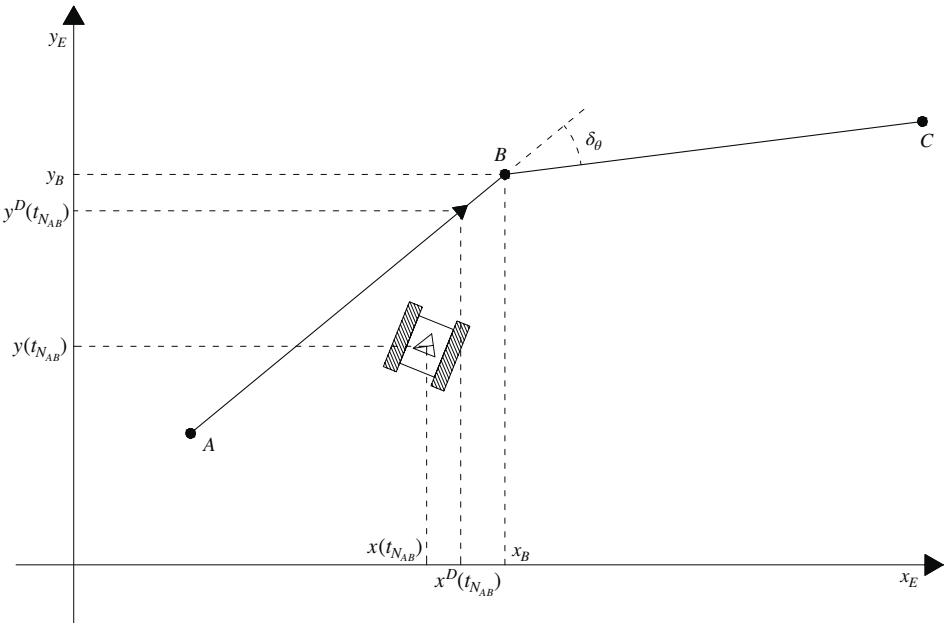


Fig. 3.4: Trajectory switch between segments $\overline{AB}$ and $\overline{BC}$. $x^D(t_{N_{AB}})(y^D(t_{N_{AB}}))$ represents the nominal robot position in $\mathbf{E}$ at time $t_{N_{AB}}$

$$\min \log \det P_{k+1}^{-1} \tag{3.64}$$

$$P_{k+1} > 0, \tau_1 \geq 0, \tau_2 \geq 0, \tau_3 \geq 0 \tag{3.65}$$

$$\begin{bmatrix} (-\phi^T P_{k+1}\phi + \tau_1 P_k - & & \\ -\tau_3 \Sigma_q^T \Sigma_q) & -\phi^T P_{k+1} H_D & -\phi^T P_{k+1} B_p \\ & (-H_D^T P_{k+1} H_D + & \\ * & + \tau_2 M_d) & -H_D^T P_{k+1} B_p \\ * & * & -B_p^T P_{k+1} B_p + \tau_3 \end{bmatrix} \geq 0 \tag{3.66}$$

$$1 - \tau_1 - \tau_2 \geq 0 \tag{3.67}$$

Proof. See Appendix.

Note that, taking into account the asymptotic stability of the system (3.34)-(3.36) and the properties of the robust D-invariant region (3.44), it can be observed that Γ_{k+1} is encompassed by Γ_0. Consequently, Lemma 2 provides a reduction in the volume of Γ_{k+1} compared to the case where $\Gamma_{k+1} = \Gamma_0$.

Given $\Gamma_{N_{AB}}$, the condition (3.60) is verified if the following condition is satisfied

$$\Gamma_B \subseteq \Gamma_0 \tag{3.68}$$

being

$$\Gamma_B = \{\xi \in \Gamma_0 : \xi^T P_B \xi \leq 1,\ P_B > 0\} \tag{3.69}$$

the smallest ellipsoidal set such that

$$\Gamma_{N_{AB}} \bigoplus \Pi = \{\xi + \Pi | \xi \in \Gamma_{N_{AB}}\} \subseteq \Gamma_B. \tag{3.70}$$

Eq.(3.70) implies that

$$(\xi + \Pi)^T P_B (\xi + \Pi) \leq 1,\ \forall \xi^T P_{N_{AB}} \xi \leq 1 \tag{3.71}$$

By recurring to the same arguments of Lemma 2, (3.69) can be obtained by solving the following minimization problem:

$$\min \log\det P_B^{-1} \tag{3.72}$$

$$P_B > 0, \tau \geq 0 \tag{3.73}$$

$$\begin{bmatrix} 1 - \Pi^T P_B \Pi - \tau & -\Pi^T P_B \\ * & -P_B + \tau P_{N_{AB}} \end{bmatrix} \geq 0 \tag{3.74}$$

If eq.(3.68) holds, the sequence of segments $\overline{AB}$ and $\overline{BC}$ is considered *feasible*.

3.2.6 Algorithm for feasible trajectory planning

This work exploits an $A^* - like$ algorithm to plan a feasible trajectory, connecting a starting point $S \in \mathcal{G}$ with final point $F \in \mathcal{G}$, consisting in a sequence of segments, possibly of minimum length, traveled at constant speed. A^* relies upon the exploration of the most promising arc according to an heuristic function estimating the lower-bound of the cost of path including arcs to be explored. If the heuristic function never overestimates path cost, as shown in [25,112,113], the algorithm guarantees the exploration of fewer nodes than any other algorithm using the same heuristic. The graph search method A^* is used here with the following heuristic function: consider the trajectory connecting the initial node S to $M \in \mathcal{G}$, denoted hereafter as c_{SM}. More precisely, M represents the last node of the trajectory under investigation. Let $\overline{MN}$ be the candidate segment to be concatenated with c_{SM}. Let $l(SM)$ be the length of c_{SM}. The structure of the heuristic function here considered is shown below:

$$f(c_{SM}, \overline{MN}, \overline{NF}) = l(SM) + \|M - N\| + \|N - F\| \tag{3.75}$$

where $\|\cdot\|$ denotes the Euclidean norm. It is easy to observe that the function (3.75) provides a lower bound on the length of the final trajectory connecting S to F by including c_{SM} and $\overline{MN}$, since the length of the trajectory $l(MF)$ connecting M and F (if it exists) is such that $l(MF) \geq \|M - F\|$. Given (3.75), the trajectory planning algorithm is shown in Fig. 3.5. The recursive function is detailed in the following Pseudocode.

Feasible Trajectory Planning Algorithm

Input: F, X, c_{opt}, c_{exp};

Output: c_{opt};

Function TRAJECTORYPLANNER(F, X, c_{opt}, c_{exp})
Be $\overline{AX}$ the last arc of c_{exp}
for all arcs $\overline{XY}$ whose origin is X **do**
  if $f(c_{exp}, \overline{XY}, \overline{YF}) < l(c_{opt}) \leftarrow$ length of c_{opt} **then**
    Given Γ_A as defined in (3.61), find $\Gamma_{N_{AX}}$ according to Lemma 2
    Given the succession of segments $\overline{AX}$ and $\overline{XY}$, compute Γ_X according to (3.72–3.73)
    if $\Gamma_X \subseteq \Gamma_0$ according to (3.68) **then**
      Append $\overline{XY}$ to c_{exp}
      if $Y = F$ is the destination node **then**
        $c_{opt} = c_{exp}$
      else
        Call TRAJECTORYPLANNER(F, Y, c_{opt}, c_{exp})
      end if
    end if
  end if
end for
return c_{opt}

3.2.7 Results — Experimental Validation

A campaign of experimental simulations was conducted using the SSTMR Jaguar V4 from Dr. Robot, Fig. 3.6, available at the Control Systems Laboratory of the University "Mediterranea" of Reggio Calabria. It is a tracked mobile robot, the gear

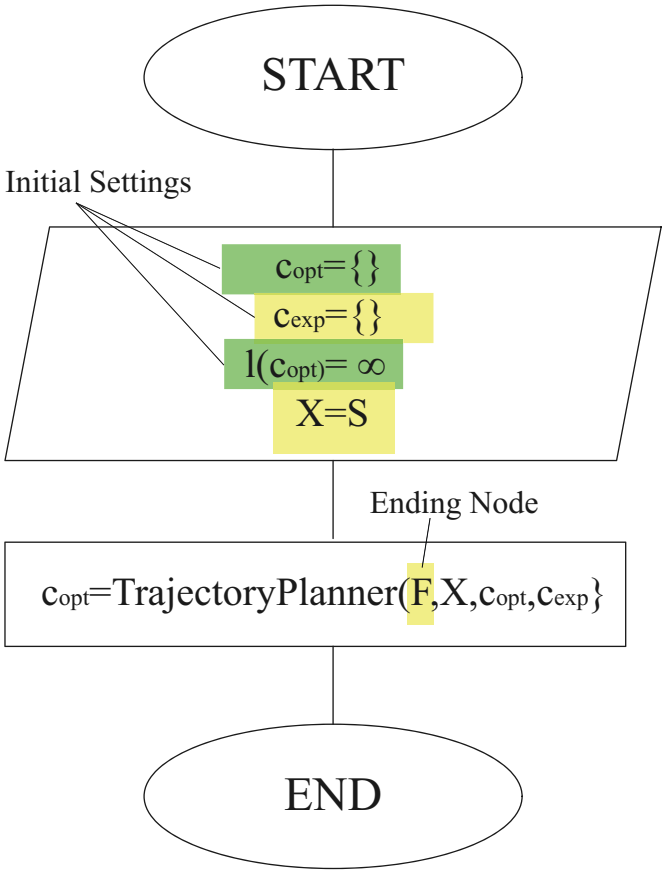


Fig. 3.5: Flowchart of the feasible trajectory planning algorithm

connecting the motors to tracks has $R = 8\,cm$, the distance between left and right tracks is $D = 50\,cm$.

It is equipped with two **LIDAR UTM-30LX** sensors, mounted on the front and backward, used by an on-board Raspberry Pi unit that estimates the robot's position and direction in **E** using the **AMCL** algorithm. The trajectory tracking control is implemented on a remote ground station connected to the robot via an 802.11n wireless data communication network using the TCP/IP protocol stack, see Fig. 3.7. A sampling time $T_s = 0.2\,s$ and a nominal forward speed $\bar{V} = 0.20\,m/s$ were chosen. Notice that, a motor control is also implemented on board to regulate the speed of the tracks. The response times of the latter are significantly smaller than T_s and are therefore considered instantaneous.

The operational scenario considered is an indoor space with an area of approximately $80\,m^2$ and is shown in Fig. 3.8. Within the operating scenario, several measurements were performed to estimate the sliding coefficients μ_r and μ_l as well as the communication delay τ caused by the data communication network, all introduced

Fig. 3.6: Skid-Steered Tracked Mobile Robot Jaguar V4 by Dr.Robot

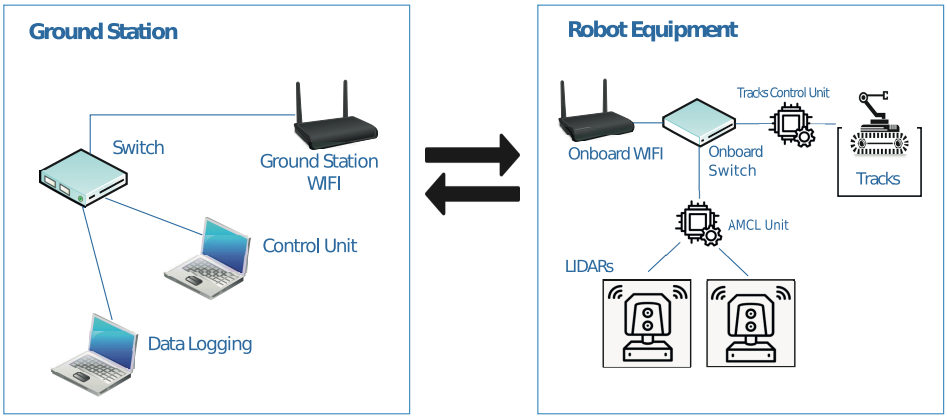


Fig. 3.7: Control framework details: trajectory tracking control action is performed on a remote computer connected to the mobile robot through a Wi-Fi communication network using TCP-IP protocol stack.

in Sect. 2.2. The measured values are as follows: $\mu_r\,(\mu_l) \in [0.8,\ 1.2]$ and according to a measurement campaign $\tau_{max} = 4$ sampling steps. In accordance with (3.27), the following lifted trajectory tracking error was defined with $n_s = 5$ components

$$\tilde{\xi}_k = \left[e_k^T\ \delta u_{k-1}^T\right]^T \tag{3.76}$$

and the uncertain discrete-time mathematical model (3.41-3.43) was calculated. For control, a proportional-integral (PI) control law was considered

$$z_{k+1} = A_C z_k + B_C \tilde{\xi}_k \tag{3.77}$$

$$\delta u_k = C_C z_k + D_C \tilde{\xi}_k \tag{3.78}$$

where $A_C = I$, $B_C = S \cdot T_s$, $C_C = Ki$, $D_C = Kp$, being $S \in \mathcal{R}^{2\times 5}$ a matrix that selects tracking errors e_x ed e_y

$$S = \begin{bmatrix} 1\,0\,0\,0\,0 \\ 0\,1\,0\,0\,0 \end{bmatrix} \tag{3.79}$$

whereas Kp and Ki are the gain matrices of proper dimensions tuned by trial and error procedure.

The following constraints were considered:

(i) maximum allowable tracking error of $0.5\,m$ on e_x and e_y and $60\,deg$ on e_θ;
(ii) $V \in [0,\, 0.4]\,m/s$ and $\omega \in [-30,\, 30]\,deg/s$

The robust ellipsoidal D-invariant region (3.44) has been calculated, the quadratic function $\|\xi|^2_{P_0} = 1$ defines its limit.

The operational scenario was discretized using a grid of approximately 1000 points spaced appropriately from obstacles to account for the actual footprint of the Jaguar V4 robot. An initial tracking error of

(a) $0.1\,m$ in each direction and $10\,deg$ for orientation were assumed.

The was used to find a feasible trajectory connecting the starting position (square) and the target (triangle), see Fig. 3.8. The resulting optimal trajectory consists of 64 segments with a total length of $24.58\,m$. To test the feasibility of the planned trajectory, 10 experimental simulations were performed varying the initial condition in accordance with the constraint (a). The results are shown in Figs. 3.8-3.11. Specifically, Fig. 3.8 shows the experimental trajectories (black lines) traversed by the robot within the operational scenario. Figs. 3.9 and 3.10 show the values assumed by the trajectory tracking error and the values of the robot's control speeds, respectively. The red dashed lines indicate the relative bounds. Fig. 3.11 shows the experimental values of $\|\xi\|^2_{P_0}$. The red line represents the boundary beyond which fault is detected. As expected, since the planned trajectory is feasible, the closed loop trajectory tracking error always stays within (3.44) and this brings the number of false alarms detected in these simulations to zero.

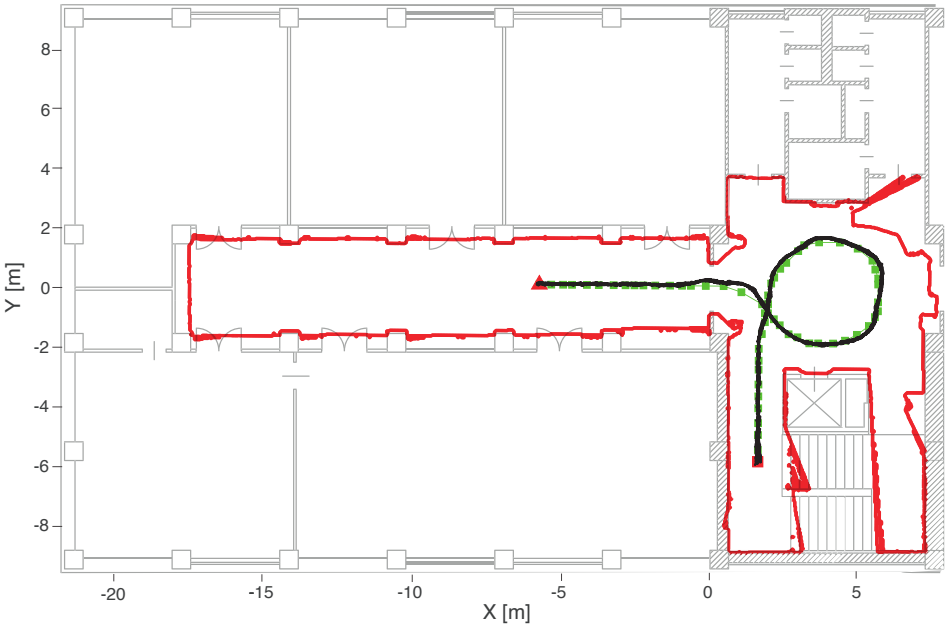


Fig. 3.8: The planned feasible trajectory (green line) connecting the starting point (red square) to the ending point (red triangle) is overlaid the original floor plan. The red line represents the boundary of the operational scenario measured using the SLAM algorithm. Black solid lines represent the trajectories performed.

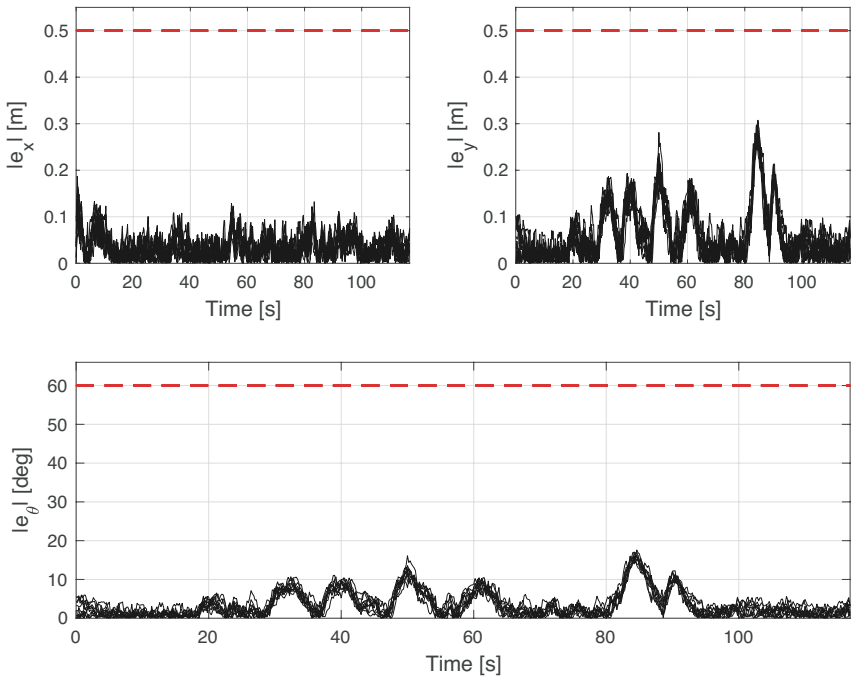


Fig. 3.9: Red dashed lines represent bounds. Black solid lines represent trajectory tracking errors.

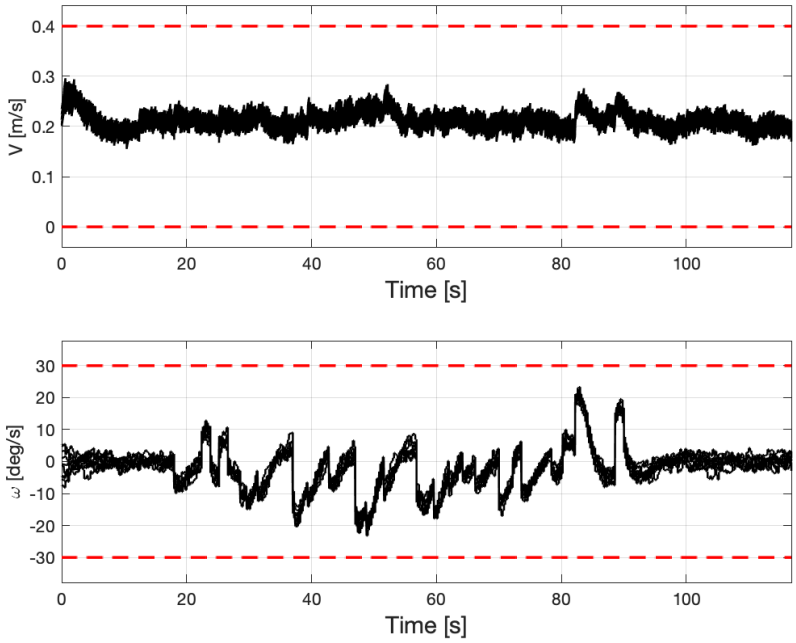


Fig. 3.10: Red dashed lines represent bounds. Black solid lines represent control speeds.

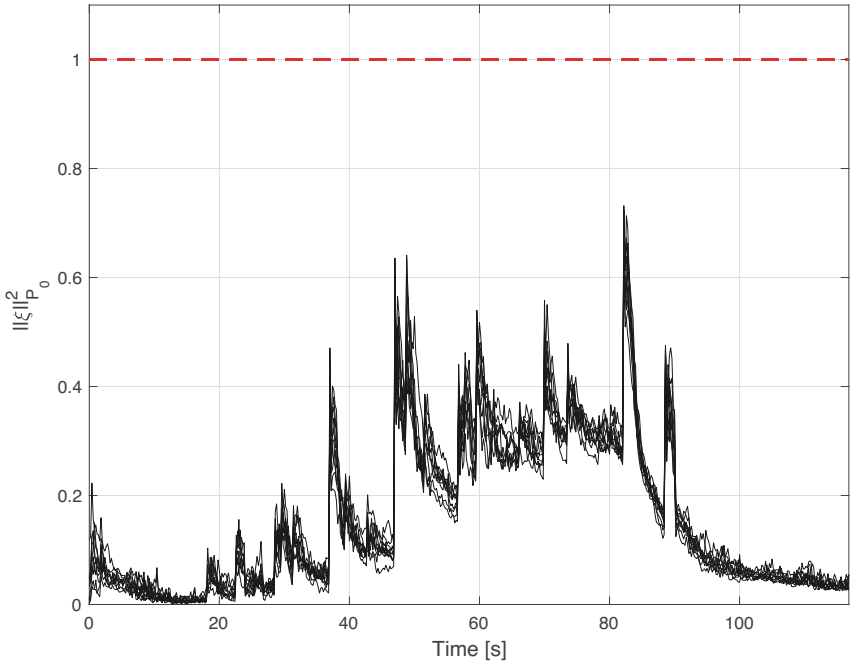


Fig. 3.11: Red dashed line represents the bound condition. Black solid lines represent the values of $\|\xi\|^2_{P_0}$ related to D-invariant region (3.44).

Methodology B

3.3 Set-Theoretic Model Predictive Control Method

This section presents the **Set-Theoretic Model Predictive Control (ST-MPC)** Method for SSTMRs operating in unknown cluttered environments and controlled through a communication network (NCS) originally proposed in [114]. The method addresses the **OAMP-UCE** problem by ensuring safe navigation from the initial to the target state, while handling uncertainties, external disturbances, and network-induced time delays. The main features of the method are summarized as follows:

- a networked control framework that provides anti-collision capabilities despite time delays occurring along the communication medium;
- the design of a robust MPC scheme based on set-theoretic techniques, guaranteeing compliance with state and input constraints even in the presence of vehicle uncertainties and unpredictable obstacles;
- formal guarantees of ultimate uniform boundedness of the regulated state trajectory and satisfaction of all constraints under disturbances and latency effects.

In summary, the purpose of **ST-MPC** is to provide a robust control architecture capable of achieving mission objectives in cluttered environments while ensuring safety and constraint fulfillment, independently of unpredictable disturbances and communication delays.

Notation and preliminaries

Given a set $S \subseteq \mathbb{R}^n$, we denote by $In[S] \subseteq S$ its inner ellipsoidal approximation, namely the ellipsoid of maximum volume that lies entirely within S.Such an approximation provides a compact and tractable representation of S, especially useful for analysis and control applications. Moreover, if $S \subseteq X \times Y \subseteq \mathbb{R}^n \times \mathbb{R}^m$, the projection

of the set S onto X is defined as $\text{Proj}_X(S) := \{x \in X \,|\, \exists y \in Y \text{ s.t. } (x,y) \in S\}$. This operation allows one to reduce the dimensionality of the problem by focusing only on the components of interest, disregarding the remaining variables.

Definition 3.3. *Given a set $W \subset \mathbb{R}^n$ and a point $x \in \mathbb{R}^n$, the distance is defined as:*

$$dist(x, W) = \inf_{w \in W} \|x - w\|_*,$$

where $*$ is any relevant norm.

Let us consider the discrete time model

$$x(t+1) = A\,x(t) + B\,u(t) \tag{3.80}$$

where $x(t) \in \mathbb{R}^{n_x}$, $u(t) \in \mathbb{R}^{n_u}$ and

$$\left[\begin{array}{cc} A & B \end{array}\right] \in \mathbf{Co}\left(\left\{\left[\begin{array}{cc} A_i & B_i \end{array}\right]\right\}_{i=1}^{n_p}\right) \tag{3.81}$$

where $\mathbf{Co}\{\cdot\}$ is the convex hull operator [109]. Moreover,

$$\begin{array}{l} u(t) \in \mathcal{U} := \{u \in \mathbb{R}^m : u^T u \leq u_{max}^2\}, \\ x(t) \in \mathcal{X} := \{x \in \mathbb{R}^n : x^T x \leq x_{max}^2\} \quad \forall t \geq 0, \end{array} \tag{3.82}$$

with $\mathcal{U}$, $\mathcal{X}$ convex and compact subsets of $\mathbb{R}^{n_u}$ and $\mathbb{R}^{n_x}$, respectively.

Definition 3.4. *[115] A set $\Xi \subseteq \mathcal{X}$ is said to be a Robust Positively Invariant set for (3.80) under constraints (3.82) if there exists a control law $u := h(x(t)) \in \mathcal{U}$ such that $\forall\, x(0) \in \Xi \rightarrow Ax(t) + Bh(x(t)) \in \Xi$, $\forall [A \;\; B]$ complying with (3.81),$\forall t \in \mathbb{Z}_+$.*

Definition 3.5. *[115] Let S be a neighborhood of the origin. The closed-loop trajectory of (3.80), namely $x_{CL}(\cdot)$, is said to be Uniformly Ultimately Bounded (UUB) in S if for all $\mu > 0$ there exist $T(\mu) > 0$ and $u(t) \in \mathcal{U}$ such that, for every $\|x(0)\| \leq \mu$, $x_{CL}(t) \in S$, $\forall$ possible realizations (3.81) and $t \geq T(\mu)$.*

Lemma 3 *Given the plant (3.80) subject to (3.82). Then, the state-feedback control law*

$$u(t) = K\,x(t - \tau(t)),\ \tau(t) \leq \bar{\tau}, \forall t \geq 0, \tag{3.83}$$

where $\tau(t)$ denotes the time-varying round-trip delay in the state-feedback loop, and $\bar{\tau} < \infty$ is a known positive scalar representing the maximum admissible delay, which stabilizes the regulated plant

$$x(t+1) = Ax(t) + B\,K\,x(t - \tau(t)) \tag{3.84}$$

and satisfies the constraints (3.82) is obtained as the solution of the following matrix inequalities

$$\begin{bmatrix} E^T P E - \mathcal{S} & \mathcal{A}^T P \\ P \mathcal{A} & P \end{bmatrix} \geq 0, \tag{3.85}$$

$$\begin{bmatrix} u_{max}^2 E^T P E & \begin{bmatrix} K^T \\ 0 \end{bmatrix} \\ \begin{bmatrix} K & 0 \end{bmatrix} & I \end{bmatrix} \geq 0 \tag{3.86}$$

$$E^T P E \geq \frac{1}{x_{max}^2} \tag{3.87}$$

where

$$\mathcal{S} := diag\{S, 0\}\,,\ S = S^T \geq 0, \mathcal{A} = \begin{bmatrix} I & I \\ (A - I + B K) & -I \end{bmatrix}.$$

Moreover, the ellipsoidal set

$$\begin{aligned} \mathcal{E} &:= Proj_x\{\bar{x} \in \mathbb{R}^{2n} \mid \bar{x}^T E^T P E^T \bar{x} \leq 1\} \\ &= \{x \in \mathbb{R}^n \mid x^T Q x \leq 1\}, Q = Q^T > 0, \end{aligned} \tag{3.88}$$

is a RPI region for the closed-loop state evolutions (3.84) such that $\mathcal{E} \subset \mathcal{X}$ *and* $K\mathcal{E} \subset \mathcal{U}$*, irrespective of any time delay occurrence* $\tau(t) \leq \bar{\tau}$*.*

Proof. For the proof, see [116].

Lemma 4 *Let* $\mathcal{E} \neq \emptyset$ *be a given RPI ellipsoidal region determined according to Lemma 3. The sequence of one-step state ahead controllable sets [115], compliant with (3.82) and with any realization of the time delay* $\tau(t) \leq \bar{\tau}$*, is given by the following family of ellipsoids:*

$$\begin{aligned} &\mathcal{E}_0 = \mathcal{E} \\ &\mathcal{E}_i = Proj_x\{In[\bar{x}_{aug} \in \mathcal{X} \times \mathcal{X} \times \mathcal{X} \times \mathcal{X} : \exists\ u \in \mathcal{U} : \\ &\qquad Proj_x\{\bar{A}_{aug}\bar{x}_{aug} + \bar{B}_{aug}u\} \in \mathcal{E}_{i-1}]\} \end{aligned} \tag{3.89}$$

$$\bar{A}_{aug} = \begin{bmatrix} I & I & 0 & 0 \\ A - I & -I & -\bar{\tau}A & \bar{\tau}A \\ 0 & 0 & 0 & I \\ 0 & 0 & 0 & A \end{bmatrix},$$

$$\bar{B}_{aug} = \begin{bmatrix} 0 & B^T & 0 & 0 \end{bmatrix}^T$$

where $x_{aug} = \begin{bmatrix} x^T & y^T & z_1^T & z_2^T \end{bmatrix}^T$*, with* y *accounting for the one-step difference* $x(t+1) - x(t), \forall t$*, and* $z_1, z_2 \in \mathcal{E}_{i-1}, \forall i$*, two slack variables taking care of all the cumulative sum vectors* $y(j),\ j = t - \bar{\tau}, \ldots, t - 1$*.*

Proof. For the proof, see [116].

3.3.1 Linear embedding of the nonlinear kinematics

In this section, a linear description of the vehicle kinematics (2.5)-(2.16) is derived. To this end, a quasi Linear Parameter Varying (LPV) description of the nonlinear kinematics (2.16) is first obtained by defining the following quantities

$$\alpha_1(t) := cos(\theta(t))\frac{\mu_l(t)+\mu_r(t)}{2} \tag{3.90}$$

$$\alpha_2(t) := cos(\theta(t))\frac{\mu_r(t)-\mu_l(t)}{4}T_d \tag{3.91}$$

$$\alpha_3(t) := sin(\theta(t))\frac{\mu_l(t)+\mu_r(t)}{2} \tag{3.92}$$

$$\alpha_4(t) := sin(\theta(t))\frac{\mu_r(t)-\mu_l(t)}{4}T_d \tag{3.93}$$

$$\alpha_5(t) := \frac{\mu_r(t)-\mu_l(t)}{T_d} \tag{3.94}$$

$$\alpha_6(t) := \frac{\mu_l(t)+\mu_r(t)}{2} \tag{3.95}$$

and rewriting the model (2.16) as

$$\dot{q}_E(t) = B(\alpha(t))u(t) \tag{3.96}$$

with $\alpha = [\alpha_1, \alpha_2, \alpha_3, \alpha_4, \alpha_5, \alpha_6]^T$ and

$$B(\alpha(t)) = \begin{bmatrix} \alpha_1(t) & \alpha_2(t) \\ \alpha_3(t) & \alpha_4(t) \\ \alpha_5(t) & \alpha_6(t) \end{bmatrix}. \tag{3.97}$$

Hence, the discrete-time counterpart obtained by using the Euler forward discretization with the sampling time T_s

$$q_E(t_{k+1}) = q_E(t_k) + G(\alpha(t_k))u(t_k) \tag{3.98}$$

with $G(\alpha(t_k)) = T_s\, B(\alpha(t_k))$, is embedded into a polytopic description according to the lines indicated in [117]:

$$G(\pi) \in \Sigma(\Pi) := \{G(\pi) = \sum_{i=1}^{L} \pi_i G_i, \forall \pi \in \Pi\} \tag{3.99}$$

where $G_i, i = 1, \ldots, L$, are the vertices of the polytopic representation $\Sigma(\Pi)$, i.e., $vert(\Sigma(\Pi))$, and the uncertain vector π belongs to the unitary simplex

$$\Pi := \{\pi \in \mathbb{R}^L : \sum_{i=1}^{L} \pi_i = 1,\ 0 \leq \pi_i \leq 1\} \tag{3.100}$$

In conclusion, all the state trajectories resulting from the kinematics (2.16) are included in the tube pertaining to the following multi-model description

$$\{I, G(\pi(t_k))\} \in \mathbf{Co}\{(I, G_1), (I, G_2) \cdots, (I, G_L)\},\ \forall t_k \tag{3.101}$$

where Co denotes the convex hull operator [109].

3.3.2 Management of delays introduced by the communication network

As far as the control architecture is concerned, the networked configuration of Fig. 2.2 is exploited. The reasons behind such a choice mainly relies on the limited computational resources available on-board the SSTMR. Hence in order to keep the robot autonomy as long as possible, it is mandatory to save battery charges by moving on the remote side most of computations pertaining to the control action. Though effective, this approach necessarily gives rise to latency effects during the data transmission, and as it is well-known, if not adequately addressed time-delays occurrences could compromise the overall control performance and, in the worst case, lead to instability. The delays can be abstractly collected into a single bounded positive scalar τ, see eq. (3.22), such that

$$\tau(t) \leq \bar{\tau}, \tag{3.102}$$

$\forall t \geq 0, \bar{\tau} < \infty$, accounting for the round-trip delay [118]. A key issue arising with the architecture of Fig. 2.2 relies on a proper time-induced delay management in order to keep the feasibility property of the control scheme. The path-planning is updated on the remote side and is exclusively based on the availability of past data $q_E(t_{k-\tau_{t_k}})$.

Therefore the underlying control action, if directly computed, could lead to collisions because the real/current obstacle scenario $\mathcal{O}^{t_k}$ could have been modified. In view of this reasoning, a *worst case* approach must be adopted to take care of all possible operating conditions both in terms of time latency and obstacle occurrences. Specifically, the idea is of resorting to the set-theoretic receding horizon control arguments developed in [116] here adequately customized to comply with above framework. To formally describe the network latency, the following definitions are used.

The delay on the command channel side is defined as $\tau^{CA}_{t_k} \in \mathbb{Z}_+$, while the delay on the measurement side $\tau^{SC}_{t_k} \in \mathbb{Z}_+$.

Moreover, $\tau^{m}_{t_k} \leq \tau^{CA}_{t_k}$ and $\tau^{c}_{t_k} \leq \tau^{SC}_{t_k}$ refer to the *ages* of the state measurement and of the command input, respectively. Then, on the plant-controller link at each time instant t_k when computing the input $u(t_k)$, the age network latency is $\tau^{NL}_{t_k} = \tau^{m}_{t_k} + \tau^{c}_{t_{k-\tau^{m}_{t_k}}}$. Since the round-trip delay $\tau^{c}_{t_{k-\tau^{m}_{t_k}}}$. is not available at the controller side, the upper bound $\bar{\tau}_c$ on the controller-actuator link during both the controller design and the command computation $u(\cdot)$ phases is considered, i.e.,

$$\tau_{t_k} = \tau^{m}_{t_k} + \bar{\tau}_c \leq \bar{\tau}, \ \forall t_k, \tag{3.103}$$

where $\bar{\tau}$ is the maximum allowable transmission interval (MATI) [105].

3.3.3 The MPC controller

In view of this analysis, the key property to ensure is that for any point q_E of the domain of attraction (DoA) of the designed controller, understood as an initial condition for the vehicle kinematics, the $\bar{\tau} - th$ step ahead state trajectory under the action of a single causal control remains confined in the disk $D(q_E, R)$ centered at q_E and with radius R, being the vision radius of the **Lidar**, i.e.,

$$q_E + \sum_{i=0}^{s-1} G^{(i)}(\pi)\bar{u} \subset D(q_E, R), \pi \in \mathit{\Pi}, s = 1, \ldots, \bar{\tau} - 1, \tag{3.104}$$

with $G^{(i)}(\pi)$ characterizing the $i-th$ state prediction. This condition must be used to ensure that, starting from $q_E(t_{k-\tau_{t_k}}) \in DoA$ and applying the command $\bar{u}$ for $\tau_{t_k} \leq \bar{\tau}$ instants consecutively, the $\tau_{t_k} - step$ state ahead prediction remains confined within the obstacle-free region $D(q_E(t_{k-\tau_{t_k}}), R)$.
A second important issue to properly take into account concerns with the obstacle occurrence during the on-line operations. This occurrence could make the current controller no longer viable. By referring to the abstract architecture of Fig. 3.12, this critical scenario occurs when the following condition holds true:

$$D(q_E(t_{k-\tau_{t_k}}), R) \nsubseteq \mathcal{O}^{t_k} \tag{3.105}$$

whose the meaning is as follows: the **Lidar** recognizes an obstacle along the DoA and the condition (3.104) cannot be guaranteed from t_k onward.

As a matter of fact, these developments prescribe that the control unit must enjoy a switching structure in charge to ensure the viability property whenever an unpredictable occlusion along the nominal path occurs. The idea is of resorting to set-theoretic arguments outlined in [116]. and here customized to the **OAMP-UCE** prescriptions.

3.3.4 Robust positively invariant (RPI) sets

First, notice that in order to comply with (3.104), the convex hulls of all $s - th$ state predictions starting from q_E under the virtual command $\bar{u}$ have to be formally defined:

$$\mathcal{X}^s(q_E) := \mathbf{Co}\left\{z + G^{(s)}(\pi)\bar{u}, \forall \pi \in \mathit{\Pi}, \forall z \in \text{vert}\left(\mathcal{X}^{s-1}(q_E)\right)\right\} \tag{3.106}$$

According to the above analysis, the following result provides a formal way to address the off-line computation of RPI regions.

Proposition 3.6. *Let $q_E(t_{k-\tau_{t_k}})$, with $\tau_{t_k} \leq \bar{\tau}$ be a state measurement, the state feedback control law*

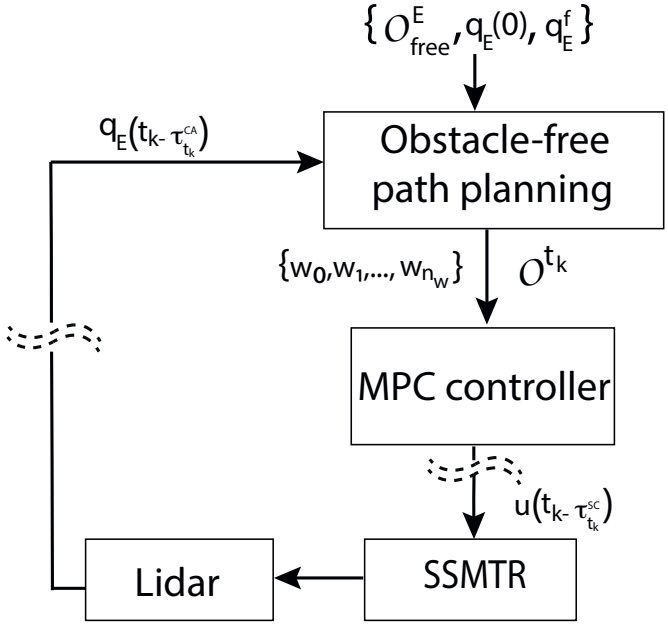


Fig. 3.12: Control architecture

$$u(t_k) = K q_E(t_{k-\tau_{t_k}}) \tag{3.107}$$

is such that the regulated plant

$$q_E(t_{k+1}) = q_E(t_k) + G(\pi(t_k)) K q_E(t_{k-\tau_{t_k}}) \tag{3.108}$$

is asymptotically stable, if the matrix inequalities

$$\begin{bmatrix} E^T P E - \mathcal{S} & \mathcal{A}_j^T P \\ P \mathcal{A}_j & P \end{bmatrix} \geq 0, \ j = 1, \ldots, L, \tag{3.109}$$

$$\begin{bmatrix} u_{max}^2 E^T P E & \begin{bmatrix} K^T \\ 0 \end{bmatrix} \\ \begin{bmatrix} K & 0 \end{bmatrix} & I \end{bmatrix} \geq 0, u_{max} = \min \left\{ \frac{1}{q_V}, \frac{1}{q_\omega} \right\}, \tag{3.110}$$

$$\begin{bmatrix} E^T(-P + \lambda_s I)E & -P\mathcal{B}_j + \lambda_s H^{(s)} & 0 \\ * & -\mathcal{B}_j^T \mathcal{B}_j + \lambda_s {H^{(s)}}^T H^{(s)} & 0 \\ * & * & 1 - \lambda_s R^2 \end{bmatrix} \geq 0, \tag{3.111}$$

$$j = 1, \ldots, L, \ s = 1 \ldots, \bar{\tau} - 1,$$

$$\lambda_s \geq 0, \ s = 1, \ldots, \bar{\tau} - 1, \tag{3.112}$$

with $\mathcal{B}_j = \begin{bmatrix} G_j K \\ 0 \end{bmatrix}$, $H^{(s)} = \sum_{i=0}^{s} G^{(i)}(\pi)K, \forall \pi \in \Pi$, $E = diag\{I, 0\}$, $\mathcal{S} = diag\{S, 0\}$, $S = S^T \geq 0$, $\mathcal{A}_j = \begin{bmatrix} I & I \\ G_j K & -I \end{bmatrix}$, $j = 1, \ldots, L$, *admit a solution. Therefore, the ellipsoidal set:*

$$\mathcal{E} := \{q_E \in \mathbb{R}^n \,|q_E^T P q_E \leq 1\}, P = P^T > 0, \tag{3.113}$$

is a RPI [115] set for the closed-loop state evolutions (3.108) complying with input and geometric constraints.

Proof. First notice that the matrix inequalities (3.109)-(3.111) are derived by resorting to the Independent-of-Delay (IOD) stability concept under the technicalities exploited in [116]. Conversely, the admissibility of the control law (3.107) within the architecture of Fig. 3.12 results from the satisfaction of (3.111) with $Kq_E(t_{k-\tau_{t_k}})$ in place of $\bar{u}$. The $\bar{\tau}$ set inclusions guarantee the set-membership of all the state predictions from $t_{k-\tau_{t_k}}$ to t_k do not overcome the disk $D(q_E(t_{k-\tau_{t_k}}), R)$. This is comes out due to the fact that for all initial conditions belonging to the segment $q_E(t_{k-\tau_{t_k}})$, $\tau_{t_k} \leq \bar{\tau}$, the controller (3.107)) can be consecutively applied for τ_{t_k} time instants without comprising closed-loop stability and constraints fulfillment. According to this reasoning and by noticing that the $s-th$ step state ahead prediction polyhedral (3.106) are known because the state conditions $q_E(t_k) \neq q_E(t_{k-\tau_{t_k}})$ are not available in the off-line phase, inequalities (3.111)-(3.112) comes out by resorting to S-procedure technicalities [119] applied to the following quadratic form implications:

$$\begin{cases} D\left(q_E(t_k) + \sum_{i=0}^{s-1} G^{(i)}(\pi)Kq_E(t_{k-\tau_{t_k}}), R\right), \\ \forall \pi \in \Pi, s = 1, \ldots, \bar{\tau} - 1, \\ \Downarrow \\ q_E(t_k) + G(\pi)Kq_E(t_{k-\tau_{t_k}}), \forall \pi \in \Pi, \end{cases}$$

where one-step-evolution (3.108) under (3.107) belongs to $\mathcal{E}$.

3.3.5 Robust one-step state ahead controllable (ROSAC) sets

Here, the following definition will be used.

Definition 3.7. *Given the plant* (3.34)-(3.36) *and a RPI target set* $\mathcal{E}$, *the set of states* i*-step controllable to* $\mathcal{T}$ *is defined via the following recursion:*

$$\begin{aligned} &\mathcal{T}_0 := \mathcal{E} \\ &\mathcal{T}_i := \{q_E : \exists u \in \mathcal{U} : q_E + G(\pi)u \in \mathcal{T}_{i-1}, \forall \pi \in \Pi\} \end{aligned} \tag{3.114}$$

where $\mathcal{T}_i$ is the set of states that can be steered into $\mathcal{T}_{i-1}$ using a single control move.

To address the discrepancy introduced by the networked configuration

$$q_E(t_{k-\tau_k}) \neq q_E(t_k) \tag{3.115}$$

the following one-step transition map applies $\forall \tau_k \in [0, \overline{\tau}]$

$$q_E(t_{k+1}) = q_E(t_{k-\tau_{t_k}}) + G(\pi(t_k))u(t_k). \tag{3.116}$$

Then, the time-delay dependency can be explicitly taken into account by considering the auxiliary state

$$\Delta q_E(t_k) = q_E(t_{k+1}) - q_E(t_k). \tag{3.117}$$

and by means of the descriptor form

$$\begin{bmatrix} q_E(t_{k+1}) \\ 0 \end{bmatrix} = \tag{3.118}$$
$$\begin{bmatrix} \Delta q_E(t_k) + q_E(t_k) \\ -\Delta q_E(t_k) + q_E(t_{k-\tau_{t_k}}) + G(\pi(t_k))u(t_k) - q_E(t_k) \end{bmatrix}$$

By resorting to the worst-case approach $\tau_{t_k} = \overline{\tau}$, one has

$$q_E(t_{k-\tau_{t_k}}) = q_E(t_k) - \sum_{j=t_k-\overline{\tau}}^{t_{k-1}} \Delta q_E(t_{k-j})$$

and the following model description comes out:

$$Ex(t_{k+1}) = \Phi x(t_k) + G_u(\pi(t_k))u(t_k) - G_{\Delta q} \sum_{j=t_k-\overline{\tau}}^{t_{k-1}} \Delta q_E(t_{k-j}) \tag{3.119}$$

with $x(t_k) = \left[q_E^T(t_k)\ \Delta q_E^T(t_k)\right]^T$ and $\Phi = \begin{bmatrix} I & I \\ 0 & -I \end{bmatrix}, G_{\Delta q} = \begin{bmatrix} 0 & I \end{bmatrix}^T$, $G_u(\pi(t_k)) = \begin{bmatrix} 0 & G(\pi(t_k)) \end{bmatrix}^T$. Then,a ROSAC sets sequence can be computed according to the following proposition.

Proposition 3.8. *Let $\mathcal{E} \neq \emptyset$ be a given ellipsoidal RPI region complying with Proposition 3.6 prescriptions and $x_{aug} = \left[q_E^T\ \Delta q_E^T\ z_1^T\ z_2^T\right]^T$ the augmented state describing the dynamics (3.119) with z_1, $z_2 \in \mathbb{R}^n$ accounting for all the cumulative sum vectors $\Delta q_E(j)$, $j = t_{k-\bar{\tau}}, \ldots, t_{k-1}$. Then, the ellipsoidal set sequence*

$$\begin{aligned} \mathcal{E}_0 &= \mathcal{E} \\ \mathcal{E}_i &= Proj_{q_E}\{In[x_{aug} \in \mathbb{R}^{4n} \text{ with } z_1, z_2 \in \mathcal{E}_{i-1} : \exists u \in \mathcal{U} : \\ &\quad Proj_{q_E}\{\Phi_{aug} x_{aug} + G_{aug}(\pi)u\} \in \mathcal{E}_{i-1}], \\ &\quad Proj_{q_E}\{x_{aug} + \sum_{i=0}^{j-1} G_{aug}^{(i)}(\pi)u\} \subset D(q_E, R), \\ &\quad j = 1, \ldots, \bar{\tau} - 1], \forall \pi \in \Pi\} \end{aligned} \tag{3.120}$$

$$\Phi_{aug} = \begin{bmatrix} I & I & 0 & 0 \\ 0 & -I & -\bar{\tau}\Phi & \bar{\tau}I \\ 0 & 0 & 0 & I \\ 0 & 0 & 0 & I \end{bmatrix},$$
$$G_{aug}(\pi) = \begin{bmatrix} 0 \; G^T(\pi) \; 0 \; 0 \end{bmatrix}^T$$
$$G^{(i)}_{aug}(\pi) = \begin{bmatrix} 0 \; G^{(i)^T}(\pi) \; 0 \; 0 \end{bmatrix}^T, \; i = 1, \ldots, \bar{\tau}$$

if non-empty, satisfies $\mathcal{E}_i \subset \mathcal{T}_i$.

Proof. By following the lines indicated in [116] and the arguments developed in Proposition 3.6 for the state predictions constraints (3.104).

3.3.6 Overlapped RPIs

A critical issue arising from using the low-demanding control strategy relies on the lack of flexibility when the time-varying obstacle scenario makes no longer viable the state trajectory tube $\bigcup_i \mathcal{E}_i$. In such a case, it's possible to determine sequences of overlapped RPIs during the on-line operations. The feasibility BMI problem (3.109)-(3.111) could be computational impracticable because it is a NP-hard problem [120]. The following step allows to mitigate such a complexity and to provide a procedure to bypass this obstacle.

Proposition 3.9. *Let* $\Xi^{t_{k-1}}$ *be a RPI set computed at the time instant* t_{k-1}*, centered at the equilibrium* $q_{E,eq}^{t_{k-1}}$ *and compliant with the constraints (3.3) and (3.104). Given the state measurement* $q_E(t_{k-\tau_k}) \in \Xi^{t_{k-1}}$ *and the vision radius* $R^{t_{k-\tau_k}}$*, such that* $D(q_E(t_{k-\tau_k}), R^{t_{k-\tau_k}})$ *is obstacle-free, then the set*

$$\Xi^{t_k} := \left\{ q_E \in \mathbb{R}^n \mid (q_E - q_{E,eq}^{t_k})^T P^{t_k} (q_E - q_{E,eq}^{t_k}) \leq 1 \right\}$$

centered at the equilibrium

$$q_{E,eq}^{t_k} := \arg \max_{q_E \in \Xi^{t_{k-1}}} \| q_{E,eq}^{t_{k-1}} - q_E \|_2 \tag{3.121}$$

and with

$$P^{t_k} := \rho^{t_k} P^{t_{k-1}}, \tag{3.122}$$

is obtained by solving the following quasi-LMI optimization

$$\rho^{t_k} := \arg \min_{\rho, K^{t_k}, \mathcal{S}} \rho \tag{3.123}$$

subject to

$$\rho > 0 \tag{3.124}$$

$$\begin{bmatrix} E^T\left(\rho P^{t_{k-1}}\right) E - \mathcal{S}\,\mathcal{A}_j^T\left(\rho P^{t_{k-1}}\right) \\ \left(\rho P^{t_{k-1}}\right)\mathcal{A}_j \qquad\qquad \rho P^{t_{k-1}} \end{bmatrix} \geq 0,\ j = 1, \ldots, L, \tag{3.125}$$

$$\begin{bmatrix} u_{max}^2\, E^T\left(\rho P^{t_{k-1}}\right) E \begin{bmatrix} {K^{t_k}}^T \\ 0 \end{bmatrix} \\ \begin{bmatrix} K^{t_k} & 0 \end{bmatrix} \qquad\qquad I \end{bmatrix} \geq 0, \tag{3.126}$$

$$\begin{bmatrix} (R^{t_{k-\tau_k}})^2 & * \\ E\begin{bmatrix} z_{s-1} \\ 0 \end{bmatrix} + G_j[K^{t_k}\,0]\begin{bmatrix} q_E(t_{k-\tau_k}) \\ 0 \end{bmatrix} & I \end{bmatrix} \geq 0, \tag{3.127}$$

$\forall z_{s-1} \in vert(\mathcal{X}^{s-1}(q_E(t_{k-\tau_k}))),\ s = 1, \ldots, \bar{\tau} - 1,$

$j = 1, \ldots, L.$

$$\textit{Moreover, it results that } \Xi^{t_{k-1}} \cap \Xi^{t_k} \neq \emptyset. \tag{3.128}$$

Proof. As the optimization (3.123)-(3.127) resolvability is concerned, it follows from the idea to soft the complexity of the BMI problem (3.109)-(3.111) by using shape and size of the previously computed set $\Xi^{t_{k-1}}$. Then by shifting $\Xi^{t_{k-1}}$ with respect to the new equilibrium (3.121), an admissible, though not optimal, solution there always exits for a sufficiently large value of the scalar ρ because in the worst case Ξ^{t_k} reduces to a neighborhood of $q_{E,eq}^{t_k}$ according to standard linearity arguments [118]. Moreover, the collision avoidance is guaranteed in virtue of using $R^{t_{k-\tau_k}}$ that represents the last admissible received radius obstacle-free, i.e., $t_{k-\tau_k}$. In fact, the new set Ξ^{t_k} will no overcome the disk $D(q_E(t_{k-\tau_k}), R^{t_{k-\tau_k}})$ thanks to (3.127), that, differently from the Proposition 3.6 (inequalities (3.111)-(3.112)), is simply translated into $\bar{\tau} - 1$ set-membership checks. Finally, the overlapping property (3.128) is directly valid by construction in virtue of the selection rule (3.121).

3.3.7 Controller design

Based on the above developments, the controller design is split in off-line and on-line phases. In the off-line phase, state trajectory tubes pertaining to all the realizations of the induced time-delay $0 \leq \tau \leq \bar{\tau}$ are computed by using Propositions 3.6 and 3.8:

$$\{\mathcal{E}_i^\tau\}_{i=0}^N,\ \ \tau = 0, 1, \ldots, \bar{\tau}. \tag{3.129}$$

For the sake of simplicity, it is assumed that each sequence of $\mathcal{E}_i^\tau$ consists of the same number of ROSAC sets. This choice is exclusively dictated by control performance reasons since the availability of the network latency may allow to significantly improve the functioning of the regulated SSTMR. Conversely, to properly address the on-line phase a time-stamp protocol, equipped with a clock synchronization mechanism,

is used to properly identify the age of each data-packet sent from the sensor and controller nodes. Specifically the event (q_E, t_{q_E}) denotes the pose measurement q_E sent at the time instant t_{q_E}, whereas (u, t_u) refers to the command u i command sent by the controller and marked by the stamp t_u. Then, at each time instant t_k the computation of the command input $u(t_k)$ on the remote side is performed by selecting the most recent data packet (q_E, t_{q_E}) and as a consequence selecting the set-membership compatible with the time induced delay τ_{t_k}, i.e., $q_E \in \{\mathcal{E}_i^{\tau_{t_k}}\}$. Then, an admissible input $u(t_k)$ is obtained as follows:

$$u(t_k) = \arg\min J_{i(t_k)}(q_E, u) \text{ s.t.} \tag{3.130}$$

$$q_E + G_j u \in \mathcal{E}_{i_{t_k}-1}^{\tau_{t_k}}, u \in \mathcal{U}, j = 1, \ldots, L, \tag{3.131}$$

with

$$J_{i(t_k)}(q_E, u) = \max_j \|q_E + G_j u\|^2_{P_{i_{t_k}-1}^{\tau_{t_k}}}. \tag{3.132}$$

On the other hand, whenever the condition (3.105) becomes true and a sequence of the overlapped RPI sets Ξ must be built during the on-line operations, the control action is computed by means of the following tracking one-step MPC problem [121]:

$$\tilde{u}(t_k) := \arg\min \max_{\pi \in \Pi} \|\tilde{q}_E(t_{k-\tau_{t_k}}) + G(\pi)\tilde{u}\|^2_{P^{t_{k+1}}} \tag{3.133}$$

s.t.

$$\tilde{q}_E(t_{k-\tau_{t_k}}) + G\tilde{u} \in \mathcal{E}^{t_{k+1}}, \; u \in \mathcal{U}, \tag{3.134}$$

with $\tilde{q}_E := q_E - q_{E,eq}^{t_{k+1}}$, $\tilde{u} := u - u_{eq}^{t_{k+1}}$ and $P^{t_{k+1}}$ the shaping positive definite matrix of $\mathcal{E}^{t_{k+1}}$.

3.3.8 Path planning in $\mathcal{O}^{free}$

In this section, the aim is to derive a sequence of obstacle-free way-points to be used as references by the MPC controller for computing the family of ROSAC sets (3.129) and, whenever necessary, during the on-line operations. Specifically, the unit must guarantee that the SSTMR can be driven from an initial position $q_E(0)$ to the target q_E^f by taking care of possible forbidden zones due to the presence of obstacles not *a-priori* detectable, see the illustrative example of Fig. 3.13. In the off-line phase, the **Path Planner** determines a sequence of way-points $\Gamma := \{w_0, w_1, \ldots, w_{n_\Gamma}\}$, with $w_0 = q_E(0)$ and $w_{n_\Gamma} = q_E^f$, by using the *RRT** algorithm discussed in [122] and exploiting the obstacle-free region $\mathcal{O}^{t_0}$. This sequence is then used to compute the state trajectory tube (3.129) by adapting the **OCSP** procedure presented in [121], see Fig. 3.13. Whenever the condition (3.105) is verified, the path Γ is no longer

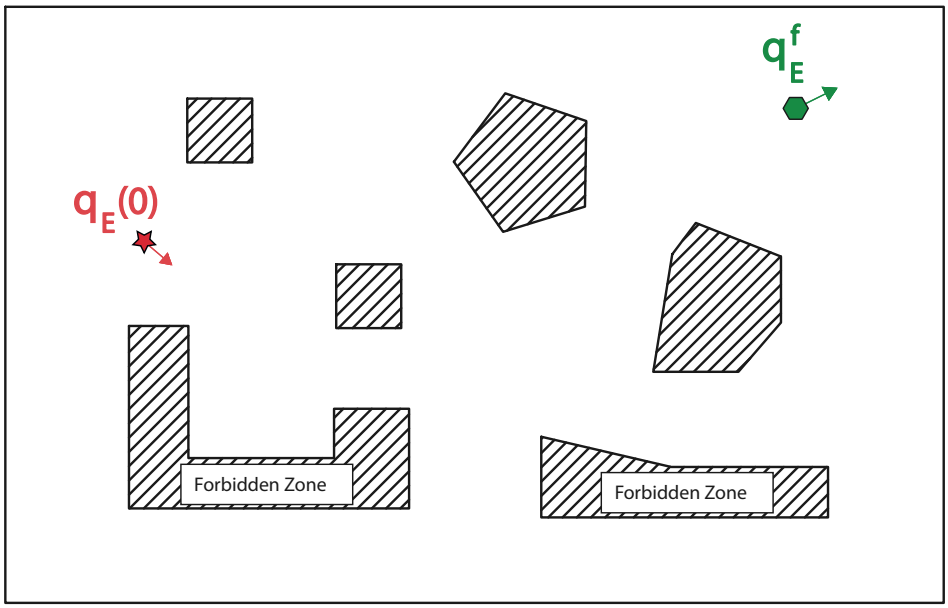


Fig. 3.13: Obstacle scenario $\mathcal{O}^{t_k}$ and operating conditions: the red star and green hexagon denote the initial and target poses, respectively.

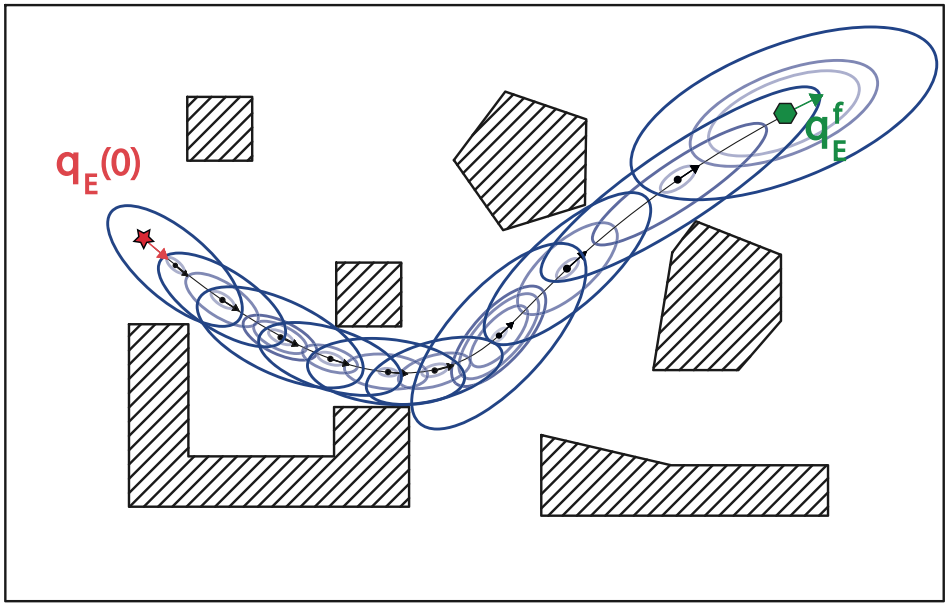


Fig. 3.14: State trajectory tube (3.129).

viable and an overlapped sequence of RPIs has to be on-line determined. This requires the computation of new admissible way-points achievable by means of the following optimization:

$$\begin{cases} w^{t_k} = arg \min\limits_{q_E \in \Xi^{t_{k-1}}} dist(q_E, \Gamma) \\ \qquad \text{subject to (3.121)} \end{cases} \tag{3.135}$$

which represents the pose at the minimum distance from the nominal path Γ. The latter is instrumental to reconnect the SSTMR to the nominal path (3.129) as soon as its viability is recovered, i.e., the set-containment (3.105) is no longer valid.

Remark 6 - Notice that by construction the problem (3.135) has always a solution since, in the worst case, it is retrieved the one determined at the previous time instant, i.e., $w^{t_{k-1}}$. □

3.3.9 A set-theoretic model predictive control scheme

All the above developments allow to write down a computable algorithm in charge to comply with the requirements of the **OAMP-UCE** problem.
Recursive feasibility and Uniformly Ultimate Boundedness of the **Navigation-MPC** Algorithm are formally stated in the following result.

Lemma 5 *Let* $\{\mathcal{E}_i^\tau\}_{i=0}^N$, $\tau = 0, 1, \ldots, \bar{\tau}$, *be non-empty families of ROSAC sets. Then, the* ***Navigation-MPC*** *Algorithm ensures constraints satisfaction and Uniformly Ultimate Boundedness for all* $\pi \in \Pi$, *for all time-delay occurrences* $\tau_{t_k} \leq \bar{\tau}$ *and for all admissible obstacle scenarios* $\mathcal{O}$.

Proof. Consider without loss of generality that at the generic instant t_k the state $q_E \in \mathcal{T}_{i(t)}^{\tau_{t_k}}$. Then by construction there exists an input vector $u \in \mathcal{U}$ such that the one-step state evolution $q_E + G(\pi)u$ belongs to $\mathcal{E}_{i_{t_k}-1}^{\tau_{t_k}}$, $\forall \pi \in \Pi$ and $\forall \tau_{t_k} \leq \bar{\tau}$. In virtue of the recursions (3.114), at the next time instant t_{k+1}, a solution $u_{t_{k+1}}$ for **Step 8** is ensured. Conversely, when the state trajectory tube is no longer admissible (condition (3.105) in **Step 10**), the feasibility is retained because Proposition 3.8 ensures the existence of an admissible, though not optimal, solution at t_{k+1}. As the collision avoidance feature is concerned, it is ensured for all time induced delays $\tau_{t_k} \leq \bar{\tau}$ thanks to the set-containment (3.104). Finally, Uniformly Ultimate Boundedness follows by using similar arguments. Under the nominal regime each initial state $q_E(0) \in \bigcup_{\tau=0}^{\bar{\tau}} \bigcup_{i=0}^N \mathcal{E}_i^\tau$ is by construction driven to the target set centered in q_E^f. On the other hand, whenever the occurrence of new obstacles $\mathcal{O}^{t_k}$ prevents the use of the state trajectory tube (3.129), overlapped RPIs can always be computed according to the prescriptions of Proposition 3.9. Therefore, in the worst case the state trajectory will be confined in $\bigcup_{\tau=0}^{\bar{\tau}} \bigcup_{i=0}^N \mathcal{E}_i^\tau$.

Navigation-MPC Algorithm

Off-line

Input: $q_E(0)$; q_E^f; $\bar{\tau}$; $\mathcal{U}$; $\mathcal{O}^{t_0}$; **Output:** $\{\mathcal{E}_i^\tau\}_{i=0}^N$, $\tau = 0, 1, \ldots, \bar{\tau}$;

1: **Compute** the nominal way-points sequence $\{w_o, w_1, \ldots, w_{n_\Gamma}\}$ by the **Path Planner**;
2: **for** $\tau = 0, 1, \ldots, \bar{\tau}$ **do**
3: **Compute** the pair $(\mathcal{E}_0^\tau, K^\tau)$ via **Proposition 1**;
4: **Compute** the sequence of ROSACs $\{\mathcal{E}_i^\tau\}_{i=1}^N$ by means of **Proposition 2**;
5: **end for**
6: **Store** $\{\mathcal{E}_i^\tau\}_{i=0}^N$, $\tau = 0, 1, \ldots, \bar{\tau}$, in the **Controller Buffer**;

On-line

Controller side -

Input: $q_E(t_{k-\tau_{t_k}})$; $\{\mathcal{E}_i^\tau\}_{i=0}^N$, $\tau = 0, 1, \ldots, \bar{\tau}$; **Output:** u_{t_k}; for all $t_k \in \mathbb{Z}_+$

1: **Acquire** the packet (q_E, t_{q_E});
2: **Determine** $\mathcal{O}^{t_k}$;
3: **if** (3.105)==false **then**
4: **Compute** $\tau_{t_k} = \tau_{t_k}^m + \bar{\tau}_c$, with $\tau_{t_k}^m = t_k - t_{q_E}$;
5: **Determine** $i(t_k) := \min\{i : q_E \in \mathcal{E}_i^{\tau_{t_k}}\}$
6: **if** $i(t_k) = 0$ **then** $u_{t_k} = K^{\tau_{t_k}} q_E$
7: **else**
8: **Solve** (3.130)-(3.131);
9: **end if**
10: **else**
11: **Compute** the new way-point w^{t_k} by (3.135);
12: **Compute** the overlapped pair $(\mathcal{E}_0^{t_k}, K^{t_k})$ via **Proposition 3**;
13: **Solve** (3.133)-(3.134);
14: **end if**
15: **Transmit** the packet (u_{t_k}, t_u);

Plant side -

Input: $u(t_{k-\tau_{t_k}})$; **Output:** $q_E(t_k)$; for all $t \in \mathbb{Z}_+$

1: **Acquire** the packet (u, t_u);
2: **Apply** u to the SSTMR;
3: **Measure** $q_E(t_k)$ by using the **Lidar**;
4: **Transmit** the packet $(q_E(t_k), t_{q_E})$;

Remark 7 - Notice that the **Navigation-MPC** Algorithm has been developed with a two-fold aim: reduce as much as possible the on-line computations by moving off-line or in the remote side most of them; ensure (guaranteed) constraints satisfaction at each time instant, formally management of latency phenomena within the networked control configuration and guarantee to safely converge to the target point. All these requirements are achieved in a formal way via the so-called set-theoretic approach [115] that is recognized to well adapt to uncertain operating conditions. A loss of performance naturally comes out because in real applications several uncertainty sources arise, nonetheless the advantage coming from the satisfaction of the above requirements highly compensates a sustainable loss of performance. Finally, fixed but unknown obstacle scenarios (as a consequence time-varying on the vehicle side) have been considered in order to be complaint with real-like operating conditions. For the sake of simplicity, the presence of moving obstacles (agents) has been omitted even if their presence can be addressed by customizing the ideas developed in [121] at the present context.

3.3.10 Results — Experimental Validation

In order to demonstrate the effectiveness of the proposed algorithm, some experimental tests have been carried out at the Control Systems Laboratory of the Mediterranean University of Reggio Calabria by using the differentially driven Jaguar V4 by Dr. Robot. It is a tracked robotic platform with a footprint of $70 \times 50\,\mathrm{cm}^2$. The gearbox connecting the motors to the tracks has a radius of 8 cm, the distance between the left and right track is 50 cm. Moreover, forward and rotational speeds satisfy the following bounds:

$$V \in [0; 0.4]\mathrm{m/s} \tag{3.136}$$

$$\omega \in [-60; 60]\mathrm{deg/s} \tag{3.137}$$

In addition, the robotic platform is equipped with a front-mounted UTM-30LX LIDAR sensor capable to detect objects within pre-assigned ranges $\rho_{min} \leq \rho \leq \rho_{max}$ and with a field of view of 360 degrees, until $R = 10\,\mathrm{m}$, and the hardware in charge to regulate the rotational speed of left and right tracks. It is assumed that the sliding coefficients μ_r and μ_l take values within the range $[0.8; 1.2]$. Note that values higher than the nominal value 1 have been considered to take into account possible tracks slippages during the deceleration phases. Then, the state trajectories of the robot nonlinear kinematics (2.5) have been embedded within those of the multi-model description (3.98)-(3.101) with $L = 12$.

Finally, the vehicle moves in the indoor space reported in Fig. 3.15 whose size $24.6 \times 4\,\mathrm{m}^2$ and is geometrically described in terms of its shell, i.e., the hyperball centered at the centroid of the shell ξ and with radius $d_{shell} = 43\,\mathrm{cm}$.

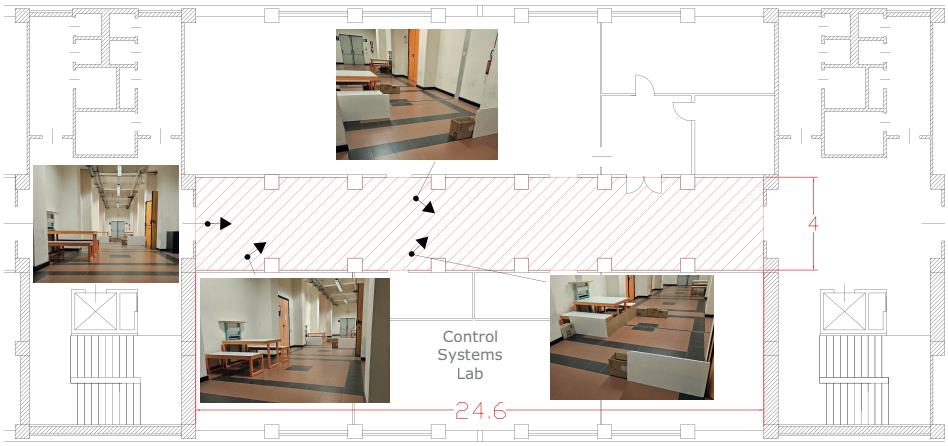


Fig. 3.15: Indoor operating environment

Hereafter, the control architecture of Fig. 3.16 is considered. Notice that pose estimation, environment map generation and command input computation tasks are completely performed on the remote station. Notice that data exchanged between vehicle and ground station are transmitted by a dedicated *802.11n* wireless data communication network that makes use of the TCP/IP protocol stack and ensures no data packet losses. In particular, the data collected by the on-board laser scanner sensor are synchronously transmitted to the ground station every sampling time $T_s = 0.4\,\mathrm{s}$. As soon as new measurements are available, a SLAM algorithm [123] is in charge to estimate the robot pose, while the control action is computed in terms while the control action is computed in terms of forward and rotational speeds. A measurement campaign has provided an estimation of the MATI pertaining to the WIFI communication medium, namely $\overline{\tau} = 2$ sampling steps.

For the proposed algorithm, the following off-line computations have been performed. First, the **Path Planner** has determined a sequence Γ of twenty way-points connecting $q_E(0) = [0\ \ 0\ \ 0]^T$ to the target $q_E^f = [15.8\ \ 1.5\ \ 0]^T$, are collected in the following Table 3.1.

As far as the controller structure is concerned, for all possible occurrences of the induced delay $\tau = 0, 1, 2$, a family of nineteen ROSAC sequences, each one consisting of $N = 40$ ellipsoids and tuned with respect the way-point w_i along the path

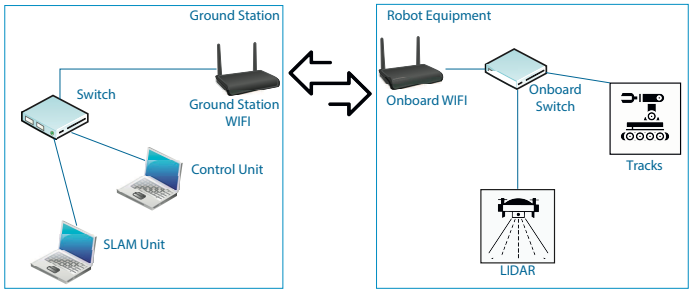


Fig. 3.16: Networked control architecture: SLAM, Controller and WIFI units.

Table 3.1: The path Γ

Way-point	$x\,[m]$	$y\,[m]$	$\theta\,[deg]$
w_0	0	0	0
w_1	1.2318	0.3316	0.2630
w_2	3.4277	0.9229	0.2630
w_3	5.4844	1.4766	0.2630
w_4	6.5000	1.7500	0.2630
w_5	6.5000	1.7500	0.2619
w_6	6.5000	1.7500	0.1233
w_7	6.5000	1.7500	−0.0208
w_8	6.5000	1.7500	−0.1912
w_9	6.5000	1.7500	−0.4636
w_{10}	7.0000	1.5000	−0.4636
w_{11}	8.0000	1.0000	−0.4636
w_{12}	8.0000	1.0000	−0.3838
w_{13}	8.0000	1.0000	−0.2414
w_{14}	8.0000	1.0000	−0.1209
w_{15}	8.0769	1.0055	0.0713
w_{16}	9.9688	1.1406	0.0713
w_{17}	15.000	1.5000	0.0713
w_{18}	15.000	1.500	0
w_{19}	15.800	1.500	0

$w_{19} \to w_0$, with $w_{19} = q_E^f$ the starting condition, have been computed by using the technicalities of Sections 3.3.4 and 3.3.5.

All the experiments are collected in Figs. 3.17-3.24. First, it is important to emphasize that, during the on-line operations, no *a-priori* information on the obstacle locations

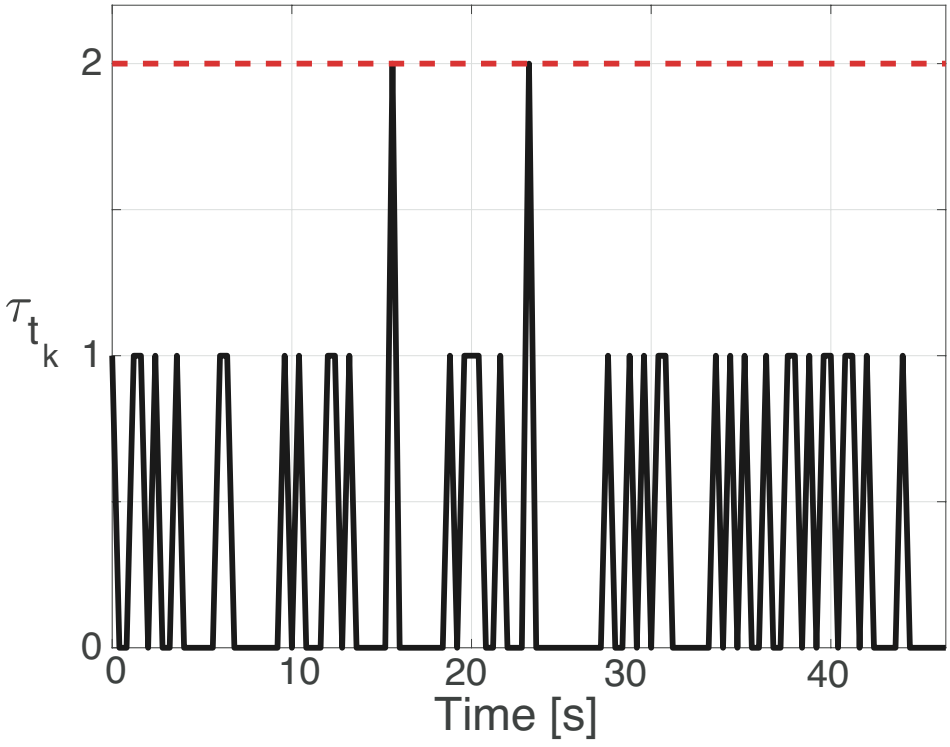


Fig. 3.17: Cumulative time-delay realizations.

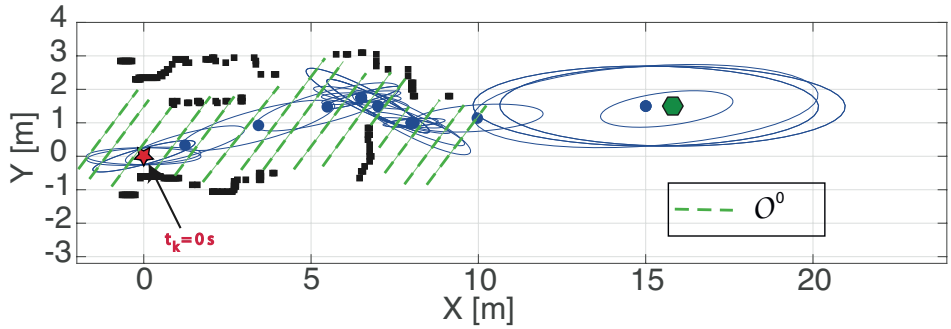


Fig. 3.18: The green hexagon denotes the target q_E^f, while the red star indicates the initial condition $q_E(0)$.

are available. Through the overall experiment, the networked cumulative delay is depicted in Fig. 3.17. At the initial time instant $t_0 = 0\,\mathrm{s}$, by resorting to the available information $q_E(0)$ and using the **LIDAR**, the SSTMR recognizes the obstacle-free region $\mathcal{O}^{t_0}$ (green dashed zone). This allows to compute the path Γ (blue circles) by the RRT* algorithm and, then, to compute the DoA (3.129), namely the blue ellipsoids. Here for the sake of clarity, the ROSAC sequences pertaining to the maximum delay $\tau = 2$ are only depicted. At $t_k = 10.4\,\mathrm{s}$, the vehicle has moved around the way-point

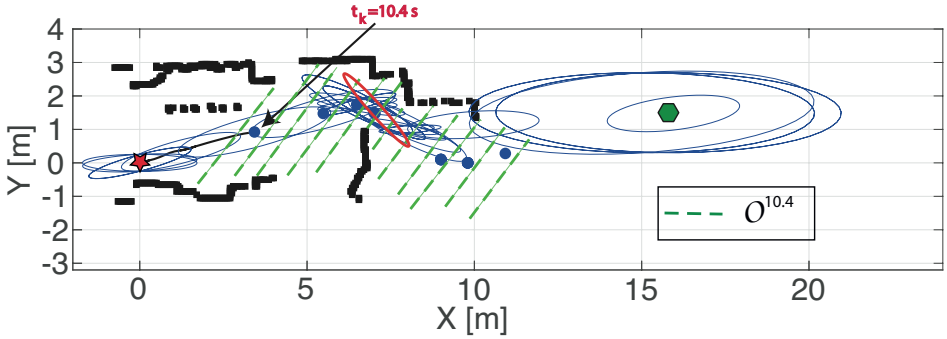


Fig. 3.19: The continuous black line accounts for the SSTMR evolution under the action of (3.130)-(3.131) moving towards overlapped RPIs.

w_2 where the critical condition (3.105) is verified and, as a consequence, the state trajectory tube resulting from the nominal path Γ no longer viable, see Fig. 3.19. Then, a new way-point $w^{10.4} = [6.5 \;\; 1.5 \;\; -0.4636]^T$ is determined by solving (3.135) and an overlapped ellipsoid $\Xi^{10.4}$ (the red region in Fig. 3.19) computed by **Step 12** of the **Navigation-MPC** algorithm. From now on, the vehicle is drive through the state trajectory tube determined by a sequence of 40 overlapped RPIs (red ellipsoids) whose the last one is centered at $w^{26.4} = [9 \;\; 0.1 \;\; -0.1244]^T$, see Fig. 3.23. At $t_k = 26.4\,\text{s}$, the orientation of the overlapped ellipsoids changes towards the nominal path Γ until at $t_k = 33.2\,\text{s}$ the ellipsoid centered at $w^{33.2} = [13.5969 \;\; 0.9492 \;\; 0.2450]^T$ (the mustard-colored one of Fig. 3.20) intersects the nominal path and condition (3.105) is no longer satisfied, so that the off-computed state trajectory can be recovered and the controller again switches to the rule (3.130)-(3.131). As a matter of fact, the resulting control action is significantly less conservative and the SSTMR asymptotically converges to the target q_E^f. This phenomenon can be also observed in Fig. 3.21 (see the shadow zone) where the input behavior is reported: in fact when the vehicle reconnects to the nominal path a saturation effect comes out. Moreover, Fig. 3.22 depicts the distance $dist(\xi, Ob_{min})$ between the SSTMR centroid and the closest obstacle at each time instant t_k, respectively. As expected, the latter is always greater than the vehicle shell d_{shell}. Finally for the sake of completeness, Figs. 3.23-3.24 show all the computed state trajectory tubes (ROSACs, RPIs and overlapped RPIs) and the robot evolution from $q_E(0)$ to q_E^f, whose length is about 16.4 m, respectively.

3.3.11 Energy consumption analysis

For the sake of completeness, the energy analysis arising from the use of the proposed **Navigation-MPC** Algorithm is here provided.

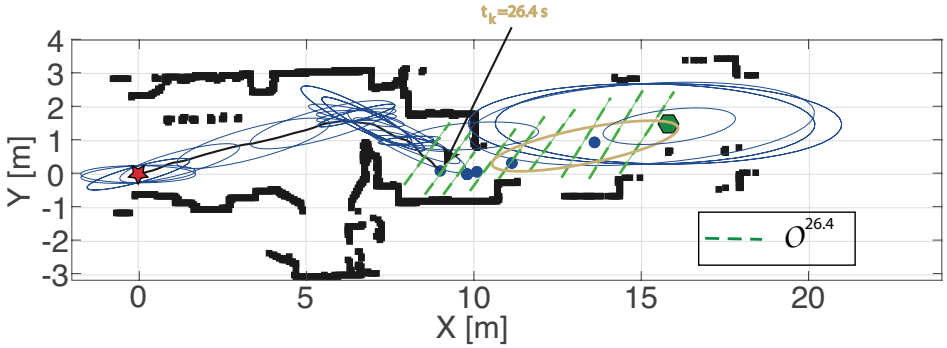


Fig. 3.20: The continuous black line accounts for the SSTMR evolution under the action of (3.133)-(3.134) moving towards the path Γ.

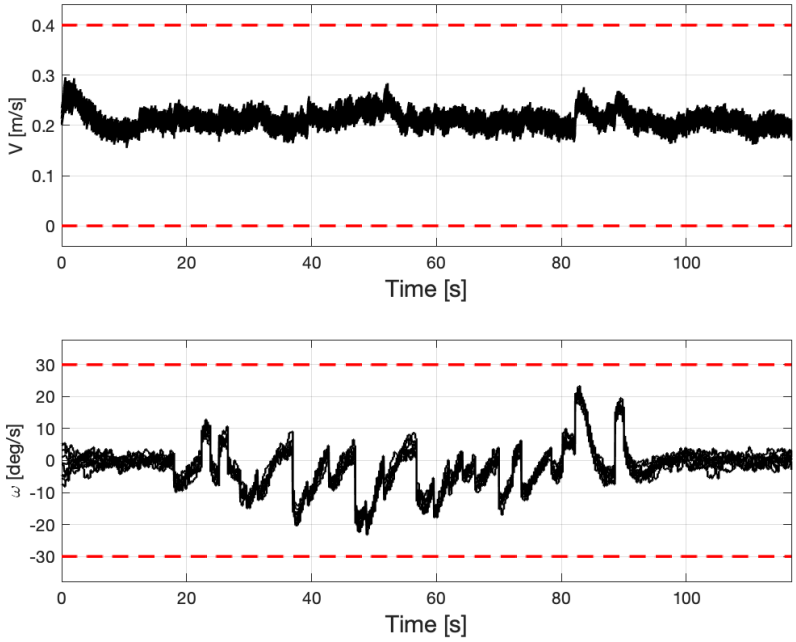


Fig. 3.21: Command inputs. The dashed red lines represent the prescribed constraints.

First of all, recall that the Jaguar V4 vehicle is equipped with a Turnigy LiPo battery whose capacity is 888 $[Wh]$ and its consumption is mainly due to the joint action of two DC motors and the on-board CPU unit. In particular, the DC motors comply with the following consumption upper bound 160 $[W]$ at a speed of 5 $[Km/h]$, while the Jetson Orin Nano operates within the range $[7,\ 25]$ $[W]$. The measurement setup is conceived by using the National Instruments NI9215 module that is capable of providing voltages within the range $[-10,\ 10]$ $[V]$ up to 100 $[kS/s]$ and a resolution

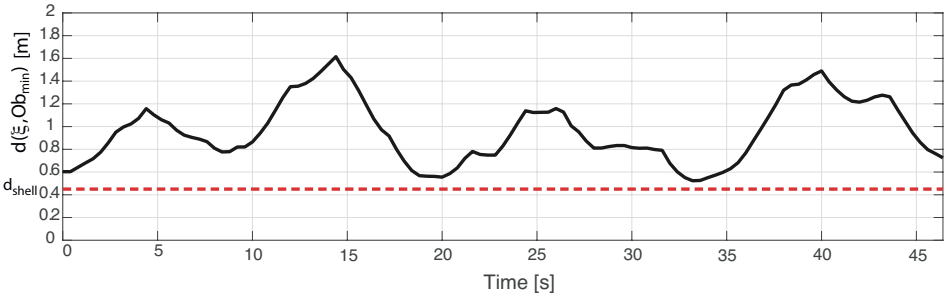


Fig. 3.22: Anti-collision distances. The dashed red line accounts for d_{shell}.

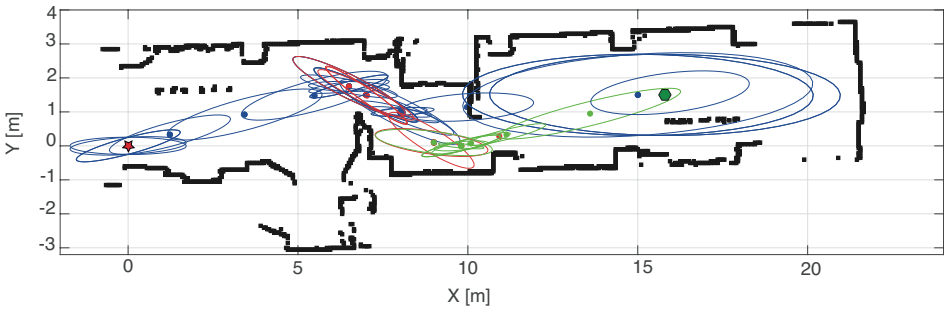


Fig. 3.23: State trajectory tubes: ROSACs families (blue ellipsoids); Overlapped RPIs (red ellipsoids) switching control rule: (3.130) – (3.131) → (3.133) – (3.134); Overlapped RPIs (green ellipsoids) switching control rule: (3.133) – (3.134) → (3.130) – (3.131).

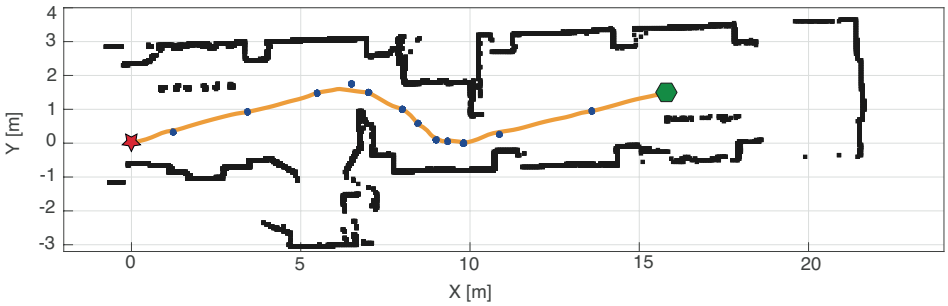


Fig. 3.24: Robot trajectory (solid orange line).

of 16 [*bit*]. Then, current and voltage transducers (the LEM CT 25-T and LEM CV 3-100/SP3, respectively) are used to get measurements compatible with the NI9215 module.

Along the whole experiment, the battery consumption has been reported in Table 3.2. Notice that the re-planning phase consists of two events at $10.4\,s$ and $26.4\,s$, respectively. As it clearly results, the re-planning phase gives rise to an extra battery

consumption of the same order as that pertaining to the control action computation. Then, it is quite reasonable that such an aspect will become critical for long endurance missions in uncertain environments.

Table 3.2: Battery consumption [Wh]

Task	Energy
Actuation	0.576
Control action	0.166
Re-planning	0.111

3.3.12 Computational complexity analysis

As far as the computational burdens are considered, it is necessary to split the discussion between off-line and on-line phase. In the off-line the main computational task concerns with the derivation the ROSAC sets. From a theoretical point of view, the required CPU time grows up linearly with the number of polytope vertices, see [124], while the average CPU time per step related to the experiment is 0.39 *sec*. Conversely, the computation of the overlapped RPI reduces to the $\mathcal{O}(n^2)$ complexity, while its CPU time per step is equal to 0.03 *sec*.

4

Set-theoretic Navigation Methods for Multi-Robot Systems

This chapter focuses on the problem of safe and feasible coordinated navigation for a multi-robot system composed of skid-steered tracked mobile robots in cluttered and unknown environments. Compared to single robots, multi-robot systems introduce additional challenges, such as ensuring collision avoidance, synchronization, and scalability, all in the presence of state and actuation constraints, model uncertainties, external disturbances, and communication delays. To address these issues, two set-theoretic methodologies are presented: the first dedicated to coordinated trajectory planning, the second to temporal synchronization of trajectories.

The structure of the chapter is as follows.

Section 4.1 formulates the problem of coordinated trajectory planning for multi-robot systems, defining the dynamic model of the individual platforms considering disturbances, uncertainties, and feasibility constraints. The problem is then reduced to a graph-based formulation, in which the coordinated trajectories must respect both collision-avoidance and actuation and state constraints.

Section 4.2 introduces the ***Set-Theoretic Feasible Coordinated Trajectory Planning (ST-FCTP)*** *Method. In 4.2.1, the reference model for multi-robot coordination is presented. The closed-loop tracking error dynamics considering communication delays and the definition of the robust positively D-invariant region are presented in 4.2.2-4.2.3. Subsequently, the feasibility conditions for coordinated trajectories are explained in 4.2.4. 4.2.5, the algorithm that solves the planning problem is formulated as a graph search algorithm, integrating feasibility conditions derived from LMI constraints. Finally, 4.2.6 reports the results of experimental simulation campaigns considering two remotely controlled tracked robots.*

Section 4.3 presents the ***Set-theoretic Time-based Trajectory Synchronization (ST-TBTS)*** *Method. In particular, 4.3.1 introduces the dynamics of tracking error in scenarios with disturbances and constraints. In 4.3.2, the definition of the positively invariant robust region is introduced. Subsequently, in 4.3.3, the synchronization method based on the imposition of optimized motion start delays, formulated as an optimization problem based on LMI constraints, considering previously planned feasible independent trajectories, is explained. In 4.3.4, the offline calculation phase of the admissible delay intervals from the online optimization phase is rewritten in the form of algorithms. Finally, Subsection 4.3.5 presents the extended numerical simulations that validate the effectiveness of the proposed synchronization approach.*

4.1 Problem Formulation

Consider the multi-robot system consisting of N remote-controlled, autonomous, skid-steered tracked units, denoted as $\{SSTMR_{(1)}, SSTMR_{(2)}, \cdots SSTMR_{(N))}\}$ operating in a common space $\mathcal{W} \subseteq \mathbf{R}^2$. Let $\mathcal{W}$ be connected. Consider an inertial reference frame $\mathbf{E}$. Let $q^E = [x^E\, y^E\, \theta^E]^T$ be a pose vector expressed in $\mathbf{E}$, defined by the position $\{x^E, y^E\}$ and the direction θ^E, see Fig. 4.1. Let us denote with $\mathcal{A}(q^{\mathbf{E}}) = \{x^E, y^E\}$ the position of the robot, and with $\mathcal{Y}(q^{\mathbf{E}}) = \theta^E$ the robot's orientation. Let $\mathcal{O}$ denote the space occupied by obstacles in $\mathcal{W}$. Let $\mathcal{O}^E_{free} = \mathcal{W} \ominus \mathcal{O}$ be the free space available to the robots. To define the space occupied by the robot body, we consider a circle with the radius $\mathcal{R}$. The space occupied by the robot with the pose q^E is denoted by $\mathcal{R}(q^E)$. Let us define the path $\mathcal{P}^E$ as a sequence of N_P poses

$$\mathcal{P}^E = \{q^E_{<i>}\}_0^{N_P-1} \tag{4.1}$$

with $i = 0 \cdots N_P - 1$. Let $\mathbf{T}_{\mathcal{P}^E}$ be the trajectory obtained by associating a time parametrization with the path $\mathcal{P}^E$. Denote with $\mathbf{T}_{\mathcal{P}^E}(t)$ the pose of the trajectory $\mathbf{T}_{\mathcal{P}^E}$ at time t with $t \in [0, t_{end}]$ where t_{end} is the time required to traverse $\mathbf{T}_{\mathcal{P}^E}$.

We assume that each $SSTMR_{(i)}$ is controlled by the commands of the forward velocities $V_{(i)}$ and the rotational velocities $\omega_{(i)}$. Let

$$u_{(i)} = \begin{bmatrix} V_{(i)} & \omega_{(i)} \end{bmatrix}^T \tag{4.2}$$

be the vector of control variables.

It is assumed that the robot $SSTMR_{(i)}$ has limited movement possibilities according to the following condition

$$u_{(i)} \in \mathcal{U}_{(i)} \tag{4.3}$$

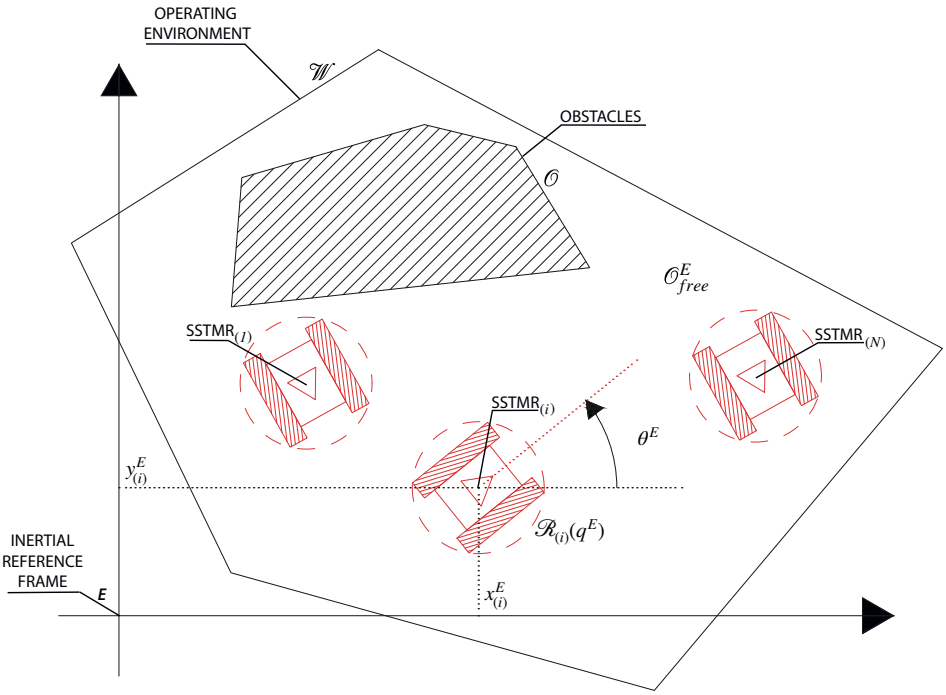


Fig. 4.1: The inertial reference system is denoted by **E**, $\mathcal{W}$ represents the operating environment, $\mathcal{O}$ denotes the occupied space, $\mathcal{O}^E_{free}$ denotes the free space, q^E denotes the pose of the generic $SSTMR$, $\mathcal{R}(q^E)$ denotes the region occupied by $SSTRM$ whose pose is q^E.

where

$$\mathcal{U}_{(i)} := \left\{ \{V_{(i)}, \omega_{(i)}\} : \underline{V}_{(i)} \leq V_{(i)} \leq \overline{V}_{(i)}, \underline{\omega}_{(i)} \leq \omega_{(i)} \leq \overline{\omega}_{(i)} \right\}. \tag{4.4}$$

being $\underline{V}_{(i)}(\underline{\omega}_{(i)})$ and $\overline{V}_{(i)}(\overline{\omega}_{(i)})$ the minimum and maximum values of the forward (rotational) speed, respectively. It is also assumed that the interaction between the tracks and the ground exposes each $SSTMR_{(i)}$ to non-negligible sliding phenomena, so that

$$\hat{u}_{(i)}(t) \neq u_{(i)}(t) \tag{4.5}$$

with $\hat{u}_{(i)}(t) = \left[\hat{V}_{(i)}(t)\ \hat{\omega}_{(i)}(t)\right]^T$ the vector of actual velocities $\hat{V}_{(i)}(t)$ and $\hat{\omega}_{(i)}(t)$ of $SSTMR_{(i)}$ at time t. The two bounded, time-varying positive coefficients, μ^r and μ^l for the used to modeled the track-soil external disturbance acting on the system are subject to the following constraint

$$\mu \in \mathcal{H} \tag{4.6}$$

where

$$\mathcal{H} = \left\{ \{\mu^r, \mu^l\} : \underline{\mu}^r \leq \mu^r \leq \overline{\mu}^r, \underline{\mu}^l \leq \mu^l \leq \overline{\mu}^l \right\}. \tag{4.7}$$

where $\underline{\mu}^r(\underline{\mu}^l)$ and $\overline{\mu}^r(\overline{\mu}^l)$ are the minimum and maximum values of the right (left) sliding coefficient. We assume that

$$\underline{\mu}^r \leq 1 \leq \overline{\mu}^r \tag{4.8}$$

$$\underline{\mu}^l \leq 1 \leq \overline{\mu}^l. \tag{4.9}$$

We also assume that each $SSTMR_{(i)}$ uses its own $\mathcal{C}_{(i)}$ trajectory tracking controller connected via a data communication network. The use of remote control contributes significantly to increasing the operational autonomy of the individual robot unit by shifting the entire computational load to a remote ground station. However, it must be pointed out that such a control architecture has significant drawbacks, mainly related to the introduction of latency, which, if not properly taken into account, can affect the operation and performance of the entire system. In this thesis, we assume that the time delay caused by the data communication channel can be abstractly summarized in a single positive scalar $\tau_{(i)}$ for each robot

$$\tau_{(i)} \in [0,\, \tau_{max}],\ i = 1, \cdots, N \tag{4.10}$$

this is the total delay in the control loop for the i-th robot. The presence of a latency (4.10) in the control loop in combination with the limited handling capabilities (4.3) of the robotic units on the one hand and the sliding phenomena (4.5) due to the interaction between tracks and ground on the other hand can lead to non-zero trajectory tracking errors. As previously explained in the previous chapter 3.2 of this thesis, we refer to this as

$$e_{(i)}(t) = q^E_{(i)}(t) - \mathbf{T}_{\mathcal{P}^E_{(i)}}(t) \tag{4.11}$$

is the difference between the pose of $SSTMR_{(i)}$ and the prescribed one at time t.

In this thesis, we assume that the closed-loop dynamics of the trajectory tracking error, i.e., considering the effect of the controller $\mathcal{C}_{(i)}$, can be described by the following discrete-time uncertain system with norm-bounded uncertainty

$$\xi_{(i)}(t_{k+1}) = \Phi\xi_{(i)}(t_k) + G_\mu \delta\mu_{(i)}(t_k) + G_p p_{(i)}(t_k) \tag{4.12}$$

$$p_{(i)}(t_k) = \Delta_{(i)}(t_k)\sigma_{(i)}(t_k) \tag{4.13}$$

$$\sigma_{(i)}(t_k) = C_\sigma \xi_{(i)}(t_k) \tag{4.14}$$

$$e_{(i)}(t_k) = F\xi_{(i)}(t_k) \tag{4.15}$$

where $\|\Delta_{(i)}(t_k)\| < 1\ \forall t_k \geq 0$ with ϕ,G_μ, C_σ and F matrices of appropriate size and $\delta\mu$ an external disturbance assumed bounded in accordance with the following ellipsoidal constraint

$$\delta\mu_{(i)} \in \Omega_\mu,\ \Omega_\mu = \{\delta\mu_{(i)} \in \mathcal{R}^{n_\mu} : \delta\mu_{(i)}^T M_\mu \delta\mu_{(i)} \leq 1,\ M_\mu > 0\}. \tag{4.16}$$

The aim is to find a solution to the following problem.

Obstacle avoidance and coordinated motion planning in unknown cluttered environments (OACMP-UCE) for N robots forming the MRS

Given a cluttered environment with free space $\mathcal{O}_{free}^E$, an initial state for each robotic platform composing the multi-robot system $q_{(i)}^E(0) \in \mathbb{R}^3$ and a target state $q_{(i)}^E(t_{end}) \in \mathbb{R}^3$, the problem consists ensuring safe and coordinated navigation for each robot from $q_{(i)}^E(0)$ to $q_{(i)}^E(t_{end})$. The strategy must always respect state and input constraints, guarantee collision avoidance with obstacles in the environment (3.9), ensure the coordinated collision free movements of the robots, and cope with uncertainties introduced by the presence of communication delays, and sliding phenomena modeled as external disturbances. The ***OACMP-UCE*** *problem thus requires addressing two key aspects:*

- *define a suitable mathematical model as in* (4.12)-(4.15) *that can describe the dynamics of the trajectory tracking error of each robotic unit by taking into account both the effects of the time delay caused by the communication channel and the time-varying sliding phenomena of the tracks on the ground;*
- *the design of a robust strategy capable of satisfying constraints and avoiding collisions despite uncertainties and latency effects;*
- *Design a strategy for platform coordination that guarantees safe and synchronized navigation in the operational environment.*

The goal is to plan/synchronize trajectories $\{\mathbf{T}_{\mathcal{P}_{(1)}^E}, \mathbf{T}_{\mathcal{P}_{(2)}^E}, \cdots, \mathbf{T}_{\mathcal{P}_{(N)}^E}\}$ that allow the $SSTMRs$ to move at the nominal velocity V_D and to guarantee :

1. *limited trajectory tracking error*

$$\mathcal{A}\left(q_{(i)}^E(t)\right) \in \mathcal{Z}\left(\mathbf{T}_{\mathcal{P}_{(i)}^E}(t)\right) \tag{4.17}$$

   $\forall t \geq 0$ with $i = 1 \cdots N$ $\forall t \geq 0$ denoting by $\mathcal{Z}\left(\mathbf{T}_{\mathcal{P}_{(i)}^E}(t)\right)$ a suitable convex set with center in $\mathbf{T}_{\mathcal{P}_{(i)}^E}(t)$;

2. *obstacle avoidance*

$$\mathcal{R}\left(q_{(i)}^E(t)\right) \subseteq \mathcal{O}_{free}^E \tag{4.18}$$

   $\forall t \geq 0$ with $i = 1 \cdots N$;

3. *collision avoidance*

$$\mathcal{R}\left(q_{(i)}^E(t)\right) \bigcap \mathcal{R}\left(q_{(j)}^E(t)\right) = \emptyset \tag{4.19}$$

   $\forall t \geq 0$, where $i, j = 1 \cdots N$ and $j \neq i$.

In order for the **OACMP-UCE** *problem to be well-defined, we assume that the following conditions are met:*

(i) there is a connected shared free space in which robots can operate

$$\mathcal{O}_{free}^{E} \neq \emptyset \tag{4.20}$$

(ii) start and end poses ensure that no collisions occur

$$\mathcal{R}\left(\mathbf{T}_{\mathcal{P}_{(i)}^{E}}(0)\right) \cap \mathcal{R}\left(\mathbf{T}_{\mathcal{P}_{(j)}^{E}}(0)\right) = \emptyset \tag{4.21}$$

$$\mathcal{R}\left(\mathbf{T}_{\mathcal{P}_{(i)}^{E}}(t_{end})\right) \cap \mathcal{R}\left(\mathbf{T}_{\mathcal{P}_{(j)}^{E}}(t_{end})\right) = \emptyset \tag{4.22}$$

with $i, j = 1 \cdots N$ *and* $j \neq i$

□

To address the **OACMP-UCE** problem, in this thesis are proposed two complementary set-theoretic methodologies for MRS composed of SSTMRs:

- **Methodology C** proposes a solution to the problem of coordinate the skid-steered units subject to sliding phenomena of a multi-robot system by planning a set of feasible coordinated trajectories. The key aspect is that the proposed solution aims to plan optimal trajectories that robustly guarantee the absence of collisions during the movement of the robots along assigned trajectories, explicitly considering the presence of errors in the trajectory tracking. The collision avoidance aspects are addressed by resorting to set-theoretic arguments. The trajectory planning aspects are reformulated in terms of solving a mixed graph-search problem and solving semi-definite programming problems with constraints expressed by linear matrix inequalities (LMIs).

- **Methodology D** presents a set-theoretic time-based trajectory synchronization method to coordinate the skid-steered units subject to sliding phenomena of a multi-robot system. The closed-loop trajectory tracking error dynamics are expressed in the form of an uncertain system subject to external disturbances and actuation constraints. Collision avoidance and coordination are achieved through synchronization of robot trajectories by imposing delays on platform departures. The procedure exploits the trajectory's feasibility property by solving optimization problems involving Linear Matrix Inequalities (LMIs) constraints.

Both approaches exploit set-theoretic arguments to systematically ensure feasibility, safety, and robustness in real-time motion planning for a MRS composed of SSTMR units.

Methodology C

4.2 A set-theoretic Feasible Coordinated Trajectory Planning Method

This section presents the **Set-Theoretic Feasible Coordinated Trajectory Planning (ST-FCTP)** Method for the SSTMR units of a MRS subject to sliding phenomena and controlled through a communication network (NCS), originally proposed in [125]. In order to solve the **OACMP-UCE** problem, the objective of the method is to generate coordinated feasible trajectories composed of a sequence of straight segments, each to be traveled at a desired speed by each robot, which do not intersect obstacles in the operating environment, guaranteeing coordinated collision free movements.

To solve the **OACMP-UCE** problem, the following issues are formally addressed:

- define a suitable mathematical model as in (4.12)-(4.15) that can describe the dynamics of the trajectory tracking error of each robotic unit by taking into account both the effects of the time delay caused by the communication channel and the time-varying sliding phenomena of the tracks on the ground;
- computing sufficient conditions that ensure that a trajectory can always be followed by a robot with an assigned dynamics that guarantees (4.17), despite the possible presence of uncertainties, actuation constraints and external disturbances that alter the robot's motion capabilities;
- formalization of an algorithm for the coordinated planning of a set of coordinate trajectories $\{\mathbf{T}_{\mathcal{P}^E_{(1)}}, \mathbf{T}_{\mathcal{P}^E_{(2)}}, \cdots, \mathbf{T}_{\mathcal{P}^E_{(N)}}\}$, which ensure robust collision-free movements (4.18)-(4.19).

4.2.1 Norm-bounded LDI of the trajectory tracking error dynamics

The kinematic model of the SSTMR, including the effect of track–soil interaction through bounded sliding coefficients, has already been derived in detail in the previous chapter 2. In the same way, this method considers the linearized open–loop trajectory tracking error dynamics, together with its representation in a local reference frame aligned with the desired path. Furthermore, the same operational set-up of Fig. 2.2has been considered, where the sampling of sensor data, the event–driven nature of the controller, and the presence of a bounded end–to–end communication delay have been explicitly modeled. Starting from these results, according to [107], the following LPV representation was used to mathematically describe the dynamics of the trajectory tracking error, explicitly taking into account the maximum possible delays caused by the data transmission channel

$$\tilde{\xi}(t_{k+1}) = \overline{A}(\Theta)\tilde{\xi}(t_k) + \overline{B}_u(\Theta)\delta u(t_k) + \overline{B}_\mu(\Theta)\delta\mu(t_k), \tag{4.23}$$

where Θ is a suitable parameters vector that depends on the delay, and where

$$\tilde{\xi}(t_k) = \left[e^T(t_k)\ \delta u^T(t_{k-1}) \cdots \delta u^T(t_{k-\overline{d}})\right]^T \in \mathcal{R}^{n_s} \tag{4.24}$$

is the vector of lifted states at time t_k, hereinafter referred to as *lifted trajectory tracking error*, with $n_s = n_e + \overline{d} \cdot n_u$, where n_e and n_u are the number of states and control inputs of (3.18), respectively. Finally, in accordance with [109], it is possible to resort to the following uncertain linear system with norm-bounded uncertainty, which embeds the dynamics of the LPV system (4.23)

$$\tilde{\xi}(t_{k+1}) = \tilde{A}\tilde{\xi}(t_k) + \tilde{B}_u\delta u(t_k) + \tilde{B}_\mu\delta\mu(t_k) + \tilde{B}_p p(t_k) \tag{4.25}$$

$$p(t_k) = \Delta(t_k)\sigma(t_k) \tag{4.26}$$

$$\sigma(t_k) = \tilde{C}_\sigma\tilde{\xi}(t_k) + \tilde{D}_{\sigma,u}\delta u(t_k) \tag{4.27}$$

$$e(t_k) = \tilde{F}\tilde{\xi}(t_k) \tag{4.28}$$

where $\|\Delta(t_k)\| < 1\ \forall t_k \geq 0$, and $\tilde{B}_p$, $\tilde{C}_\sigma$, $\tilde{D}_{\sigma,u}$ and $\tilde{F}$ are matrices of proper dimensions.

4.2.2 Closed-loop trajectory tracking error dynamics

In order to define the closed-loop trajectory tracking error dynamics, the control law for trajectory tracking must be taken into account. Without loss of generality, we assume that we use a discrete-time LTI controller with the following structure

$$z(t_{k+1}) = A_C z(t_k) + B_C \tilde{\xi}(t_k) \tag{4.29}$$

$$\delta u(t_k) = C_C z(t_k) + D_C \tilde{\xi}(t_k) \tag{4.30}$$

where $z \in \mathbb{R}^{n_z}$ is the vector of state variables of the control system. Let

$$\xi(t_k) = \left[\tilde{\xi}^T(t_k)\ z^T(t_k)\right]^T \in \mathbb{R}^{n_\xi} \tag{4.31}$$

be the augmented vector of the state variables of the closed control loop with $n_\xi = n_z + n_s$. By combining the equations (4.29)-(4.30) and (4.25)-(4.27), we obtain the following uncertain mathematical model with norm-bounded uncertainty in the presence of external disturbances

$$\xi(t_{k+1}) = \Phi\xi(t_k) + G_\mu \delta\mu(t_k) + G_p p(t_k) \tag{4.32}$$

$$p(t_k) = \Delta(t_k)\sigma(t_k) \tag{4.33}$$

$$\sigma(t_k) = C_\sigma \xi(t_k) \tag{4.34}$$

$$e(t_k) = F\xi(t_k) \tag{4.35}$$

where $\|\Delta(t_k)\| < 1\ \forall t_k \geq 0$, $\phi = \begin{bmatrix} \tilde{A} + \tilde{B}_u D_C & \tilde{B}_u C_C \\ B_C & A_C \end{bmatrix}$,

$$G_\mu = \begin{bmatrix} \tilde{B}_\mu \\ \mathbf{0_{n_c \times n_\mu}} \end{bmatrix}, G_p = \begin{bmatrix} \tilde{B}_p \\ \mathbf{0_{n_c \times n_s}} \end{bmatrix},$$

$C_\sigma = \left[\tilde{C}_\sigma + \tilde{D}_{\sigma,u} D_C\ \tilde{D}_{\sigma,u} C_C\right]$ and $F = \left[\tilde{F}\ \mathbf{0_{n_e \times (\overline{d} \cdot n_u + n_c)}}\right]$.

4.2.3 Definition of the positive robust D-invariant region

At the price cost of a certain degree of conservatism, suppose that the constraints (4.3) and (4.6) can be recast into the the following ellipsoidal form

$$\xi \in \Omega_\xi, \Omega_\xi = \{\xi \in \mathcal{R}^{n_\xi} : \xi^T S_\xi \xi \leq 1,\ S_\xi > 0\}. \tag{4.36}$$

$$\delta\mu \in \Omega_\mu,\ \Omega_\mu = \{\delta\mu \in \mathcal{R}^{n_\mu} : \delta\mu^T M_\mu \delta\mu \leq 1,\ M_\mu > 0\}. \tag{4.37}$$

Consider a given convex constraint on the allowable trajectory tracking error

$$[e_x\ e_y]^T \in \mathcal{Z}\,(\mathbf{0})\,. \tag{4.38}$$

According to (4.35), (4.31), (4.24) and (4.11), eq.(4.38) can be rewritten in the following form

$$\xi \in \Omega_{\mathcal{Z}}, \Omega_{\mathcal{Z}} = \{\xi \in \mathcal{R}^{n_\xi} : \xi^T S_{\mathcal{Z}} \xi \leq 1,\ S_{\mathcal{Z}} \geq 0\}. \tag{4.39}$$

According to (4.17) this means that

$$\mathcal{A}\left(q^{E}(t_k)\right) \in \mathcal{Z}\left(\mathbf{T}_{\mathcal{P}^E}(t_k)\right), \forall t_k \geq t_0. \tag{4.40}$$

where $\mathcal{Z}\left(\mathbf{T}_{\mathcal{P}^E}(t_k)\right)$ is the region $\mathcal{Z}\left(\mathbf{0}\right)$, which is centered in $\mathcal{A}\left(\mathbf{T}_{\mathcal{P}^E}(t_k)\right)$ and rotated by $\mathcal{Y}\left(\mathbf{T}_{\mathcal{P}^E}(t_k)\right)$. Finally, according to [111], we assume that a robust D-invariant ellipsoidal region can be defined

$$\mathit{\Gamma}_0 = \{\xi \in \mathcal{R}^{n_\xi} : \xi^T P_0 \xi \leq 1 \;\; P_0 > 0\} \subseteq \Omega_\xi \bigcap \Omega_{\mathcal{Z}}. \tag{4.41}$$

which fulfills the following property: if $\xi(t_0) \in \mathit{\Gamma}_0$ then $\xi(t_k) \in \mathit{\Gamma}_0 \; \forall t_k \geq t_0$ regardless of the allowable uncertainty of (4.32)-(4.34) and the external disturbances (4.37).

4.2.4 Trajectory feasibility

In this section, we first outline the set-theoretic arguments necessary to define the sufficient conditions for planning a feasible trajectory for a single $SSTRM$. Since the full procedure has already been detailed in Section 3.2.5, we provide only the essential references here, while a detailed explanation of the algorithm for coordinated feasible trajectory planning is given in the following.

Consider the path shown in Fig. 4.2, which consists of a sequence of straight line segments $\overline{W_0W_1}$ and $\overline{W_1W_2}$. Let $\mathbf{E}$ be the reference system centered in W_0 and such that W_1 lies on the x-axis. Let $\mathbf{E}$ be defined by the following two pose sequences $\mathcal{P}_1^E = \{q_k^E\}_0^1$ and $\mathcal{P}_2^E = \{q_k^E\}_2^3$ where $\mathcal{A}(q_{<0>}^E) = W_0$, $\mathcal{Y}(q_{<0>}^E) = 0$, $\mathcal{A}(q_{<1>}^E) = W_1$, $\mathcal{Y}(q_{<1>}^E) = 0$, $\mathcal{A}(q_{<2>}^E) = W_1$, $\mathcal{Y}(q_{<2>}^E) = \hat{\theta}^E$, $\mathcal{A}(q_{<3>}^E) = W_2$ and $\mathcal{Y}(q_{<3>}^E) = \hat{\theta}^E$. Let $\mathbf{T}_{\mathcal{P}_1^E}$ and $\mathbf{T}_{\mathcal{P}_2^E}$ be the trajectories obtained from $\mathcal{P}_1^E$ and $\mathcal{P}_2^E$, respectively, using the following time parametrization. Following eq. (3.58) let $\mathcal{N}_1$ be the maximum positive integer so that

$$\mathcal{N}_1 \leq \frac{\|W_0 - W_1\|_2}{V_D \cdot T_s}, \tag{4.42}$$

then, consider a second reference frame $\mathbf{L}$ centered in W_1 and obtained from $\mathbf{E}$ by the roto-translation (3.10) where the matrix R_E^L is defined as in (3.11) being $\theta_0^E = \hat{\theta}^E$. Let $\mathcal{N}_2$ be the maximum positive integer such that

$$\mathcal{N}_2 \leq \frac{\|W_2 - W_1\|_2}{V_D \cdot T_s}. \tag{4.43}$$

Let $\mathbf{T}_{\mathcal{P}_1^E}$ be the trajectory expressed in $\mathbf{E}$ for which

$$\mathbf{T}_{\mathcal{P}_1^E}(t) = \begin{bmatrix} V_D \cdot t & 0 & 0 \end{bmatrix}^T \tag{4.44}$$

with $t \in [0, \mathcal{N}_1 \cdot T_s)$, while the trajectory $\mathbf{T}_{\mathcal{P}_2^E}$ can be expressed more conveniently in the reference frame $\mathbf{L}$ as

$$\mathbf{T}_{\mathcal{P}_2^L}(t) = \begin{bmatrix} V_D \cdot (t - \mathcal{N}_1 \cdot T_s) & 0 & 0 \end{bmatrix}^T \tag{4.45}$$

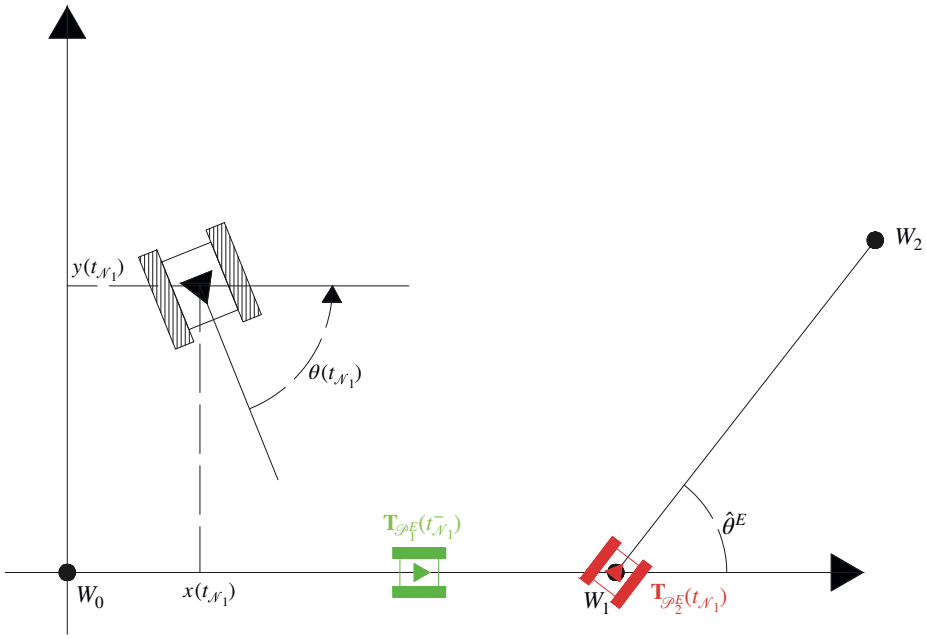


Fig. 4.2: Path segment switch from $\overline{W_0W_1}$ to $\overline{W_1W_2}$. The pose of the robot $q^E(t_{\mathcal{N}_1})$ is shown in black. The green robot represents the pose at time $\mathbf{T}_{\mathcal{P}_1^E}(t_{\mathcal{N}_1}^-)$. The red robot represents the pose $\mathbf{T}_{\mathcal{P}_2^E}(t_{\mathcal{N}_1})$.

with $t \in [\mathcal{N}_1 \cdot T_s, (\mathcal{N}_1 + \mathcal{N}_2) \cdot T_s)$. Consider the dynamics of the trajectory tracking error (4.32)-(4.35).

The succession of trajectories $\mathbf{T}_{\mathcal{P}_1^E}$ and $\mathbf{T}_{\mathcal{P}_2^E}$ is considered feasible if

$$\xi(t_k) \in \Gamma_0, \forall t_k \in [0, (\mathcal{N}_1 + \mathcal{N}_2) \cdot T_s]. \tag{4.46}$$

Remark 8 - The procedure to prove the feasibility condition in (4.46) follows exactly the same set-theoretic approach already presented for Methodology A 3.2 in Section 3.2.5. Therefore, for the sake of brevity, the derivation of the admissible tracking error sets, the computation of the instantaneous transition offset due to pose changes, and the associated SDP problem with LMI constraints are omitted.

4.2.5 Coordinated feasible trajectories planning algorithm

Let Ψ be a discretization by n_Ψ points of the free space $\mathcal{O}_{free}^E$. Let us denote by

$$c = [x_c^E \ y_c^E]^T \in \mathcal{O}_{free}^E \tag{4.47}$$

the generic point of Ψ expressed in $\mathbf{E}$. Let us now introduce the set $\Psi^{(N)} = \Psi \times \Psi \times \Psi \times \cdots \times \Psi$. Let us define the weighted oriented graph $\mathcal{G}^{(N)}$ whose nodes correspond to the elements of $\Psi^{(N)}$. Given two generic nodes

$$g_a = \{c^E_{a_1}, c^E_{a_2}, \cdots, c^E_{a_N}\} \in \Psi^{(N)} \tag{4.48}$$

$$g_b = \{c^E_{b_1}, c^E_{b_2}, \cdots, c^E_{b_N}\} \in \Psi^{(N)}, \tag{4.49}$$

the arc connecting g_a with g_b is denoted by $\mathcal{W}_{ab}$. Note that $\mathcal{W}_{ab}$ identifies the set of N segments connecting $c^E_{a_i}$ to $c^E_{b_i}$, where $i = 1, \cdots, N$. To restrict the density of the graph $\mathcal{G}^{(N)}$, assume that the arc $\mathcal{W}_{ab}$ only exists if

$$c^E_{b_i} \in \mathcal{S}_{\overline{L}}(c^E_{a_i}),\ i = 1 \cdots N \tag{4.50}$$

where $\mathcal{S}_{\overline{L}}(c^E)$ is the largest circle with a radius less than or equal to $\overline{L}$ that is centered in c^E and completely contained in $\mathcal{O}^E_{free}$. Let $c^E_{I_i}(c^E_{F_i}) \in \Psi$ be the initial (final) position for the i-th trajectory. The solution of the planning problem of the N coordinated feasible trajectories starting from $c^E_{I_i}$ and ending in $c^E_{F_i}$, possibly of minimum length, which guarantee the avoidance of collisions between the robots, is thus obtained by recursively exploring the $\mathcal{G}^{(N)}$.

Given a sequence of nodes of $\mathcal{G}^{(N)}$, according to the notation introduced in Sect.4.2.4, let $\mathcal{P}^{(N)}$ be a sequence of poses representing the N paths to be traversed at constant forward velocity V_D. Consider an $SSTMR_{(i)}$, denote by $\mathbf{T}_{\mathcal{P}^E_{(i)}}$ its trajectory and by $q^E_{(i)}(t)$ its pose at generic time t. In accordance with Sect. 4.2.4 it is assumed that all trajectories $\mathbf{T}_{\mathcal{P}^E_{(i)}}$ with $i = 1 \cdots N$ are feasible. According to (4.17)- (4.18) it is assumed that the following conditions are checked

$$\mathcal{A}\left(q^E_{(i)}(t)\right) \in \mathcal{Z}(\mathbf{T}_{\mathcal{P}^E_{(i)}}(t)) \tag{4.51}$$

$$\mathcal{R}\left(q^E_{(i)}(t)\right) \subseteq \mathcal{O}^E_{free} \tag{4.52}$$

and finally, in accordance with (4.19), given $SSTMR_{(i)}$ and $SSTMR_{(j)}$ with $i, j = 1, \cdots, N$ and $i \neq j$, it is guaranteed that $\forall q^E_{(i)}(t) : \mathcal{A}\left(q^E_{(i)}(t)\right) \in \mathcal{Z}(\mathbf{T}_{\mathcal{P}^E_{(i)}}(t))$ and $\forall q^E_{(j)}(t) : \mathcal{A}\left(q^E_{(j)}(t)\right) \in \mathcal{Z}(\mathbf{T}_{\mathcal{P}^E_{(j)}}(t))$ with $t \geq 0$

$$\mathcal{R}(q^E_{(i)}(\cdot)) \bigcap \mathcal{R}(q^E_{(j)}(\cdot)) = \emptyset \tag{4.53}$$

see Fig. 4.3. Let $l_{\mathcal{P}^{(N)}} = \sum_{i=1}^{N} l_{\mathcal{P}^E_{(i)}}$, where $l_{\mathcal{P}^E_{(i)}}$ is the length of $\mathbf{T}_{\mathcal{P}^E_{(i)}}$. The exploration of $\mathcal{G}^{(N)}$ aims to find an alternative sequence of nodes (if any) that identifies a new sequence of poses $\overline{\mathcal{P}}^{(N)}$, which is characterized by the same properties of $\mathcal{P}^{(N)}$, but for which the following condition is satisfied

$$l_{\overline{\mathcal{P}}^{(N)}} < l_{\mathcal{P}^{(N)}}.$$

To achieve this, a recursive exploration of the graph $\mathcal{G}^{(N)}$ was chosen in this work, which uses an algorithm of research A^*.

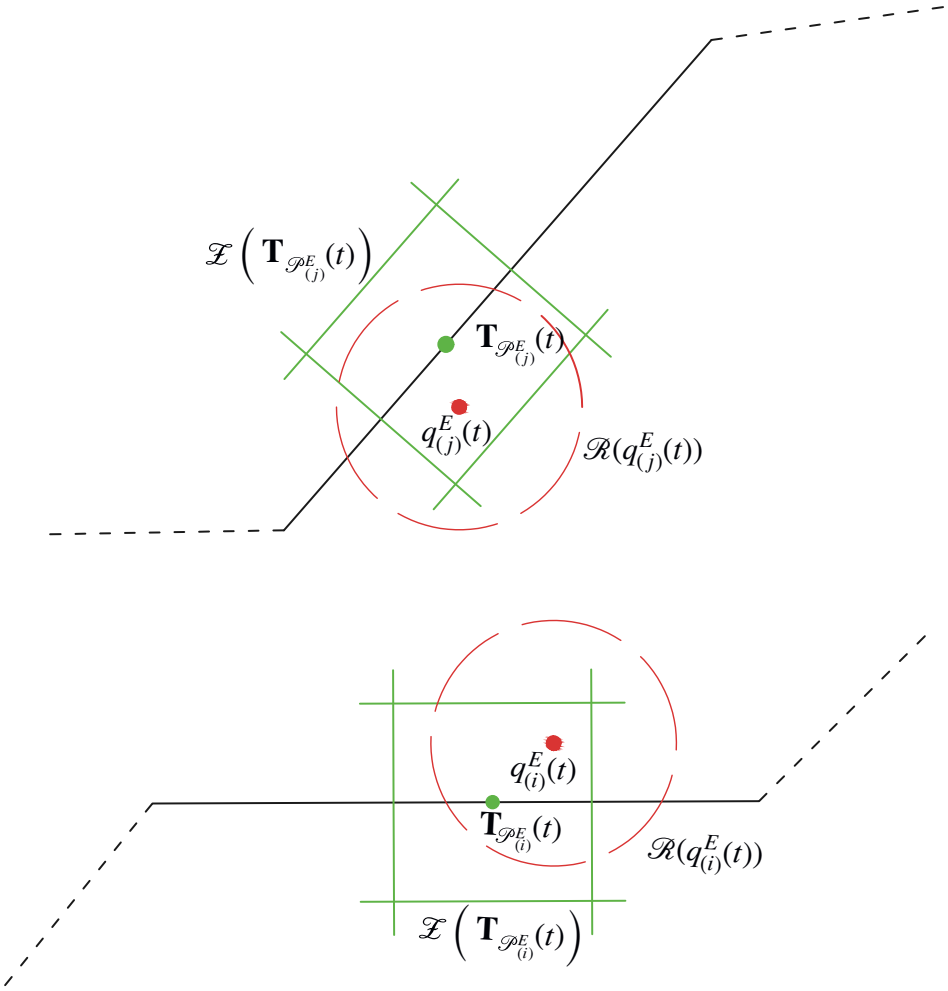


Fig. 4.3: The red dots represent the position of the robots. The red circles represent the space occupied by the robots. The green dots represent the points along the trajectories. The green rectangle represents the regions where the robots are guaranteed to be.

4.2.6 Results—Experimental Validation

In order to demonstrate the effectiveness of the algorithm described in Sect. 4.2.5, some planning results obtained for a multi-robot system, involving two robotic platforms, are discussed here. Let us consider the experimental setup depicted in Fig. 4.4. It consists of two tracked skid-steered robotic units mod. Jaguar v4 by Dr.Robot. The robot has dimensions $0.6 \times 0.7\,\mathrm{m}^2$. Specifically, each robotic platform is connected to its control unit via a common 802.11n data communication channel on a wireless network using the TCP/IP protocol stack. The controllers are responsible for zeroing the trajectory tracking errors, which are estimated using a motion capture

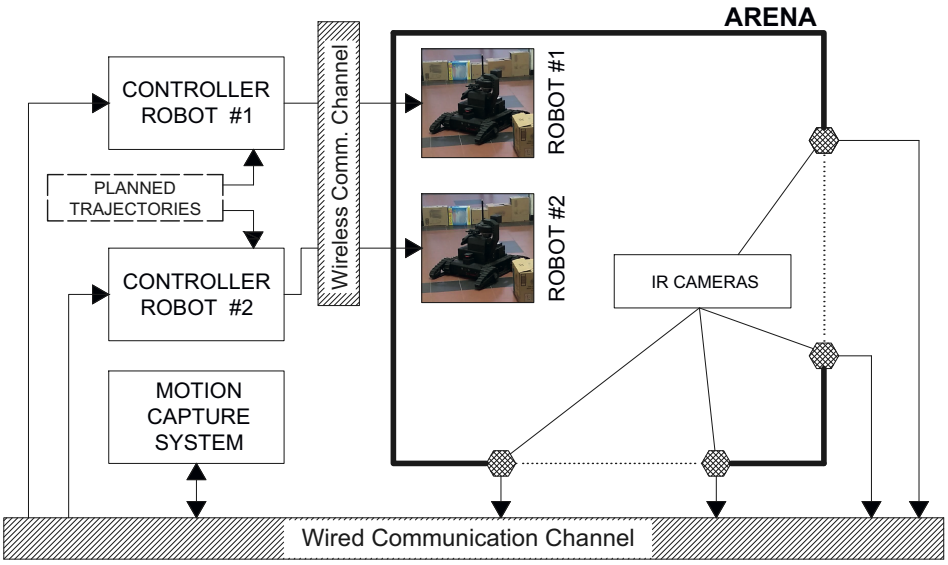


Fig. 4.4: Experimental setup architecture.

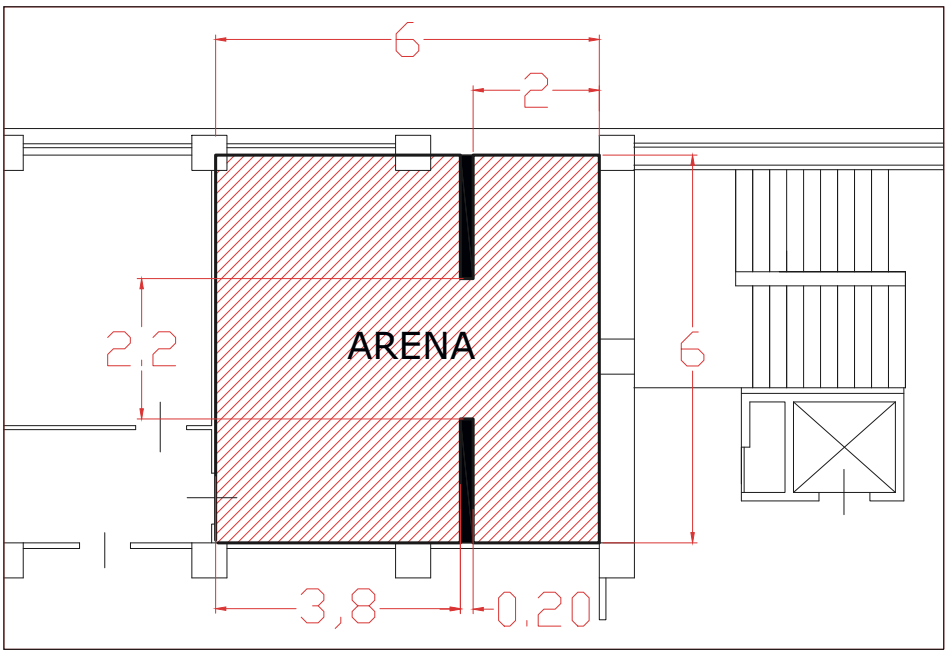


Fig. 4.5: Experimental Operating Scenario.

system by VICON. The motion capture systems send all relevant information to the controllers via a wired communication channel using the TCP/IP protocol stack. The pose of the robots is measured synchronously every $T_S = 0.2\ s$. The robots operate asynchronously by applying a control action as soon as it is available and maintaining it until a new command (ZOH) is received. In accordance with Sect. 2.2, the

mathematical model proposed in (4.25)-(4.28) has been defined. For this purpose, a series of experimental measurements were carried out in advance to estimate both the maximum delay introduced by data communication channels (4.10) and the allowable values of the sliding coefficients in the operating scenario, see (4.6). In particular, it is assumed that the sliding coefficients μ_r and μ_l can deviate from the unit nominal value of $\pm 25\%$ and that $\tau_{max} \leq 2T_s$. All values required to define the uncertain mathematical model can be found in Tab. 4.1.

Table 4.1: Experimental Scenario: relevant values.

Parameter	min	max	typ.	unit
R	-	-	0.08	m
D	-	-	0.5	m
e_x	-0.35	0.35	-	m
e_y	-0.35	0.35	-	m
e_θ	-60	60	-	deg
$\mathcal{R}$	-	-	0.46	m
V	0	0.4	0.2	m/s
ω	-35	35	0	deg/s
$\mu_r(\mu_l)$	0.75	1.25	1	-
τ	$20 \cdot 10^{-3}$	$280 \cdot 10^{-3}$	-	s
T_s	-	-	0.2	s
V_D	-	-	0.2	m/s

Each robotic unit uses the following discrete-time proportional-integral (PI) control law for trajectory tracking:

$$z(t_{k+1}) = z(t_k) + S \cdot \tilde{\xi}(t_k) \cdot T_s, \tag{4.54}$$

$$\delta_u(t_k) = K_i \cdot z(t_k) + K_p \cdot \tilde{\xi}(t_k). \tag{4.55}$$

where $\tilde{\xi}(t_k) = \left[e^T(t_k) \;\; \delta_u^T(t_{k-1}) \;\; \delta_u^T(t_{k-2})\right]^T$ is the lifted trajectory tracking error (4.24) required to account for delays introduced by the architecture of control system and $S = \left[I_2 \; 0_{2\times 5}\right]$,

$$K_i = -\begin{bmatrix} 0.091 & 0 \\ 0 & 0.019 \end{bmatrix}, \tag{4.56}$$

$$K_p = -\begin{bmatrix} 0.402 & 0.001 & 0 & 0.01 & 0 & 0.025 & 0 \\ 0 & 0.821 & 0.5 & 0 & -0.007 & 0 & 0 \end{bmatrix}. \tag{4.57}$$

Based on the above information, the mathematical model of the dynamics of the robot's closed-loop trajectory tracking error (4.32)-(4.35) was defined. Finally, according to Sect. 4.2.2 the robust ellipsoidal region (4.41) was calculated. For the planning purposes, the operating environment shown in Fig. 4.5 with a size of about $6 \times 6\ m^2$ was considered. Taking into account the robot's dimensions, the circular safety zone has a radius $\mathcal{R} = 0.46\,m$, see Sect. 4.1. Based on the maximum permissible errors in trajectory tracking, according to Sect. 4.2.5, the region $\mathcal{Z}(\mathbf{0})$, see (4.17) and (4.38), takes the following form

$$\mathcal{Z}(\mathbf{0}) := \left\{ \{e_x, e_y\} \in \mathbf{R}^2 : J \cdot \begin{bmatrix} e_x \\ e_y \end{bmatrix} \leq b \right\} \tag{4.58}$$

where

$$J = \begin{bmatrix} 1 & -1 & 0 & 0 \\ 0 & 0 & 1 & -1 \end{bmatrix}^T$$

and $b = \begin{bmatrix} e_{x,max} & -e_{x,min} & e_{y,max} & -e_{y,min} \end{bmatrix}^T$, see Tab. 4.1. The free space was discretized, be Ψ the discretization using a regular grid with 515 points with a resolution of 0.25 m. According to Sect. 4.2.5, the weighted oriented graph $\mathcal{G}^{(2)}$ was defined by assuming $\overline{L} = 2$m.

In Figs. 4.6-4.7 are shown the results of the planning algorithm. The planning result obtained by applying the method proposed in this paper is denoted by **TP-MRS** and represented by black lines. For the first robot, the identified path consists of 12 segments with a length of $8,59$ m. For the second robot, the identified path consists of 11 segments with a length of 6.26 m. Both paths must be traveled at the prescribed nominal speed $V_D = 0.2$ m/s. As can be seen in Fig.4.7, see black solid lines, the collision threshold $2 \cdot \mathcal{R}$ is never violated and then the (4.19) is always guaranteed.

Two alternative planning methods, referred to as **TP-D** and **TP-F**, were considered in order to provide a comparison that allows a better evaluation of the results obtained. The method **TP-D**, see red lines in Figs. 4.6-4.7, uses Dijkstra's classical algorithm to plan two independent paths connecting given starting and ending points. The **TP-D** does not consider any temporal parameterization of the paths in the planning phase, furthermore the **TP-D** uses an ad-hoc reduced grid of points to account for the constraint (4.18).

The method **TP-F** uses the set-based approach proposed in [25] to perform the independent planning of two trajectories defined by a sequence of path segments to be traversed at the nominal velocity V_D, guaranteeing the fulfillment of the constraints (4.17)-(4.18). The results are shown in Figs. 4.6-4.7, see green lines.

The results of the comparison are shown in Tab. 4.2. As expected, the algorithm **TP-**

D identifies the two shortest trajectories in the shortest time. However, even under nominal conditions (i.e., zero trajectory tracking error) the collision threshold is violated, see red solid line in Fig.4.7. The algorithm **TP-F** calculates two independent feasible, non-coordinated trajectories that guarantee a limited tracking error (4.17). The trajectories calculated with **TP-F** are shorter than those obtained with **TP-MRS** and the planning time is also shorter, yet the collision threshold is violated due to the lack of coordination between the planned trajectories, see green solid line in Fig.4.7.

When comparing **TP-D**, **TP-F** and **TP-MRS**, it turns out that the latter is the only way to guarantee the absence of collisions during the robots' movements. As expected, this result is achieved at the price of an increase in the total length of the trajectories combined with a significant increase in computation time.

Table 4.2: Comparison summary table.

Algorithm	Total Length	Coll. Avd.	Feas.	Planning Time
TP-D	24.76 m	No	No	1.89 s
TP-F	25.16 m	No	Yes	23.02 s
TP-MRS	27.78 m	Yes	Yes	107.89 s

Figs 4.8-4.9 show the results obtained with 10 experimental runs using the experimental setup described above. These were performed to stress the properties of the trajectories planned by **TP-MRS** through experimental simulations. Fig. 4.8 shows the trajectories executed by the robots with different starting poses in order to change the initial trajectory tracking errors. Fig. 4.9 shows the distance between the geometric centers of the robots. The collision threshold is shown in red dashed line. As designed, the distance between the robots is always greater than the collision threshold. Figs. 4.10 and 4.11 show the trajectory tracking errors and the control velocities of the robots, respectively. As the reader can see, all the values fulfills the relative constraints indicated by the red dashed lines. Finally, to better illustrate the coordinated movement of the two robots, Fig. 4.12 shows the poses $q^E_{(1)}$ and $q^E_{(2)}$ of the two robots at different times during one of the experimental runs performed. The solid green rectangles denote the regions $\mathcal{Z}_{\mathbf{T}_{\mathcal{P}^E_{(1)}}}$ and $\mathcal{Z}_{\mathbf{T}_{\mathcal{P}^E_{(2)}}}$. The red dashed circles represent the circular regions $\mathcal{R}(q^E_{(1)})$ and $\mathcal{R}(q^E_{(2)})$ that enclose the footprint of the robots.

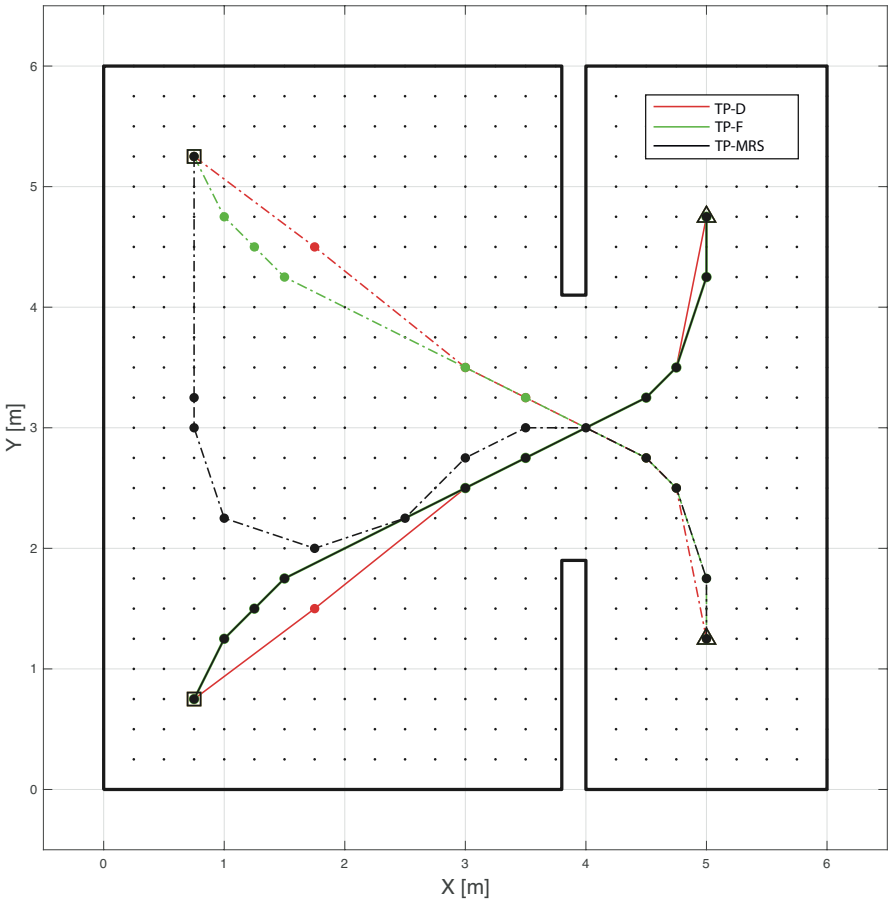


Fig. 4.6: The red and green solid lines show the sequence of segments composing the coordinated planned trajectories. Squares and triangles show the starting and ending points respectively.

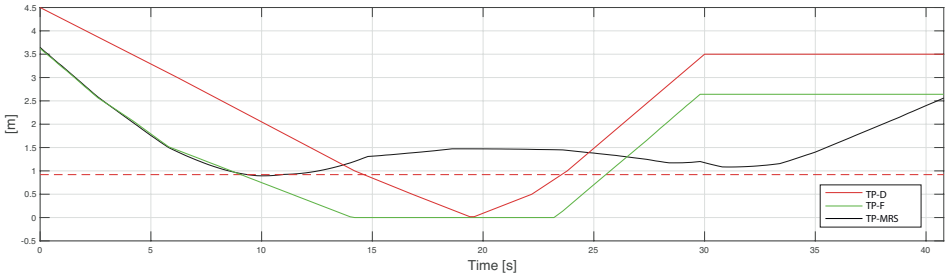


Fig. 4.7: Black (**TP-MRS**) and green (**TP-F**) solid lines represent the worst-case separation distances between robots. Red (**TP-D**) solid line represents the distance between the robots under nominal conditions (i.e. zero trajectory tracking error). The red dashed-dot line indicates the collision threshold.

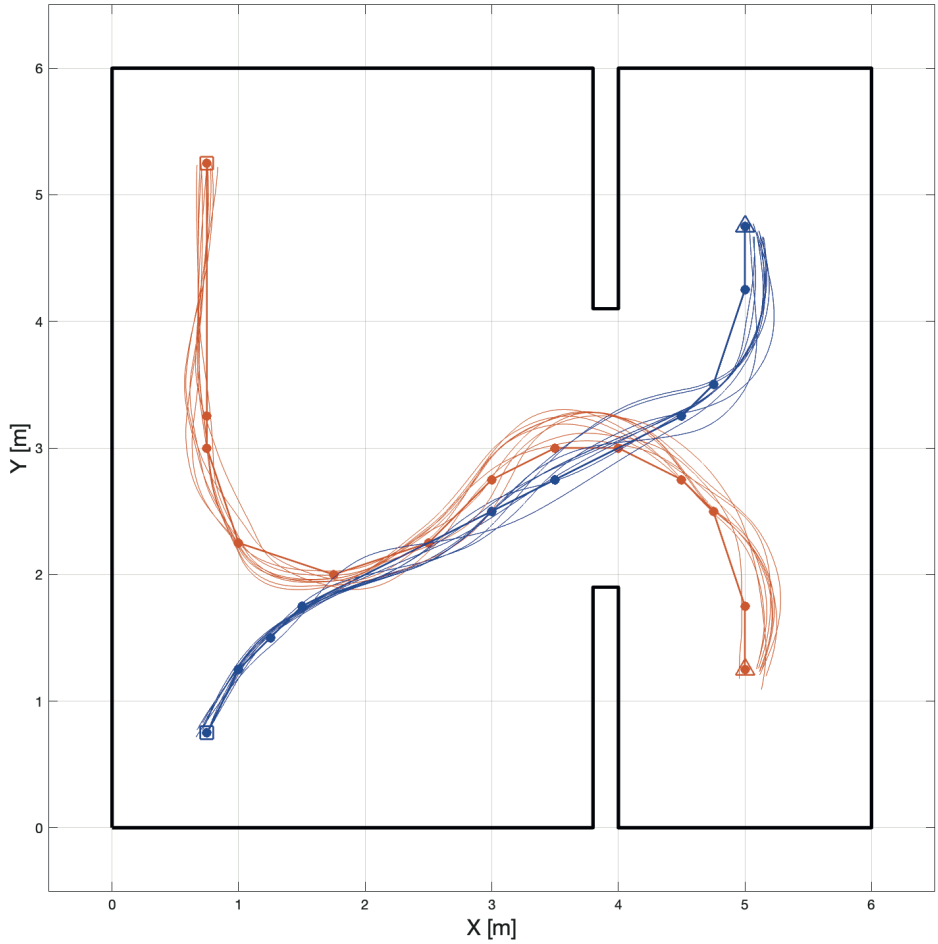


Fig. 4.8: Experimental example. The solid orange (blue) lines represent the paths executed by Robot #1 (#2) in different numerical runs at different starting poses.

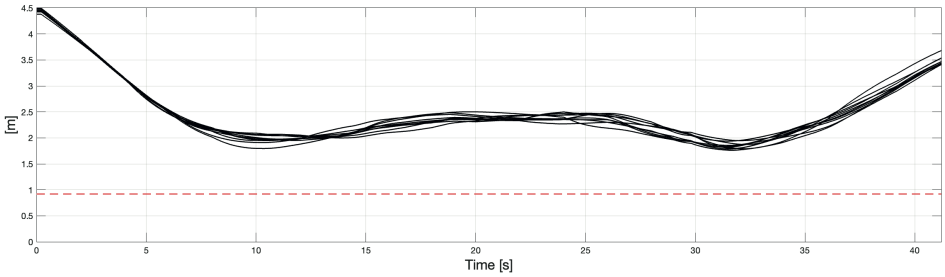


Fig. 4.9: Experimental runs. Experienced distance between the geometric centers of the robots. The red dashed-dot line denotes the collision distance γ.

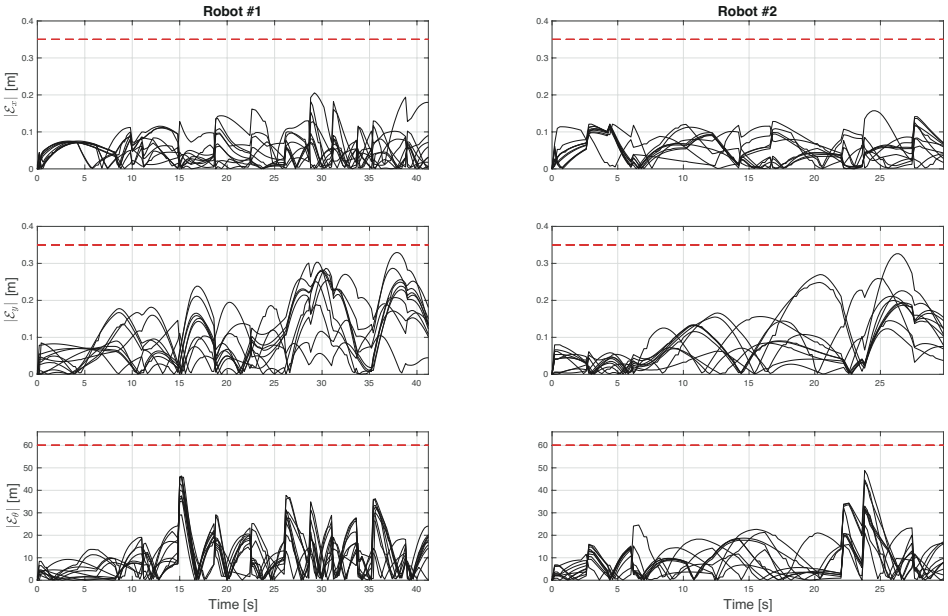


Fig. 4.10: Experimental runs. Trajectories tracking errors. Dashed lines represent allowable bounds.

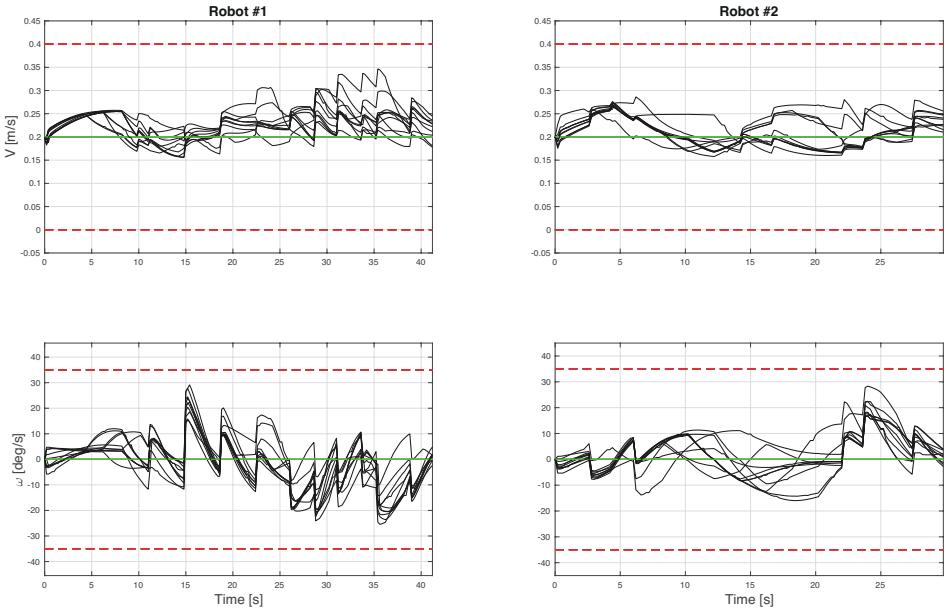


Fig. 4.11: Experimental runs. Control speeds. The dashed lines represent allowable bounds. The green solid lines represent the prescribed forward and rotational speeds along the trajectories

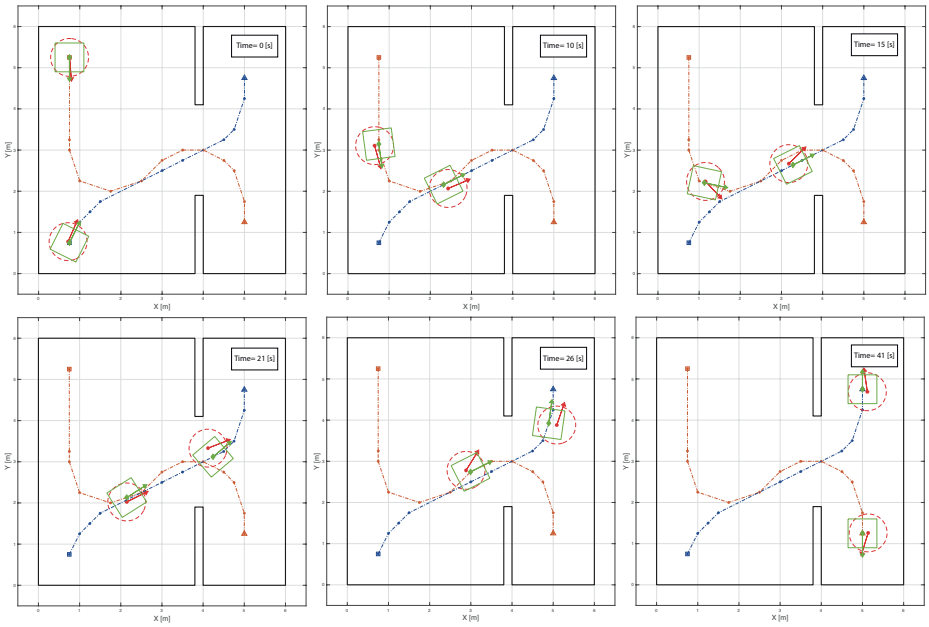


Fig. 4.12: Details of an experimental run. Snapshots at different times. The dashed orange (blue) line represents the path taken by Robot #1 (#2). The squares and triangles show the starting and ending points respectively. Solid green rectangles represent the regions $\mathcal{Z}(\mathbf{T}_{\mathcal{P}^E_{(1)}})$ and $\mathcal{Z}(\mathbf{T}_{\mathcal{P}^E_{(2)}})$, while dashed red circles represent the space occupied by the robots. Green arrows represent $\mathbf{T}_{\mathcal{P}^E_{(1)}}$ and $\mathbf{T}_{\mathcal{P}^E_{(2)}}$. Red arrows indicate the pose of the robots.

Methodology D

4.3 Set-theoretic Time-based Trajectory Synchronization Method

This section presents the **Set-theoretic Time-based Trajectory Synchronization (ST-TBTS)** Method to coordinate the skid-steered units subject to slippage phenomena of a multi-robot system, originally proposed in [126]. In order to solve the **OACMP-UCE** problem, the objective of the method is to synchronize independent feasible trajectories composed of a sequence of straight segments, each to be traveled at a desired speed by each robot, which do not intersect obstacles in the operating environment, guaranteeing coordinated collision free movements. Collision avoidance and coordination are achieved through synchronization of robot trajectories by imposing delays on platform departures. The algorithm operates in separate offline and online phases. In the offline phase, the time delay intervals useful for synchronization are calculated based on the planned feasible trajectories. Then, in the online phase, the actual delay is calculated by solving an optimization problem that minimizes the occupancy time of the shared operational space. To implement the proposed set-theoretic time-based trajectory synchronization method, the following issues are formally addressed:

- modeling the closed-loop trajectory tracking error dynamics of each skid–steered unit as an uncertain system affected by external disturbances and actuation constraints, while explicitly accounting for sliding phenomena;
- ensuring collision avoidance and coordination of the multi-robot system by synchronizing the trajectories through the imposition of suitable departure delays;
- building of a scheduling set-theoretic algorithm for synchronizing a set of feasible independent trajectories $\{\mathbf{T}_{\mathcal{P}^E_{(1)}}, \mathbf{T}_{\mathcal{P}^E_{(2)}}, \cdots, \mathbf{T}_{\mathcal{P}^E_{(N)}}\}$, to guarantee robust collision-free movements for the robotic platforms.

4.3.1 Closed loop trajectory tracking error dynamics

Similar to the methodology described above, this methodology also considers the kinematic model of the SSTMR, which explicitly takes into account the track-ground interaction modeled by limited sliding coefficients, derived entirely in the previous Chapter 2. Assume that each robot has no reactive capabilities implemented, and no robots exchange information with other robots in the system. The dynamics describing the closed-loop trajectory tracking error for a single robot following a straight trajectory with constant velocity can be expressed by an uncertain discrete linear time-invariant system with norm-bounded uncertainty [109] subject to an external disturbance as follows

$$\xi_{k+1} = \phi\xi_k + H_d d_k + B_p p_k, \tag{4.59}$$

$$p_k = \Delta_k q_k, \tag{4.60}$$

$$q_k = \Sigma_q \xi_k, \tag{4.61}$$

where $\|\Delta_k\| < 1 \ \forall k \geq 0$, $\phi = \begin{bmatrix} \tilde{A} + \tilde{B}D_C & \tilde{B}C_C \\ B_C & A_C \end{bmatrix}$, $H_d = \begin{bmatrix} \tilde{B}_d & 0 \end{bmatrix}^T$, $B_p = \begin{bmatrix} \tilde{B}_p & 0 \end{bmatrix}^T$ and $\Sigma_q = \begin{bmatrix} \tilde{C}_q + \tilde{D}_q D_C & \tilde{D}_q C_C \end{bmatrix}$.

4.3.2 Definition of the positive robust D-invariant region

Assume that admissible state and control input variations are constrained according to the following ellipsoidal sets:

$$\xi \in \Omega_\xi, \ \Omega_\xi = \{\xi \in \mathcal{R}^{n_\xi} : \xi^T S_\xi \xi \leq 1, \ S_\xi > 0\}, \tag{4.62}$$

$$\delta u \in \Omega_u, \ \Omega_u = \{\delta u \in \mathcal{R}^{n_u} : \delta u^T S_u \delta u \leq 1, \ S_u > 0\}. \tag{4.63}$$

In addition, n_d external perturbations acting on (4.59)-(4.61) can be bounded according to the following ellipsoidal condition

$$d \in \Omega_d, \ \Omega_d = \{d \in \mathcal{R}^{n_d} : d^T M_d d \leq 1, \ M_d > 0\}. \tag{4.64}$$

Finally, according to [111], it's possible to define the robust D-invariant region

$$\Gamma_0 = \{\xi \in \mathcal{R}^{n_\xi} : \xi^T P_0 \xi \leq 1 \ \ P_0 > 0\}. \tag{4.65}$$

Remark 9 - Region (4.41) is robust to the disturbance d and defines a positively invariant ellipsoidal set in the augmented state space, ensuring $\xi \in \Gamma_0$ for all $t \geq 0$, despite uncertainties and admissible disturbances affecting (4.59)-(4.61), and thus guaranteeing the feasibility property to the trajectories.

4.3.3 Set-theoretic time-based synchronization method

The purpose of this thesis is to exploit set-theoretic methods to solve engineering problems related to mobile robotics, in particular, in this section, the problem of coordinating SSTMR platforms that share a common operating space. In confined spaces where coordinated planning using the 4.2 methodology is not possible, the proposed solution considers coordination in the form of temporal synchronization of the independent trajectories assigned to the robots. In this methodology, the authors specifically aim to exploit the guarantee that the state of the system (4.59)–(4.61) belongs to the ellipsoidal D-invariant set (4.65), which ensures the following limited tracking error condition:

$$\mathcal{A}\left(q^{E}(t_k)\right) \in \mathcal{Z}\left(\mathbf{T}_{\mathcal{P}^E}(t_k)\right), \forall t_k \geq t_0. \tag{4.66}$$

where $\mathcal{Z}\left(\mathbf{T}_{\mathcal{P}^E}(t_k)\right)$ is the region $\mathcal{Z}\left(\mathbf{0}\right)$, which is centered in $\mathcal{A}\left(\mathbf{T}_{\mathcal{P}^E}(t_k)\right)$ and rotated by $\mathcal{Y}\left(\mathbf{T}_{\mathcal{P}^E}(t_k)\right)$, thus allowing the formulation of the following collision avoidance condition:

$$\mathcal{R}\left(q^{E}_{(i)}(t)\right) \bigcap \mathcal{R}\left(q^{E}_{(j)}(t)\right) = \emptyset. \tag{4.67}$$

Therefore, the fundamental prerequisite for the applicability of the method for synchronizing trajectories by scheduling robot departures lies in feasible independent planning of trajectories for robots, which in this work was carried out using the algorithm presented in 3.2. Readers are invited to review the details of the methodology. The feasibility properties of the planned trajectories, ensures that for each trajectory the following feasibility condition holds:

Definition 4.1. *If the initial state of the system $\xi(t_0) \in \Gamma_0$, then $\xi(t_k) \in \Gamma_k \subseteq \Gamma_0$ $\forall t_k \geq t_0$, regardless of the allowable uncertainty of (4.59)-(4.61) and the external disturbances* (4.64)*, along the complete trajectory.*

Proof. For the proof, see Lemma 1.

For each robot the feasible independent trajectory $\mathbf{T}_{\mathcal{P}^E}$, planned in $\mathcal{O}^{free}$, is represented by a succession of the robot's desired pose, hereinafter labeled as waypoints:

$$\mathbf{T}_{\mathcal{P}^E} = [w(t_0), \cdots, w(t_k), \cdots, w(t_\epsilon)] \in \mathrm{R}^{(\epsilon+1)\times 3}, \tag{4.68}$$

each of them with the following form $w(t) = [x(t)\ \ y(t)\ \ \theta(t)]$, and where $w(t_0)$ $(w(t_\epsilon))$ is the starting (ending) desired pose. For each couple of planned trajectories $\mathbf{T}_{\mathcal{P}^E_{(i)}}$ and

$\mathbf{T}_{\mathcal{P}^E_{(j)}}$ with $i, j = 1, \cdots, N$, and $j \neq i$ robotic platforms, where $\mathbf{T}_{\mathcal{P}^E_{(i)}}(t_0) \neq \mathbf{T}_{\mathcal{P}^E_{(j)}}(t_0)$, and $\mathbf{T}_{\mathcal{P}^E_{(i)}}(t_\epsilon) \neq \mathbf{T}_{\mathcal{P}^E_{(j)}}(t_\epsilon)$, the following procedure is applied.

A geometric distance is imposed between the sets to which the states of the systems are guaranteed to belong, such that the collision avoidance condition (4.66) is ensured. To ensure a minimum geometric distance between the ellipsoids associated with points along the robots' trajectories, the problem is solved by considering couple of trajectories. For each pair, a constrained minimization problem and a constrained maximization problem are solved to identify the intervals where collisions could occur. By determining the critical points along the trajectories where the minimum distance is not guaranteed, and exploiting the fact that the trajectories are piecewise straight and traversed at a constant assigned forward speed, it is possible to define departure time intervals for one robot relative to the other that ensure collision-free motion along the trajectories.

4.3.4 Algorithm for trajectory synchronization

Collision avoidance and coordination of the skid-steered platforms composing the MRS are addressed in this work by synchronizing the platforms through time scheduling of robot departures, ensuring a minimum safety distance between the robots during the trajectory tracking. The procedure consists of *offline*, and *online* phases.

- In the *offline* phase, mutual delays between platforms that ensure no collision during robot movement are calculated on the basis of planned feasible trajectories. Assume that for each robotic platform a trajectory is preliminarily planned. The trajectory is composed of a sequence of straight segments that must be traveled by the robot at constant assigned forward velocity v_D. For each combination of feasible independent planned trajectories, the mutual departure delays are calculated to ensure that there are no collisions between the platforms.
- In the *online* phase, the delay calculated for feasible synchronization of trajectories are used in the *online* phase of the algorithm to schedule the motion starts of the robotic platforms according to the instant of time when the activation request occurs. The delay value is calculated by minimizing the overall duration of the operations.

Offline phase:

Consider two generic trajectories

$$\mathbf{T}_{\mathcal{P}^E_{(i)}} = \{S_i, \cdots, A_i, B_i, \cdots F_i\}, \tag{4.69}$$

$$\mathbf{T}_{\mathcal{P}^E_{(j)}} = \{S_j, \cdots, A_j, B_j, \cdots F_j\}, \tag{4.70}$$

where S_i (S_j) is the starting pose of the i-th (j-th) trajectory, and F_i (F_j) is the final pose of the i-th (j-th) trajectory, with A_i, B_i, (A_j, B_j) intermediate waypoints. It is possible to calculate the length of the generic segment $\overline{A_iB_i}$ as follows

$$l_{A_iB_i} = \sqrt{(x_{A_i} - x_{B_i})^2 + (y_{A_i} - y_{B_i})^2}. \tag{4.71}$$

Let N_{AB} be the maximum positive integer for which the following inequality holds

$$N_{A_iB_i} \leq \frac{l_{A_iB_i}}{v_D \cdot T_s} \tag{4.72}$$

where T_s is the sampling time. The trajectory segments composing $\mathbf{T}_{\mathcal{P}^E_{(i)}}$ and $\mathbf{T}_{\mathcal{P}^E_{(j)}}$, are discretized using the interpolation parameters $\alpha, \beta \in [0, 1]$, such that the waypoints along the trajectory can be rewritten as follows:

$$w(\alpha) = (1 - \alpha)A_i + \alpha B_i, \tag{4.73}$$

and

$$w(\beta) = (1 - \beta)A_j + \beta B_j. \tag{4.74}$$

The purpose of this work is to identify all pairs of points for which the collision avoidance condition (4.67) is not met and to calculate the minimum and maximum permissible delays between platform departures that do allow it to be guaranteed. To find these collision extremes the following mixed integers optimization problems involving LMIs constrains are solved:

1) Collision lower bound

$$\underline{\beta} = \min_{\beta,\alpha,q_i,q_j} \beta \tag{4.75}$$

s.t.

$$\alpha \in \{0, \frac{1}{N_{AB}}, \frac{2}{N_{AB}}, \cdots, 1\} \tag{4.76}$$

$$\beta \geq 0, \tag{4.77}$$

$$\beta \leq 1, \tag{4.78}$$

$$\begin{bmatrix} 1 & * \\ R(\theta_\alpha)^T(q_i - w(\alpha)) & \tilde{P}_0^{-1} \end{bmatrix} \geq 0 \tag{4.79}$$

$$\begin{bmatrix} 1 & * \\ R(\theta_\beta)^T(q_j - w(\beta)) & \tilde{P}_0^{-1} \end{bmatrix} \geq 0 \tag{4.80}$$

$$\begin{bmatrix} 2\gamma & (q_i - q_j)^T M^T \\ * & I \end{bmatrix} \geq 0 \tag{4.81}$$

$$\tag{4.82}$$

2) Collision upper bound

$$\overline{\beta} = \max_{\beta,\alpha,q_i,q_j} \beta \tag{4.83}$$

s.t.

$$\alpha \in \{0, \frac{1}{N_{AB}}, \frac{2}{N_{AB}}, \cdots, 1\} \tag{4.84}$$

$$\beta \geq 0, \tag{4.85}$$

$$\beta \leq 1, \tag{4.86}$$

$$\begin{bmatrix} 1 & * \\ R(\theta_\alpha)^T(q_i - w(\alpha)) & \tilde{P}_0^{-1} \end{bmatrix} \geq 0 \tag{4.87}$$

$$\begin{bmatrix} 1 & * \\ R(\theta_\beta)^T(q_j - w(\beta)) & \tilde{P}_0^{-1} \end{bmatrix} \geq 0 \tag{4.88}$$

$$\begin{bmatrix} 2\gamma & (q_i - q_j)^T M^T \\ * & I \end{bmatrix} \geq 0 \tag{4.89}$$

where $\tilde{P}_0$, following [116], denotes the inner ellipsoidal approximation of the projection of the set (4.65) onto x, y, and θ components, $R(\theta_\alpha)$ and $R(\theta_\beta)$ are rotational matrices defined as in eq. (3.11), the matrix M is the following

$$M = \begin{bmatrix} 1\ 0\ 0 \\ 0\ 1\ 0 \\ 0\ 0\ 0 \end{bmatrix}, \tag{4.90}$$

and γ is the radius of the circle $\mathcal{R}\left(q^E_{(i)}(t)\right)$ accounting for the physical dimensions of the robotic platforms calculated as follows:

$$\gamma = \sqrt{\left(\frac{h_1}{2}\right)^2 + \left(\frac{h_2}{2}\right)^2} \tag{4.91}$$

with h_1,h_2 being the width and length of the robot.

It is important to note that the maximum longitudinal and lateral tracking errors are taken into account in the procedure by the constraints (4.79)-(4.80) and (4.87)-(4.88), and the footprint of the platform by (4.81) and (4.89), thus ensuring the presence of a safe distance, between the mobile robots, for every realization of the dynamic state of the system, that provides no collision. See Fig. 4.13 for a 2D simplified graphical representation.

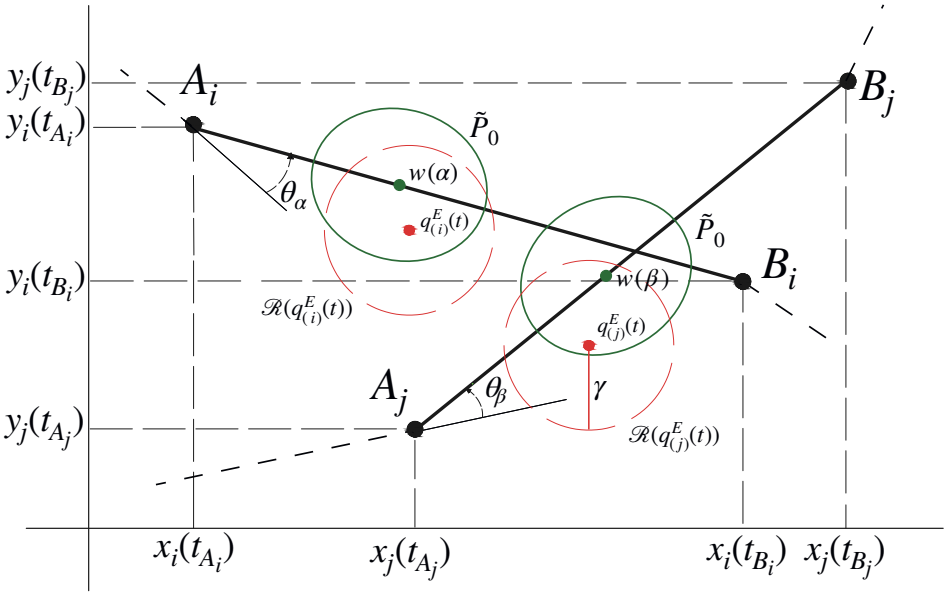


Fig. 4.13: 2D Graphical representation. $\overline{A_iB_i}$ and $\overline{A_jB_j}$ represent two segments of two different trajectories. The green dots represent $w(\alpha)$ and $w(\beta)$. Dark green ellipsoids are a 2D representation of the $\tilde{P}_0$ sets to which the state of the system is guaranteed to belong. The solid red line represents γ, the radius of the circle $\mathcal{R}\left(q^E_{(i)}(t)\right)$ accounting for the physical dimensions of the robotic platforms.

Temporal representation: Departure Delays

The minimum and maximum values of β resulting from the minimization and maximization problems represent the extremes of the range where the collisions could

occur. By considering that the trajectories are traversed by the robots at assigned constant forward speed v_D, it is possible to calculate the instants of time corresponding to the values of $\underline{\beta}$ and $\overline{\beta}$ as:

$$t(\beta) = \frac{\sqrt{(x_\beta - x_{A_j})^2 + (y_\beta - y_{A_j})^2}}{v_D} + t_{A_j}, \tag{4.92}$$

where t_{A_j} is the instant of time when the robot reaches A_j. Calculating the global minimum $t(\underline{\beta})$ and maximum $t(\overline{\beta})$, that considers all the segments that make up the trajectories, the minimum $\underline{\Delta}_{i,j}$ and maximum $\overline{\Delta}_{i,j}$ delay value between the departure times of the skid-steered robotic platforms are calculated as follows:

$$\overline{\Delta}_{i,j} = t_{A_i} + t(\alpha) - t_{A_j} - t(\overline{\beta}), \tag{4.93}$$

$$\underline{\Delta}_{i,j} = t_{A_i} + t(\alpha) - t_{A_j} - t(\underline{\beta}), \tag{4.94}$$

and finally organized into two matrices $\underline{\mathbf{\Delta}}$ and $\overline{\mathbf{\Delta}}$ with the following form:

$$\underline{\mathbf{\Delta}} = \begin{bmatrix} 0 & \underline{\Delta}_{1,2} & \cdots & \underline{\Delta}_{1,i} & \underline{\Delta}_{1,j} & \cdots & \underline{\Delta}_{1,N} \\ \underline{\Delta}_{2,1} & 0 & \cdots & \underline{\Delta}_{2,i} & \underline{\Delta}_{2,j} & \cdots & \underline{\Delta}_{2,N} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \underline{\Delta}_{i,1} & \cdots & \cdots & 0 & \underline{\Delta}_{i,j} & \cdots & \underline{\Delta}_{i,N} \\ \underline{\Delta}_{j,1} & \cdots & \cdots & \underline{\Delta}_{j,i} & 0 & \cdots & \underline{\Delta}_{j,N} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \underline{\Delta}_{N,1} & \cdots & \cdots & \cdots & \cdots & \cdots & 0 \end{bmatrix}, \tag{4.95}$$

$$\overline{\mathbf{\Delta}} = \begin{bmatrix} 0 & \overline{\Delta}_{1,2} & \cdots & \overline{\Delta}_{1,i} & \overline{\Delta}_{1,j} & \cdots & \overline{\Delta}_{1,N} \\ \overline{\Delta}_{2,1} & 0 & \cdots & \overline{\Delta}_{2,i} & \overline{\Delta}_{2,j} & \cdots & \overline{\Delta}_{2,N} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \overline{\Delta}_{i,1} & \cdots & \cdots & 0 & \overline{\Delta}_{i,j} & \cdots & \overline{\Delta}_{i,N} \\ \overline{\Delta}_{j,1} & \cdots & \cdots & \overline{\Delta}_{j,i} & 0 & \cdots & \overline{\Delta}_{j,N} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \overline{\Delta}_{N,1} & \cdots & \cdots & \cdots & \cdots & \cdots & 0 \end{bmatrix}. \tag{4.96}$$

These values represent the minimum and maximum time that the j-th robot can/should wait before starting up, taking into account the presence of the i-th robot. Specifically, the j-th robot can go at most up to $\underline{\Delta}_{i,j}$ seconds later, or it must wait to leave at least $\overline{\Delta}_{i,j}$ seconds to ensure no collisions. The waiting time determines the order in which the robots pass through the critical sections of the trajectory; it is

possible to obtain negative values of $\Delta_{i,j}$, which represent the time advance that one robot must have over the other to avoid collision.

Online phase:

Consider a N robot system. Without loss of generality we assume that $t_{req,N}$ is the instant at which the activation of the N-th robot is requested. Let $t_{act,i}$, with $i = 1, \cdots, N-1$, be the startup times of the first $N-1$ robots. With the aim of freeing up the operational space in the shortest possible time, thus minimizing the total time for carrying out operations, the startup time of the N-th robot is calculated according to the following linear programming problem that guarantees the minimum delay with respect to $t_{req,N}$

$$\Delta_{t_N} = min \Delta_t \tag{4.97}$$

s.t.

$$\Delta_t \geq 0 \tag{4.98}$$

$$t_{req,N} + \Delta_t - t_{act,i} \leq \underline{\Delta}_{N,i},\ i = 1, \cdots, N-1 \tag{4.99}$$

or

$$\Delta_t \geq 0 \tag{4.100}$$

$$t_{req,N} + \Delta_t - t_{act,i} \geq \overline{\Delta}_{N,i},\ i = 1, \cdots, N-1 \tag{4.101}$$

4.3.5 Results—Numerical Validation

In order to test and evaluate the proposed scheduling algorithm several numerical simulation were performed using the MATLAB Simulink environment.

The skid-steered robots are subject to model uncertainties and external disturbances due to varying terrain sliding coefficients. In all the numerical simulations performed, a communication delay due to the data communication system was considered, which was randomly varied over time with a uniform distribution, and whose maximum and minimum values are compatible with those given in Table 4.4. In addition, the initial tracking error was also randomly varied. Let consider the virtual operating environment shown in Fig. 4.14, discretized with a grid of about 700 points with a resolution of $0.5\,m$. The operating environment was divided into 3 regions, each characterized by the sliding coefficients given in Tab 4.3.

The nominal forward velocity assigned to robotic platforms is set at 0.2 m/s. All relevant values for figuring out the simulation are given in Tab. 4.4. By considering a physical footprint of each robot of $0.50 \times 0.70\ m^2$, the minimum distance required

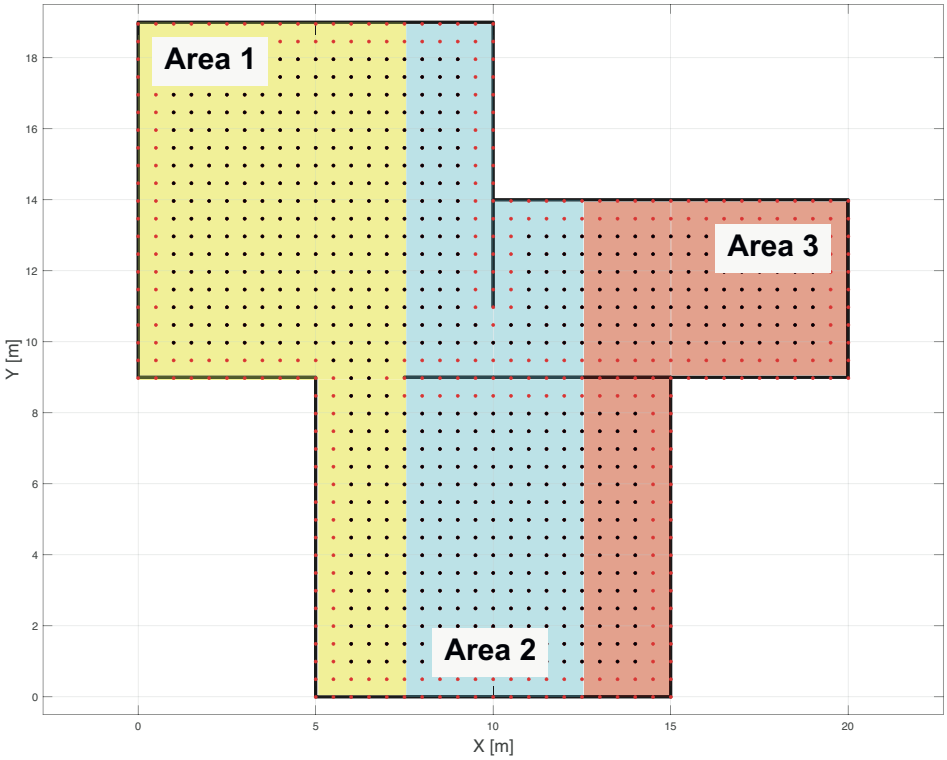


Fig. 4.14: Virtual operating environment. The points represent the grid that discretizes the environment. Black points are utilized for path planning, while red points have been excluded to ensure collision avoidance with the boundaries and obstacles. Three areas with different sliding coefficients are assumed.

Table 4.3: Numerical scenario: sliding coefficients values.

	Area 1	Area 2	Area 3
Acceleration	$\mu_r = 0.75$ $\mu_l = 0.9$	$\mu_r = 0.95$ $\mu_l = 0.95$	$\mu_r = 0.85$ $\mu_l = 0.75$
Deceleration	$\mu_r = 1.25$ $\mu_l = 1.1$	$\mu_r = 1.05$ $\mu_l = 1.05$	$\mu_r = 1.15$ $\mu_l = 1.25$

to avoid collisions results in $2\gamma = 0.86\,m$. Finally, in accordance with Sect. 4.3.4, the calculation of the delay matrices $\overline{\Delta}$ and $\underline{\Delta}$ gave the following results:

$$\underline{\Delta} = \begin{bmatrix} 0 & -109.15 & -154.34 \\ -101.71 & 0 & -169.42 \\ -56.07 & -92.01 & 0 \end{bmatrix}, \tag{4.102}$$

$$\overline{\Delta} = \begin{bmatrix} 0 & 7.88 & -5.01 \\ 54.81 & 0 & 4.32 \\ 110.14 & 83.64 & 0 \end{bmatrix} \tag{4.103}$$

Table 4.4: Simulation relevant values.

Parameter	Minimum value	Maximum value	Unit
e_x	-0.35	0.35	m
e_y	-0.45	0.45	m
e_θ	-60	60	deg
V	0	0.4	m/s
ω	-35	35	deg/s
τ	0	$400 \cdot 10^{-3}$	s
t_{act}	0	60	s

Each planned trajectory was simulated 10 times, and, in every simulation, the activation request times of robots #2 and #3 were randomly varied with a Gaussian distribution in the range $t_{act} = [0, 60]$ s, while robot # 1 is always activated in time instant $t_{act} = 0$. The relevant timing data of the simulations are summarized in Tab. 4.5. The results of the simulations are illustrated in Figs. 4.15-4.18, which highlight key aspects of the method.

Fig. 4.15 showcases the inter-platform distances during the simulations. Notably, the collision avoidance safety distance 2γ was always maintained, even in scenarios where the trajectories intersected or the robots operated in close proximity. This result underscores the effectiveness of the synchronization algorithm in coordinating the robots' movements while adhering to safety constraints.

Fig. 4.16 shows the trajectory tracking errors for all robots during the simulations. These errors remained consistently within the allowable limits, demonstrating the robustness of the feasible planning algorithm used, a key aspect for the application of the proposed solution.

Fig. 4.17 provides profiles of the forward and rotational velocities of the control algorithm. As can be seen, the values remain in the assigned range, demonstrating the effectiveness of the method even in the case of limited actuation capabilities.

Finally, to emphasize the aspect of coordination, Fig. 4.18 shows the pose of the robots at the time when the minimum distance between robot #2 and robot #3, corresponding to $t = 190\,s$, of a specific collected Run, was recorded. The results are based on the data collected during Run #8. The ellipsoidal sets (4.65) to which the

Table 4.5: Relevant simulations timing values.

	Robot #2		Robot #3	
Run #	Request time [s]	Activation time [s]	Request time [s]	Activation time[s]
1	41.03	41.56	47.02	168.72
2	33.10	33.61	35.01	160.78
3	8.27	10.36	13.06	137.54
4	2.50	10.36	6.41	137.53
5	4.95	10.37	43.17	137.54
6	21.27	21.61	58.27	148.78
7	39.74	40.03	14.64	167.20
8	40.81	41.22	31.67	168.38
9	36.15	36.74	45.03	163.90
10	4.01	10.36	56.36	137.50

robot pose belongs are highlighted in the figure. It can be seen that even when they turn out to be closest, the minimum distance imposed by the procedure is correctly met. Overall, these results validate the proposed method as a reliable solution for synchronization and coordination of multi-robot trajectories under realistic conditions, confirming the method's ability to ensure safety, accuracy, and robustness in various operational scenarios.

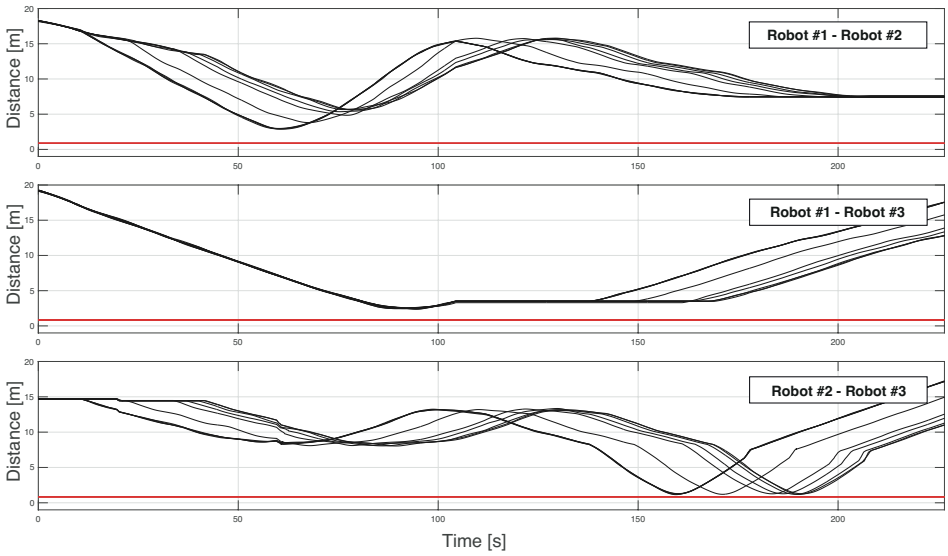


Fig. 4.15: Actual distances between the robots. The solid red line highlights the threshold value 2γ.

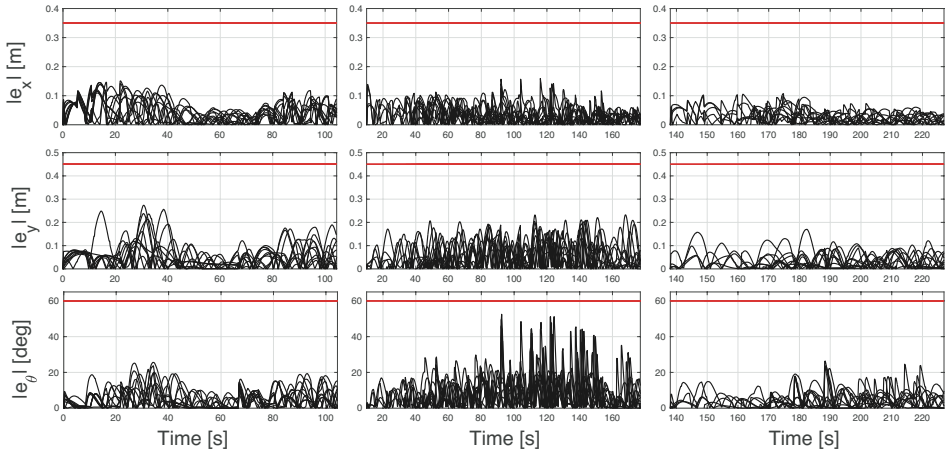


Fig. 4.16: Trajectory tracking errors. From left to right the values recorded for robot #1, #2, and #3 respectively. The solid red lines represent allowable bounds.

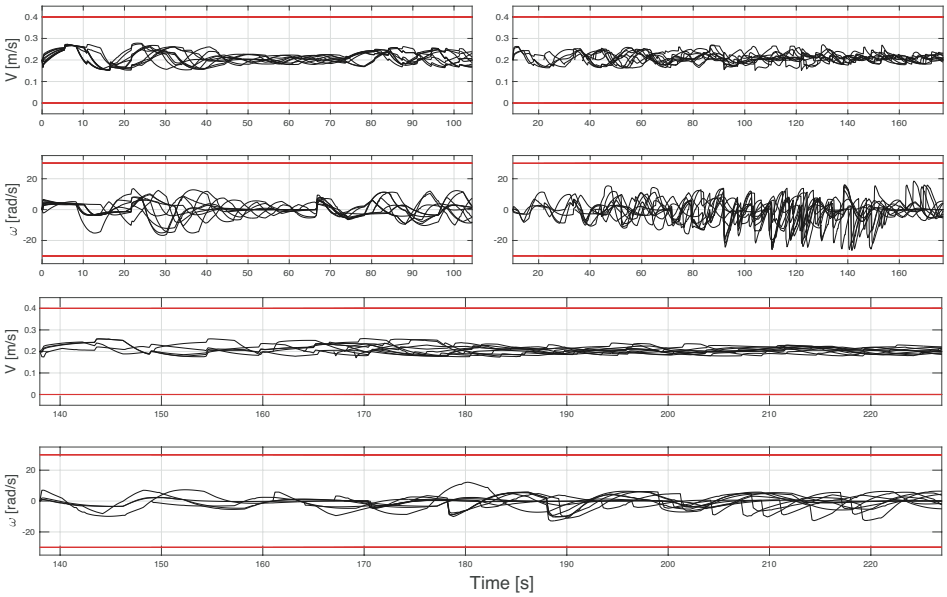


Fig. 4.17: Values of the forward and rotational velocities of the robots. The plots in the top-left, top-right, and bottom-center show the values for robots # 1, # 2, and # 3, respectively. The solid red lines represent allowable bounds.

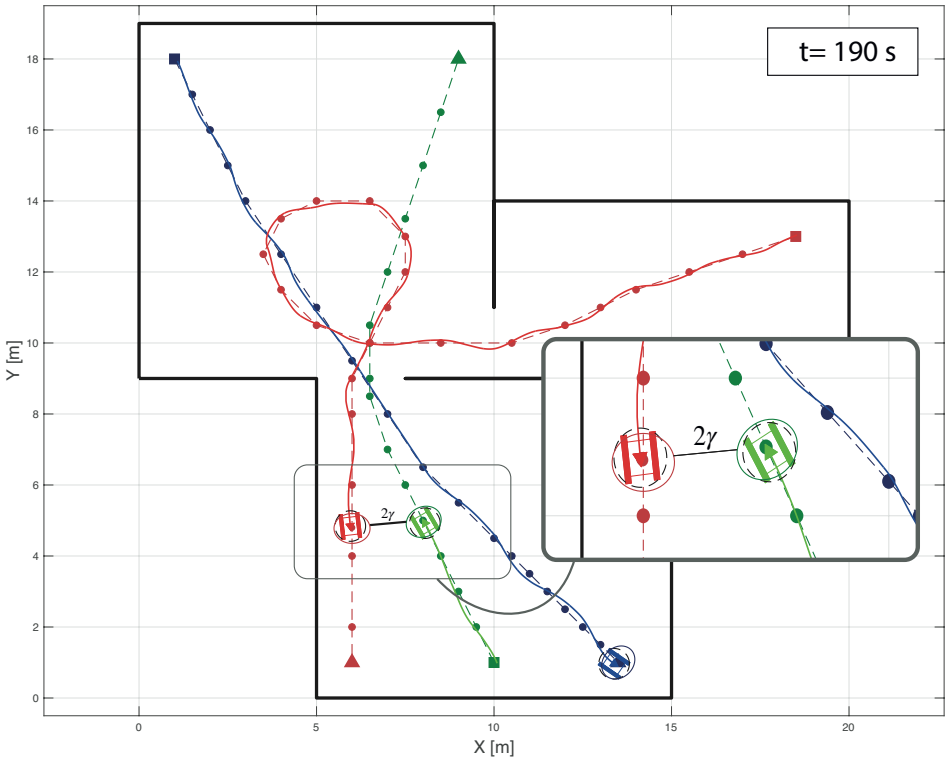


Fig. 4.18: Run #8. Snapshot at the time instant $t = 190\,s$. Dashed line represents assigned trajectories for the robot #1, #2, and #3. Solid lines represents the executed trajectories. Squares and triangles represent the starting and target points. The blue, red, and green ellipsoids represent the D-invariant regions for robot #1, #2, and #3, respectively. Black dashed circles account for the physical footprints of the robot and have radius γ. The black line between the ellipsoids represents the distance recorded, which turns out to be equal to the threshold for collision avoidance 2γ.

5

Conclusions

We conclude this thesis with a summary of the main research contributions and conclusions. We then present future research avenues.

5.1 Summary

This thesis addresses the problem of trajectory and motion planning for remote-controlled skid-steer tracked mobile robots, with the aim of ensuring the feasibility and safety of navigation taking into account the presence of model uncertainties, external disturbances, and actuation constraints in complex environments. The proposed approaches exploit set-theoretic arguments in different ways, with the aim of ensuring robustness, feasibility, and safety in both single-robot and multi-robot scenarios.

The contributions can be summarized as follows:

- **Set-Theoretic Feasible Trajectory Planning (ST-FTP)**
  It solves the problem of planning the optimal trajectory for a networked skid-steered tracked mobile robot. First, a feedback control action is considered that can solve the trajectory tracking problem, taking into account network delay, external disturbances, and actuation constrains. A recursive algorithm is then proposed to find the optimal trajectory that complies with the prescribed control constraints. The feasibility of the trajectory is guaranteed by solving optimization problems involving LMI constraints.
- **Set-Theoretic Model Predictive Control (ST-MPC):**
  A control architecture that ensures safe navigation in cluttered environments under the presence of model uncertainties, disturbances, and network-induced delays. The method integrates set-theoretic predictive control with receding horizon principles to handle obstacle avoidance, delayed information, and robustness to uncertainties, guaranteeing constraint satisfaction and safety.

- **Set-Theoretic Feasible Coordinated Trajectory Planning (ST-FCTP):** A coordinated planning framework for multi-robot systems that ensures collision-free execution of feasible trajectories despite tracking errors, disturbances, and uncertainties. The problem is formulated as a combination of graph search and convex optimization with LMI constraints, ensuring that coordination and collision avoidance are achieved robustly.
- **Set-Theoretic Time-Based Trajectory Synchronization (ST-TBTS):** A temporal coordination methodology that synchronizes robot trajectories by exploiting feasibility properties and solving optimization problems involving LMI and linear programming constraints. The approach minimizes the occupancy time of shared spaces while guaranteeing robustness and collision avoidance.

All contributions have been validated through extensive numerical simulations and experimental campaigns with skid-steered tracked robots available at the Automatic Control Laboratory of the Mediterranean University of Reggio Calabria, demonstrating their practical applicability in real-world scenarios.

5.2 Discussion

The research presented in this thesis demonstrates that methodologies based on set-theoretic arguments can effectively address the challenges of safe navigation for rigid-steering tracked robots in both single-robot and multi-robot contexts. In particular, the results highlight the following points:

- By solving the trajectory planning problem, the integration of a feasibility condition into graph search algorithms allows the optimal path from a starting point to an end point to be found, taking into account control performance and always ensuring a minimum safety distance from obstacles, even in the presence of model uncertainties, communication channel delays, implementation constraints, and external disturbances affecting the system.
- By addressing the control problem of safe navigation through a model predictive control approach integrated with control techniques that exploit set-theoretic arguments, it is possible to manage network-induced delays, unpredictable obstacles, and dynamic uncertainties, ensuring compliance with constraints and robust navigation performance even in crowded environments subject to temporal variations.
- By reformulating the coordinated trajectory planning problem as a graph search integrated with feasibility conditions expressed via LMIs, the proposed set-

theoretic methodology enables multiple robots to safely and reliably operate in shared environments, explicitly accounting for trajectory tracking errors, uncertainties, and external disturbances.

- By solving the synchronization problem through an optimization-based procedure, the introduction of time delays on robot departures ensures the coordinated use of shared operational spaces. This time-based set-theoretic strategy guarantees feasibility and robustness while minimizing the occupancy of common areas, thus improving safety and efficiency in multi-robot missions even in presence of model uncertainties, network delay, external disturbances, and actuation constrains

5.3 Future Research and Challenges

The results obtained open up several perspectives for future research related to the proposed methodologies.

A first direction that we are interested in investigating concerns the extension of the ST-MPC to multi-robot scenarios. Although the proposed approach focuses on the single-robot case, the advantages of predictive control combined with set-theoretic arguments could be further exploited to manage coordination, safety, and robustness in cooperative missions for multi-robot systems. This extension would allow more complex tasks to be tackled, in which multiple platforms must operate jointly in dynamic and uncertain environments exchanging information through communication networks subject to delays.

Another challenging extension concerns the computational scalability and online adaptability of set-theoretic methods. The proposed approaches guarantee security and robustness in the presence of uncertainties and model disturbances, but unfortunately their computational complexity can increase significantly in complex environments or as the number of robots involved grows. Future research could therefore investigate approximate formulations that preserve formal guarantees while reducing complexity. Such developments would further improve the applicability of set-theoretic tools in large-scale, uncertain, and highly dynamic scenarios.

Appendix

Proof of Lemma 1

Proof. To prove that Γ_0 is robustly positively invariant even in the presence of the allowable disturbance d, see Fig. 1 the following condition must be satisfied

$$V(k+1) - V(k) \leq 0 \tag{1}$$

$$\forall d : d^T M_d d \leq 1 \tag{2}$$

being

$$V(k) = \xi_k^T P_0 \xi_k \leq 1 \tag{3}$$

where ξ_k represents the state of (3.34)-(3.36) at the time k. According to (3.44), eq. (1) can be rewritten in the following form

$$\zeta^T \begin{bmatrix} \xi^T P_0 \xi - \xi^T \phi^T P_0 \phi \xi & -\xi^T \phi^T P_0 H_D & -\xi^T \phi^T P_0 B_p \\ * & -H_D^T P_0 H_D & -H_D^T P_0 B_p \\ * & * & -B_p^T P_0 B_p \end{bmatrix} \zeta \geq 0 \tag{4}$$

being $\zeta = \begin{bmatrix} 1 & d^T & p^T \end{bmatrix}^T$. Eq. (2) can be rewritten as

$$\zeta^T \begin{bmatrix} 1 & 0 & 0 \\ * & -M_d & 0 \\ * & * & 0 \end{bmatrix} \zeta \geq 0 \tag{5}$$

Finally, condition $p^T p \leq \xi^T \Sigma_q^T \Sigma_q \xi$ can be rewritten as

$$\zeta^T \begin{bmatrix} \xi^T \Sigma_q^T \Sigma_q \xi & 0 & 0 \\ * & 0 & 0 \\ * & * & -I \end{bmatrix} \zeta \geq 0 \tag{6}$$

By recurring to classical arguments of S-Procedure [109], if there are two positive scalars $\tau_1, \tau_2 \geq 0$ satisfying the following condition

$$\begin{bmatrix} \xi^T(P_0 - \phi^T P_0 \phi - \\ -\tau_1 \Sigma_q^T \Sigma_q)\xi - \tau_2 & -\xi^T \phi^T P_0 H_D & -\xi^T \phi^T P_0 B_p \\ * & -H_D^T P_0 H_D + \tau_2 M_d & -H_D^T P_0 B_p \\ * & * & -B_p^T P_0 B_p + \tau_1 I \end{bmatrix} \geq 0 \tag{7}$$

then (4) is fulfilled for each ζ satisfying (5) and (6).

If

$$\begin{bmatrix} -H_D^T P_0 H_D + \tau_2 M_d & -H_D^T P_0 B_p \\ * & -B_p^T P_0 B_p + \tau_1 I \end{bmatrix} > 0 \tag{8}$$

by applying the Schur's Complement eq.(7) can be rewritten in the compact form

$$\xi^T \tilde{P} \xi \geq \tau_2 \tag{9}$$

being

$$\begin{aligned} \tilde{P} =& P_0 - \phi^T P_0 \phi - \tau_1 \Sigma_q^T \Sigma_q - \\ & - \begin{bmatrix} H_D^T P_0 \phi \\ B_p^T P_0 \phi \end{bmatrix}^T \begin{bmatrix} -H_D^T P_0 H_D + \tau_2 M_d & -H_D^T P_0 B_p \\ * & -B_p^T P_0 B_p + \tau_1 I \end{bmatrix}^{-1} \begin{bmatrix} H_D^T P_0 \phi \\ B_p^T P_0 \phi \end{bmatrix} \end{aligned} \tag{10}$$

If $\tilde{P} \geq 0$, eq. (9) represents the outer region of the ellipsoid

$$\Gamma_1 = \{\xi : \xi^T \tilde{P} \xi < \tau_2\} \tag{11}$$

According to Schur's complement, $\tilde{P} \geq 0$ if the following LMI holds

$$\begin{bmatrix} P_0 - \phi^T P_0 \phi - \tau_1 \Sigma_q^T \Sigma_q & \phi^T P_0 H_D & \phi^T P_0 B_p \\ * & -H_D^T P_0 H_D + \tau_2 M_d & -H_D^T P_0 B_p \\ * & * & -B_p^T P_0 B_p + \tau_1 I \end{bmatrix} \geq 0 \tag{12}$$

To minimize the volume of Γ_1, solve the following GEVP problem

$$\min_{\tau_2} \tau_2 \tag{13}$$

$$\tau_1 \geq 0 \tag{14}$$

$$\tau_2 \geq 0 \tag{15}$$

s.t.

$$\begin{bmatrix} P_0 - \phi^T P_0 \phi - \tau_1 \Sigma_q^T \Sigma_q & \phi^T P_0 H_D & \phi^T P_0 B_p \\ * & -H_D^T P_0 H_D + \tau_2 M_d & -H_D^T P_0 B_p \\ * & * & -B_p^T P_0 B_p + \tau_1 I \end{bmatrix} \geq 0 \tag{16}$$

$$\begin{bmatrix} -H_D^T P_0 H_D + \tau_2 M_d & -H_D^T P_0 B_p \\ * & -B_p^T P_0 B_p + \tau_1 I \end{bmatrix} > 0 \tag{17}$$

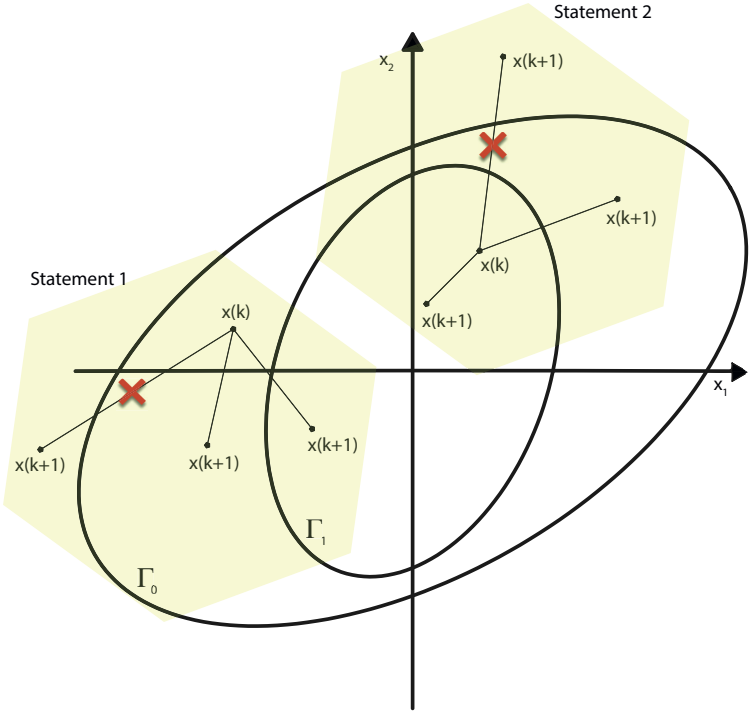


Fig. 1: Graphical representation of the two statements in a generic two-dimensional state example. $x(k)$ represents the current state whereas $x(k+1)$ the successive. x_1 and x_2 are the two components of vector x.

To impose that Γ_1 is internal to Γ_0, see Fig. 1, the **Statement 1** must be fulfilled

$$\tilde{P} \geq \tau_2 P_0 \tag{18}$$

Finally, to prove that Γ_0 is robust positively invariant even in presence of disturbance d the following **Statement 2** must be held

$$V(k+1) \leq 1, \forall \xi : \xi^T \tilde{P} \xi < \tau_2, \forall d : d^T M_d d \leq 1 \tag{19}$$

see Fig. 1. Be $\chi = \begin{bmatrix} 1 & \xi^T & d^T & p^T \end{bmatrix}^T$, eq. (19) can be rewritten as

$$\chi^T \begin{bmatrix} 1 & 0 & 0 & 0 \\ * & -\phi^T P_0 \phi & -\phi^T P_0 H_D & -\phi^T P_0 B_p \\ * & * & -H_D^T P_0 H_D & -H_D^T P_0 B_p \\ * & * & * & -B_p^T P_0 B_p \end{bmatrix} \chi \geq 0 \tag{20}$$

Condition $\xi^T \tilde{P} \xi \leq \tau_2$ can be rewritten as

$$\chi^T \begin{bmatrix} \tau_2 & 0 & 0 & 0 \\ * & -\tilde{P} & 0 & 0 \\ * & * & 0 & 0 \\ * & * & * & 0 \end{bmatrix} \chi \geq 0 \tag{21}$$

Condition $d^T M_d d \leq 1$ can be rewritten as

$$\chi^T \begin{bmatrix} 1 & 0 & 0 & 0 \\ * & 0 & 0 & 0 \\ * & * & -M_d & 0 \\ * & * & * & 0 \end{bmatrix} \chi \geq 0 \tag{22}$$

Moreover, condition $p^T p \leq \xi^T \Sigma_q^T \Sigma_q \xi$ can be rewritten as

$$\chi^T \begin{bmatrix} 0 & 0 & 0 & 0 \\ * & \Sigma_q^T \Sigma_q & 0 & 0 \\ * & * & 0 & 0 \\ * & * & * & -I \end{bmatrix} \chi \geq 0 \tag{23}$$

According to S-procedure, (19) is robustly satisfied if there is a solution to the following feasibility problem

$$\sigma_1 \geq 0 \tag{24}$$

$$\sigma_2 \geq 0 \tag{25}$$

$$\sigma_3 \geq 0 \tag{26}$$

$$1 - \sigma_1 \tau_2 - \sigma_2 \geq 0 \tag{27}$$

$$\begin{bmatrix} -\phi^T P_0 \phi + \sigma_1 \tilde{P} - \sigma_3 \Sigma_q^T \Sigma_q & -\phi^T P_0 H_D & -\phi^T P_0 B_p \\ * & -H_D^T P_0 H_D + \sigma_2 M_d & -H_D^T P_0 B_p \\ * & * & -B_p^T P_0 B_p + \sigma_3 I \end{bmatrix} \geq 0 \tag{28}$$

Proof of Lemma 2

Proof. Let $\alpha = \begin{bmatrix} 1 & \xi_k^T & d_k^T & p_k^T \end{bmatrix}^T$. The condition (3.63) can be rewritten in the following convenient form

$$\alpha^T \begin{bmatrix} 1 & 0 & 0 & 0 \\ * & -\phi^T P_{k+1}\phi & -\phi^T P_{k+1} H_D & -\phi^T P_{k+1} B_p \\ * & * & -H_D^T P_{k+1} H_D & -H_D^T P_{k+1} B_p \\ * & * & * & -B_p^T P_{k+1} B_p \end{bmatrix} \alpha \geq 0 \tag{.29}$$

The condition (3.62) can be rewritten as

$$\alpha^T \begin{bmatrix} 1 & 0 & 0 & 0 \\ * & -P_k & 0 & 0 \\ * & * & 0 & 0 \\ * & * & * & 0 \end{bmatrix} \alpha \geq 0 \tag{.30}$$

The condition (3.43) becomes

$$\alpha^T \begin{bmatrix} 1 & 0 & 0 & 0 \\ * & 0 & 0 & 0 \\ * & * & -M_d & 0 \\ * & * & * & 0 \end{bmatrix} \alpha \geq 0 \tag{.31}$$

Finally, from norm-bound uncertain representation (3.34)-(3.36), the condition $p_k^T p_k \leq \xi_k^T \Sigma_q^T \Sigma_q \xi_k$ can be rewritten as

$$\alpha^T \begin{bmatrix} 0 & 0 & 0 & 0 \\ * & \Sigma_q^T \Sigma_q & 0 & 0 \\ * & * & 0 & 0 \\ * & * & * & -I \end{bmatrix} \alpha \geq 0 \tag{.32}$$

Recalling the arguments of the S-procedure [109], the smallest region (3.63) can be determined by solving the minimization problem (3.64)-(3.67).

References

1. M. Hoy, A. S. Matveev, and A. V. Savkin, "Algorithms for collision-free navigation of mobile robots in complex cluttered environments: a survey," *Robotica*, vol. 33, no. 3, pp. 463–497, 2015.
2. R. Chai, K. Chen, B. Hua, Y. Lu, Y. Xia, X.-M. Sun, G.-P. Liu, and W. Liang, "A two phases multiobjective trajectory optimization scheme for multi-ugvs in the sight of the first aid scenario," *IEEE Transactions on Cybernetics*, vol. 54, no. 9, pp. 5078–5091, 2024.
3. D. T. Fasiolo, L. Scalera, E. Maset, and A. Gasparetto, "Towards autonomous mapping in agriculture: A review of supportive technologies for ground robotics," *Robotics and Autonomous Systems*, vol. 169, p. 104514, 2023.
4. T. d. J. M. Sanguino, "50 years of rovers for planetary exploration: A retrospective review for future directions," *Robotics and Autonomous Systems*, vol. 94, pp. 172–185, 2017.
5. P. P.-Y. Wu, D. Campbell, and T. Merz, "Multi-objective four-dimensional vehicle motion planning in large dynamic environments," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 41, no. 3, pp. 621–634, 2010.
6. M. Shahriari and M. Biglarbegian, "Toward safer navigation of heterogeneous mobile robots in distributed scheme: A novel time-to-collision-based method," *IEEE transactions on cybernetics*, vol. 52, no. 9, pp. 9302–9315, 2021.
7. E. A. Keller and D. E. DeVecchio, *Natural hazards: earth's processes as hazards, disasters, and catastrophes*. Routledge, 2019.
8. R. R. Murphy, "Activities of the rescue robots at the world trade center from 11-21 september 2001," *IEEE Robotics & Automation Magazine*, vol. 11, no. 3, pp. 50–61, 2004.
9. B. Doroodgar, Y. Liu, and G. Nejat, "A learning-based semi-autonomous controller for robotic exploration of unknown disaster scenes while searching for victims," *IEEE Transactions on Cybernetics*, vol. 44, no. 12, pp. 2719–2732, 2014.
10. S. M. LaValle, *Planning algorithms*. Cambridge university press, 2006.
11. P. E. Hart, N. J. Nilsson, and B. Raphael, "A formal basis for the heuristic determination of minimum cost paths," *IEEE transactions on Systems Science and Cybernetics*, vol. 4, no. 2, pp. 100–107, 1968.
12. A. Nash, K. Daniel, S. Koenig, and A. Felner, "Theta^*: Any-angle path planning on grids," in *AAAI*, vol. 7, pp. 1177–1183, 2007.

13. D. Dolgov, S. Thrun, M. Montemerlo, and J. Diebel, "Path planning for autonomous vehicles in unknown semi-structured environments," *The international journal of robotics research*, vol. 29, no. 5, pp. 485–501, 2010.
14. H. Liu, G. Wu, L. Zhou, W. Pedrycz, and P. N. Suganthan, "Tangent-based path planning for uav in a 3-d low altitude urban environment," *IEEE Transactions on Intelligent Transportation Systems*, vol. 24, no. 11, pp. 12062–12077, 2023.
15. S. M. LaValle and J. J. Kuffner Jr, "Randomized kinodynamic planning," *The international journal of robotics research*, vol. 20, no. 5, pp. 378–400, 2001.
16. S. Karaman and E. Frazzoli, "Sampling-based algorithms for optimal motion planning," *The international journal of robotics research*, vol. 30, no. 7, pp. 846–894, 2011.
17. S. S. Ge and Y. J. Cui, "New potential functions for mobile robot path planning," *IEEE Transactions on Robotics and Automation*, vol. 16, no. 5, pp. 615–620, 2000.
18. Z. Qu, J. Wang, and C. E. Plaisted, "A new analytical solution to mobile robot trajectory generation in the presence of moving obstacles," *IEEE transactions on robotics*, vol. 20, no. 6, pp. 978–993, 2004.
19. A. Zamuda, J. D. H. Sosa, and L. Adler, "Constrained differential evolution optimization for underwater glider path planning in sub-mesoscale eddy sampling," *Applied Soft Computing*, vol. 42, pp. 93–118, 2016.
20. M. Neshat, G. Sepidnam, M. Sargolzaei, and A. N. Toosi, "Artificial fish swarm algorithm: A survey of the state-of-the-art, hybridization, combinatorial and indicative applications," *Artificial Intelligence Review*, vol. 42, no. 4, pp. 965–997, 2014.
21. J. Chen, W. Zhan, and M. Tomizuka, "Autonomous driving motion planning with constrained iterative lqr," *IEEE Transactions on Intelligent Vehicles*, vol. 4, no. 2, pp. 244–254, 2019.
22. H. Kurniawati, Y. Du, D. Hsu, and W. S. Lee, "Motion planning under uncertainty for robotic tasks with long time horizons," *International Journal of Robotics Research*, vol. 30, no. 3, pp. 308–323, 2011.
23. C. Danielson, K. Berntorp, A. Weiss, and S. Di Cairano, "Robust motion planning for uncertain systems with disturbances using the invariant-set motion planner," *IEEE Transactions on Automatic Control*, vol. 65, no. 10, pp. 4456–4463, 2020.
24. Y. Ning, M. Yue, J. Shangguan, and J. Zhao, "Optimal trajectory planning method for the navigation of wip vehicles in unknown environments: Theory and experiment," *IEEE Transactions on Cybernetics*, vol. 53, no. 10, pp. 6317–6328, 2022.
25. V. Scordamaglia, V. A. Nardi, and A. Ferraro, "A feasible trajectory planning algorithm for a network controlled robot subject to skid and slip phenomena," in *2019 24th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)*, pp. 933–940, IEEE, 2019.
26. Z. Wang, A. Ahmad, R. Quirynen, Y. Wang, A. Bhagat, E. Zeino, Y. Zushi, and S. Di Cairano, "Motion planning and model predictive control for automated tractor-

trailer hitching maneuver," in *2022 IEEE Conference on Control Technology and Applications (CCTA)*, pp. 676–682, IEEE, 2022.

27. C. Danielson, K. Berntorp, S. Di Cairano, and A. Weiss, "Motion-planning for unicycles using the invariant-set motion-planner," in *2020 American Control Conference (ACC)*, pp. 1235–1240, IEEE, 2020.
28. C. Danielson, A. Weiss, K. Berntorp, and S. Di Cairano, "Path planning using positive invariant sets," in *2016 IEEE 55th Conference on Decision and Control (CDC)*, pp. 5986–5991, IEEE, 2016.
29. A. Weiss, C. Danielson, K. Berntorp, I. Kolmanovsky, and S. Di Cairano, "Motion planning with invariant set trees," in *2017 IEEE Conference on Control Technology and Applications (CCTA)*, pp. 1625–1630, IEEE, 2017.
30. R. Marino, S. Scalzi, and M. Netto, "Nested pid steering control for lane keeping in autonomous vehicles," *Control Engineering Practice*, vol. 19, no. 12, pp. 1459–1467, 2011.
31. S. X. Yang, A. Zhu, G. Yuan, and M. Q. Meng, "A bioinspired neurodynamics-based approach to tracking control of mobile robots," *IEEE Transactions on Industrial Electronics*, vol. 59, no. 8, pp. 3211–3220, 2012.
32. H. Pan, W. Sun, H. Gao, and X. Jing, "Disturbance observer-based adaptive tracking control with actuator saturation and its application," *IEEE Transactions on Automation Science and Engineering*, vol. 13, no. 2, pp. 868–875, 2016.
33. C. Hu, H. Jing, R. Wang, F. Yan, and M. Chadli, "Robust h_∞ output-feedback control for path following of autonomous ground vehicles," *Mechanical Systems and Signal Processing*, vol. 70-71, pp. 414–427, 2016.
34. W. Tsui, M. S. Masmoudi, F. Karray, I. Song, and M. Masmoudi, "Softcomputing-based embedded design of an intelligent wall/lane-following vehicle," *IEEE/ASME Transactions on Mechatronics*, vol. 13, no. 1, pp. 125–135, 2008.
35. T. M. Howard, C. J. Green, and A. Kelly, "Receding horizon model-predictive control for mobile robot navigation of intricate paths," in *Field and Service Robotics, 7th International Conference*, (Berlin, Heidelberg), pp. 69–78, Springer Berlin Heidelberg, 2010.
36. J. Ji, A. Khajepour, W. W. Melek, and Y. Huang, "Path planning and tracking for vehicle collision avoidance based on model predictive control with multiconstraints," *IEEE Transactions on Vehicular Technology*, vol. 66, no. 2, pp. 952–964, 2016.
37. E. Stefanini, L. Palmieri, A. Rudenko, T. Hielscher, T. Linder, and L. Pallottino, "Efficient context-aware model predictive control for human-aware navigation," *IEEE Robotics and Automation Letters*, 2024.
38. D. Q. Mayne, E. C. Kerrigan, E. J. vanWyk, and P. Falugi, "Tube-based robust nonlinear model predictive control," *International Journal of Robust and Nonlinear Control*, vol. 21, no. 11, pp. 1341–1353, 2011.
39. Q. Zou, Q. Sun, L. Chen, B. Nie, and Q. Li, "A comparative analysis of lidar slam-based indoor navigation for autonomous vehicles," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 7, pp. 6907–6921, 2021.

40. S. T. Arzo, D. Sikeridis, M. Devetsikiotis, F. Granelli, R. Fierro, M. Esmaeili, and Z. Akhavan, "Essential technologies and concepts for massive space exploration: Challenges and opportunities," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 59, no. 1, pp. 3–29, 2022.
41. X. M. Zhang, Q. L. Han, X. Ge, D. Ding, L. Ding, D. Yue, and C. Peng, "Networked control systems: A survey of trends and techniques," *IEEE/CAA Journal of Automatica Sinica*, vol. 7, no. 1, pp. 1–17, 2019.
42. K. Masaba and A. Q. Li, "Gvgexp: Communication-constrained multi-robot exploration system based on generalized voronoi graphs," in *2021 International Symposium on Multi-Robot and Multi-Agent Systems (MRS)*, pp. 146–154, IEEE, 2021.
43. L. Zhang, Z. Zhang, R. Siegwart, and J. J. Chung, "Distributed pdop coverage control: Providing large-scale positioning service using a multi-robot system," *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 2217–2224, 2021.
44. N. Bartolini, A. Coletta, and G. Maselli, "On task assignment for early target inspection in squads of aerial drones," in *2019 IEEE 39th International Conference on Distributed Computing Systems (ICDCS)*, pp. 2123–2133, IEEE, 2019.
45. A. Khamis, A. Hussein, and A. Elmogy, "Multi-robot task allocation: A review of the state-of-the-art," *Cooperative robots and sensor networks 2015*, pp. 31–51, 2015.
46. J. Leitner, "Multi-robot cooperation in space: A survey," *2009 Advanced Technologies for Enhanced Quality of Life*, pp. 144–151, 2009.
47. J. P. Queralta, J. Taipalmaa, B. C. Pullinen, V. K. Sarker, T. N. Gia, H. Tenhunen, M. Gabbouj, J. Raitoharju, and T. Westerlund, "Collaborative multi-robot systems for search and rescue: Coordination and perception," *arXiv preprint arXiv:2008.12610*, 2020.
48. X. Dai, L. Jiang, and Y. Zhao, "Cooperative exploration based on supervisory control of multi-robot systems," *Applied Intelligence*, vol. 45, pp. 18–29, 2016.
49. C. Yu, B. D. 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, 2009.
50. M. Ji and M. Egerstedt, "Distributed coordination control of multiagent systems while preserving connectedness," *IEEE Transactions on Robotics*, vol. 23, no. 4, pp. 693–703, 2007.
51. J.-M. McNew, E. Klavins, and M. Egerstedt, "Solving coverage problems with embedded graph grammars," in *Hybrid Systems: Computation and Control: 10th International Workshop, HSCC 2007, Pisa, Italy, April 3-5, 2007. Proceedings 10*, pp. 413–427, Springer, 2007.
52. J. Cortes, S. Martinez, and F. Bullo, "Spatially-distributed coverage optimization and control with limited-range interactions," *ESAIM: Control, Optimisation and Calculus of Variations*, vol. 11, no. 4, pp. 691–719, 2005.

53. S. Susca, F. Bullo, and S. Martinez, "Monitoring environmental boundaries with a robotic sensor network," *IEEE Transactions on Control Systems Technology*, vol. 16, no. 2, pp. 288–296, 2008.
54. N. Michael and V. Kumar, "Controlling shapes of ensembles of robots of finite size with nonholonomic constraints," *Proceedings of Robotics: Science and Systems IV*, 2008.
55. Z. Yan, N. Jouandeau, and A. A. Cherif, "A survey and analysis of multi-robot coordination," *International Journal of Advanced Robotic Systems*, vol. 10, no. 12, p. 399, 2013.
56. L. Liu, X. Wang, X. Yang, H. Liu, J. Li, and P. Wang, "Path planning techniques for mobile robots: Review and prospect," *Expert Systems with Applications*, vol. 227, p. 120254, 2023.
57. Q. Yao, Y. Tian, Q. Wang, and S. Wang, "Control strategies on path tracking for autonomous vehicle: State of the art and future challenges," *IEEE Access*, vol. 8, pp. 161211–161222, 2020.
58. J. R. Sanchez-Ibanez, C. J. Pérez-del Pulgar, and A. García-Cerezo, "Path planning for autonomous mobile robots: A review," *Sensors*, vol. 21, no. 23, p. 7898, 2021.
59. A. Madridano, A. Al-Kaff, D. Martín, and A. De La Escalera, "Trajectory planning for multi-robot systems: Methods and applications," *Expert Systems with Applications*, vol. 173, p. 114660, 2021.
60. X. Bai, W. Yan, M. Cao, and D. Xue, "Distributed multi-vehicle task assignment in a time-invariant drift field with obstacles," *IET Control Theory & Applications*, vol. 13, no. 17, pp. 2886–2893, 2019.
61. A. Erokhin, V. Erokhin, S. Sotnikov, and A. Gogolevsky, "Optimal multi-robot path finding algorithm based on a," in *Proceedings of the Computational Methods in Systems and Software*, pp. 172–182, Springer, 2018.
62. L. E. Kavraki, P. Svestka, J.-C. Latombe, and M. H. Overmars, "Probabilistic roadmaps for path planning in high-dimensional configuration spaces," *IEEE transactions on Robotics and Automation*, vol. 12, no. 4, pp. 566–580, 1996.
63. R. Cui, Y. Li, and W. Yan, "Mutual information-based multi-auv path planning for scalar field sampling using multidimensional rrt," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 46, no. 7, pp. 993–1004, 2015.
64. O. Khatib, "Real-time obstacle avoidance for manipulators and mobile robots," in *Autonomous robot vehicles*, pp. 396–404, Springer, 1986.
65. Z. Ying and L. Xu, "Leader-follower formation control and obstacle avoidance of multi-robot based on artificial potential field," in *The 27th Chinese Control and Decision Conference (2015 CCDC)*, pp. 4355–4360, IEEE, 2015.
66. S. S. Ge and Y. J. Cui, "Dynamic motion planning for mobile robots using potential field method," *Autonomous robots*, vol. 13, pp. 207–222, 2002.
67. R. Olfati-Saber, J. A. Fax, and R. M. Murray, "Consensus and cooperation in networked multi-agent systems," *Proceedings of the IEEE*, vol. 95, no. 1, pp. 215–233, 2007.

68. R. C. Arkin, *Behavior-based robotics*. MIT press, 1998.

69. A. D. Ames, X. Xu, J. W. Grizzle, and P. Tabuada, "Control barrier function based quadratic programs for safety critical systems," *IEEE Transactions on Automatic Control*, vol. 62, no. 8, pp. 3861–3876, 2016.

70. D. Fox, W. Burgard, and S. Thrun, "The dynamic window approach to collision avoidance," *IEEE Robotics & Automation Magazine*, vol. 4, no. 1, pp. 23–33, 1997.

71. S. Quinlan and O. Khatib, "Elastic bands: Connecting path planning and control," in *[1993] Proceedings IEEE International Conference on Robotics and Automation*, pp. 802–807, IEEE, 1993.

72. G. Antonelli, F. Arrichiello, and S. Chiaverini, "The null-space-based behavioral control for autonomous robotic systems," *Intelligent Service Robotics*, vol. 1, no. 1, pp. 27–39, 2008.

73. M. Mattei and V. Scordamaglia, "Task priority approach to the coordinated control of a team of flying vehicles in the presence of obstacles," *IET Control Theory & Applications*, vol. 6, no. 13, pp. 2103–2110, 2012.

74. I. Ulrich and J. Borenstein, "Vfh/sup*: Local obstacle avoidance with look-ahead verification," in *Proceedings 2000 ICRA. Millennium Conference. IEEE International Conference on Robotics and Automation. Symposia Proceedings (Cat. No. 00CH37065)*, vol. 3, pp. 2505–2511, IEEE, 2000.

75. R. Simmons, "The curvature-velocity method for local obstacle avoidance," in *Robotics and Automation, 1996. Proceedings., 1996 IEEE International Conference on*, vol. 4, pp. 3375–3382, IEEE, 1996.

76. K. Kant and S. W. Zucker, "Toward efficient trajectory planning: The path-velocity decomposition," *The international journal of robotics research*, vol. 5, no. 3, pp. 72–89, 1986.

77. K. Kant and S. W. Zucker, "Planning collision-free trajectories in time-varying environments: a two-level hierarchy," *The Visual Computer*, vol. 3, no. 5, pp. 304–313, 1988.

78. S. Liu, J. Shen, W. Tian, J. Lin, P. Li, and B. Li, "Balanced task allocation and collision-free scheduling of multi-robot systems in large spacecraft structure manufacturing," *Robotics and Autonomous Systems*, vol. 159, p. 104289, 2023.

79. X. Wang, M. Kloetzer, C. Mahulea, and M. Silva, "Collision avoidance of mobile robots by using initial time delays," in *2015 54th IEEE Conference on Decision and Control (CDC)*, pp. 324–329, IEEE, 2015.

80. J. Yi, H. Wang, J. Zhang, D. Song, S. Jayasuriya, and J. Liu, "Kinematic modeling and analysis of skid-steered mobile robots with applications to low-cost inertial-measurement-unit-based motion estimation," *IEEE transactions on robotics*, vol. 25, no. 5, pp. 1087–1097, 2009.

81. A. Al-Jarrah, M. Salah, and F. Almomani, "Controlling a skid-steered tracked mobile robot with slippage using various control schemes," in *2019 20th International Conference on Research and Education in Mechatronics (REM)*, pp. 1–7, IEEE, 2019.

82. S. A. A. Moosavian and A. Kalantari, "Experimental slip estimation for exact kinematics modeling and control of a tracked mobile robot," in *2008 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 95–100, IEEE, 2008.

83. V. Scordamaglia, A. Ferraro, F. Tedesco, and G. Franzè, "Set-theoretic approach for autonomous tracked vehicles involved in post-disaster first relief operations," in *2024 10th International Conference on Control, Decision and Information Technologies (CoDIT)*, pp. 2006–2011, IEEE, 2024.

84. N. S. Krishna, A. George, A. John, and A. Sudheer, "Controller design for a skid-steered robot and mapping for surveillance applications," in *Proceedings of the 2017 3rd International Conference on Advances in Robotics*, pp. 1–7, 2017.

85. J. L. Martínez, A. Mandow, J. Morales, S. Pedraza, and A. García-Cerezo, "Approximating kinematics for tracked mobile robots," *The International Journal of Robotics Research*, vol. 24, no. 10, pp. 867–878, 2005.

86. J. Y. Wong, *Theory of ground vehicles*. John Wiley & Sons, 2022.

87. Y. Tian and N. Sarkar, "Control of a mobile robot subject to wheel slip," *Journal of Intelligent & Robotic Systems*, vol. 74, no. 3, pp. 915–929, 2014.

88. S. Al-Milli, L. D. Seneviratne, and K. Althoefer, "Track–terrain modelling and traversability prediction for tracked vehicles on soft terrain," *Journal of Terramechanics*, vol. 47, no. 3, pp. 151–160, 2010.

89. I. Zohar, A. Ailon, and R. Rabinovici, "Mobile robot characterized by dynamic and kinematic equations and actuator dynamics: Trajectory tracking and related application," *Robotics and Autonomous Systems*, vol. 59, no. 6, pp. 343–353, 2011.

90. D. Wang and C. B. Low, "Modeling and analysis of skidding and slipping in wheeled mobile robots: Control design perspective," *IEEE Transactions on Robotics*, vol. 24, no. 3, pp. 676–687, 2011.

91. D. Lhomme-Desages, C. Grand, and J. Guinot, "Trajectory control of a four-wheel skid-steering vehicle over soft terrain using physical interaction model," in *Proceedings of the IEEE International Conference on Robotics and Au- tomation*, 2007.

92. Z. Zong, Y. H. Zweiri, and L. D. Seneviratne, "Non-linear observer for slip estimation of skid-steering vehicles," in *Proceedings of the IEEE International Conference on Robotics and Automation*, 2006.

93. I. Motte and G. Campion, "A slow manifold approach for the control of mobile robots not satisfying the kinematic constraints," *IEEE Transactions on Robotics and Automation*, vol. 16, no. 6, pp. 875–880, 2000.

94. S. T. Peng, J. Sheu, and C. C. Chang, "On one approach to constraining wheel slip for the autonomus control of a 4ws/4wd," in *Proceedings of the International Conference on Control Applications*, 2004.

95. M. Michalek, P. Dutkiewicz, M. Kielczewski, and D. Pazderski, "Trajectory tracking for a mobile robot with skid-slip compensation in the vector-field orientation control system," *Int. J. Appl. Math. Comput. Sci.*, vol. 19, no. 4, pp. 547–559, 2009.

96. M. Corradini, T. Leo, and G. Orlando, "obust stabilization of a mobile robot violating the nonholonomic constraint via quasi-sliding modes," in *Proceedings of the Amer- ican Control Conference*, 1999.

97. R. Lenain, B. Thuilot, C. Cariou, and P. Martinet, "High accuracy path tracking for vehicles in presence od sliding: Application to farm vehicle automatic guidance for agricultural tasks," *Autonomous Robots*, vol. 21, no. 1, pp. 79–97, 2006.

98. A. Mandow, J. L. Martinez, J. Morales, J. L. Blanco, A. Garcia-Cerezo, and J. Gonzalez, "Experimental kinematics for wheeled skid-steer mobile robots," in *Proceedings of the International Conference on Intelligent Robots and Systems*, 2007.

99. A. De Luca, G. Oriolo, and M. Vendittelli, "Control of wheeled mobile robots: An experimental overview," *RAMSETE: articulated and mobile robotics for services and technologies*, pp. 181–226, 2002.

100. M. Reinstein, V. Kubelka, and K. Zimmermann, "Terrain adaptive odometry for mobile skid-steer robots," in *2013 IEEE International Conference on Robotics and Automation*, pp. 4706–4711, May 2013.

101. H. E. Sevil, P. Desai, A. Dogan, and B. Huff, "Modeling of an unmanned ground vehicle for autonomous navigation and obstacle avoidance simulations," in *ASME 2012 5th Annual Dynamic Systems and Control Conference joint with the JSME 2012 11th Motion and Vibration Conference*, vol. 1, pp. 529–534, American Society of Mechanical Engineers Digital Collection, 10 2012.

102. S. Dogru and L. Marques, "Estimation of rotational speeds of skid-steered wheeled mobile robots using an improved kinematic model," in *2017 IEEE International Conference on Autonomous Robot Systems and Competitions (ICARSC)*, pp. 73–78, April 2017.

103. H. Zhang, H. Yang, H. Zhao, and H. Cheng, "Networked trajectory tracking control for a nonlinear wheeled mobile robot," in *2021 IEEE International Conference on Unmanned Systems (ICUS)*, pp. 45–50, IEEE, 2021.

104. A. Bemporad, M. Heemels, M. Johansson, *et al.*, *Networked control systems*, vol. 406. Springer, 2010.

105. G. C. Walsh, O. Beldiman, and L. G. Bushnell, "Asymptotic behavior of nonlinear networked control systems," *IEEE transactions on automatic control*, vol. 46, no. 7, pp. 1093–1097, 2002.

106. V. Scordamaglia and V. A. Nardi, "A set-based trajectory planning algorithm for a network controlled skid-steered tracked mobile robot subject to skid and slip phenomena," *Journal of Intelligent & Robotic Systems*, vol. 101, no. 1, pp. 1–20, 2021.

107. W. P. Heemels, N. Van De Wouw, R. H. Gielen, M. Donkers, L. Hetel, S. Olaru, M. Lazar, J. Daafouz, and S. Niculescu, "Comparison of overapproximation methods for stability analysis of networked control systems," in *Proceedings of the 13th ACM international conference on Hybrid systems: computation and control*, pp. 181–190, 2010.

108. A. Ferraro, V. A. Nardi, and V. Scordamaglia, "Mathematical model of a remotely controlled skid-slip tracked mobile robot," *arXiv preprint arXiv:2303.03033*, 2023.

109. S. Boyd, L. El Ghaoui, E. Feron, and V. Balakrishnan, *Linear matrix inequalities in system and control theory*. SIAM, 1994.

110. F. Blanchini, "Constrained control for systems with unknown disturbances," in *Robust Control System Techniques and Applications, Part 2 of 2* (C. LEONDES, ed.), vol. 51 of *Control and Dynamic Systems*, pp. 129–182, Academic Press, 1992.

111. I. Kolmanovsky, E. G. Gilbert, *et al.*, "Theory and computation of disturbance invariant sets for discrete-time linear systems," *Mathematical problems in engineering*, vol. 4, pp. 317–367, 1998.

112. R. Dechter and J. Pearl, "Generalized best-first search strategies and the optimality of a," *Journal of the ACM (JACM)*, vol. 32, no. 3, pp. 505–536, 1985.

113. V. A. Nardi, A. Ferraro, and V. Scordamaglia, "Feasible trajectory planning algorithm for a skid-steered tracked mobile robot subject to skid and slip phenomena," in *2018 23rd International Conference on Methods & Models in Automation & Robotics (MMAR)*, pp. 120–125, IEEE, 2018.

114. V. Scordamaglia, A. Ferraro, and G. Franzè, "Autonomous tracked vehicles operating in cluttered and unknown environments: A networked set-theoretic receding horizon control strategy," *IEEE Transactions on Cybernetics*, 2025.

115. F. Blanchini, S. Miani, *et al.*, *Set-theoretic methods in control*, vol. 78. Springer, 2008.

116. G. Franzè, F. Tedesco, and D. Famularo, "Model predictive control for constrained networked systems subject to data losses," *Automatica*, vol. 54, pp. 272–278, 2015.

117. A. Sadeghzadeh and R. Tóth, "Improved embedding of nonlinear systems in linear parameter-varying models with polynomial dependence," *IEEE Transactions on Control Systems Technology*, vol. 31, no. 1, pp. 70–82, 2022.

118. L. A. Montestruque and P. Antsaklis, "Stability of model-based networked control systems with time-varying transmission times," *IEEE Transactions on Automatic Control*, vol. 49, no. 9, pp. 1562–1572, 2004.

119. V. Yakubovich, "Nonconvex optimization problem: The infinite-horizon linear-quadratic control problem with quadratic constraints," *Systems & Control Letters*, vol. 19, no. 1, pp. 13–22, 1992.

120. J. G. VanAntwerp and R. D. Braatz, "A tutorial on linear and bilinear matrix inequalities," *Journal of process control*, vol. 10, no. 4, pp. 363–385, 2000.

121. G. Franze and W. Lucia, "A receding horizon control strategy for autonomous vehicles in dynamic environments," *IEEE Transactions on Control Systems Technology*, vol. 24, no. 2, pp. 695–702, 2015.

122. S. Karaman, M. R. Walter, A. Perez, E. Frazzoli, and S. Teller, "Anytime motion planning using the rrt," in *2011 IEEE international conference on robotics and automation*, pp. 1478–1483, ieee, 2011.

123. Z. Xuexi, L. Guokun, F. Genping, X. Dongliang, and L. Shiliu, "Slam algorithm analysis of mobile robot based on lidar," in *2019 Chinese Control Conference (CCC)*, pp. 4739–4745, IEEE, 2019.

124. W. Lucia, D. Famularo, and G. Franze, "A set-theoretic reconfiguration feedback control scheme against simultaneous stuck actuators," *IEEE Transactions on Automatic Control*, vol. 63, no. 8, pp. 2558–2565, 2017.

125. V. Scordamaglia and A. Ferraro, "A set-based method for planning coordinated trajectories for skid-steered robotic units subject to constraints, uncertainties and external disturbances," *IEEE Access*, 2025.

126. A. Ferraro, C. De Capua, and V. Scordamaglia, "Set-theoretic time-based trajectory synchronization approach for skid-steered robotic units subject to constraints, uncertainties, and external disturbances," in *2025 10th International Conference on Control, Decision and Information Technologies (CoDIT)*, pp. –, IEEE, 2025.

SOMMARIO
L'uso sempre più diffuso di robot autonomi in scenari quali sorveglianza, ricerca e soccorso e logistica richiede strategie di navigazione che garantiscano sicurezza, robustezza e fattibilità anche in presenza di vincoli di attuazione rigorosi, disturbi esterni, incertezze di modello e ritardi di comunicazione.
Questa ricerca si concentra sullo sviluppo di metodologie basate sulla teoria degli insiemi - *set-theoretic* -per la pianificazione della traiettoria e del movimento di robot mobili cingolati a guida differenziale controllati da remoto con l'obiettivo di garantire la feasibility e la sicurezza della navigazione in presenza di incertezze del modello, disturbi esterni e vincoli di attuazione in ambienti complessi.
Vengono proposti due metodi che risolvono il problema della navigazione sicura in presenza di incertezze del modello, disturbi esterni e vincoli di attuazione per il caso di un singolo robot, e due metodi che risolvono il problema della navigazione sicura in presenza di incertezze del modello, disturbi esterni e vincoli di attuazione per sistemi multi-robot, ai quali si aggiungono le sfide cruciali della gestione del coordinamento e della prevenzione delle collisioni.
Tutte le metodologie proposte sono state validate tramite simulazioni numeriche e sperimentali con piattaforme reali disponibili presso il Laboratorio di Controlli Automatici dell'Università Mediterranea di Reggio Calabria.

ABSTRACT
The increasingly widespread use of autonomous mobile robots in scenarios such as surveillance, search and rescue, and logistics requires navigation strategies that guarantee safety, robustness, and feasibility even in the presence of stringent constraints, external disturbances, model uncertainties, and communication delays.
This research focuses on the development of *set-theoretic* methodologies for trajectory and motion planning of remotely controlled skid-steered tracked mobile robots, with the aim of ensuring the feasibility and safety of navigation in the presence of model uncertainties, external disturbances, and actuation constraints in complex environments.
Two methods are proposed that solve the problem of safe navigation in the presence of model uncertainties, external disturbances, and actuation constraints for the single-robot case, and two methods that solve the problem of safe navigation in the presence of model uncertainties, external disturbances, and actuation constraints for multi-robot systems, to which are added the crucial challenges of coordination management and collision prevention to maintain safe and cooperative operations.
All the proposed methodologies have been validated through extensive numerical simulations and real-world experiments using skid-steered tracked mobile robots available at the Automatic Control Laboratory of the Mediterranean University of Reggio Calabria.