



NATIONAL TECHNICAL UNIVERSITY OF ATHENS

PHD DISSERTATION

Distributed Cooperation of Multiple Robots under Operational Constraints via Lean Communication

Author:
Panagiotis VLANTIS

Supervisor:
Prof. Kostas J.
KYRIAKOPOULOS

*A thesis submitted in fulfillment of the requirements
for the degree of Doctor of Philosophy*

in the

Control Systems Laboratory
School of Mechanical Engineering



Operational Programme
Human Resources Development,
Education and Lifelong Learning

Co-financed by Greece and the European Union



This research is co-financed by Greece and the European Union (European Social Fund - ESF) through the Operational Programme «Human Resources Development, Education and Lifelong Learning» in the context of the project “Strengthening Human Resources Research Potential via Doctorate Research” (MIS-5000432), implemented by the State Scholarships Foundation (IKY)

January 2020



NATIONAL TECHNICAL UNIVERSITY OF ATHENS

PHD DISSERTATION

Distributed Cooperation of Multiple Robots under Operational Constraints via Lean Communication

Author:

Panagiotis VLANTIS

Supervisor:

Prof. Kostas J.
KYRIAKOPOULOS

Examination Committee:

Prof. I. Antoniadis
Prof. K. J. Kyriakopoulos
Prof. E. Papadopoulos
Prof. Ch. Sarimveis
Assoc. Prof. C. Tzafestas
Assist. Prof. G. Papalambrou
Prof. N. M. Sgouros

*A thesis submitted in fulfillment of the requirements
for the degree of Doctor of Philosophy*

in the

Control Systems Laboratory
School of Mechanical Engineering

January 2020



ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ

ΔΙΔΑΚΤΟΡΙΚΗ ΔΙΑΤΡΙΒΗ

Κατανεμημένη Συνεργασία Πολλαπλών Ρομπότ υπό Λειτουργικούς Περιορισμούς μέσω Ισχνής Επικοινωνίας

Συγγραφέας:
Παναγιώτης Βλαντής

Επιβλέπων:
Καθ. Κώστας Κυριακόπουλος

Η διατριβή υποβάλλεται εκπληρώνοντας τις απαιτήσεις για την απονομή
του Διδακτορικού Διπλώματος

σύμφωνα με το

Εργαστήριο Αυτομάτου Ελέγχου
Σχολή Μηχανολόγων Μηχανικών



Επιχειρησιακό Πρόγραμμα
Ανάπτυξη Ανθρώπινου Δυναμικού,
Εκπαίδευση και Διά Βίου Μάθηση

Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης



Το έργο συγχρηματοδοτείται από την Ελλάδα και την Ευρωπαϊκή Ένωση (Ευρωπαϊκό Κοινωνικό Ταμείο) μέσω του Επιχειρησιακού Προγράμματος «Ανάπτυξη Ανθρώπινου Δυναμικού, Εκπαίδευση και Διά Βίου Μάθηση», στο πλαίσιο της Πράξης «Ενίσχυση του ανθρώπινου ερευνητικού δυναμικού μέσω της υλοποίησης διδακτορικής έρευνας» (MIS-5000432), που υλοποιεί το Ίδρυμα Κρατικών Υποτροφιών (ΙΚΥ)

Ιανουάριος 2020

NATIONAL TECHNICAL UNIVERSITY OF ATHENS

Abstract

School of Mechanical Engineering

Doctor of Philosophy

Distributed Cooperation of Multiple Robots under Operational Constraints via Lean Communication

by Panagiotis VLANTIS

The autonomous operation of robots inside obstacle cluttered environments constitutes an actively studied research topic and autonomous navigation is undeniably a fundamental aspect of it. Moreover, as the tasks that robots are called to fulfill grow in complexity by the day, the employment of multi-agent robotic systems, which generally exhibit higher robustness and versatility than their single-robot counterparts, progressively increases. Thus, the autonomous navigation problems that need to be addressed become more and more challenging, increasing the need for more efficient and robust path and motion planning schemes.

In this dissertation, we employ hierarchical decomposition techniques, coupled with suitably designed adaptive configuration space exploration schemes for building versatile and complete planners, capable of computing “paths” connecting two given configurations, or determine the absence of feasible solutions, after a finite amount of steps. In addition, distributed control laws based on artificial potential fields, which can elegantly realize the generated high-level plans, are devised for addressing the navigation problem within complex, planar workspaces with guaranteed convergence properties. Finally, we demonstrate the efficacy of the proposed methodologies for addressing real-world problems, such as coordinating the motion of multiple robots operating within the same environment, as well as coordinating the transportation of an object by a team of cooperating mobile manipulators.

ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ

Περίληψη

Σχολή Μηχανολόγων Μηχανικών

Διδακτορική Διατριβή

Κατανεμημένη Συνεργασία Πολλαπλών Ρομπότ υπό Λειτουργικούς Περιορισμούς μέσω
Ισχύος Επικοινωνίας

Παναγιώτης Βλαντής

Η αυτόνομη λειτουργία των ρομπότ εντός περίπλοκων χώρων εργασίας αποτελεί ένα επίκαιρο θέμα έρευνας και η αυτόνομη πλοήγηση είναι αναμφισβήτητα ένα θεμελιώδες κομμάτι αυτής. Επιπλέον, καθώς οι εργασίες που τα ρομπότ καλούνται να εκπληρώσουν αυξάνονται σε πολυπλοκότητα μέρα με τη μέρα, η χρήση πολύ-ρομποτικών συστημάτων, τα οποία εμφανίζουν γενικά υψηλότερη ευρωστία και ευελιξία, αυξάνεται προοδευτικά. Ως εκ τούτου, τα προβλήματα αυτόνομης πλοήγησης που πρέπει να επιλυθούν γίνονται όλο και πιο απαιτητικά, αυξάνοντας την ανάγκη για πιο αποτελεσματικά και σθεναρά σχήματα σχεδιασμού πορείας και κίνησης.

Σε αυτή την διατριβή, χρησιμοποιούμε ιεραρχικές μεθόδους αποσύνθεσης, σε συνδυασμό με κατάλληλα σχεδιασμένα συστήματα εξερεύνησης του χώρου διαμόρφωσης, για την κατασκευή ευέλικτων αλγορίθμων, ικανών να βρουν διαδρομές που συνδέουν δύο δεδομένες διαμορφώσεις, ή να καθορίσουν την απουσία εφικτής λύσης, μετά από ένα πεπερασμένο πλήθος βημάτων. Επιπλέον, κατανεμημένοι νόμοι ελέγχου που βασίζονται σε τεχνητά δυναμικά πεδία, οι οποίοι μπορούν να υλοποιήσουν ασφαλώς τις προδιαγραφές υψηλού επιπέδου, σχεδιάζονται για την αντιμετώπιση του προβλήματος πλοήγησης εντός σύνθετων, επίπεδων χώρων εργασίας με εγγυημένες ιδιότητες σύγκλισης. Τέλος, επιδεικνύουμε την αποτελεσματικότητα των προτεινόμενων μεθοδολογιών με την αντιμετώπιση πρακτικών προβλημάτων, όπως το συντονισμό της κίνησης πολλών ρομπότ που λειτουργούν εντός του ίδιου περιβάλλοντος, καθώς και τον συντονισμό της μεταφοράς ενός αντικειμένου από ομάδα συνεργαζόμενων κινούμενων ρομποτικών βραχιόνων.

Acknowledgements

This Ph.D. Dissertation could not have been completed without the support of a lot of people.

Foremost, I would like to truly thank my PhD advisor Prof. Kostas J. Kyriakopoulos for putting his on faith me and giving me the opportunity to work on such a demanding yet interesting topic. His counsel has been an indispensable guide during this journey and I am particularly thankful to him for also granting me the freedom to explore my own directions, however silly they seemed at first.

I would also like to thank my committee members, Prof. Evangelos Papadopoulos, and Prof. Ioannis Antoniadis for standing by ready to offer their support at any and all times.

I cannot thank enough Dr. Charalampos Bechioulis and my dear friend Constantinos Vrohidis for their exceptional contributions to my thesis. Our time together has contributed greatly to shaping my current mentality and attitude to numerous matters. Also, I am especially grateful to Assist. Prof. George C. Karras, for our invaluable discussions have helped me not loose focus of the things that matter.

Additionally, I would like to thank the State Scholarships Foundation (I.K.Y.) for financially supporting this research.

Needless to say, I also want to thank my mother Maria, my father Vasilis, my sister Victoria, and my uncle Ioannis for supporting my choices during these difficult times. Of course, I cannot leave out my close friends Nikos Martsoukos, Michalis Mathioudakis, Ioannis Moraitis, Gikas Thomas, Marios Vasileiadis, Nikos Kavalopoulos, Aggelos Markou and Panagiotis Bitsikas who would always find the time to listen to my problems.

Last but definitely not least, I am especially grateful to my friends and colleagues from the Control Systems Lab Michalis Logothetis, Kostas Alevizos, Sotiris Aspragathos, Panagiotis Grondas, Panos Marandos, Shahab Heshmati-Alamdari and George Zogopoulos-Papaliakos, who filled each day of this journey with witty and funny moments that I will recall upon fondly.

Contents

Abstract	iii
Acknowledgements	vii
1 Introduction	1
1.1 Motivation	1
1.2 State-of-the-Art and Related Work	3
1.3 Outline	7
2 Navigation of Disk-Shaped Robots on the Plane	11
2.1 Introduction	11
2.2 Problem Formulation	12
2.3 Harmonic Maps for Planar Navigation	13
2.4 Control Design	17
2.4.1 Artificial Harmonic Potential Fields	18
2.4.2 Adaptive Laws	19
2.4.3 Stability Analysis	21
2.5 Extensions	22
2.5.1 Unicycle Robot Kinematics	22
2.5.2 Atlas of Harmonic Maps	23
2.6 Simulations and Experimental Results	26
2.6.1 Simulations — Full Workspace Transformation	26
2.6.2 Simulations — Atlas of Harmonic Maps	27
2.6.3 Experiments	27
3 Navigation of Polygonal-Shaped Robots on the Plane	39
3.1 Introduction	39
3.2 Problem Formulation	40
3.3 Control Design	41
3.3.1 Configuration Space Decomposition	41
3.3.2 Hybrid Control Scheme	44
3.3.3 Harmonic-based Navigation Functions and Maps	47
3.4 Stability Analysis	49
3.5 Simulation Results	50
4 Coordinating the Motion of Multiple Independent Disks	55

4.1	Introduction	55
4.2	Problem Formulation	56
4.3	Control Design	57
4.3.1	Configuration Space Decomposition	57
4.3.2	High-Level Planner	60
4.3.3	Velocity Control Law	61
4.4	Simulation Results	63
5	Cooperative Load Transportation	69
5.1	Introduction	69
5.2	Preliminaries	70
5.3	Problem Formulation	71
5.3.1	Mobile Manipulator Kinematics	72
5.3.2	Mobile Manipulator Dynamics	74
5.4	Control Design	74
5.4.1	Configuration Space Decomposition	75
5.4.2	Distributed Control Law	81
	Object's Position	83
	Object's Orientation	84
	Manipulators	84
5.5	Stability Analysis	87
5.6	Simulation Results	90
6	Conclusions and Future Work	103
	Appendices	105
A	Proofs of chapter 2	107
A.1	Proof of Proposition 1	107
A.2	Proof of Proposition 2	108
A.3	Proof of Proposition 3	109
A.4	Proof of Proposition 4	110
A.5	Proof of Proposition 5	110
A.6	Proof of Theorem 2	111
A.7	Proof of Theorem 3	112
A.8	Proof of Theorem 4	113
B	Proofs of chapter 3	115
B.1	Proof of Proposition 6	115
B.2	Proof of Proposition 7	115
B.3	Proof of Proposition 8	116
B.4	Proof of Theorem 5	116
C	List of Publications	119

Bibliography	123
---------------------	------------

Extended Greek Summary	137
-------------------------------	------------

List of Figures

2.1	Mobile robot navigating within a maze.	11
2.2	Transformation of a real workspace onto a punctured disk.	14
2.3	Discretization of an arbitrary workspace's boundary using line segment elements. By convention, the outer boundary is considered to be clockwise oriented whereas inner boundaries as counter-clockwise oriented. The normal direction of each element is depicted using green colored vectors. Furthermore, the values of the two linear shape functions $G_{0,24,1}$ and $G_{0,24,2}$ are plotted along the length of the corresponding element $E_{0,24}$	15
2.4	The partition of a complex workspace into overlapping subsets along with the corresponding graph and the transformation T_2 of the second partition \mathcal{P}_2	24
2.5	Simulation results using a full workspace transformation. The robot starts from an arbitrary location and navigates to different goal configurations (colored crosses). The resulted trajectories are plotted in both real and transformed workspaces with different color.	28
2.6	The evolution of ψ along the robot trajectories.	29
2.7	The resulted robot trajectories using an atlas of the workspace.	29
2.8	Robotnik Summit XL	30
2.9	Map of the robot's workspace used during the experiments, obtained using SLAM.	31
2.10	Trajectory of the unicycle robot in the real workspace executed during the first experimental case.	32
2.11	Trajectory of the unicycle robot in the transformed workspace's partitions executed during the first experimental case.	33
2.12	Trajectory of the unicycle robot in the real workspace executed during the second experimental case.	34
2.13	Trajectory of the unicycle robot in the transformed workspace's partitions executed during the second experimental case.	35
2.14	Trajectory of the unicycle robot in the real workspace executed during the third experimental case.	36
2.15	Trajectory of the unicycle robot in the transformed workspace's partitions executed during the third experimental case.	37

3.1	Team of mobile manipulators forming a virtual, non-convex robotic platform.	39
3.2	$\overline{\mathcal{W}}^{\mathcal{I}}$ (blue colored regions) and $\underline{\mathcal{W}}^{\mathcal{I}}$ (red colored regions) of the free space available to the Y-shaped robot (bottom-right) for $\mathcal{I} = [315^\circ, 360^\circ)$	42
3.3	Workspace consisting of two rooms connected via a corridor. The $\overline{\mathcal{W}}^{\mathcal{I}}$ (blue colored regions) and $\underline{\mathcal{W}}^{\mathcal{I}}$ (red colored regions) were constructed via the Swept-Area method for (a) $\mathcal{I} = [0^\circ, 360^\circ)$ and (b) $\mathcal{I} = [67.5^\circ, 90^\circ)$. Corresponding $\underline{\mathcal{R}}(\mathcal{I})$ and $\overline{\mathcal{R}}(\mathcal{I})$ of the rectangular robot (bottom) can be seen below each workspace.	43
3.4	Possible pairs of consecutive cells in Π	47
3.5	Motion profile executed by the robot. The purple line corresponds to the path traversed by the robot's reference point p . Blue and red poses correspond to robot's initial and desired configurations, respectively, whereas gray and green poses correspond to intermediate configurations captured at regular time intervals.	52
3.6	State of robot over time. Vertical gray lines indicate transitions of the hybrid controller.	53
3.7	Examples of $\mathcal{F}(\mathcal{C}^{\mathcal{I}})$ (blue) and $\mathcal{G}(\mathcal{C}^{\mathcal{I}})$ (green) generated by the planner.	54
4.1	Multiple disk-shaped robots navigating within a complex, planar workspace.	55
4.2	Over-approximation $\mathcal{R}_i(\mathcal{Z})$ (resp. $\mathcal{R}_i(\mathcal{Z}')$) and under-approximation $\underline{\mathcal{R}}_i(\mathcal{Z})$ (resp. $\underline{\mathcal{R}}_i(\mathcal{Z}')$) of the footprint of robot i when swept over $\mathcal{Z} \subset \mathbb{R}^2$ (resp. \mathcal{Z}).	57
4.3	Example of a hierarchical configuration space decomposition for a system of two identical robots. The green area corresponds to the configuration space $\mathcal{A}_i(\mathcal{W})$ of each robot. For the sake of simplicity, we assume $\mathcal{H}_1 = \mathcal{H}_2$. The workspace slice \mathcal{S}_1 corresponding to robot 1 consists of a single simple cell ($\mathcal{C}_{\mathcal{S}_1}$) whereas slice \mathcal{S}_2 corresponding to robot 2 consists of two cells ($\mathcal{C}_{\mathcal{S}_2,1}, \mathcal{C}_{\mathcal{S}_2,2}$). The compound cell ($\mathcal{C}_{\mathcal{S}_1}, \mathcal{C}_{\mathcal{S}_2,1}$) is labeled as mixed because \mathcal{R}_1 and \mathcal{R}_2 may intersect when $p_1 \in \mathcal{C}_{\mathcal{S}_1}$ and $p_2 \in \mathcal{C}_{\mathcal{S}_2,2}$ since $\overline{\mathcal{R}}_1(\mathcal{C}_{\mathcal{S}_1}) \cap \overline{\mathcal{R}}_2(\mathcal{C}_{\mathcal{S}_2,1}) \neq \emptyset$, whereas ($\mathcal{C}_{\mathcal{S}_1}, \mathcal{C}_{\mathcal{S}_2,2}$) is marked as admissible.	58
4.4	Two adjacent compound cells $\hat{\mathcal{C}}_1 = (\mathcal{C}_1^1, \mathcal{C}_1^2)$ and $\hat{\mathcal{C}}_2 = (\mathcal{C}_2^1, \mathcal{C}_2^2)$. In order robot 1 (resp. robot 2) to successfully move from \mathcal{C}_1^1 to \mathcal{C}_2^1 (resp. from \mathcal{C}_1^2 to \mathcal{C}_2^2) it has to reach any point of $\mathcal{G}_{1,1}$ (resp. $\mathcal{G}_{2,1}$).	62
4.5	Executed trajectories of the 2 robot case. Squares indicate initial positions whereas the corresponding goal positions are depicted using crosses.	64

4.6	Executed trajectories of the 4 robot case. Squares indicate initial positions whereas the corresponding goal positions are depicted using crosses.	64
4.7	Executed trajectories of the 6 robot case. Squares indicate initial positions whereas the corresponding goal positions are depicted using crosses.	65
4.8	Executed trajectories of the 8 robot case. Squares indicate initial positions whereas the corresponding goal positions are depicted using crosses.	65
4.9	Executed trajectories of the 10 robot case. Squares indicate initial positions whereas the corresponding goal positions are depicted using crosses.	66
4.10	The initial robot positions $p_{\text{init},i}$, $i \in \mathcal{I}_8$ and calculated initial compound cell $\hat{\mathcal{C}}_{\text{init}}$	66
4.11	The desired robot positions $p_{\text{des},i}$, $i \in \mathcal{I}_8$ and calculated goal compound cell $\hat{\mathcal{C}}_{\text{goal}}$	67
5.1	Cooperative object transportation by two robots.	69
5.2	Typical workspace.	72
5.3	Example of robotic system consisting of two mobile manipulators carrying a rectangular object. Each platform is equipped with a 2-link manipulator, which is able to rotate about the joint with the base.	73
5.4	Over- and under-approximations of a robotic system (a) consisting of two components corresponding to slices: $\{0\} \times [2\pi - 1, 1]$ (b and f), $\{0\} \times [2\pi - 0.2, 0.2]$ (c and g), $[3\pi/2, \pi/2] \times [2\pi - 0.2, 0.2]$ (d and h), $[2\pi - 0.2, 0.2] \times [2\pi - 0.2, 0.2]$ (e and i).	79
5.5	Initial and goal configuration of the robotic system and object, respectively.	91
5.6	Path executed by the robotic system during the simulations (blue line), as well as the footprint of the robotic system at various instants.	93
5.7	Evolution of the object's position over time. The vertical dashed lines indicate transitions between consecutive cells.	94
5.8	Evolution of the object's orientation over time (solid line), as well as the corresponding performance functions and	95
5.9	Object's linear velocity	96
5.10	Object's angular velocity	97
5.11	Total force applied to the object by the robots.	98
5.12	Total torque applied to the object by the robots.	99
5.13	Evolution of joint value 11 with corresponding lower and upper bounds.	100
5.14	Evolution of joint value 21 with corresponding lower and upper bounds.	101

Chapter 1

Introduction

1.1 Motivation

The autonomous operation of robotic platforms inside cluttered environments constitutes an actively studied research topic with autonomous navigation being undeniably a fundamental aspect of it. Moreover, as the tasks that robots are called to fulfill grow in complexity by the day, the employment of multi-agent robotic systems, which generally exhibit higher robustness and versatility than their single-robot counterparts, progressively increases. Thus, the robot navigation problems that need to be addressed become more and more challenging, increasing the need for more efficient and robust path and motion planning schemes. In addition, in order to design efficient and provably correct navigation schemes, one inevitably requires to model and understand the complicated topology of high-dimensional configuration spaces resulting from robotics systems with several degrees of freedom.¹

A wide range of techniques has been employed in the literature for designing control schemes that enable robots to navigate within complex environments, with configuration space motion planning [33] being among the earliest. Given an explicit representation of the system's configuration space, several planning and control schemes, such as ones based on exact cell decomposition approaches [53, 8] and roadmap construction approaches [12], can be used for safely steering the robot toward its target configuration, satisfying the prescribed task specifications. However, describing exactly the configuration space even of trivial robotics systems in terms of standard geometric and topological models, may prove to be a daunting task, if not an impossible one. On the other hand, approximate decomposition techniques, such as Slice Projection (SP) [69, 70] and Hierarchical Approximate Cell Decomposition (HACD) [10], avoid these issue by constructing instead an approximation of the configuration space defined as the union of finite, simpler geometrical shapes, such as hyper-rectangles and polytopes, at the expense of accuracy and completeness. In addition, the complexity of motion planning is known to grow exponentially as the

¹The configuration space of a robotic system is the set of all possible robot states that are free of collisions and satisfy any imposed kinematic and dynamic constraints [69].

dimension of the configuration space increases [12]. As such, configuration space navigation techniques are generally restricted to very low dimensional and simple settings.

Sampling-based motion planning [54, 40] constitutes another modern alternative solution that manages to circumvent the aforementioned issues as it requires instead an implicit definition of the configuration space, which is generally much easier to provide. On the other hand, these methodologies tackle solely the corresponding path finding problem and provide an open-loop solution that another class of controllers must track as closely as possible in order to ensure invariance of the robotic system's free configuration space. Moreover, this class of planners are known to have a hard time finding solutions to problems which are subject to relatively tight constraints (e.g., narrow corridors). Another deficiency of sampling-based planners is also their inability of determining infeasibility of a given problem, thus failing to answer connectivity queries within a specified amount of time provides no insight about the reason behind it.

Finally, another actively researched class of controllers tackles the path and motion planning sub-problems simultaneously by designing robot commands according to an underlying Artificial Potential Field (APF). Originally introduced in [46], APFs in general provide a simple means of encoding collision avoidance specifications and devising reactive control schemes, with their negated gradient functioning as a reference motion direction that drives the robot towards the desired goal configuration. Despite its intuitive nature, as shown in [49], this class of controllers suffers unavoidably from the presence of unwanted equilibria induced by the workspace's topology and whose region of attraction may not be trivial. Also, artificial potential fields admitted in a closed-form expression are generally restricted to geometrically simple spaces, thus elaborate transformations suitable for navigation need to be designed for mapping real workspaces onto such ideal worlds. On top of that, most artificial potential fields and transformations alike require tedious off-line tuning of certain design parameters in order to eliminate unwanted local minima and render the transformation a diffeomorphism.

In brief, the path and motion planning schemes for robot navigation mentioned above fundamentally differ from each other. Furthermore, each family exhibits a different set of advantages and disadvantages over the rest: sampling-based methods can address intricate problems but fail to exploit local properties of the configuration space configuration space decomposition methods use an explicit representation to ensure completeness at the expense of high computational cost, and artificial potential fields can produce closed-form control laws with guaranteed safety and convergence properties for topologically and geometrically simple workspaces. Hence, the question that naturally arises is whether it is possible to combine the strengths of these methods in order to design more versatile autonomous robot navigation schemes. In this dissertation, we answer this question in the affirmative as

we employ hierarchical decomposition techniques, coupled with suitably designed adaptive exploration schemes, to build versatile and complete planners that can answer connectivity queries, or determine their infeasibility, after a finite amount of steps. In addition, we equip the considered robotic systems with distributed control laws based on artificial potential fields which can safely realize the generated high-level plans, addressing the navigation problem within complex, planar workspaces with guaranteed convergence properties. We also remark that, in this work, we design more sophisticated high-level planners by also taking advantage of the proposed low-level control laws' ability to easily handle environments with arbitrary connectedness and shape, thus eliminating the need for partitioning the configuration space into unnecessarily simple components, such as rectangularoids. Finally, we demonstrate the efficacy of the proposed methodologies for addressing real-world problems, such as coordinating the motion of multiple robots operating within the same environment, as well as coordinating the transportation of an object by a team of cooperating mobile manipulators.

1.2 State-of-the-Art and Related Work

The autonomous navigation problem of robots in cluttered environments is a well studied one and various methods addressing it can be found in the vast literature. Although no clear distinction exists, a categorization that is frequently adopted classifies the alternative motion planning schemes into three wide categories [53]: a) Sampling-based Motion Planning, b) Combinatorial Motion Planning, and c) Feedback-based Motion Planning.

The first family of planners and, undoubtedly, the most actively studied of the three in modern literature, includes methodologies such as probabilistic roadmaps (PRMs) [40] and rapidly exploring random trees (RRTs) [54]. The probabilistic roadmap method, originally developed independently at different sites [79, 42, 41, 74], was the first to emerge and get adopted, mainly due to its simplicity and versatility. This approach comprises of two distinct stages. During the first, pre-processing stage, the configuration space is sampled for collision-free placements, which are added as nodes to a graph. Pairs of promising nodes are selected and a (usually simplistic) local planner is employed for connecting such placements with a feasible path, if one can be found, which corresponds to an edge of the graph. This process is repeated for a specified amount of time, which is hopefully sufficient for the resulting graph to encapsulate the connectedness of the actual configuration space. Then, during the second stage, queries are imposed to the planner in the form of pairs of initial and final configurations. The planner then tries to connect the given configurations to the readily computed graph and, if successful, then standard graph search algorithms are employed for finding a sequence of connected nodes between the given configurations, assuming they belong to the same connected component of the aforementioned graph.

The basic PRM description leaves some details to be filled in, like how to sample the underlying space or what local planner to employ. Different aspects of this methodology have been investigated and several improvements have been proposed over the original abstract scheme, such as Gaussian sampling [9], Bridge-Test sampling [37, 36, 103], the Small-Step Retraction method [93], Lazy PRM [7], Obstacle-Based PRM [75, 120], and many more [87, 90, 121, 103, 29, 81, 48]. The rapidly exploring random trees method, which was originally introduced in [54, 52, 55] for coping with dynamical systems subject to Pfaffian constraints, constitutes a more versatile, sampling-based alternative to PRM allowing one to address the navigation problem for more intricate classes of dynamical systems and restrictions. Optimal variants of the aforementioned methods, in the sense of finding the shortest path subject to a given metric, were initially presented in [39]. On the other hand, sampling-based methods strongly rely on fast collision detectors, efficient nearest neighbor and graph search algorithms, effective sampling strategies and informative metric selection [61, 56] to operate efficiently. Moreover, these methodologies are generally having a hard time addressing problems with constricted configuration spaces (e.g., workspaces densely occupied by obstacles or narrow corridors). To alleviate these issues various attempts have been made. In [3], a novel sensory steering algorithm was designed which used local Voronoi decomposition of the workspace to significantly improve the path planning performance of sampling based algorithms near difficult regions such as narrow passages. In [95] and [94], the authors propose a scheme that samples entire manifolds instead of isolated configurations, which are, in turn, used for approximating the configuration space's connectivity graph, thus allowing the planner to perform significantly better even in tight workspaces. In order to address multi-robot scenarios, when a common graph representation of the workspace is shared among the agents, efficient methodologies for coordinating their transitions were proposed in [89, 110, 119]. In [104], a methodology was presented which addresses cases where the motion of each robot is restricted to a distinct graph by building a composite roadmap (i.e., the Cartesian product of the individual graphs). Furthermore, more efficient extensions of this approach, which work on implicitly defined composite roadmaps and potentially, lower dimensional configuration spaces, can be found in [117, 118, 101, 99].

Moving on to combinatorial planning schemes, exact cell decomposition and algebraic approaches, such as [77, 76, 43, 96, 31, 98, 102] to name but a few, were among the earliest to be considered. In arguably the most exemplary works of this class [107, 97], the authors tackle the path finding problem for a robot which is able to translate and rotate within polygonal environments by building a graph of critical robot configurations [57, 44], i.e., configurations at which the robot has at least two points in contact with the workspace's boundary, which encapsulates the topological structure of the configuration space's projection onto the robot's workspace. Also, a similar approach was presented in [106] for coordinating the motion of teams of two and three disk shaped robots. Unfortunately, careful implementation of such exact

cell decomposition algorithms, generally achieves a complexity doubly exponential w.r.t. the degrees of freedom of the robotic system [17]. A more elaborate solution, known as the Roadmap Algorithm, which reduces this complexity to singly exponential time, was presented by Canny in [12]. For the special case of point and ball shaped robots, operating within a compact, simply connected workspace, a novel control scheme was introduced in [1, 4] which, by partitioning the system's configuration space into strata based on nearest neighbor relations and constructing binary tree representations of those, manages to reduce the computational complexity of the motion planning problem down to $O(n^2)$ time.

Nevertheless, despite their elegance, the complexity of such exact methods renders them impractical for addressing cases with non-trivial shapes and sizes of robotic teams. To that end, approximate cell decomposition techniques [71, 10] have been successfully employed for tackling robot navigation problems with complex, high-dimensional configuration spaces and generally exhibit fast exploration capabilities when coupled with hierarchical adaptive subdivision schemes guided by suitable heuristics. More specifically, Slice Projection (SP) methodologies, like the ones presented in [69, 70], consist of partitioning the robot's configuration space into slices, with respect to a given axis (e.g., the robot's orientation), and projecting those slices to obtain safe, lower dimensional under-approximations of the robot's free space. On the other hand, Hierarchical Approximate Cell Decomposition (HACD), originally introduced in [10], constitutes a tool for robot path planning which builds a decomposition of the robot's configuration space made of rectangloid cells at successive levels of approximation. A major advantage of the former over the later is that SP generates, in general, a smaller amount of cells since the geometry represented by a slice does not need to be any further approximated by simpler shapes, although, this comes at the cost of requiring more elaborate techniques, compared to rectangloids, in order to safely navigate within a projected slice. Additionally, heuristics [122] designed for use with HACD methods enable the design of more efficient algorithms, able to adaptively refine the configuration space's representation, that have been successfully employed for addressing various tasks [11, 105].

Feedback-based planners constitute another actively researched class of controllers which tackle the path and motion planning sub-problems simultaneously by designing robot commands according to some sort of reactive policy. Online optimal control strategies, like receding horizon control [35, 78, 50], are an example quite popular in practice since they can accommodate for realistic robot kinematic and dynamic models [72, 16, 59], and address systems with several degrees of freedom [68, 34, 26]. In addition, examples of artificial vector fields include but are not limited to methodologies based on Lyapunov-like barrier functions [83, 111, 60], Prescribed Performance Control (PPC) schemes [116, 6, 112, 115] and dipolar vector fields [86, 85, 84]. Also, a novel approach based on Power Diagrams which can be used for designing tune-free vector fields for navigation within sufficiently curved,

convex workspaces is also presented in [1, 2, 5]. Unfortunately, the majority of the solutions mentioned above possess guaranteed convergence properties only for initializations of the problem “close enough” to the desired state or under rather strict assumptions (e.g., convex obstacles, trivial workspaces, etc), effectively sacrificing completeness for a reactivity and safety. Another methodology of this class of particular interest involves the design of reactive control laws based on Artificial Potential Fields (APFs). Originally introduced in [46], APFs provide, in general, a simple means of encoding collision avoidance specifications, with their negated gradient functioning as a reference motion direction that drives the robot towards the desired goal configuration. As shown in [49], despite its intuitive nature, this class of controllers suffers unavoidably from the presence of unwanted equilibria induced by the workspace’s topology and whose region of attraction may not be trivial. In their seminal work [91], Rimon and Koditschek presented a family of APFs called Navigation Functions (NF) for point and sphere worlds,² as well as a constructive transformation for mapping workspaces cluttered by trees of star-shaped obstacles³ into such worlds. However, certain design parameters required tedious tuning to eliminate unwanted local minima and render the transformation a diffeomorphism. The NF framework has been successfully adapted since then to address a variety of problems for different kinds and number of robots [15, 108, 109, 92, 27, 32, 58]. Although provably correct, this solution suffers in practice by the fact that the allowable values of the design parameters may cause both the potential and the corresponding transformation to vary abruptly close to the obstacles (the issue of “disappearing valleys” [91]), thus pushing the trajectories of the robot very close to them. Methods for remedying such drawbacks or relaxing some generally conservative requirements of NFs, up to a certain degree, are presented in [66, 18, 23]. Additionally, attempts to extend the NF framework directly to non-sphere worlds can be found in [19, 24, 25, 88].

Artificial Harmonic Potential Fields (AHPF) constitute an interesting subclass of APFs, since they are free of unwanted local minima by construction. However, no simple way exists for constructing safe, w.r.t. obstacle avoidance, harmonic potentials even for simple workspaces. AHPFs suitable for navigation in realistic environments were originally utilized in [13, 14], where computationally expensive numerical techniques were employed to solve the associated Dirichlet and Neumann problems. Several extensions of the aforementioned methodology have followed [73, 30, 28], addressing issues such as numerical precision and computation, dynamic environments, etc. The panel method was employed in [47] and [21] to build harmonic potentials to coordinate the motion of a single and multiple robots in polygonal environments. In [22, 80], well known closed-form solutions of the uncompressed fluid flow around simple geometries were used in order to safely drive a robot among

²A Euclidean sphere world of dimension N is formed by removing from the interior of a large N -dimensional ball a finite number of non-overlapping smaller balls.

³A star is a set which possesses a point from which all the rays cross the boundary only once.

moving obstacles. Harmonic potential fields have also been used in [100, 38] for addressing the Simultaneous Localization and Mapping problem (SLAM) by coordinating the robot motion in unknown environments. Moreover, a methodology based on the evaluation of the harmonic potential field's streamlines is used in [114] for mapping a multiply connected workspace to a disk, collapsing inner obstacles to line segments or arcs. In a recent work [62], the problem of designing closed form harmonic potentials in sphere worlds is addressed by the introduction of a diffeomorphism [67, 65], which allows mapping such workspaces to the euclidean plane with some of its points removed. Finally, extensions of this work addressing topologically complex 3-dimensional workspaces or multi-robot scenarios, by introducing appropriate constructive workspace transformations, can be found in [63] and [64], respectively.

1.3 Outline

The organization of this dissertation is as follows.

Chapter 2:

In this chapter, we present a novel control scheme for addressing the navigation problem for a robot operating within a static, compact and planar workspace with arbitrary connectedness and shape. Our goal is to design a control law that can successfully drive the robot to any valid goal configuration, starting from almost any initial configuration, without colliding with the obstacles occupying the workspace.

To cope with the workspace's topology, we present a methodology for building a harmonic map of the workspace onto a punctured disk. This transformation, which to the best of our knowledge, has not been employed before for addressing the robot navigation problem, is unlike transformations used in the related literature (e.g., [91] and [64]), as it does not require any sort of decomposition of the workspace into simpler geometries (e.g., trees of stars). Moreover, contrary to [91, 64, 65] where an explicit representation of the workspace boundaries is required (i.e., as the level sets of a sufficiently smooth function), our approach only needs a sufficiently fine polygonal workspace description, which can be easily acquired in practice via SLAM. Although the construction of this map relies on computationally expensive numerical techniques, it only needs to be calculated offline once for a given static workspace due to the fact that it depends neither on the robot's initial nor on the final configurations, contrary to approaches where different harmonic potentials have to be computed for each robot's destination.

In addition, we design closed-form AHPFs for robot navigation, extending the methodology presented in [62] by the introduction of an appropriate adaptive laws for their parameters, which enables addressing bounded workspaces and controlling, up to a certain degree, the tradeoff between high obstacle repulsiveness and convergence rate. Moreover, we exploit novel tools on dynamical systems' analysis to study

the stability of degenerate equilibria of the closed-loop system. Finally, we adapt our methodology to the class of differential drive robots, which are commonly encountered in real-world applications and propose an elegant way to render the problem of addressing large workspaces tractable.

Chapter 3:

As realistic robots hardly ever can be accurately modeled as disks, in this chapter, we present a hybrid control scheme in order to address the navigation problem for a robotic platform whose shape can be adequately represented by a polygon. Like before, the robot is assumed to operate within a compact workspace with arbitrary connectedness and shape.

The proposed control scheme consists of two components: a high-level planner, which is responsible for finding a sequence of configuration space cells that the robot can safely traverse in order to reach its goal, and a low-level kinematic controller which is tasked with driving the robot from one cell to the next, while ensuring invariance of the free configuration space. To do so, given an initial and desired robot configuration, we begin with constructing two types of approximations of the robot's configuration space based on slice projections obtained by adaptively refining the range of permitted robot orientations, while searching for sequences of connected cells in each approximation that contain these two configurations. In order to successfully find such a path of cells, we design a heuristic used by our algorithm for choosing slices whose subdivision potentially increases our knowledge regarding the connectivity of the configuration space's portion directly related to our problem. Furthermore, the proposed algorithm, which is resolution complete by design, is also able to determine when a given problem is not feasible in a finite amount of iterations, assuming that the robot's collision-free configuration space is open. Having thus obtained a sequence of connected cells, we design decoupled control laws for the robot's linear and angular velocities which can safely drive the robot from one cell to the next, till the goal configuration is reached. Particularly, in order to regulate the robot's position, we employ the harmonic transformation and adaptive harmonic potential field scheme presented in [chapter 2](#), which integrates elegantly with the sort of configuration space representation used in this chapter.

Chapter 4:

As the popularity of multi-agent robotic systems grows by the day, due to the display of higher robustness and versatility compared to single-robot alternatives, the need for more efficient forms of planning grows by the day. In this chapter, we adopt the concepts presented in the previous chapters, in order to design a novel control scheme for coordinating the navigation of a team of disk-shaped robots that operate within an arbitrary, obstacle cluttered, planar workspace.

More specifically, given an initial and desired configuration for the robotic team, we design a high-level planner, based on Hierarchical Cell Decomposition techniques,

which is tasked with the exploration of the system's configuration space to discover a sequence of cells, that the robots can safely traverse, towards the desired configurations. One of the strong points of the proposed algorithm is the use of a suitable labeling mechanism for selecting the regions of the configuration space to be subdivided at each iteration. Particularly, by computing an over- and an under-approximation of each robot's footprint, in a manner similar to the one presented in [chapter 3](#), our algorithm can determine which cells may contain feasible configurations of the system, while automatically discarding cells that are determined to contain none. Finally, having obtained a sequence of traversable cells, we equip our system with a distributed low-level control law based on harmonic maps and adaptive harmonic potential fields, introduced in [chapter 2](#), which guarantee safe and almost-global convergence from one cell to the next. In addition, as our scheme does not require explicit knowledge of the entire system's state in order to compute control inputs for each individual robot, it significantly reduces the communication requirements of the robotic team.

Chapter 5:

In this chapter, we present a methodology for planning the motion of a team of mobile manipulators carrying an object within a compact planar workspace. More specifically, as the object is allowed to translate and rotate and each robot is equipped with a manipulator consisting of one or more moving links, our scheme must take into consideration the varying shape of the robotic system in order to devise a plan that can safely drive the object to a specified configuration. To this end, we extend the methodology introduced in [chapter 3](#) in order to build a sequence of configuration space cells, each of which defines an allowable set of configurations of the object, as well as explicit intervals for each manipulator's states. Furthermore, appropriately designed under- and over-approximations of the free configuration space are used in order to guide the configuration space's exploration without loss of completeness. In addition, we couple methodologies based on Reference Governors and Prescribed Performance Control with harmonic maps, presented in [chapter 2](#), in order to design a distributed control law for realizing the transitions specified by the high-level planner, which possesses guaranteed invariance and almost global convergence properties. Furthermore, the proposed low-level control law does not require continuous information exchange between the robots, which rely only on measurements of the object's configuration and their own states.

Chapter 2

Navigation of Disk-Shaped Robots on the Plane



FIGURE 2.1: Mobile robot navigating within a maze.

2.1 Introduction

In this chapter, we address the navigation problem for a robot operating within a static, compact, planar workspace which has arbitrary connectedness and shape. Particularly, we design a control law which can safely drive the robot to a specific goal configuration from almost all initial configurations. To cope with the workspace's topology, we construct a harmonic map of the workspace onto a punctured disk, which, unlike the transformations proposed in [91] and [64], does not require the decomposition of the workspace into trees of stars. Moreover, contrary to [91, 64, 65] where an explicit representation of the workspace boundaries is required (i.e., as the level sets of a sufficiently smooth function), our approach only needs a sufficiently fine polygonal workspace description, which can be easily acquired in practice via SLAM. In addition, we employ closed-form AHPFs for robot navigation, similar to the ones presented in [62], but modified accordingly by introducing appropriate

adaptive laws for their parameters to address efficiently bounded workspaces and control, up to a certain degree, the tradeoff between high obstacle repulsiveness and convergence rate. Notice also that, unlike the methodologies presented in [14, 73], we only need to solve a computationally expensive problem once for a given static workspace, independently of the robot's initial and goal configurations. Also, we elaborate on the construction of the proposed transformation using numerical techniques, and particularly the Panel Method. Finally, we adapt our methodology to the class of differential drive robots, which are commonly encountered in real-world applications and propose an algorithm to render the problem of addressing large workspaces tractable.

The outline of this chapter is as follows. In Section 2.2, we rigorously formulate the problem and in Section 2.3, we present the methodology for constructing harmonic maps suitable for navigation. In Section 2.4, we describe the proposed control law for addressing the robot navigation problem and elaborate on its stability. Furthermore, in Section 2.5, we present methodologies for extending the proposed control scheme to differential drive robots as well as for reducing the complexity when addressing large and composite workspaces. Finally, in Section 2.6, we demonstrate the efficacy of the proposed control scheme via extensive simulation and experimental results.

Notation: We use $\mathcal{D}_r(x)$ to denote the open disk with radius $r > 0$ centered at $x \in \mathbb{R}^2$. Additionally, \mathcal{D} and $\partial\mathcal{D}$ denote the closed disk and circle with unit radii centered at the origin of \mathbb{R}^2 , respectively. Also, let $\mathcal{I}_N \triangleq \{1, 2, \dots, N\}$ and $\mathcal{I}_N^* \triangleq \{0\} \cup \mathcal{I}_N$. Given sets $A, B, \subseteq \mathbb{R}^n$, we use $\text{cl}(A)$, ∂A , $\text{int}(A)$, \bar{A} to denote the closure, boundary, interior and complement of A w.r.t. \mathbb{R}^n , respectively, and $A \setminus B$ to denote the complement of B w.r.t. A . Furthermore, we use $\mathbf{0}_N$ and $\mathbf{1}_N$ to denote the all-zeros and all-ones column vectors of length N , respectively, and $\mathbf{0}_{N \times M}$ to denote the $N \times M$ zero matrix. We also define $\mathbf{1}_{N \times M}^k$, $k \in \mathcal{I}_M$ as the $N \times M$ matrix whose k -th column is equal to $\mathbf{1}_N$ and every other column being equal to $\mathbf{0}_N$. Given a vector function $f(x)$, we use $\nabla_x f$ to denote its Jacobian matrix. Also, given an arc C , we use $\text{len}(C)$ to denote its length. Finally, we will say that a set A is attractive (resp. repulsive) under a potential field ψ when there exists a point $p_0 \notin \text{cl}(A)$ such that if we initialize at p_0 and move along the negated gradient of ψ , we will converge (resp. will not converge) to ∂A .

2.2 Problem Formulation

We consider a robot operating within a compact workspace $\mathcal{W} \subset \mathbb{R}^2$, bounded by a single outer and a finite set of inner disjoint Jordan curves¹, which correspond to the

¹A Jordan curve is a non-self-intersecting continuous planar curve.

boundaries of static obstacles. It is assumed that \mathcal{W} can be written as

$$\mathcal{W} = \overline{\mathcal{W}_0} \setminus \bigcup_{i \in \mathcal{I}_N} \mathcal{W}_i \quad (2.1)$$

where $\mathcal{W}_i, i \in \mathcal{I}_N^*$ denote regions of \mathbb{R}^2 that the robot cannot occupy. Particularly, the complement of \mathcal{W}_0 is considered to be a bounded, simply connected region that may also include a strict subset of its own boundary² and $\mathcal{W}_1, \mathcal{W}_2, \dots, \mathcal{W}_N$ are assumed to be closed, simply connected compact sets that are contained in $\overline{\mathcal{W}_0}$ and are pairwise disjoint. Let $p = [x, y]^T \in \mathbb{R}^2$ denote the robot's position and assume that the robot's motion is described by the single integrator model:

$$\dot{p} = u \quad (2.2)$$

where $u \in \mathbb{R}^2$ is the corresponding control input vector.

Problem 1. *Our goal is to design a control law to successfully drive a robot with kinematics (2.2) towards a given goal configuration $p_d \in \mathcal{W}$, from almost any feasible initial configuration $p_{\text{init}} \in \mathcal{W}$, while ensuring collision avoidance, i.e., $p(t) \in \mathcal{W}$ for all $t \geq 0$.*

Remark 1. *The results presented in this chapter can be readily employed for the navigation of disk robots with radius $R > 0$ by appropriately augmenting the workspace boundaries with the robot's size.*

2.3 Harmonic Maps for Planar Navigation

In this section, we present a methodology that maps the robot's workspace onto a punctured unit disk, where the robot's control law is designed. Particularly, our goal is to construct a transformation $T : \text{cl}(\mathcal{W}) \mapsto \mathcal{D}$, from the closure of the robot's configuration space $\text{cl}(\mathcal{W})$ to the unit disk \mathcal{D} , with the following properties:

1. $T(\cdot)$ maps the outer boundary $\partial\mathcal{W}_0$ to the unit circle $\partial\mathcal{D}$,
2. $T(\cdot)$ maps the boundary $\partial\mathcal{W}_i, i \in \mathcal{I}_N$ of each obstacle to a distinct point $q_i = [u_i, v_i]^T \in \text{int}(\mathcal{D})$,
3. $T(\cdot)$ is a diffeomorphism for all $p \in \text{int}(\mathcal{W})$.

To that end, we propose a harmonic map $T(p) = [u(p), v(p)]^T$, (i.e., $u(p)$ and $v(p)$ are both harmonic function), which, to the best of our knowledge, has never before been employed in a motion planning framework addressing multiply connected workspaces in this manner. The existence of such map was proved in [20] and sufficient conditions rendering a harmonic map a diffeomorphism were extracted as

²This corresponds to cases when we wish to place the robot's goal configuration on some part of the workspace outer boundary which is not physically occupied by an actual obstacle, e.g., the door of a compartment (refer to [Subsection 2.5.2](#) for more details).

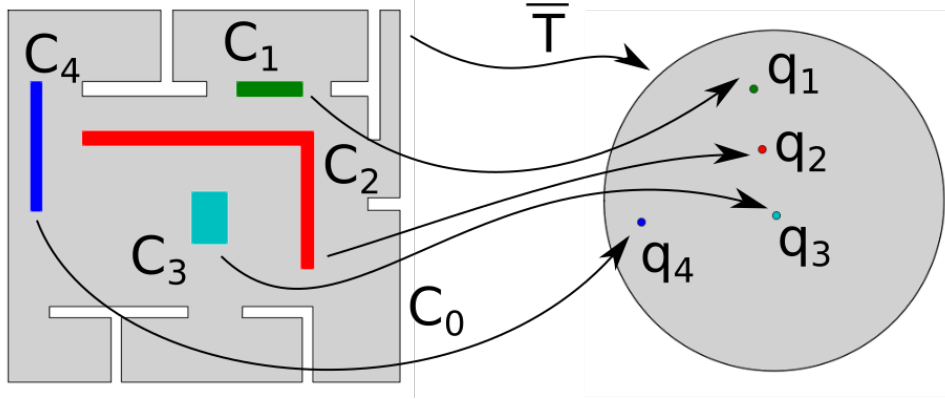


FIGURE 2.2: Transformation of a real workspace onto a punctured disk.

part of the proof. We will now summarize those findings expressed for real-valued, vector functions in the form of a theorem for the sake of completeness.

Theorem 1. (Theorem 2 in [20]) Let $\mathcal{W} \subset \mathbb{R}^2$ be a finitely connected open domain bounded by Jordan curves $\partial\mathcal{W}_0, \partial\mathcal{W}_1, \dots, \partial\mathcal{W}_N$, where $\partial\mathcal{W}_0$ is the outer boundary component. Let $\Omega \in \mathbb{R}^2$ be a bounded convex domain. Suppose that \bar{f} is an orientation-preserving continuous weak homeomorphism of $\partial\mathcal{W}_0$ onto $\partial\Omega$. Then, there exists a harmonic function $f : \mathcal{W} \mapsto \Omega$ that satisfies the boundary conditions:

$$f(p) = \bar{f}(p), \quad \forall p \in \partial\mathcal{W}_0, \quad (2.3)$$

$$\int_{\partial\mathcal{W}_i} \frac{\partial f}{\partial n} ds = 0, \quad \forall i \in \mathcal{I}_N \quad (2.4)$$

where $n = n(p)$ denotes the unit vector that is normal to the boundary at the point $p \in \partial\mathcal{W}_i$, $i \in \mathcal{I}_N$, and maps \mathcal{W} injectively onto Ω with N points removed.

What this theorem practically means is that given any orientation-preserving, weak homeomorphism from the workspace outer boundary $\partial\mathcal{W}_0$ to the boundary of any given convex set, we can construct a diffeomorphism that maps $\text{cl}(\mathcal{W})$ to the target set and which collapses each obstacle onto a separate point q_i within its interior (see Figure 2.2). Thus, in order to construct the requested transformation of $\text{cl}(\mathcal{W})$ onto the unit disk \mathcal{D} , first we need to provide a map $\bar{T} : \partial\mathcal{W}_0 \mapsto \partial\mathcal{D}$ for its boundary³ (i.e., $\bar{T}(p) \triangleq [\bar{u}(p), \bar{v}(p)]^T$), and solve the two corresponding boundary value problems for $u(p)$ and $v(p)$ subject to the constraints (2.3) and (2.4). Notice that the coordinates of q_i (i.e., the images of the internal obstacles) are not explicitly specified, but are enforced by the problem's solution as a result of satisfying the aforementioned constraints.

³This particular transformation \bar{T} can be easily constructed for any given planar Jordan curve C by 1) arbitrarily selecting a point p_o on C , 2) defining $\ell(p), \forall p \in C$ as the length of the arc $\widehat{p_o p}$, assuming one travels from p_o to p on C while having the curve's interior to its left, and 3) choosing $\bar{T}(p) = [\cos(2\pi\ell(p)/L), \sin(2\pi\ell(p)/L)]^T$, where $L = \text{len}(C)$.

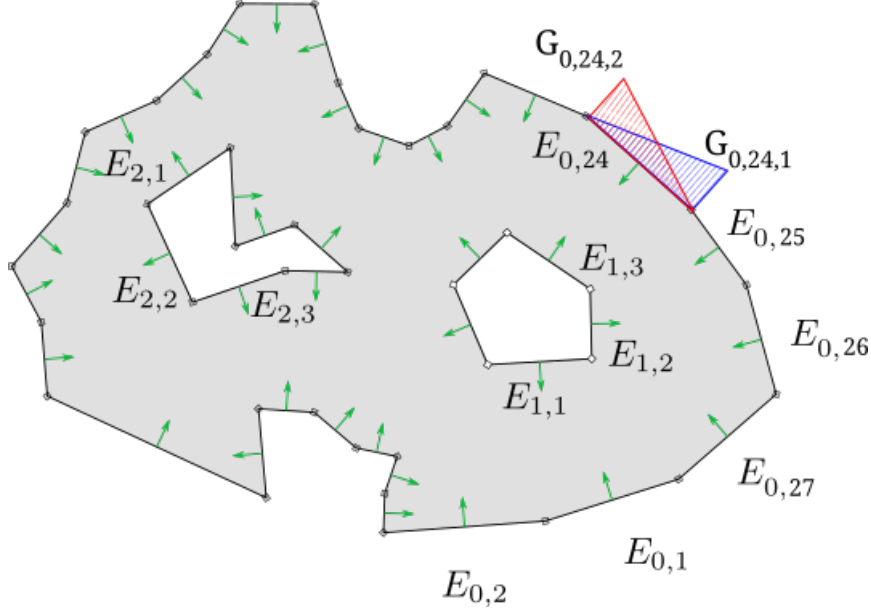


FIGURE 2.3: Discretization of an arbitrary workspace's boundary using line segment elements. By convention, the outer boundary is considered to be clockwise oriented whereas inner boundaries are counter-clockwise oriented. The normal direction of each element is depicted using green colored vectors. Furthermore, the values of the two linear shape functions $G_{0,24,1}$ and $G_{0,24,2}$ are plotted along the length of the corresponding element $E_{0,24}$.

In order to construct the harmonic functions $u(p)$ and $v(p)$, we employ numerical techniques and particularly the Panel Method.⁴ Accordingly, we approximate each of the components of $T(p) = [u(p), v(p)]^T$ by a sum of simple harmonic source distributions as follows:

$$u(p) \approx \sum_{i=0}^N \sum_{j=1}^{\tilde{M}_i} \sum_{l=1}^{\tilde{L}} C_{ijl}^x H_{ijl}(p) \quad (2.5)$$

$$v(p) \approx \sum_{i=0}^N \sum_{j=1}^{\tilde{M}_i} \sum_{l=1}^{\tilde{L}} C_{ijl}^y H_{ijl}(p)$$

$$H_{ijl}(p) = \int_{E_{ij}} G_{ijl}(s) \ln(\|p - \tilde{p}_{i,j}(s)\|) ds \quad (2.6)$$

which are obtained by subdividing separately the workspace's outer and inner boundaries into $\tilde{M}_0, \tilde{M}_1, \dots, \tilde{M}_N$ number of elements (see Figure 2.3), where \tilde{L} is the number of control parameters per element, E_{ij} denotes the j -th element of the i -th boundary's approximation, $\tilde{p}_{i,j}(s) \in E_{ij}$ is a bijective parameterization of E_{ij} by its length (i.e., $\tilde{p}_{i,j} : [0, \text{len}(E_{ij})] \mapsto E_{ij}$), $G_{ijl} : [0, \text{len}(E_{ij})] \mapsto \mathbb{R}$ is the shape function corresponding to the l -th control parameter of E_{ij} , and $C_{ijl}^x, C_{ijl}^y \in \mathbb{R}$ are control parameters that need to be appropriately selected so that T satisfies Properties 1–3, for all $l \in \mathcal{J}_{\tilde{L}}, j \in \mathcal{J}_{\tilde{M}_i}$ and $i \in \mathcal{J}_N^*$. It is worth noting that for common choices of G_{ijl} (e.g., constant or linear shape functions) and simple types of E_{ij} (e.g., line segments),

⁴Similar formulations can be obtained employing other techniques such as the Boundary Element Method (BEM), Finite Element Method (FEM) or Finite Differences Method (FDM).

the integral in (2.6) can be easily evaluated to obtain a closed-form expression for H_{ij} ⁵. Generally, this comes at the expense of requiring more elements to sufficiently approximate the exact transformation, as opposed to selecting more elaborate (e.g., quadratic) shape functions or element geometries. To obtain the unknown control parameters as well as the images of the workspace's inner obstacles, one needs to solve the two independent linear systems of equations:

$$AX = B_x, \quad AY = B_y \quad (2.7)$$

for the unknown vectors

$$\begin{aligned} X &= [C_{0,1,1}^x, \dots, C_{1,1,1}^x, \dots, C_{N,\tilde{M}_N,\tilde{L}}^x, \mathbf{u}_1, \dots, \mathbf{u}_N]^T \\ Y &= [C_{0,1,1}^y, \dots, C_{1,1,1}^y, \dots, C_{N,\tilde{M}_N,\tilde{L}}^y, \mathbf{v}_1, \dots, \mathbf{v}_N]^T. \end{aligned} \quad (2.8)$$

The matrix A and r.h.s. vectors B_x, B_y are constructed by selecting a set of arbitrary points $\tilde{p}_{i,j}^*$ ⁶ such that a) $\tilde{p}_{i,j}^* \in \partial\mathcal{W}_i$ for all $j \in \mathcal{J}_{\tilde{N}_i}$ and $i \in \mathcal{J}_N^*$, where $\tilde{N}_i, i \in \mathcal{J}_N^*$ denotes the number of points selected on the corresponding boundary components of \mathcal{W} , and b) $\sum_{i \in \mathcal{J}_N^*} \tilde{N}_i$ is equal to the amount of unknown parameters (i.e., $\sum_{i \in \mathcal{J}_N^*} \tilde{M}_i \tilde{L}$), on which the conditions (2.3) and (2.4) are evaluated as follows:

$$A = \begin{bmatrix} A_0, & \mathbf{0}_{\tilde{N}_0 \times N} \\ A_1, & -\mathbf{1}_{\tilde{N}_1 \times N}^1 \\ \vdots & \vdots \\ A_N, & -\mathbf{1}_{\tilde{N}_N \times N}^N \\ A_n, & \mathbf{0}_{N \times N} \end{bmatrix}, \quad B_x = \begin{bmatrix} B_{x0} \\ \mathbf{0}_{\tilde{N}_1} \\ \vdots \\ \mathbf{0}_{\tilde{N}_N} \\ \mathbf{0}_N \end{bmatrix}, \quad B_y = \begin{bmatrix} B_{y0} \\ \mathbf{0}_{\tilde{N}_1} \\ \vdots \\ \mathbf{0}_{\tilde{N}_N} \\ \mathbf{0}_N \end{bmatrix} \quad (2.9)$$

$$A_k = \begin{bmatrix} H_{0,1,1}(\tilde{p}_{k,1}^*) & \cdots & H_{N,\tilde{M}_N,\tilde{L}}(\tilde{p}_{k,1}^*) \\ H_{0,1,1}(\tilde{p}_{k,2}^*) & \cdots & H_{N,\tilde{M}_N,\tilde{L}}(\tilde{p}_{k,2}^*) \\ \vdots & \vdots & \vdots \\ H_{0,1,1}(\tilde{p}_{k,\tilde{N}_k}^*) & \cdots & H_{N,\tilde{M}_N,\tilde{L}}(\tilde{p}_{k,\tilde{N}_k}^*) \end{bmatrix}, \quad \forall k \in \mathcal{J}_N^* \quad (2.10)$$

$$A_n = \begin{bmatrix} \sum_k^{\tilde{N}_1} \frac{\partial H_{0,1,1}}{\partial n}(\tilde{p}_{1,k}^*) & \cdots & \sum_k^{\tilde{N}_1} \frac{\partial H_{N,\tilde{M}_N,\tilde{L}}}{\partial n}(\tilde{p}_{1,k}^*) \\ \sum_k^{\tilde{N}_2} \frac{\partial H_{0,1,1}}{\partial n}(\tilde{p}_{2,k}^*) & \cdots & \sum_k^{\tilde{N}_2} \frac{\partial H_{N,\tilde{M}_N,\tilde{L}}}{\partial n}(\tilde{p}_{2,k}^*) \\ \vdots & \vdots & \vdots \\ \sum_k^{\tilde{N}_N} \frac{\partial H_{0,1,1}}{\partial n}(\tilde{p}_{N,k}^*) & \cdots & \sum_k^{\tilde{N}_N} \frac{\partial H_{N,\tilde{M}_N,\tilde{L}}}{\partial n}(\tilde{p}_{N,k}^*) \end{bmatrix} \quad (2.11)$$

⁵As an illustration, for a line segment element E_{ij} with two control parameters (i.e., $\tilde{L} = 2$), a typical choice for linear shape functions is $G_{ij1}(s) = s/\text{len}(E_{ij})$, $G_{ij2}(s) = (1-s)/\text{len}(E_{ij})$ and $\tilde{p}_{i,j}(s) = \tilde{p}_{i,j,A} + \tilde{p}_{i,j,B}s/\text{len}(E_{ij})$ for the corresponding parameterization, where $\tilde{p}_{i,j,A}, \tilde{p}_{i,j,B}$ are the element's end-points.

⁶A typical strategy is to select the points $\tilde{p}_{i,j}^*$ uniformly on the outer and inner boundaries of the given domain.

$$B_{x0} = \begin{bmatrix} \bar{u}(\tilde{p}_{0,1}^*) \\ \bar{u}(\tilde{p}_{0,2}^*) \\ \vdots \\ \bar{u}(\tilde{p}_{0,\tilde{N}_0}^*) \end{bmatrix} \quad B_{y0} = \begin{bmatrix} \bar{v}(\tilde{p}_{0,1}^*) \\ \bar{v}(\tilde{p}_{0,2}^*) \\ \vdots \\ \bar{v}(\tilde{p}_{0,\tilde{N}_0}^*) \end{bmatrix} \quad (2.12)$$

Notice that the directional derivatives in (2.11) are computed along the unit vectors that are normal to the boundary at $\tilde{p}_{i,j}^*$ according to the convention depicted in Figure 2.3. Moreover, by selecting a sufficiently large number of sufficiently small elements, the overall approximation error between the solution of the aforementioned linear problem and the exact transformation defined in Theorem 1 can be rendered arbitrarily small.

2.4 Control Design

To address Problem 1, we equip the robot with the aforementioned transformation $q = T(p)$ from the closure of its configuration space \mathcal{W} onto the unit disk \mathcal{D} and an artificial potential field $\psi(q, k)$ augmented with an adaptive control law $\dot{k} = f_k(q, k)$ for its parameters $k = [k_d, k_1, k_2, \dots, k_N]^T$. The robot velocity control law is calculated as follows:

$$u = -\mathcal{K}_u s(q, k) J^{-1}(p) \nabla_q \psi(q, k) \quad (2.13)$$

where $J(p)$ denotes the Jacobian matrix of $T(p)$, $s(p, k) \geq 0$ is a continuously differentiable gain function given by:

$$s(q, k) = \gamma \sigma_p \left(\frac{1 - \|q\|}{\epsilon_p} \right) + (1 - \gamma) \sigma_v \left(\frac{(\nabla_q \psi)^T q}{\epsilon_v + \|\nabla_q \psi\| \|q\|} \right) \quad (2.14)$$

with

$$\sigma_p(x) = \begin{cases} x^2(3 - 2x), & \text{if } x \leq 1 \\ 1, & \text{if } x > 1 \end{cases}, \quad (2.15)$$

$$\sigma_v(x) = \begin{cases} x^2, & \text{if } x \geq 0 \\ 0, & \text{if } x < 0 \end{cases} \quad (2.16)$$

and $\mathcal{K}_u, \gamma, \epsilon_p, \epsilon_v$ are scalar constants such that $\mathcal{K}_u, \epsilon_v > 0$ and $\gamma, \epsilon_p \in (0, 1)$. More specifically, $s(q, k)$ consists of two individual terms with the first vanishing as the robot approaches the workspace's outer boundary (and its distance from the unit circle is less than ϵ_p) whereas the second vanishes when the robot's velocity points away from the disk's center. The scalar parameter γ can be used for adjusting the contribution of each respective term of $s(q, k)$. Finally, ψ is a harmonic artificial potential field defined on the image $T(\mathcal{W})$ of the workspace \mathcal{W} and whose negated gradient $-\nabla_q \psi(q, k)$ defines the direction of the robot's motion in the real workspace \mathcal{W} via the inverse Jacobian $J^{-1}(p)$. By design, the resultant vector field precludes collisions between the robot and the workspace's inner obstacles and renders the

goal configuration almost globally attractive except for a set of measure zero initial configurations. However, since \mathcal{W}_0 may not be repulsive under ψ for an arbitrary, fixed selection of k , we also introduce the adaptive law $f_k(p, k)$ which, along with $s(p, k)$, guarantee forward invariance of the workspace without compromising the convergence and stability properties of the overall system. The following subsections elaborate on each component of the proposed control law individually.

2.4.1 Artificial Harmonic Potential Fields

Following a procedure similar to the one proposed in [62], we construct an artificial harmonic potential field on the disk space \mathcal{D} from point sources placed at the desired configuration $q_d = T(p_d)$ as well as at the points $q_i = T(\partial\mathcal{W}_i)$, $\forall i \in \mathcal{I}_N$ that correspond to the inner obstacles, as follows:

$$\phi = k_d \ln \left(\frac{\|q - q_d\|}{2} \right) - \sum_{i=1}^N k_i \ln \left(\frac{\|q - q_i\|}{2} \right) \quad (2.17)$$

where $k_d > 0$ and $k_i \geq 0$ denote harmonic source strengths which vary according to adaptive laws that are presented later. An interesting property of the above potential field, which stems from the maximum principle for harmonic functions, is that, for fixed k , the only minima of ϕ are located at q_d and, possibly, at infinity. As a direct consequence of this property, the hessian $\nabla_q^2 \phi$ computed at a non-degenerate critical point of ϕ in our domain's interior, has one positive and one negative eigenvalue with the same magnitude, e.g., λ and $-\lambda$ with $\lambda > 0$.

Next, we define a reference potential field ψ based on ϕ , which is given by

$$\psi = \frac{1 + \tanh(\phi/w)}{2} \quad (2.18)$$

where w is a positive scaling constant. Note that ψ maps the extended real line to the closed interval $[0, 1]$. As $\tanh(\phi/w)$ is a strictly increasing function, the only critical points of ψ are the ones inherited from ϕ with their indices preserved. Furthermore, the gradient of ψ w.r.t. q , given by:

$$\nabla_q \psi = \frac{1 - (\tanh(\phi/w))^2}{2w} \nabla_q \phi \quad (2.19)$$

is well-defined and bounded for all $q \in \mathcal{D}$.

If the workspace was radially unbounded, selecting k fixed with $k_d > \sum_{i=1}^N k_i$ would render the potential field (18) sufficient for navigation. The author in [62] addresses bounded workspaces that are diffeomorphic to sphere worlds by simply mapping the outer bounding circle to infinity. For reasons that will become apparent later, we would like to be able to place q_d on regions of $\partial\mathcal{D}$ that are not physically occupied by obstacles (such as passages to other compartments, see for example

Subsection 2.5.2), thus we cannot follow the same procedure since that would render the effect of the sole attractor on the robot null. Instead, we design appropriate adaptive laws for the parameters k of ϕ , to render the outer boundary repulsive and establish the forward completeness of the proposed scheme for all time.

Before proceeding with the definition of the adaptive law, we first state two propositions that will be used in the subsequent analysis, the proofs of which can be found in the Appendix.

Proposition 1. *Let $k_d > 0$ and $q' \in \partial\mathcal{D} \setminus \{q_d\}$. There exists $k' > 0$ such that if $k_i < k'$, $\forall i \in \mathcal{I}_N$, then q' is repulsive under ψ .*

Proposition 2. *If k_i are non-negative and bounded, there exists $k'_d > 0$ such that ψ is Morse for all $k_d \geq k'_d$.*

2.4.2 Adaptive Laws

We now present the adaptive law $\dot{k} = f_k(q, k)$ that updates the parameters of the potential field ψ . It's primary goal is to render: a) the workspace outer boundary repulsive and b) any critical point of ϕ in the vicinity of the robot non-degenerate; a property that will be used later in the analysis. In particular, we consider f_k of the form:

$$\begin{aligned} \dot{k}_d &= \zeta_1(\lambda + \|\nabla_q \phi\|; \epsilon_1) \\ \dot{k}_i &= (\bar{k}_i - k_i) w_i \ell_i g_i - \mathcal{K}_k k_i h_i w_0 (g_0 + \zeta_1(s; \epsilon_2)), \quad \forall i \in \mathcal{I}_N \end{aligned} \quad (2.20)$$

where w_i and g_i , $i \in \mathcal{I}_N^*$, as well as h_i , $i \in \mathcal{I}_N$, are functions to be defined later, \bar{k}_i , $i \in \mathcal{I}_N$, are desired upper bounds for k_i , λ denotes the non-negative eigenvalue of $\nabla_q^2 \phi$, \mathcal{K}_k is a positive control gain, and ϵ_1, ϵ_2 are small positive constants. The continuously differentiable switch $\zeta_1(x; \epsilon)$ and functions $\ell_i(q)$ are respectively given by:

$$\zeta_1(x; \epsilon) = 1 - \sigma_p(x/\epsilon) \quad (2.21)$$

$$\ell_i(q) = -\frac{\mathcal{K}_i s(q, k)}{\ln\left(\frac{\|q - q_i\|}{2}\right)}. \quad (2.22)$$

According to **Proposition 1**, our first requirement can be accomplished by designing f_k to reduce k_i as the robot approaches $\partial\mathcal{D}$. To do so without compromising the inherent inner obstacle collision avoidance properties of ϕ , we need to also ensure that each k_i does not vanish within some neighborhood of q_i , for all $i \in \mathcal{I}_N$. To that end, firstly we define g_i , employing the smoothly vanishing function defined in (16), to serve as pseudo-metrics of the alignment between the robot's velocity and the directions towards the goal and inner obstacles, respectively, given by:

$$g_i = \sigma_v(\bar{g}_i), \quad \forall i \in \mathcal{I}_N^* \quad (2.23)$$

with

$$\begin{aligned}\bar{g}_0 &= \frac{1}{4} \left(\alpha \|\nabla_q \psi\| \|q - q_d\| - (\nabla_q \psi)^T (q - q_d) \right) \\ \bar{g}_i &= \frac{1}{2} (\nabla_q \psi)^T (q - q_i), \quad \forall i \in \mathcal{I}_N\end{aligned}\tag{2.24}$$

where $\alpha \in (0, 1]$ is a fixed constant that is used for selecting the desired alignment between the robot's motion and the direction to the goal. We also define the accompanying weights w_i as follows to ensure that only one term of (2.20) dominates as the robot approaches a particular boundary of \mathcal{W} :

$$\begin{aligned}w_0 &= \frac{\zeta_2(\bar{w}_0; \epsilon_3)}{\bar{w}_0 + \sum_{j=1}^N (\bar{k}_j \bar{w}_j)} \\ w_i &= \frac{\bar{w}_i}{\bar{w}_0 + \sum_{j=1}^N (\bar{k}_j \bar{w}_j)}, \quad \forall i \in \mathcal{I}_N\end{aligned}\tag{2.25}$$

with

$$\bar{w}_i = \bar{r}_i / (r_i + \bar{r}_i), \quad \forall i \in \mathcal{I}_N^*,\tag{2.26}$$

$$r_i = \begin{cases} (1 - \|q\|)^2, & \text{if } i = 0, \\ \|q - q_i\|^2, & \text{if } i \in \mathcal{I}_N \end{cases}\tag{2.27}$$

$$\bar{r}_i = \sqrt[m]{\sum_{j \neq i} (r_j)^m}, \quad \forall i \in \mathcal{I}_N\tag{2.28}$$

$$\zeta_2(x; \epsilon) = \begin{cases} 0, & \text{if } x < \epsilon \\ \left(\frac{x-\epsilon}{1-\epsilon}\right)^2 (3 - 2\frac{x-\epsilon}{1-\epsilon}), & \text{if } \epsilon \leq x \leq 1 \\ 1, & \text{otherwise} \end{cases}\tag{2.29}$$

for a scalar constant $\epsilon_3 \in (0, 1)$ in (2.25) and some integer $m < -1$ in (2.28) that serves as a smooth under-approximation of $\min_{j \neq i} (r_j), i \in \mathcal{I}_N$. Finally, the weights $h_i, i \in \mathcal{I}_N$ are defined as follows:

$$h_i = 1 + \frac{\sigma_v(\bar{h}_i)}{1 + \sum_{j \in \mathcal{I}_N} \sigma_v(\bar{h}_j)}\tag{2.30}$$

with

$$\bar{h}_i = k_i \left(1 - (\tanh(\phi/w))^2\right)^2 \left(\frac{q_d - q}{\|q_d - q\|^2}\right)^T \frac{q_i - q}{\|q_i - q\|^2}\tag{2.31}$$

whose purpose is to accelerate the decay of those k_i that contribute the most to the component of $\nabla_q \psi$ that pushes the robot toward the workspace's outer boundary.

Regarding the second requirement, as shown in **Proposition 1**, selecting k_d above a certain threshold is sufficient to render ϕ free of degenerate equilibria. On the other hand, for a given \bar{k}_i , increasing k_d steers the robot closer to the workspace's inner obstacles. Nevertheless, since the robot may never actually enter the vicinity of a degenerate equilibrium, instead of setting k_d sufficiently large a priori, the adaptive

law for the parameter k_d is introduced to increase k_d only when it is actually needed, thus potentially alleviating the aforementioned shortcoming.

2.4.3 Stability Analysis

Let us consider the overall system:

$$\dot{z} = f_z(z) \quad (2.32)$$

where $z = (q, k)$, $f_z(z) = (f_q, f_k)$ with $f_q = Ju$. Also, let Ω denote the image of \mathcal{W} under T , i.e., $\Omega = T(\mathcal{W})$. Notice that Ω consists of $\text{int}(\mathcal{D})$, possibly with a subset of $\partial\mathcal{D}$, and with the points q_i removed. In this section, we elaborate on the stability properties of (2.32) under the proposed control scheme (2.13) and (2.20). First, we formalize the safety properties of the closed-loop system dynamics, which guarantee that our robot does not collide with any obstacle.

Proposition 3. *The workspace \mathcal{W} is invariant under the dynamics (2.32) with control laws (2.13) and (2.20), i.e., $p(t) \in \mathcal{W}$, for all $t \geq 0$.*

Proof. For the proof, refer to the Appendix. □

Having eliminated the possibility of the robot colliding with the workspace's boundaries, we proceed by showing that all critical points of ψ , where (2.32) may converge to, are either non-degenerate saddles or q_d . Additionally, we show that the latter is a stable equilibrium.

Proposition 4. *The artificial potential ψ is decreasing along the trajectories of the closed-loop system and its time derivative vanishes only at its critical points. Additionally, the preimage of q_d is a set of stable equilibria of (2.2).*

Proof. For the proof, refer to the Appendix. □

Proposition 5. *Let $z^* = (q^*, k^*)$ be a critical point of the closed-loop system dynamics with $q^* \in \Omega \setminus \{q_d\}$. Then, q^* is a non-degenerate saddle point of ψ .*

Proof. For the proof, refer to the Appendix. □

Finally, we conclude this section with the main theoretical findings.

Theorem 2. *The control law (2.13) and (2.20) addresses Problem 1, i.e., system (2.2) under the control law (2.13) and (2.20) converges safely to q_d , for almost all initial configurations.*

Proof. For the proof, refer to the Appendix. □

2.5 Extensions

In this section we present certain extensions of the proposed approach to: a) address the safe navigation problem for unicycle robots which are frequently encountered in many application domains, and b) tackle computational complexity issues that affect the numerical computation of the harmonic map presented in [Section 2.3](#) as the size of the workspace increases.

2.5.1 Unicycle Robot Kinematics

In this subsection, we consider robots whose motion is subjected to pfaffian constraints of the form:

$$\begin{aligned}\dot{p} &= \hat{n}_\theta v \\ \dot{\theta} &= \omega\end{aligned}\tag{2.33}$$

where $\theta \in [0, 2\pi)$ denotes the robot's orientation, $\hat{n}_\theta = [\cos(\theta), \sin(\theta)]^T$, and $v, \omega \in \mathbb{R}$ are control inputs corresponding to the robot's linear and angular velocities, respectively. First, let us define the robot's kinematics in the image of the configuration space via the proposed transformation, as follows:

$$\begin{aligned}\dot{q} &= \hat{n}_{\tilde{\theta}} \tilde{v} \\ \dot{\tilde{\theta}} &= \tilde{\omega}\end{aligned}\tag{2.34}$$

and notice that the robot's orientations θ and $\tilde{\theta}$ are related by:

$$\hat{n}_{\tilde{\theta}} = \frac{J(p)\hat{n}_\theta}{\|J(p)\hat{n}_\theta\|}.\tag{2.35}$$

To safely drive the robot to its goal configuration, we consider the following control laws:

$$\begin{aligned}\tilde{v} &= -\mathcal{K}_v s_v(q, \tilde{\theta}, k) (\hat{n}_{\tilde{\theta}})^T \nabla_q \psi(q, k) \\ \tilde{\omega} &= -\mathcal{K}_\omega (\hat{n}_{\tilde{\theta}}^\perp)^T \nabla_q \psi(q, k)\end{aligned}\tag{2.36}$$

with $\mathcal{K}_v, \mathcal{K}_\omega \in \mathbb{R}$ positive constant gains, $\hat{n}_{\tilde{\theta}}^\perp = [-\sin(\tilde{\theta}), \cos(\tilde{\theta})]^T$ and

$$\begin{aligned}s_v(q, \tilde{\theta}, k) &= \gamma \sigma_p \left(\frac{1 - \|q\|}{\epsilon_p} \right) + \\ & (1 - \gamma) \sigma_v \left(\frac{(\hat{n}_{\tilde{\theta}}^T \nabla_q \psi) \hat{n}_{\tilde{\theta}}^T q}{\epsilon_v + |\hat{n}_{\tilde{\theta}}^T \nabla_q \psi| \|q\|} \right).\end{aligned}\tag{2.37}$$

Additionally, we need to employ a modified version of the adaptive laws for the potential field's parameters, which is obtained by substituting s with s_v in [\(2.20\)](#)

and (2.22) and $\bar{g}_i, i \in \mathcal{I}_N^*$, with

$$\begin{aligned}\bar{g}_{v,0} &= \frac{1}{4} \left(\alpha \left| \hat{n}_\theta^T \nabla_q \psi \right| \|q - q_d\| - \left(\hat{n}_\theta^T \nabla_q \psi \right) \hat{n}_\theta^T (q - q_d) \right) \\ \bar{g}_{v,i} &= \frac{1}{2} \left(\hat{n}_\theta^T \nabla_q \psi \right) \hat{n}_\theta^T (q - q_i), \quad \forall i \in \mathcal{I}_N\end{aligned}\tag{2.38}$$

respectively, in (2.23). Finally, by expressing the aforementioned control laws to the robot's actual configuration space, we obtain:

$$\begin{aligned}v &= v\tilde{v} \\ \omega &= \omega_{dq} + \omega_{d\tilde{\theta}}\end{aligned}\tag{2.39}$$

where ω_{dq} and $\omega_{d\tilde{\theta}}$ are terms corresponding to angular velocities induced by translational and rotational motion of the robot in the workspace's image, respectively, given by:

$$\begin{aligned}\omega_{dq} &= -\tilde{v}v^2 \left(\left(J \begin{bmatrix} \hat{n}_\theta & v\hat{n}_\theta^\perp \end{bmatrix} \right)^{-1} \frac{\partial}{\partial \hat{n}_\theta} J \hat{n}_\theta \right)^T \begin{bmatrix} 0 \\ 1 \end{bmatrix} \\ \omega_{d\tilde{\theta}} &= \tilde{\omega} \left(\left(J \begin{bmatrix} \hat{n}_\theta & v\hat{n}_\theta^\perp \end{bmatrix} \right)^{-1} \hat{n}_\theta^\perp \right)^T \begin{bmatrix} 0 \\ 1 \end{bmatrix}\end{aligned}\tag{2.40}$$

with $v = \|J^{-1}\hat{n}_\theta\|$ and $\frac{\partial}{\partial \hat{n}_\theta} J$ denoting the directional derivative of J along \hat{n}_θ .

The stability properties of the aforementioned closed-loop system dynamics are formalized below:

Theorem 3. *The workspace \mathcal{W} is invariant under the dynamics of (2.33) equipped with the proposed control law. Additionally, the robot will asymptotically converge either to an interior critical point of ϕ or to the pre-image of q_d , which is stable.*

Proof. For the proof, refer to the Appendix. □

Remark 2. *The result of Theorem 3 is weaker compared to that of Theorem 2 since there is no guarantee that the set of configurations which converge to a critical point of ϕ (other than the pre-image of q_d) has Lebesgue measure zero.*

2.5.2 Atlas of Harmonic Maps

As the size of the workspace increases, the problem of computing the transformation T grows in complexity as well, because the resources required by commonly employed numerical techniques that can solve the problem presented in Section 2.3 are polynomial in the number of elements used for representing \mathcal{W} . Alternatively, to cope with large workspaces efficiently, we propose instead the construction of an atlas $\mathcal{A} \triangleq \{(\mathcal{P}_i, T_i) \mid i \in \mathcal{I}_{N_A}\}$ obtained by separating the workspace \mathcal{W} into N_A overlapping subsets $\mathcal{P}_i \subset \mathcal{W}$, such that $\bigcup_{i \in \mathcal{I}_{N_A}} \mathcal{P}_i = \mathcal{W}$ and constructing a separate harmonic map T_i for each \mathcal{P}_i (see Figure 2.4). This essentially allow us to

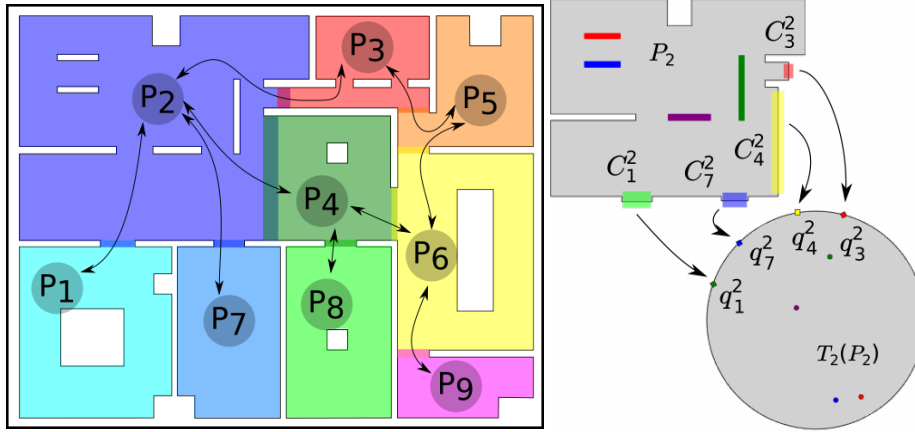


FIGURE 2.4: The partition of a complex workspace into overlapping subsets along with the corresponding graph and the transformation T_2 of the second partition \mathcal{P}_2 .

solve many small (and computationally less intensive) problems instead of a large one, thus reducing the overall resources required for addressing a given workspace. Therefore, given such a partitioning of \mathcal{W} , we define the graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ where $\mathcal{V} = \{\mathcal{P}_i \mid i \in \mathcal{I}_{N_A}\}$ denotes the set of corresponding nodes (workspace partitions) and \mathcal{E} denotes the set of edges between the elements of \mathcal{V} , with each edge indicating a feasible transition from one partition to another, i.e., $(i, j) \in \mathcal{E}$ if and only if $(\text{cl}(\mathcal{P}_i) \cap \text{cl}(\mathcal{P}_j)) \neq \emptyset$. Notice that \mathcal{G} is undirected by definition, i.e., $(i, j) \in \mathcal{E}$ only if $(j, i) \in \mathcal{E}$. Additionally, since the workspace is connected, \mathcal{G} should also be connected. Thus, for a given atlas \mathcal{A} , an initial configuration p_{init} and a final configuration p_d , we can employ standard graph search algorithms to obtain a sequence of indices $\mathcal{S} = \{s_1, s_2, \dots, s_n\}$ corresponding to partitions that the robot can traverse to reach its goal.⁷ Additionally, note that, since the partitioning of \mathcal{W} does not need to be fine, the size of \mathcal{G} will generally be small, rendering the cost of finding \mathcal{S} negligible.

We now concentrate on how the transition between two consecutive elements of \mathcal{S} is implemented. Let $\mathcal{C}_{i,j} \triangleq \text{cl}(\mathcal{P}_i) \cap \text{cl}(\mathcal{P}_j)$ denote the common region of $\text{cl}(\mathcal{P}_i)$ and $\text{cl}(\mathcal{P}_j)$ and $\mathcal{B}_{i,j} \triangleq \partial\mathcal{P}_i \cap \mathcal{P}_j$ denote the set of points on the boundary of \mathcal{P}_i that also belong to \mathcal{P}_j and are not occupied by obstacles, for all $i \in \mathcal{I}_{N_A}$ and all j such that $(i, j) \in \mathcal{E}$. Without loss of generality, we assume that \mathcal{A} is constructed such that the sets $\mathcal{B}_{\ell,i} \cap \mathcal{B}_{\ell,j}$ are either empty or consist of isolated points. We note that in order to successfully complete the transition between two consecutive nodes \mathcal{P}_i and \mathcal{P}_j of \mathcal{S} , it suffices for the robot to reach any single point of $\mathcal{B}_{i,j}$ from \mathcal{P}_i . We also observe that each $\mathcal{B}_{i,j}$ may consist of one or more disjoint components $\mathcal{B}_{i,j}^\ell$, $\ell \in \mathcal{L}(i, j)$, with $\mathcal{L}(i, j)$ being some valid indexing of those. By exploiting the fact that **Theorem 1** imposes a weak homeomorphism requirement on \bar{T}_i , we can construct each T_i such that each disjoint subset of $\partial\mathcal{P}_i$ collapses into a separate point, i.e., $T_i(\mathcal{B}_{i,j}^\ell) = q_{i,j}^\ell \in \partial\mathcal{D}$ (see

⁷In general, more than one such sequence of partitions may exist connecting the initial and the final configurations. However, the selection of one that corresponds to some sort of “optimal” path is beyond the scope of this work.

Figure 2.4), which, in turn, implies that selecting $q_{i,j}^\ell$ as an intermediate goal configuration suffices to render the entire $\mathcal{B}_{i,j}^\ell$ attractive. Building upon this fact, for each consecutive pair of \mathcal{P}_i and \mathcal{P}_j in \mathcal{S} , we (arbitrarily) select a $\mathcal{B}_{i,j}^\ell$ and we construct a transformation $T_i : \mathcal{P}_i \mapsto \mathcal{D}$, with $q^{[i]} = T_i(p)$, and artificial potential field $\phi_i(q^{[i]}, k^{[i]})$ with goal configuration $q_d^{[i]} = q_{i,j}^\ell$. Additionally, to smooth the transition between consecutive partitions, when they overlap, we propose the following modified control law for the robot:

$$u = u^{[i]} + \eta_{c,i,j} \cdot \eta_{t,i,j} \cdot u^{[j]}, \quad \forall p \in \mathcal{C}_{i,j} \quad (2.41)$$

where $u^{[i]}$ and $u^{[j]}$ denote the control inputs as defined in (2.13) and evaluated using ψ_i, T_i and ψ_j, T_j , respectively, the function $\eta_{t,i,j} : \mathcal{C}_{i,j} \mapsto [0, 1]$ is any smooth bump function such that

$$\eta_{t,i,j}(p) = \begin{cases} 0, & \text{if } p \in \mathcal{B}_{j,i} \\ 1, & \text{if } p \in \mathcal{B}_{i,j} \end{cases} \quad (2.42)$$

and

$$\eta_{c,i,j}(p, k^{[i]}, k^{[j]}) = \begin{cases} \frac{(\zeta_{i,j})^2}{\epsilon_4 + (\zeta_{i,j})^2}, & \text{if } \zeta_{i,j} \geq 0 \\ 0, & \text{if } \zeta_{i,j} < 0 \end{cases} \quad (2.43)$$

with $\zeta_{i,j} = (\nabla_p \phi_i)^T \cdot (\nabla_p \psi_j)$ and $\epsilon_4 > 0$ being a fixed parameter. What this modification essentially does is to incrementally add an extra component, with the direction of $\nabla_p \psi_j$, to the robot's velocity when that component is cosine similar⁸ with $\nabla_p \psi_i$. We note that $\eta_{c,i,j} \rightarrow 1$ and $\eta_{t,i,j} \rightarrow 1$ as the robot approaches the boundary of the corresponding partition. We also remark that once the robot has completed its transition to \mathcal{P}_j , we do not concern ourselves with $u^{[i]}$ anymore, i.e., $u = u^{[j]}$ even if p returns to $\mathcal{C}_{i,j}$. The overall scheme employed for navigating a holonomic robot to its goal configuration using an atlas constructed as described above can be seen in [Algorithm 1](#).

Regarding the stability analysis of the modified system, by following the same procedure as in [Subsection 2.4.3](#) and by virtue of $\eta_{c,i,j}$, it is trivial to verify the following statement.

Theorem 4. *System (2.2) equipped with [Algorithm 1](#) converges safely to a given goal configuration $p_d \in \mathcal{W}$ from almost all initial configurations $p_{\text{init}} \in \mathcal{W}$.*

Proof. For the proof, refer to the Appendix. □

⁸Two vectors u and v are cosine similar if their inner product is positive.

Algorithm 1 Atlas-based motion planning scheme for a holonomic robot.

Require: $\mathcal{A}, p_{\text{init}}, p_d$
 $\mathcal{S} \leftarrow \text{FINDPATHTOGOAL}(\mathcal{G}, p_{\text{init}}, p_d)$
Initialize $k^{[s]}$ for all $s \in \mathcal{S}$.
for all i in \mathfrak{I}_{n-1} **do**
 $s, s' \leftarrow s_i, s_{i+1}$
 Select (arbitrary) ℓ such that $\ell \in \mathcal{L}(s, s')$.
 Place goal configuration of ψ_s at $q_{s,s'}^\ell$.
end for
Place goal configuration of ψ_{s_n} at $T_{s_n}(p_d)$.
 $\ell \leftarrow 1$
loop
 if $\ell = n$ or $p \in \mathcal{P}_{s_\ell} \setminus \mathcal{P}_{s_{\ell+1}}$ **then**
 Update p using (2.13) and $k^{[s_\ell]}$ using (2.20).
 else if $p \in \mathcal{C}_{s_\ell, s_{\ell+1}}$ **then**
 Update p using (2.41) with $i = s_\ell$ and $j = s_{\ell+1}$.
 Update $k^{[s_\ell]}$ and $k^{[s_{\ell+1}]}$ using (2.20).
 else
 $\ell \leftarrow \ell + 1$
 end if
end loop

2.6 Simulations and Experimental Results

In order to demonstrate the efficacy of the proposed control scheme, we have conducted various simulation and experimental studies, the results of which are presented in this section. The algorithm that computes the harmonic transformation and its Jacobian was implemented in C++ while the proposed control protocols were implemented in Python. All simulations were carried out on a PC with an Intel i5 processor operating at 2.2Ghz, 4GB RAM, running a GNU/Linux operating system.

2.6.1 Simulations — Full Workspace Transformation

In the first case study, a single transformation of the entire workspace (see Figure 2.4) was constructed and the robot was instructed to navigate to various goal configurations starting from the same initial position. The initial configuration and the parameters of our controller were selected such as to better demonstrate the guaranteed collision avoidance properties of our scheme. Particularly, the initial values for the parameters of the adaptive law were selected as $k_d = 20$, $k_i = 1$ and $\bar{k}_i = 20$ for all $i \in \mathfrak{I}_{10}$. The values of the remaining parameters were $\mathcal{K}_u = 100$, $w = 20$, $\mathcal{K}_k = 100$, $\alpha = 1$, $\epsilon_p = 0.025$, $\epsilon_v = 0.1$, $\gamma = 0.7$, $\epsilon_1 = 0.01$, $\epsilon_2 = 0.1$, $\epsilon_3 = 0.1$. The goal configurations and the trajectories executed by the robot, both in the real and transformed workspace, are illustrated in Figure 2.5. Furthermore, Figure 2.6 depicts the potential ψ of each trajectory. The simulations were conducted using the Euler method with 10ms step. Regarding the computational complexity of the control scheme, the

construction of the harmonic transformation for this large workspace that was carried out offline once, required 5.4s to complete, for a sufficient approximation of the workspace boundary with 3680 segments. Finally, the online computation of the transformation $T(p)$ and its Jacobian $J(p)$ required an average of 6.0ms per step.

2.6.2 Simulations — Atlas of Harmonic Maps

In this case study, we partitioned the aforementioned workspace into separate rooms (see [Figure 2.4](#)) and constructed a harmonic transformation T_i for each room. The robot initialized at the same position with the previous study and it was instructed to navigate towards the same set of individual goal configurations. The initial values selected for the parameters of the adaptive law were $k^{[i]} = N^{[i]} + 3$, $k_j^{[i]} = 1$ and $\bar{k}_j^{[i]} = k_j^{[i]}$ for all $j \in \mathcal{J}_{N^{[i]}}$ and $i \in \mathcal{J}_{N_A}$, where $N^{[i]}$ denotes the amount of obstacles inside the corresponding partition. All remaining control parameters were selected as in [Subsection 2.6.1](#). The trajectories of the robot are depicted in [Figure 2.7](#). The time spend to construct the corresponding harmonic transformations varied from 0.019s to 0.211s (depending on the amount of elements required for sufficiently approximating each room, ranging between 320 and 1000 segments) and was significantly much less than the full map construction of the previous case. Additionally, the online computation of $T_i(p)$ and $J_i(p)$ in each of these rooms required an average time between 1.0ms and 2.2ms per step, respectively. Finally, it should be noted that in this case the workspace inner obstacles were mapped to points further away from the boundaries of the partitions, which is an interesting result as it alleviates possible numerical issues that may arise in the computation of the transformation near the obstacles.

2.6.3 Experiments

In order to verify the results presented in [Subsection 2.5.1](#), real experiments were conducted on a non-holonomic robotic platform “Robotnik Summit-XL” (see [Figure 2.8](#)) operating within the workspace that is depicted in [Figure 2.9](#). The boundaries of the workspace were obtained using readily available SLAM algorithms and were later augmented with the robot’s shape (approximated by a disk). The workspace was partitioned into six overlapping subsets and the robot was instructed to visit three different goal configurations, each located in a different room. An off-the-shelf localization algorithm was employed for estimating the robot’s position and orientation using its on-board sensors (laser scanners and RGB-D cameras), providing feedback at approximately 5 Hz to the robot’s linear and angular commanded velocities. Notice that our algorithm successfully managed to drive the robot safely to its specified goal configurations, as one can verify from the trajectories of the robot executed in the real workspace (see [Figure 2.10](#), [Figure 2.12](#), [Figure 2.14](#)), as well as in the corresponding transformed workspace partitions (see [Figure 2.11](#), [Figure 2.13](#),

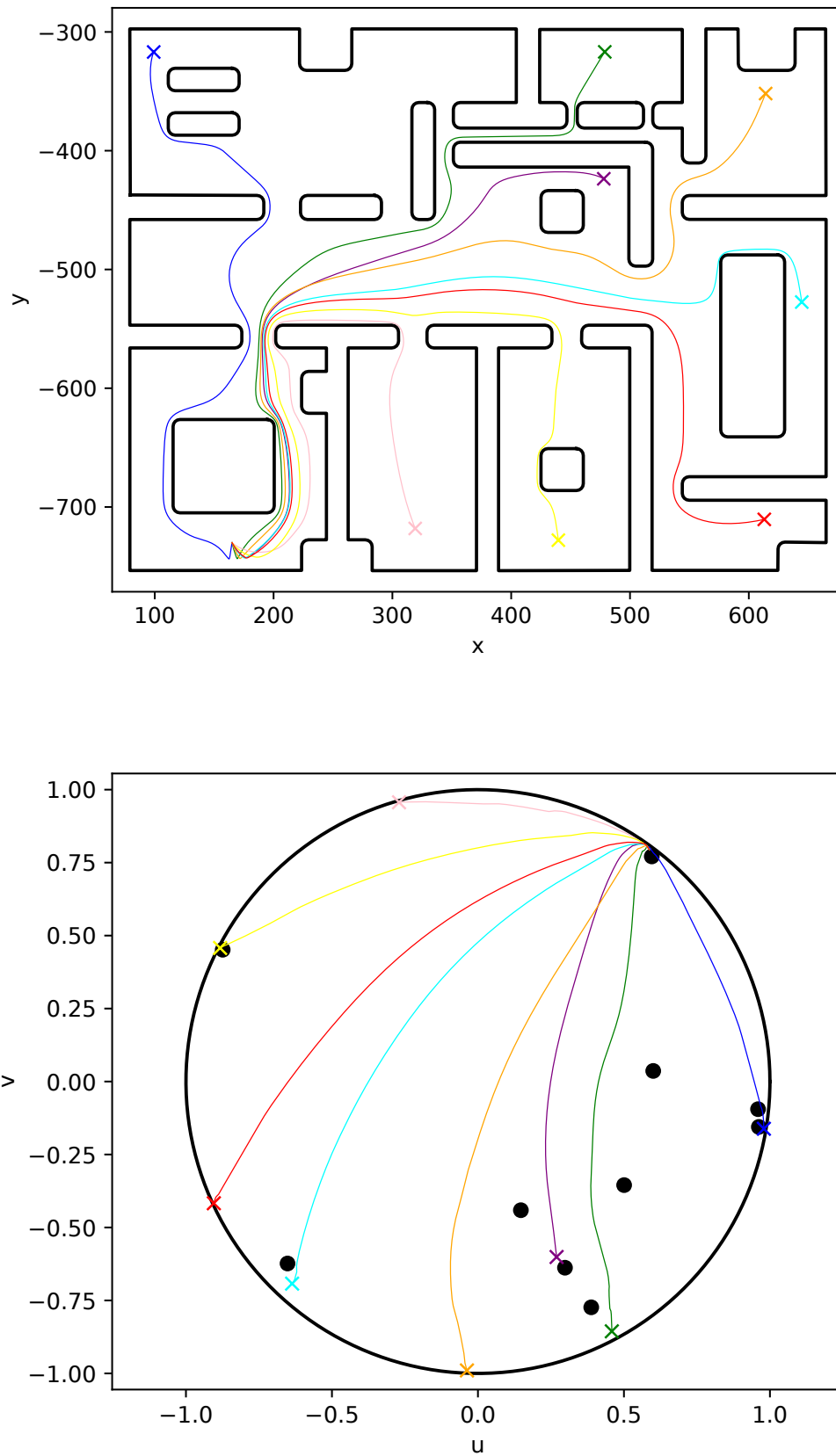


FIGURE 2.5: Simulation results using a full workspace transformation. The robot starts from an arbitrary location and navigates to different goal configurations (colored crosses). The resulted trajectories are plotted in both real and transformed workspaces with different color.

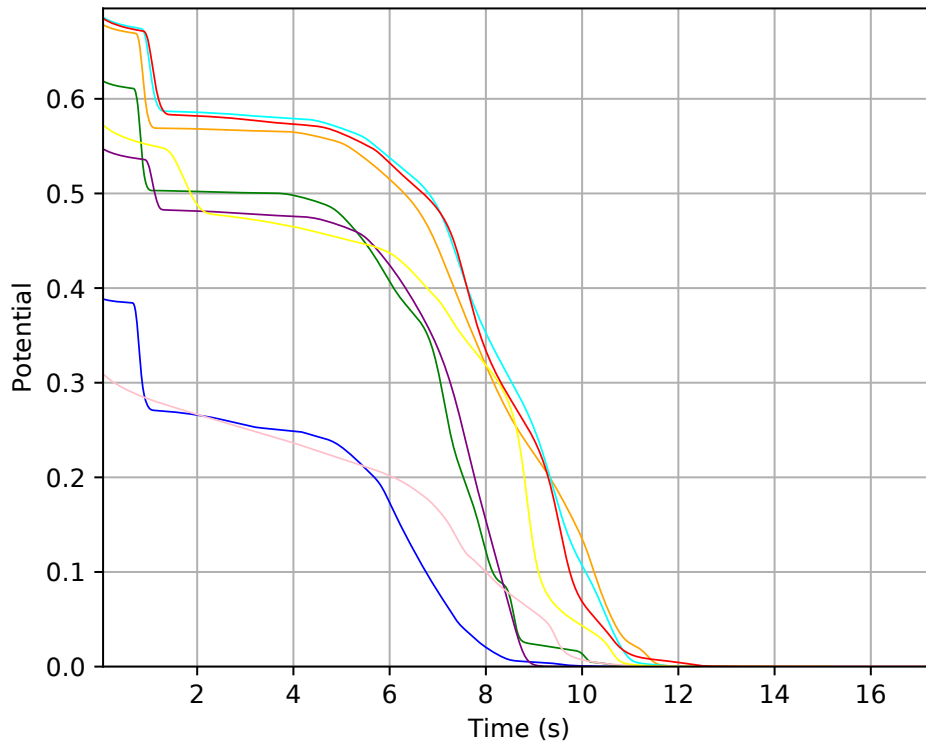
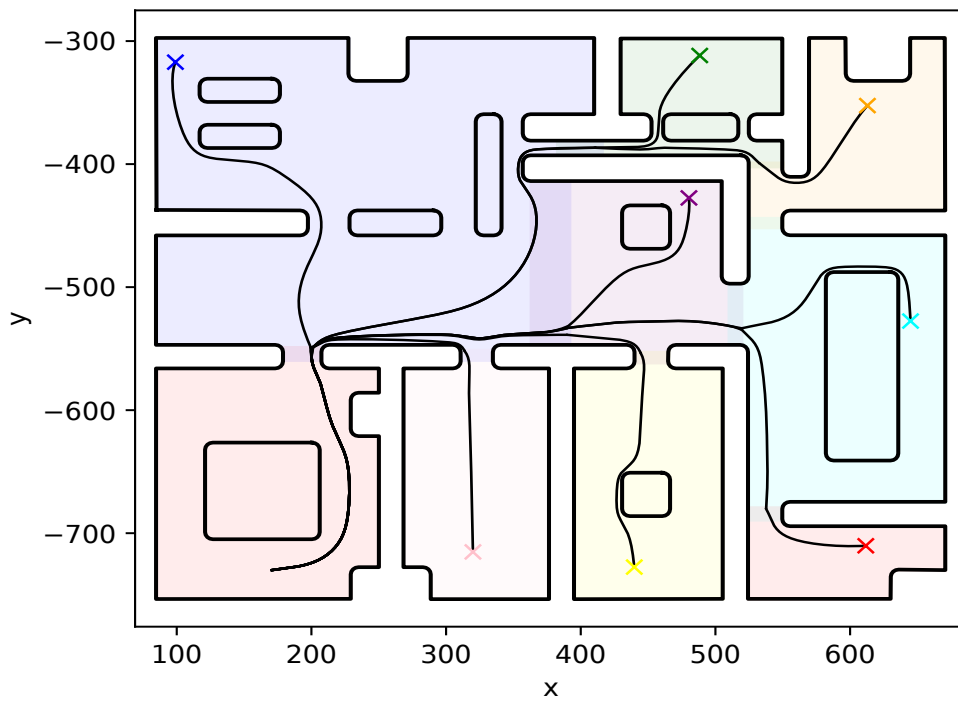
FIGURE 2.6: The evolution of ψ along the robot trajectories.

FIGURE 2.7: The resulted robot trajectories using an atlas of the workspace.



FIGURE 2.8: Robotnik Summit XL

Figure 2.15). However, an issue that needs to be pointed out is the oscillating behavior that the robot exhibited in the configuration space's image — particularly, in p_1 and p_2 of ???. Such behavior is attributed both to: a) the relative slow update of the robot's pose estimation, and b) the inversion of the Jacobian which is generally ill-conditioned close to narrow passages of the domain. Nevertheless, such shortcomings can be alleviated by a better choice of partitions, i.e., by partitioning the domain into more subsets with less complex shapes.



FIGURE 2.9: Map of the robot's workspace used during the experiments, obtained using SLAM.

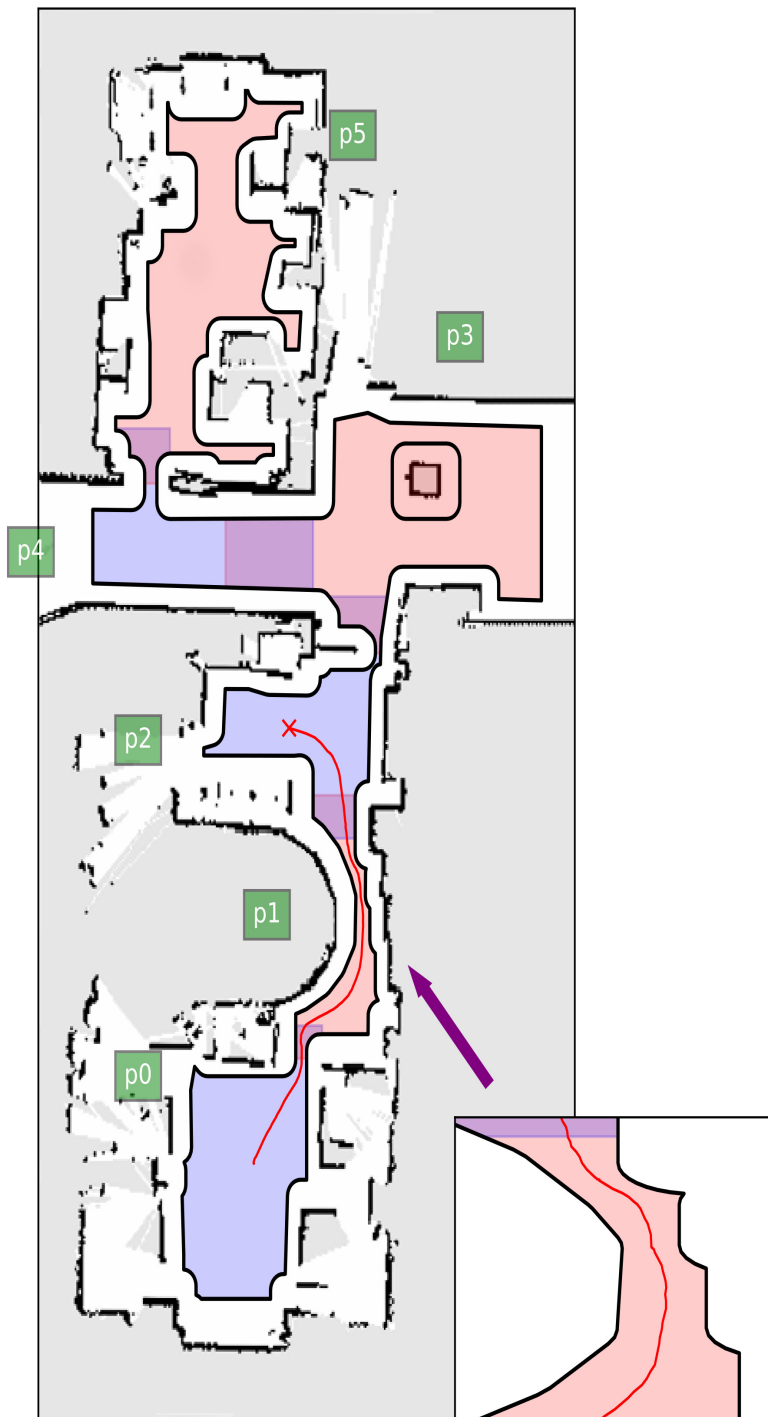


FIGURE 2.10: Trajectory of the unicycle robot in the real workspace executed during the first experimental case.

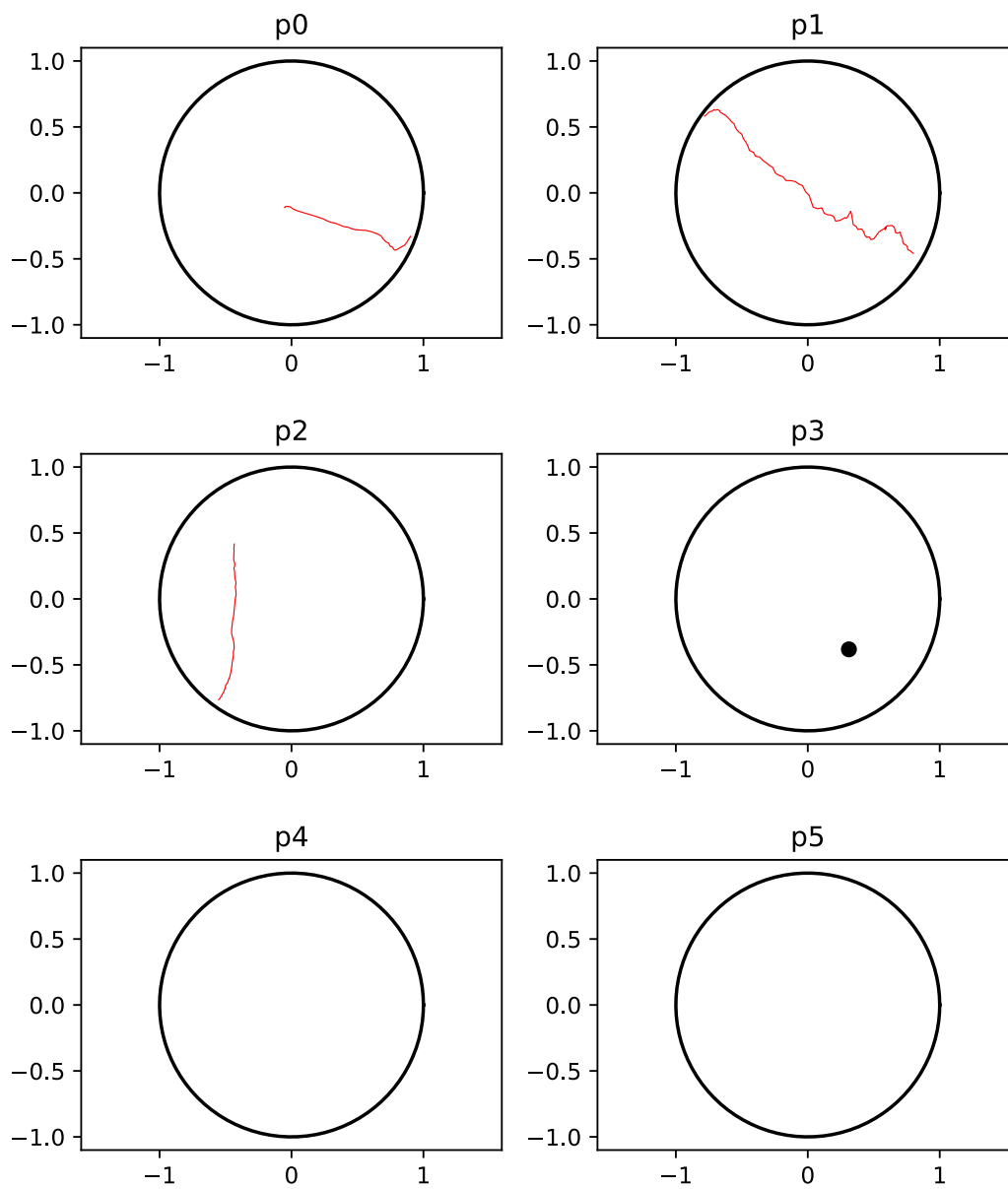


FIGURE 2.11: Trajectory of the unicycle robot in the transformed workspace's partitions executed during the first experimental case.

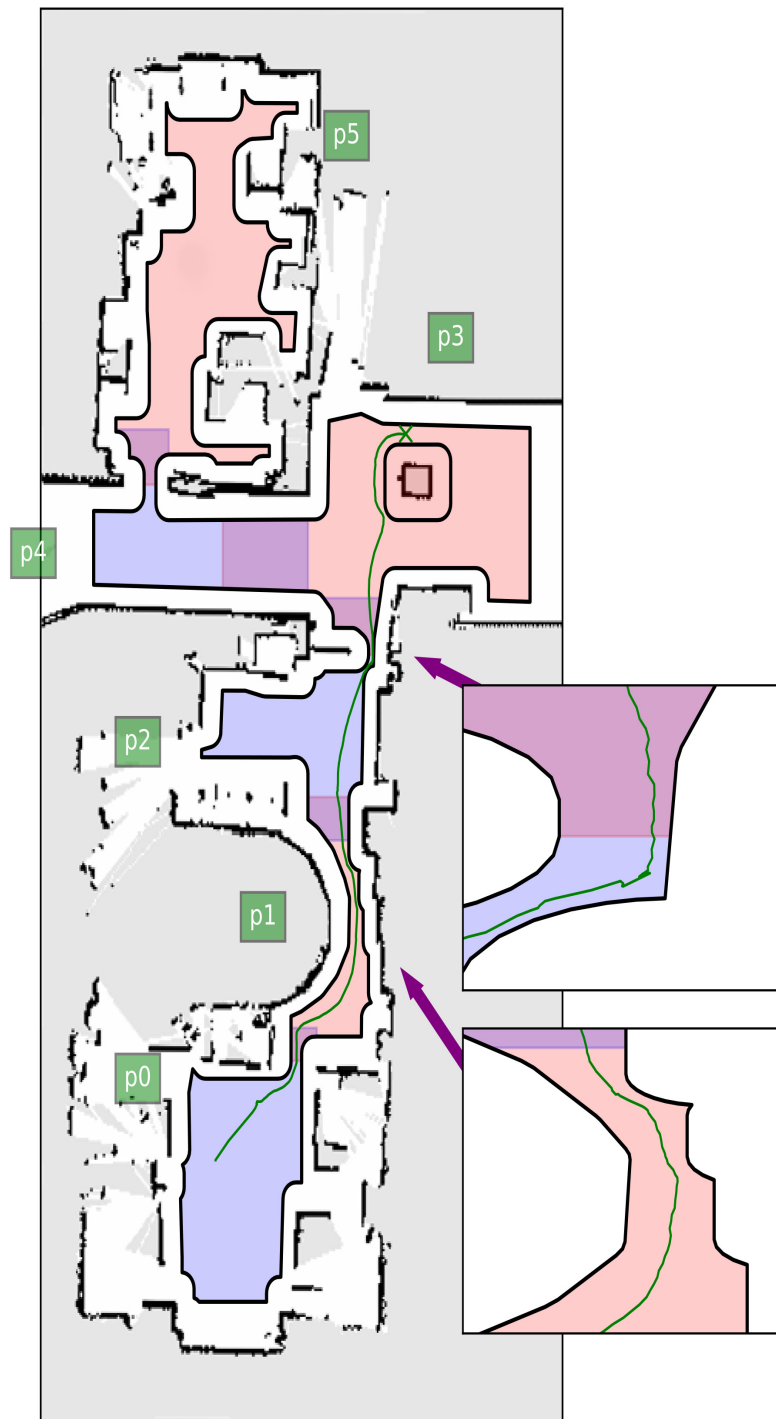


FIGURE 2.12: Trajectory of the unicycle robot in the real workspace executed during the second experimental case.

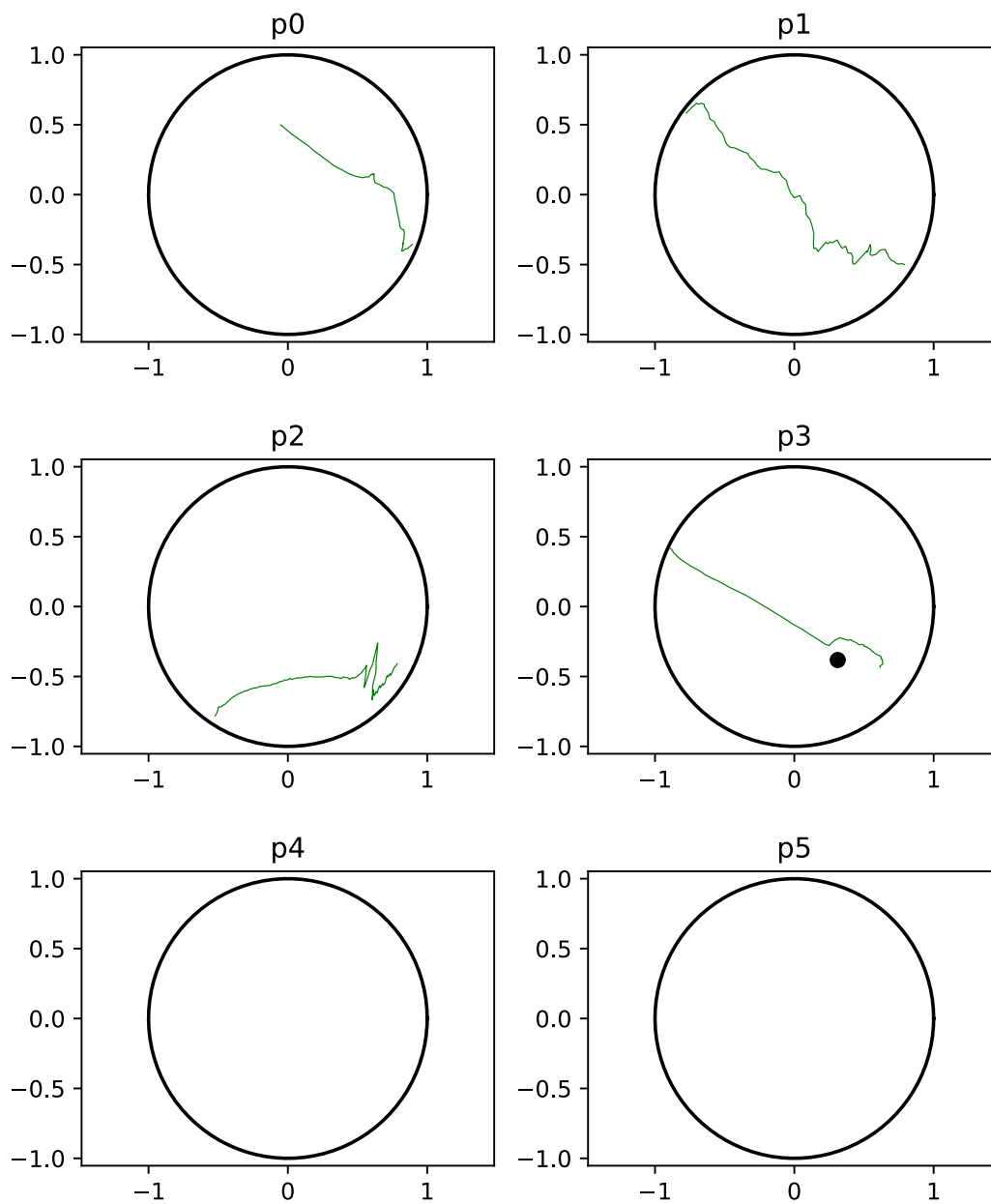


FIGURE 2.13: Trajectory of the unicycle robot in the transformed workspace's partitions executed during the second experimental case.

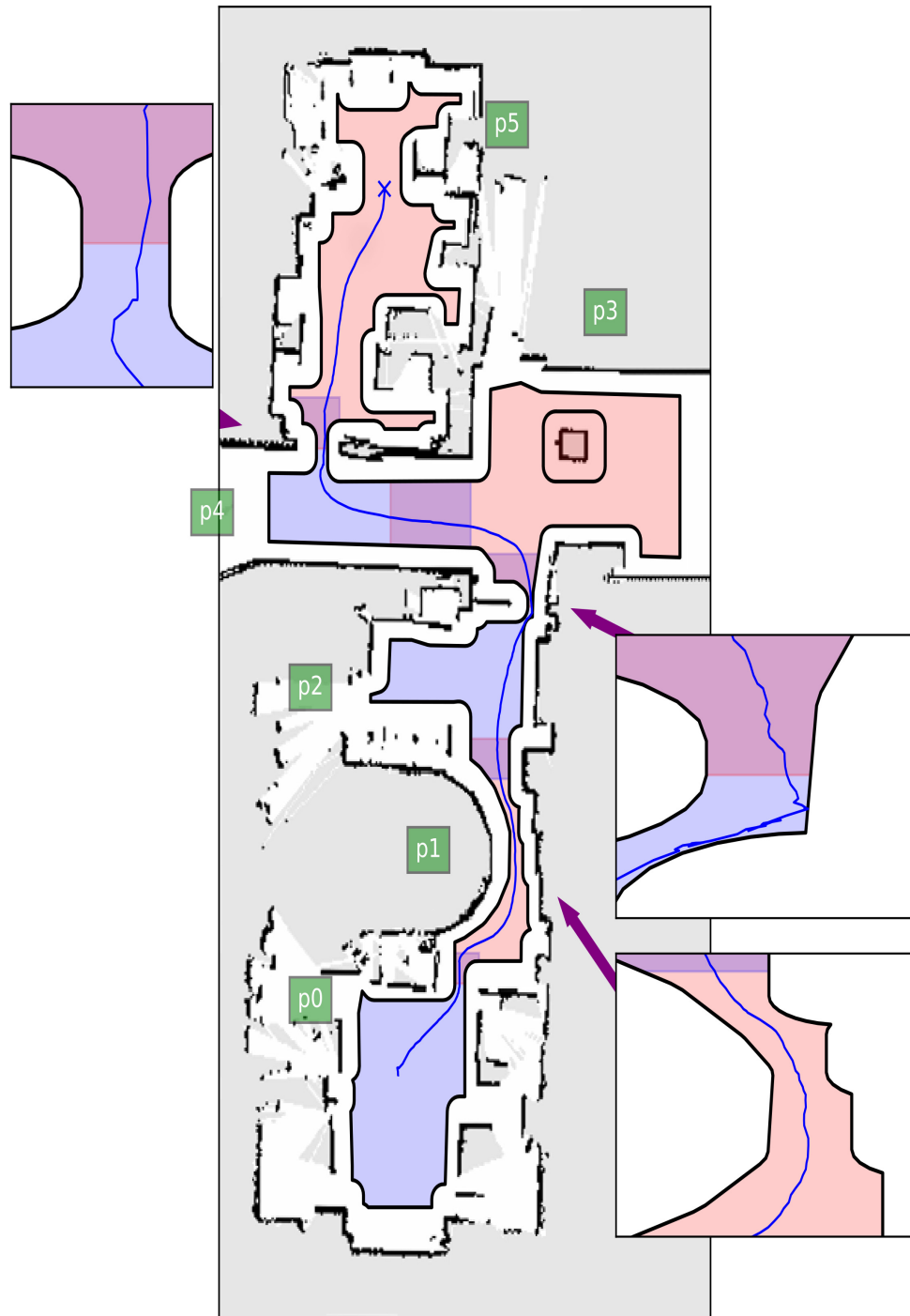


FIGURE 2.14: Trajectory of the unicycle robot in the real workspace executed during the third experimental case.

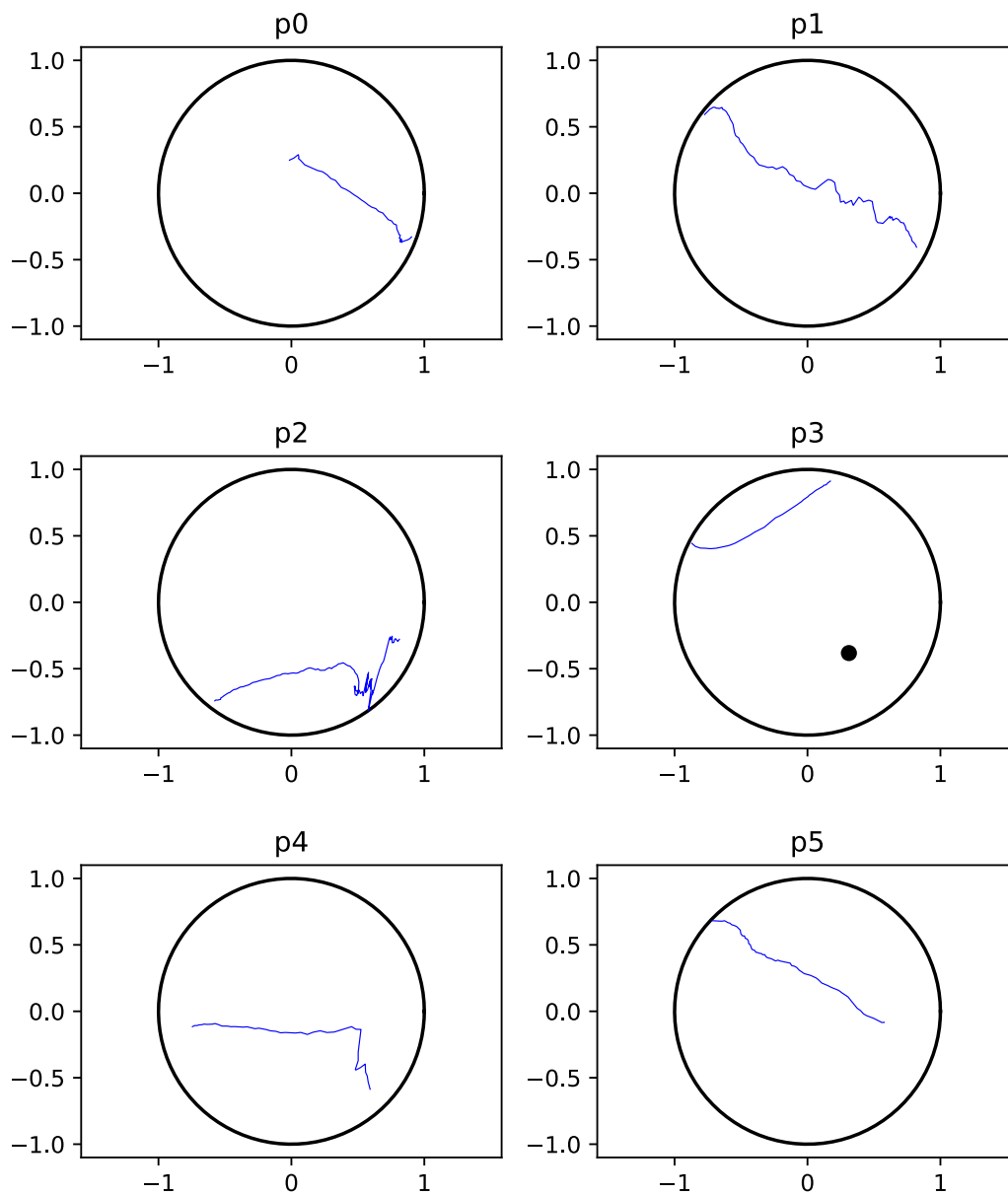


FIGURE 2.15: Trajectory of the unicycle robot in the transformed workspace's partitions executed during the third experimental case.

Chapter 3

Navigation of Polygonal-Shaped Robots on the Plane

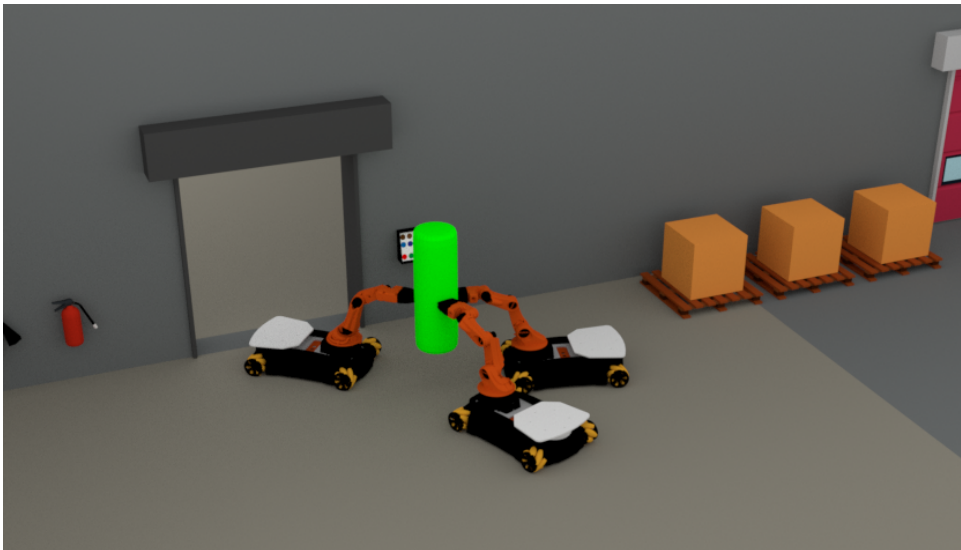


FIGURE 3.1: Team of mobile manipulators forming a virtual, non-convex robotic platform.

3.1 Introduction

In this chapter, a hybrid control scheme is presented in order to address the navigation problem for a planar robotic platform of arbitrary shape that is moving inside an obstacle cluttered workspace. Given an initial and desired robot configuration, we propose a methodology based on approximate configuration space decomposition techniques that makes use of heuristics to adaptively refine a partition of the configuration space into non-overlapping, adjacent slices. Furthermore, we employ appropriate workspace transformations and adaptive potential field based control laws that integrate elegantly with the type of configuration space representation used, in order to safely navigate within a given cell and successfully cross over to the next, for almost all initial configurations, until the desired configuration is reached.

Finally, we present simulation results that demonstrate the efficacy of the proposed control scheme.

The outline of this chapter is as follows. At the end of this section, we define some preliminary notions and notation used throughout this chapter. In [Section 4.2](#), we explicitly formulation the problem addressed in this chapter. In [Section 4.3](#), we elaborate on the proposed approximate cell decomposition methodology as well as the hybrid control law designed for ensuring safe transitions between sequentially reachable sets of goal configurations. An analysis of the completeness of the proposed control scheme is given in [Section 3.4](#). Finally, simulation results demonstrating the efficacy of the proposed control scheme are presented in [Section 4.4](#).

Notation: Given a set \mathcal{S} , let $\partial\mathcal{S}$ denote that set's boundary. We also define \mathcal{D} and $\partial\mathcal{D}$ as the disk and circle with unit radius centered at the origin of \mathbb{R}^2 , respectively. Given two sets A and B , we use $\text{cl}(A)$, $\text{int}(A)$, A/B and $A \ominus B$ to denote the closure of A , the interior of A , the component of B not belonging in A and the Minkowski difference of A from B , respectively. Finally, given a vector function $f(x)$ we use $\nabla_x f$ to denote its Jacobian matrix.

3.2 Problem Formulation

We consider a robot moving within a compact workspace $\mathcal{W} \subset \mathbb{R}^2$ occupied by a set of disjoint, fixed inner obstacles \mathcal{O}_i , $i \in \mathcal{I}_o \triangleq \{1, 2, \dots, N_o\}$. We assume that the robot's boundary $\partial\mathcal{R}$, the workspace's outer boundary $\partial\mathcal{W}_0$ and the boundaries $\partial\mathcal{W}_i$ of the static obstacles are polygonal Jordan curves, where $\partial\mathcal{W}_i \triangleq \partial\mathcal{O}_i$, $\forall i \in \mathcal{I}_o$. Furthermore, the robot \mathcal{R} is assumed to be a closed, 2-manifold of \mathbb{R}^2 , whereas the workspace \mathcal{W} is assumed to be a closed subset of \mathbb{R}^2 . We will use $\mathcal{F}_{\mathcal{W}}$ and $\mathcal{F}_{\mathcal{R}}$ to denote two arbitrarily embedded coordinate frames in the workspace and the robot, respectively, and we will refer to the origin of $\mathcal{F}_{\mathcal{R}}$ as the reference point of the robot. Let $z = [x, y, \theta]^T$ denote the configuration of the robot, specifying the relative position $p = [x, y]^T \in \mathbb{R}^2$ and the orientation $n = [\cos \theta, \sin \theta]^T$, $\theta \in [0, 2\pi)$, of $\mathcal{F}_{\mathcal{R}}$ with respect to $\mathcal{F}_{\mathcal{W}}$. A single integrator model characterizes the robot's oriented motion as follows:

$$\begin{aligned} \dot{p} &= u \\ \dot{\theta} &= \omega \end{aligned} \tag{3.1}$$

where $u \in \mathbb{R}^2$ is the control input vector corresponding to the robot's linear velocity and $\omega \in \mathbb{S}^1$ is the desired angular velocity.

The robot's configuration space \mathcal{CS} is a manifold diffeomorphic to $\mathbb{R}^2 \times \mathbb{S}^1$ and each obstacle \mathcal{O}_i corresponds to a closed region \mathcal{O}_i^c inside that, given by $\mathcal{O}_i^c \triangleq \{z | \mathcal{R}(z) \cap \mathcal{O}_i \neq \emptyset\}$, where $\mathcal{R}(z)$ denotes the subset of \mathcal{W} occupied by the robot at configuration z (for brevity, $\mathcal{R}(\theta)$ will be used instead of $\mathcal{R}([0, 0, \theta])$ wherever is

deemed preferable). The robot's free configuration space $\mathcal{CS}_{\text{free}}$ is thus defined by $\mathcal{CS}_{\text{free}} \triangleq \mathcal{CS} / \cup_{i \in \mathcal{I}_o} \mathcal{O}_i^c$.

Our goal is to design a control law that can successfully drive the robot to a given goal configuration $z_{\text{final}} \in \mathcal{CS}_{\text{free}}$ for almost all initial configurations $z_{\text{init}} \in \mathcal{CS}_{\text{free}}$, while ensuring that the robot's trajectory remains safe, i.e., $p(t) \in \mathcal{CS}_{\text{free}}$ for all $t \geq 0$.

3.3 Control Design

To address the aforementioned problem, we design a hybrid control scheme which consists of: a) a high-level controller that given an initial (z_{init}) and final (z_{final}) configuration, can compute a sequence of reachable intermediate goals for the robot, if a solution to the above problem exists, or determine its infeasibility otherwise, and b) a low-level controller which utilizes harmonic based artificial potential fields and appropriate workspace transformations in order to safely navigate the robot to each of these sequentially reachable configurations. More specifically, the high-level controller, presented in [Subsection 3.3.1](#), constructs a partitioning of $\mathcal{CS}_{\text{free}}$ by adaptively subdividing S^1 until a sequence Π of connected cells containing z_{init} and z_{final} is found (if one exists). Then, for each cell in Π , goals for the robot's position p and orientation θ are computed separately, as described in [Subsection 3.3.2](#), and pairs of independent low-level controllers using harmonic transformations and adaptive artificial potential fields, described in [Subsection 3.3.3](#), are employed.

3.3.1 Configuration Space Decomposition

In this section, we provide a detailed description of the proposed methodology for constructing a hierarchical approximation of the robot's configuration space. More specifically, we propose an algorithm for partitioning $\mathcal{CS}_{\text{free}}$ into non-overlapping slices which are obtained by adaptively subdividing the set of valid robot's orientations. A free configuration space's slice $\mathcal{S}^{\mathcal{I}}$ corresponding to an interval $\mathcal{I} = [a, b)$ of rotation angles is defined by $\mathcal{S}^{\mathcal{I}} \triangleq \{(x, y, \theta) | \theta \in \mathcal{I}\} \cap \mathcal{CS}_{\text{free}}$. In general, $\mathcal{S}^{\mathcal{I}}$ consists of zero or more disjoint sets of robot configurations with arbitrary connectedness which will be referred henceforth to as cells. Thus, we associate to each slice $\mathcal{S}^{\mathcal{I}}$ an indexing $\mathcal{K}^{\mathcal{I}}$ of its cells $\mathcal{C}_k^{\mathcal{I}}$ and we will use $\mathcal{L}(\mathcal{S}^{\mathcal{I}})$ to denote the set of its cells, i.e., $\mathcal{C}_k^{\mathcal{I}} \in \mathcal{L}(\mathcal{S}^{\mathcal{I}}), \forall k \in \mathcal{K}^{\mathcal{I}}$.

Two slices $\mathcal{S}^{\mathcal{I}}$ and $\mathcal{S}^{\mathcal{J}}$ are said to be adjacent if $\text{cl}(\mathcal{I}) \cap \text{cl}(\mathcal{J}) \neq \emptyset$. Additionally, we will say that two cells $\mathcal{C}^{\mathcal{I}}$ and $\mathcal{C}^{\mathcal{J}}$, belonging to adjacent slices, are connected if $\text{cl}(\mathcal{C}^{\mathcal{I}}) \cap \text{cl}(\mathcal{C}^{\mathcal{J}}) \neq \emptyset$. In order to avoid the implications introduced by the geometric complexity of \mathcal{CS} , we will also construct and use two additional, simpler approximations of the free configuration space, namely an over-approximation $\overline{\mathcal{CS}}_{\text{free}}$ and an under-approximation $\underline{\mathcal{CS}}_{\text{free}}$, which are made similarly of slices and cells and the purpose of each will become apparent later in this section.

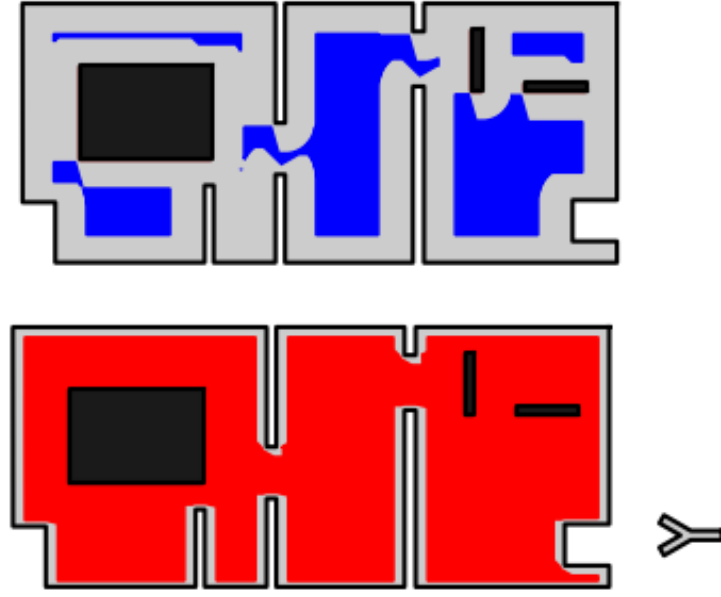


FIGURE 3.2: $\overline{\mathcal{W}}^{\mathcal{I}}$ (blue colored regions) and $\underline{\mathcal{W}}^{\mathcal{I}}$ (red colored regions) of the free space available to the Y-shaped robot (bottom-right) for $\mathcal{I} = [315^\circ, 360^\circ)$.

The method devised in this chapter for constructing the aforementioned approximations requires the existence of two functions, denoted by \bar{f} and \underline{f} , for computing an over-approximation $\overline{\mathcal{S}}^{\mathcal{I}}$ and an under-approximation $\underline{\mathcal{S}}^{\mathcal{I}}$ of the free space corresponding to a given slice $\mathcal{S}^{\mathcal{I}}$. Apart from $\underline{\mathcal{S}}^{\mathcal{I}} \subseteq \mathcal{S}^{\mathcal{I}} \subseteq \overline{\mathcal{S}}^{\mathcal{I}}$, these functions must be selected such that $\bar{f}(\mathcal{S}^{\mathcal{I}}) \triangleq \overline{\mathcal{S}}^{\mathcal{I}} \rightarrow \mathcal{S}^{\mathcal{I}}$ and $\underline{f}(\mathcal{S}^{\mathcal{I}}) \triangleq \underline{\mathcal{S}}^{\mathcal{I}} \rightarrow \mathcal{S}^{\mathcal{I}}$ as the length of \mathcal{I} goes to zero. A simple approach for constructing such sets is by first computing appropriate approximations $\overline{\mathcal{W}}^{\mathcal{I}}$ and $\underline{\mathcal{W}}^{\mathcal{I}}$ of the free space of $\mathcal{S}^{\mathcal{I}}$ projected on the xy -plane, and then extruding along the θ -axis. Thus, the sets $\overline{\mathcal{S}}^{\mathcal{I}}$ and $\underline{\mathcal{S}}^{\mathcal{I}}$ are given by

$$\begin{aligned} \overline{\mathcal{S}}^{\mathcal{I}} &= \{[x, y, \theta]^T \mid [x, y]^T \in \overline{\mathcal{W}}^{\mathcal{I}}, \theta \in \mathcal{I}\} \\ \underline{\mathcal{S}}^{\mathcal{I}} &= \{[x, y, \theta]^T \mid [x, y]^T \in \underline{\mathcal{W}}^{\mathcal{I}}, \theta \in \mathcal{I}\} \end{aligned} \quad (3.2)$$

In order to compute $\overline{\mathcal{W}}^{\mathcal{I}}$ and $\underline{\mathcal{W}}^{\mathcal{I}}$, several algorithms can be employed, such as the *Projection Method* [122] or the *Swept-Area Method* [122]. In this chapter, the later was preferred due to its simplicity and computational efficiency over the former. As such, $\overline{\mathcal{W}}^{\mathcal{I}}$ and $\underline{\mathcal{W}}^{\mathcal{I}}$ can be computed using

$$\begin{aligned} \overline{\mathcal{W}}^{\mathcal{I}} &= \mathcal{W} \ominus \overline{\mathcal{R}}(\mathcal{I}), \\ \underline{\mathcal{W}}^{\mathcal{I}} &= \mathcal{W} \ominus \underline{\mathcal{R}}(\mathcal{I}), \end{aligned} \quad (3.3)$$

where $\overline{\mathcal{R}}(\mathcal{I})$, $\underline{\mathcal{R}}(\mathcal{I})$ are sets given by

$$\begin{aligned}\overline{\mathcal{R}}(\mathcal{I}) &= \{p \mid \exists \theta \in \mathcal{I} : p \in \mathcal{R}(\theta)\}, \\ \underline{\mathcal{R}}(\mathcal{I}) &= \{p \mid \forall \theta \in \mathcal{I} : p \in \mathcal{R}(\theta)\},\end{aligned}\tag{3.4}$$

representing the subsets of \mathbb{R}^2 that the robot occupies either for at least one or for every orientation in \mathcal{I} , respectively (see Figure 3.2). The notions of adjacency and connectedness defined above for exact slices can also be easily extended for their over- and under-approximations.

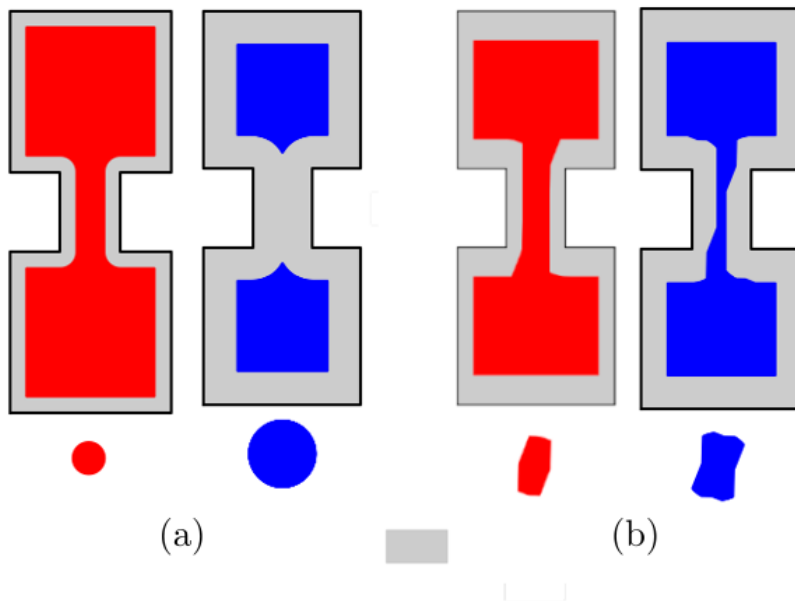


FIGURE 3.3: Workspace consisting of two rooms connected via a corridor. The $\overline{\mathcal{W}}^{\mathcal{I}}$ (blue colored regions) and $\underline{\mathcal{W}}^{\mathcal{I}}$ (red colored regions) were constructed via the Swept-Area method for (a) $\mathcal{I} = [0^\circ, 360^\circ)$ and (b) $\mathcal{I} = [67.5^\circ, 90^\circ)$. Corresponding $\underline{\mathcal{R}}(\mathcal{I})$ and $\overline{\mathcal{R}}(\mathcal{I})$ of the rectangular robot (bottom) can be seen below each workspace.

At this point, it should be noted that as one subdivides the configuration space into more and more slices, the corresponding approximations $\overline{\mathcal{C}\mathcal{S}}_{\text{free}}$ and $\underline{\mathcal{C}\mathcal{S}}_{\text{free}}$ constructed using the methodology described above converge to $\mathcal{C}\mathcal{S}_{\text{free}}$, which ensures that this approach is resolution complete. Thus, by sufficiently subdividing $\mathcal{C}\mathcal{S}_{\text{free}}$, a sequence of under-approximation cells connecting any given two configurations of the robot will be found, as long as one exists in the first place (see Figure 3.3). Instead of blindly subdividing $\mathcal{C}\mathcal{S}_{\text{free}}$ in search of such sequence, we propose an alternative algorithm, based on the approach presented in [122], which makes also use of the space's over-approximation to provide a heuristic for choosing the next slice for subdivision that can potentially yield a desired path. Additionally, it is also able to recognize whether such a path even exists.

Let \mathcal{P} be a set consisting of \mathcal{N} angle intervals $\mathcal{I}_1, \mathcal{I}_2, \dots, \mathcal{I}_{\mathcal{N}}$. We will say that \mathcal{P} is a partition of $\mathcal{C}\mathcal{S}_{\text{free}}$ if $\mathcal{S}^{\mathcal{I}_1}, \mathcal{S}^{\mathcal{I}_2}, \dots, \mathcal{S}^{\mathcal{I}_{\mathcal{N}}}$ are non-overlapping and $\mathcal{C}\mathcal{S}_{\text{free}} = \bigcup_{k=1}^{\mathcal{N}} \mathcal{S}^{\mathcal{I}_k}$.

Additionally, given a partition \mathcal{P} , a path Π of cells is defined as a finite sequence of cells such that $\mathcal{I}_k \in \mathcal{P}$ and the cells of each consecutive pair $(\mathcal{C}_k^{\mathcal{I}_k}, \mathcal{C}_{k+1}^{\mathcal{I}_{k+1}})$ are connected for all $k = 1, 2, \dots, N$. One can verify from the definitions of $\overline{\mathcal{S}}$ and $\underline{\mathcal{S}}$ that, for any given partition \mathcal{P} of \mathcal{CS} and a pair of initial z_{init} and final z_{final} configurations, the following statements hold:

1. If there exists a path of under-approximation cells in \mathcal{P} containing these configurations, then a solution to our problem exists.
2. If there exists a path of over-approximation cells in \mathcal{P} containing these two configurations, then our problem may admit a solution.
3. If there is no path of over-approximation cells in \mathcal{P} containing these two configurations, then our problem is unsolvable.

We can now present [Algorithm 2](#) that is employed for finding a path of under-approximation cells connecting a given initial and final configuration of the robot. We begin by approximating $\mathcal{CS}_{\text{free}}$, $\overline{\mathcal{CS}}_{\text{free}}$, $\underline{\mathcal{CS}}_{\text{free}}$ by slices spanning the whole \mathbb{S}^1 . Then, we search for cells $\underline{\mathcal{C}}_{\text{init}}^{\mathcal{I}}$ and $\underline{\mathcal{C}}_{\text{final}}^{\mathcal{J}}$ of $\underline{\mathcal{CS}}_{\text{free}}$, potentially subdividing \mathcal{P} that contain the robot's initial and final configurations, respectively. If no such pair of cells exists, indicated by either $\overline{\mathcal{C}}_{\text{init}}^{\mathcal{I}}$ and $\overline{\mathcal{C}}_{\text{final}}^{\mathcal{J}}$ being empty, our initial problem is obviously infeasible and the algorithm terminates. Otherwise, we try to connect $\underline{\mathcal{C}}_{\text{init}}^{\mathcal{I}}$ and $\underline{\mathcal{C}}_{\text{final}}^{\mathcal{J}}$ using the under-approximation cells in \mathcal{P} . If this attempt fails, then we try instead to find a path of over-approximation cell connecting $\overline{\mathcal{C}}_{\text{init}}^{\mathcal{I}}$ and $\overline{\mathcal{C}}_{\text{final}}^{\mathcal{J}}$. If such a path cannot be found, this also implies that no solution exists and the algorithm terminates. Otherwise, a heuristic is utilized for selecting an interval in \mathcal{P} to be subdivided and the process starts anew.

The heuristic used, which can be seen in [Algorithm 3](#), selects which interval of \mathcal{P} to subdivide by trying to form a new path using cells belonging in $\underline{\mathcal{CS}}_{\text{free}}$ which are also subsets of the cells of $\overline{\mathcal{CS}}_{\text{free}}$ that belong to the sequence given to it. After isolating adjacent slices containing cells that cannot be connected, the largest corresponding angle interval gets then subdivided and the function terminates.

3.3.2 Hybrid Control Scheme

Given now a path Π of cells obtained by [Algorithm 2](#), we describe how to build a sequence of low-level controllers which can safely navigate the robot, from one cell to another, starting from z_{init} , till the goal configuration z_{final} is reached. Let $\mathcal{C}^{\mathcal{I}}$ be a cell in Π and let $\mathcal{P}(\mathcal{C}^{\mathcal{I}})$ denote the projection of $\mathcal{C}^{\mathcal{I}}$ on the xy -plane. It should be noted that $\mathcal{P}(\overline{\mathcal{C}}^{\mathcal{I}}) = \overline{\mathcal{W}}^{\mathcal{I}}$ and $\mathcal{P}(\underline{\mathcal{C}}^{\mathcal{I}}) = \underline{\mathcal{W}}^{\mathcal{I}}$. We begin by noting that by construction $\mathcal{C}^{\mathcal{I}}$ is an under-approximation of the actual free configuration space such that

$$\mathcal{C}^{\mathcal{I}} = \{[x, y, \theta]^T \mid [x, y]^T \in \mathcal{P}(\mathcal{C}^{\mathcal{I}}) \text{ and } \theta \in \mathcal{I}\}. \quad (3.5)$$

Algorithm 2 Algorithm that searches for a path of cells containing specified configurations.

```

function CONNECTCONFIGS( $z_{\text{init}}, z_{\text{final}}$ )
   $\mathcal{P} \leftarrow \{[0, 2\pi)\}$ 
  loop
     $\mathcal{P}, \underline{\mathcal{C}}_{\text{init}}^{\mathcal{I}}, \bar{\mathcal{C}}_{\text{init}}^{\mathcal{I}} \leftarrow \text{FINDUACELL}(z_{\text{init}}, \mathcal{P})$ 
     $\mathcal{P}, \underline{\mathcal{C}}_{\text{final}}^{\mathcal{J}}, \bar{\mathcal{C}}_{\text{final}}^{\mathcal{J}} \leftarrow \text{FINDUACELL}(z_{\text{final}}, \mathcal{P})$ 

    if  $\bar{\mathcal{C}}_{\text{init}}^{\mathcal{I}} = \emptyset$  or  $\bar{\mathcal{C}}_{\text{final}}^{\mathcal{J}} = \emptyset$  then
      return None
    end if
     $\text{solution} \leftarrow \text{CONNECTUACELLS}(\underline{\mathcal{C}}_{\text{init}}^{\mathcal{I}}, \underline{\mathcal{C}}_{\text{final}}^{\mathcal{J}})$ 
    if  $\text{solution}$  is empty then
       $\text{path} \leftarrow \text{CONNECTOACELLS}(\bar{\mathcal{C}}_{\text{init}}^{\mathcal{I}}, \bar{\mathcal{C}}_{\text{final}}^{\mathcal{J}})$ 

      if  $\text{path}$  is empty then
        return None
      else
         $\mathcal{P} \leftarrow \text{REFINE}(\mathcal{P}, \text{path}, \{\underline{\mathcal{C}}_{\text{init}}^{\mathcal{I}}\}, \{\underline{\mathcal{C}}_{\text{final}}^{\mathcal{J}}\})$ 
      end if
    else
      return  $\text{solution}$ 
    end if
  end loop
end function

```

This implies that $\mathcal{P}(\mathcal{C}^{\mathcal{I}})$ is actually a non-empty, compact region of \mathbb{R}^2 with arbitrary connectedness and that the position and the orientation of the robot can change independently, as long as they both remain in $\mathcal{C}^{\mathcal{I}}$, without compromising the safety of the robot.

Exploiting this fact, we can decouple the low-level control laws of the robot's linear and angular velocities. Thus, for each intermediate cell of Π , all we need to do is find independent goal regions for the robot's position and desired ranges for its orientation that can ensure safe transition from that cell to the next. Let us consider a pair of consecutive cells $\mathcal{C}^{\mathcal{I}}$ and $\mathcal{C}^{\mathcal{J}}$ in Π . Regarding the robot's position, in order to safely cross from $\mathcal{C}^{\mathcal{I}}$ to $\mathcal{C}^{\mathcal{J}}$, it is sufficient that p reaches the set $\mathcal{G}(\mathcal{C}^{\mathcal{I}})$ which is defined as $\mathcal{G}(\mathcal{C}^{\mathcal{I}}) \triangleq \mathcal{P}(\mathcal{C}^{\mathcal{I}}) \cap \mathcal{P}(\mathcal{C}^{\mathcal{J}})$, which is not empty by construction. Moreover, $\mathcal{G}(\mathcal{C}^{\mathcal{I}})$ is generally made of one or more disjoint subsets of arbitrary connectedness. Therefore, as long as the robot's position reaches either of these subsets, the robot can cross to the next cell.

Let $\mathcal{F}(\mathcal{C}^{\mathcal{I}})$ denote all configurations of $\mathcal{C}^{\mathcal{I}}$ that do not belong to the goal set $\mathcal{G}(\mathcal{C}^{\mathcal{I}})$, i.e., $\mathcal{F}(\mathcal{C}^{\mathcal{I}}) \triangleq \mathcal{P}(\mathcal{C}^{\mathcal{I}}) \setminus \mathcal{P}(\mathcal{C}^{\mathcal{J}})$. When $p \in \mathcal{F}(\mathcal{C}^{\mathcal{I}})$, we need to design an appropriate control law for u that can safely drive p to a reachable subset of the boundary of $\mathcal{G}(\mathcal{C}^{\mathcal{I}})$. When more than one of such disjoint goal subsets are reachable from the

Algorithm 3 Heuristic choosing next slice for subdivision.

```

function REFINE( $\mathcal{P}$ ,  $path$ ,  $src$ ,  $dst$ )
  if  $len(path) = 1$  then
    return SUBDIVIDE( $\mathcal{P}$ ,  $path[0]$ )
  else
     $cells \leftarrow \{\}$ 
     $connected \leftarrow \{\}$ 
    if  $len(path) == 2$  then
       $cells \leftarrow dst$ 
    else
       $\bar{\mathcal{C}}^{\mathcal{J}} \leftarrow path[1]$ 
      for all  $\underline{\mathcal{C}}^{\mathcal{J}}$  in  $\mathcal{L}(\underline{\mathcal{S}}^{\mathcal{J}})$  do
         $cells \leftarrow cells \cup \{\underline{\mathcal{C}}^{\mathcal{J}}\}$ 
      end for
    end if
    for all  $\underline{\mathcal{C}}^{\mathcal{I}}$  in  $src$  do
      for all  $\underline{\mathcal{C}}^{\mathcal{J}}$  in  $cells$  do
        if  $\underline{\mathcal{C}}^{\mathcal{I}} \cap \underline{\mathcal{C}}^{\mathcal{J}} \neq \emptyset$  then
           $connected \leftarrow connected \cup \{\underline{\mathcal{C}}_k^{\mathcal{J}}\}$ 
        end if
      end for
    end for
    if  $|connected| > 0$  then
      return REFINE( $\mathcal{P}$ ,  $path[1:]$ ,  $connected$ ,  $dst$ )
    else
      return SUBDIVIDELONGEST( $\mathcal{P}$ ,  $\mathcal{I}$ ,  $\mathcal{J}$ )
    end if
  end if
end function

```

connected component of $\mathcal{F}(\mathcal{C}^{\mathcal{I}})$ that contains p , one of them is arbitrarily (though deterministically) selected and assigned as the goal set.¹ Let $\mathcal{F}_C(\mathcal{C}^{\mathcal{I}})$ and $\mathcal{G}_p(\mathcal{C}^{\mathcal{I}})$ denote the connected component of $\mathcal{F}(\mathcal{C}^{\mathcal{I}})$ containing p and the assigned goal set, respectively (see Figure 3.4). We consider the following cases of possible goal sets, which need to be addressed differently during the corresponding low-level control design:

1. $\mathcal{G}_p(\mathcal{C}^{\mathcal{I}})$ is a point in $\mathcal{C}^{\mathcal{I}}$.
2. $\mathcal{G}_p(\mathcal{C}^{\mathcal{I}})$ is an inner boundary of $\mathcal{F}_C(\mathcal{C}^{\mathcal{I}})$.
3. $\mathcal{G}_p(\mathcal{C}^{\mathcal{I}})$ is a contiguous, measurable subset of $\mathcal{F}_C(\mathcal{C}^{\mathcal{I}})$ outer boundary.
4. $\mathcal{G}_p(\mathcal{C}^{\mathcal{I}})$ is equal to $\mathcal{F}_C(\mathcal{C}^{\mathcal{I}})$ outer boundary.

Regarding the goal set corresponding to the robot's orientation, we note that it is equal to $cl(\mathcal{I}) \cap cl(\mathcal{J})$. At this point, for the given combination of $\mathcal{F}_C(\mathcal{C}^{\mathcal{I}})$ and $\mathcal{G}_p(\mathcal{C}^{\mathcal{I}})$, we can design a vector field based control law $f_p(p, k)$ for the robot's position and,

¹A more sophisticated approach for choosing goal regions could be employed but that exceeds the scope of the current work.

accordingly, a simple proportional control law $f_\theta(\theta)$ for its orientation, with overall guaranteed convergence and safety properties, as described in [Subsection 3.3.3](#). Finally, when $\mathcal{C}^{\mathcal{I}}$ corresponds to the terminal cell of Π , the aforementioned control law f_p and f_θ is employed once again for driving the robot to $z_{\text{final}} = (p_d, \theta_d)$ by simply setting $\mathcal{G}_p(\mathcal{C}^{\mathcal{I}}) = \{p_d\}$ and $\mathcal{G}_\theta(\mathcal{C}^{\mathcal{I}}) = \{\theta_d\}$.

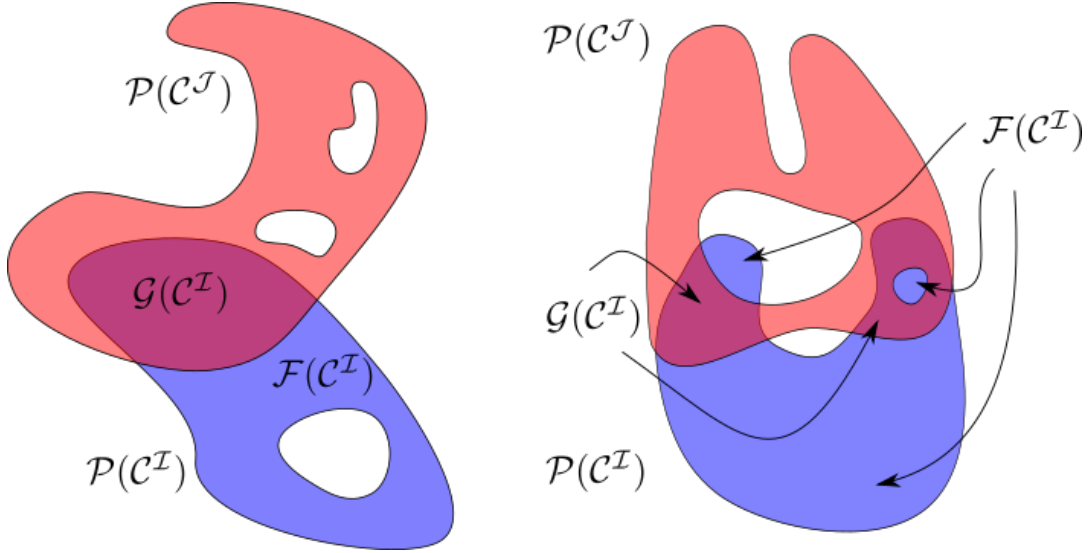


FIGURE 3.4: Possible pairs of consecutive cells in Π .

3.3.3 Harmonic-based Navigation Functions and Maps

In this section we provide a brief overview of the transformation and the adaptive vector field which were originally presented in [113] and are used in the design of the robot's linear velocity controllers.

Let \mathcal{F} be a given compact, multiply connected domain bounded by $N_o^{\mathcal{F}} + 1$ disjoint Jordan curves, namely $\partial\mathcal{F}_0, \partial\mathcal{F}_1, \dots, \partial\mathcal{F}_{N_o^{\mathcal{F}}}$, where $\partial\mathcal{F}_0$ is the enclosing outer boundary of \mathcal{F} . A harmonic map $T(p) = [u(p), v(p)]^T$ (i.e., $u(p)$ and $v(p)$ are harmonic function), is then constructed for \mathcal{F} , satisfying the following properties: a) maps the outer boundary $\partial\mathcal{F}_0$ of \mathcal{F} to the unit circle $\partial\mathcal{D}$, b) maps all inner boundaries $\partial\mathcal{F}_i$ to distinct points $q_i = [u_i, v_i]^T \in \text{int}(\partial\mathcal{D})$, and c) is a diffeomorphism for all $p \in \text{int}\mathcal{F}$. The computation of T requires a orientation-preserving weak homeomorphism $\bar{T} : \partial\mathcal{F}_0 \mapsto \partial\mathcal{D}$ which can be trivially constructed as needed. Then, the corresponding boundary value problem can be solved, e.g., by employing numerical techniques such as the Finite Element Method (FEM) or the Boundary Element Method (BEM), in order to obtain $u(p)$ and $v(p)$.

In this subsection, such transformations are utilized for mapping each connected set $\mathcal{F}_C(\mathcal{C}^{\mathcal{I}})$ of non-goal configurations corresponding to a cell $\mathcal{C}^{\mathcal{I}}$. Thus, depending on the type of the corresponding $\mathcal{G}_p(\mathcal{C}^{\mathcal{I}})$ the aforementioned methodology may need to be adapted accordingly. More specifically, if $\mathcal{G}_p(\mathcal{C}^{\mathcal{I}})$ is a contiguous subset of the outer boundary of $\mathcal{F}_C(\mathcal{C}^{\mathcal{I}})$, then \bar{T} must be chosen such that all $p \in \mathcal{G}_p(\mathcal{C}^{\mathcal{I}})$

get mapped to the same configuration q_d on the boundary. When $\mathcal{G}_p(\mathcal{C}^I)$ is equal to the entire outer boundary of $\mathcal{F}_C(\mathcal{C}^I)$, an additional diffeomorphism T^* mapping the unit circle to its center and vice-versa must be applied on the image of T . Thus, the composition $T' = T^* \circ T$ must be used instead of T for mapping $\mathcal{F}_C(\mathcal{C}^I)$ to \mathcal{D} . In this case, the goal configuration q_d of the transformed cell can be placed simply on the center of \mathcal{D} . A simple choice for T^* is as follows:

$$T^*(x) = \frac{x}{c + \|x\|} \circ \frac{x}{\|x\|^2} \circ \frac{x}{1 - \|x\|} \quad (3.6)$$

where c some positive constant. Finally, considering the case where $\mathcal{G}_p(\mathcal{C}^I)$ is equal to an inner boundary of $\mathcal{F}_C(\mathcal{C}^I)$, no further actions need to be made as the goal set will get automatically collapsed into a single point, onto which q_d will be placed.

Given now the corresponding transformation (T or T') mapping $\mathcal{F}_C(\mathcal{C}^I)$ to \mathcal{D} , we define a harmonic potential field ϕ by placing point harmonic sources upon the corresponding goal configuration q_d and the transformed inner obstacles $q_i = T(\partial\mathcal{F}_i)$, which is given by

$$\phi = k_d \cdot \ln\left(\frac{\|q - q_d\|}{2}\right) - \sum_{i=1}^{N_o^F} k_i \cdot \ln\left(\frac{\|q - q_i\|}{2}\right) \quad (3.7)$$

where $k_d > 0$ and $k_i \geq 0$ are adaptively varying parameters. We subsequently define a bounded reference potential field ψ based on ϕ , defined as

$$\psi = \frac{1 + \tanh(w \cdot \phi)}{2}, \quad (3.8)$$

whose gradient is given by

$$\nabla_q \psi = w \cdot \frac{1 - \tanh(w \cdot \phi)^2}{2} \cdot \nabla_q \phi, \quad (3.9)$$

where w is some positive scaling constant. We can now equip the robot with the following control law for its linear velocity:

$$u = f_p(p, k) \triangleq -\mathcal{K}_v \cdot s_\ell(q, k) \cdot J^{-1}(p) \cdot \nabla_q \psi(q, k) \quad (3.10)$$

where $q = T(p)$, \mathcal{K}_v is a positive constant, $J(p)$ denotes the jacobian matrix of T , ψ is an artificial potential field augmented with an adaptive control law $\dot{k} = f_k(p, k)$ for its parameters $k = [k_d, k_1, k_2, \dots, k_{\mathcal{N}_{\mathcal{F}_C(\mathcal{C}^I)}}]^T$, where $\mathcal{N}_{\mathcal{F}_C(\mathcal{C}^I)}$ is the number of holes in $\mathcal{F}_C(\mathcal{C}^I)$, and $s_\ell(p, k) \geq 0$ is a continuously differentiable function given by

$$s_\ell(q, k) = (1 - \|q\|) + \left(\max\left((\nabla_q \psi)^T \cdot q, 0\right)\right)^2. \quad (3.11)$$

Furthermore, the adaptive law designed for the parameters of ψ can be computed by

$$\begin{aligned} \dot{k}_d &= H_{\epsilon_1}(\lambda) \\ \dot{k}_i &= (\bar{k}_i - k_i) W_i c_i h_i - k_i (W_0 h_0 + H_{\epsilon_2}(s_\ell)), \forall i \in N_o^{\mathcal{F}} \end{aligned} \quad (3.12)$$

and its purpose is to guarantee invariance of \mathcal{F} without compromising the convergence properties of ψ . For a detailed definition of the adaptive laws \dot{k}_d and \dot{k}_i , as well as for the safety and stability proofs of f_p , refer to [chapter 2](#).

Finally, regarding the robot's orientation, we employ a simple proportional control law of the form

$$\omega = f_\theta(\theta) \triangleq -\mathcal{K}_\omega \cdot d_s(\theta, \theta_d) \quad (3.13)$$

where $d_s(\cdot, \cdot)$ is a continuously differentiable function that calculates the shortest signed distances between two given angles and θ_d is the mean value of the interval $\mathcal{G}_\theta(\mathcal{C}^I)$ (which is either singleton or a non-trivial connected subset of S^1).

3.4 Stability Analysis

In this section, we provide an analysis of the high-level planner, presented in [Subsection 3.3.1](#), w.r.t. completeness, as well as an analysis of stability properties of the overall proposed control scheme. Beginning with the former, we shall first state a fundamental property of the robot's configuration space.

Proposition 6. *The free configuration space $\mathcal{CS}_{\text{free}}$ of the robot \mathcal{R} is an open subset of \mathcal{CS} .*

Proof. For the proof, refer to the Appendix. □

Next, we need to take a closer look at the continuity of the approximation yielding functions \bar{f} and \underline{f} . Let $\text{dist}(p, \mathcal{C})$ be the shortest distance between a point p and the set \mathcal{C} , i.e.

$$\text{dist}(p, \mathcal{C}) = \min_{q \in \mathcal{C}} |p - q|.$$

Considering the area $\overline{\mathcal{R}}([\theta_l, \theta_u])$ swept by the robot rotating 2-manifold, it can be readily seen that the distance between any point $p \in \mathcal{R}(\theta)$ and $\overline{\mathcal{R}}([\theta_l, \theta_u])$, with $\theta_l \leq \theta \leq \theta_u$, is zero, since $\mathcal{R}(\theta) \subseteq \overline{\mathcal{R}}([\theta_l, \theta_u])$ by construction. In addition, we can show that $\overline{\mathcal{R}}([\theta_l, \theta_u])$ is continuous w.r.t. its arguments:

Proposition 7. *Given any $\theta_o \in S^1$ and any $\epsilon > 0$, there exists $\delta > 0$ such that $\text{dist}(q, \mathcal{R}(\theta_o)) < \epsilon$ for all $q \in \overline{\mathcal{R}}([\theta_o, \theta])$ and θ such that $|\theta_o - \theta| < \delta$.*

Proof. For the proof, refer to the Appendix. □

Contrary to $\overline{\mathcal{R}}([\theta_l, \theta_u])$, it can be readily seen that the distance between any point $p \in \underline{\mathcal{R}}([\theta_l, \theta_u])$ and $\mathcal{R}(\theta)$, with $\theta_l \leq \theta \leq \theta_u$, is zero, since $\underline{\mathcal{R}}([\theta_l, \theta_u]) \subseteq \mathcal{R}(\theta)$ by

construction. Continuity of $\underline{\mathcal{R}}([\theta_l, \theta_u])$ w.r.t. its arguments, for sufficiently small $|\theta_l - \theta_u|$, can also be shown as follows:

Proposition 8. *Given any $\theta_o \in \mathbb{S}^1$ and any $\varepsilon > 0$, there exists $\delta > 0$ such that $\text{dist}(p, \underline{\mathcal{R}}([\theta_o, \theta])) < \varepsilon$ (resp. $\text{dist}(p, \underline{\mathcal{R}}([\theta, \theta_o])) < \varepsilon$) for all $p \in \mathcal{R}(\theta_o)$ and θ such that $|\theta_o - \theta| < \delta$.*

Proof. For the proof, refer to the Appendix. □

The main results of this work follows.

Theorem 5. *Algorithm 2 is complete, i.e., if there exists a solution to the given navigation problem, the proposed planner is bound to return a sequence of under-approximation cells after a finite number of steps. Furthermore, if no solution exists, the planner is bound to conclude so in finite time.*

Proof. For the proof, refer to the Appendix. □

Theorem 6. *The proposed control scheme will drive the robot \mathcal{R} to the specified goal configuration z_{final} starting from almost any initial configuration z_{init} , assuming that z_{init} and z_{final} belong to the same connected component of $\mathcal{CS}_{\text{free}}$, while ensuring collision avoidance with the workspace's boundary.*

Proof. According to **Theorem 5**, we know that if the given problem is feasible, then our planner will return a sequence of adjacent under-approximation cells leading to z_{final} . The control law employed for the navigation within each cell is locally Lipschitz and possesses guaranteed almost-global convergence to the cell-specific goal configuration while ensuring invariance of the current cell, as shown in **Theorem 2**. Since after a transition from one cell to the next has been completed, the robot is by design guaranteed not to return to it, this concludes the proof. □

3.5 Simulation Results

In this section we present simulation results demonstrating the efficacy of the methodology presented in this chapter. More specifically, we consider a scenario involving a Y-shaped robotic platform which must navigate across a complex workspace to a desired configuration. The algorithm introduced in **Subsection 3.3.1** was used for obtaining a valid path of cells connecting the robot's initial and desired configurations. A total of 30 slices were used for sufficiently approximating the configuration space in order to complete this task, corresponding to angle intervals of lengths ranging from π to $\pi/32$ rad. The hybrid control law, presented in **Subsection 3.3.2**, was then employed for constructing corresponding control laws for the robot's linear and angular velocities, as described in **Subsection 3.3.3**. The motion profile executed by the robot can be seen in **Figure 3.5**, verifying the guaranteed convergence and safety

properties of our control scheme. Additionally, the evolution of the robot's configuration z over time as well as the time instances when a transition between two consecutive cells took place can be seen in [Figure 3.6](#). Finally, some typical examples of cells that were generated by the planner can be seen in [Figure 3.7](#).

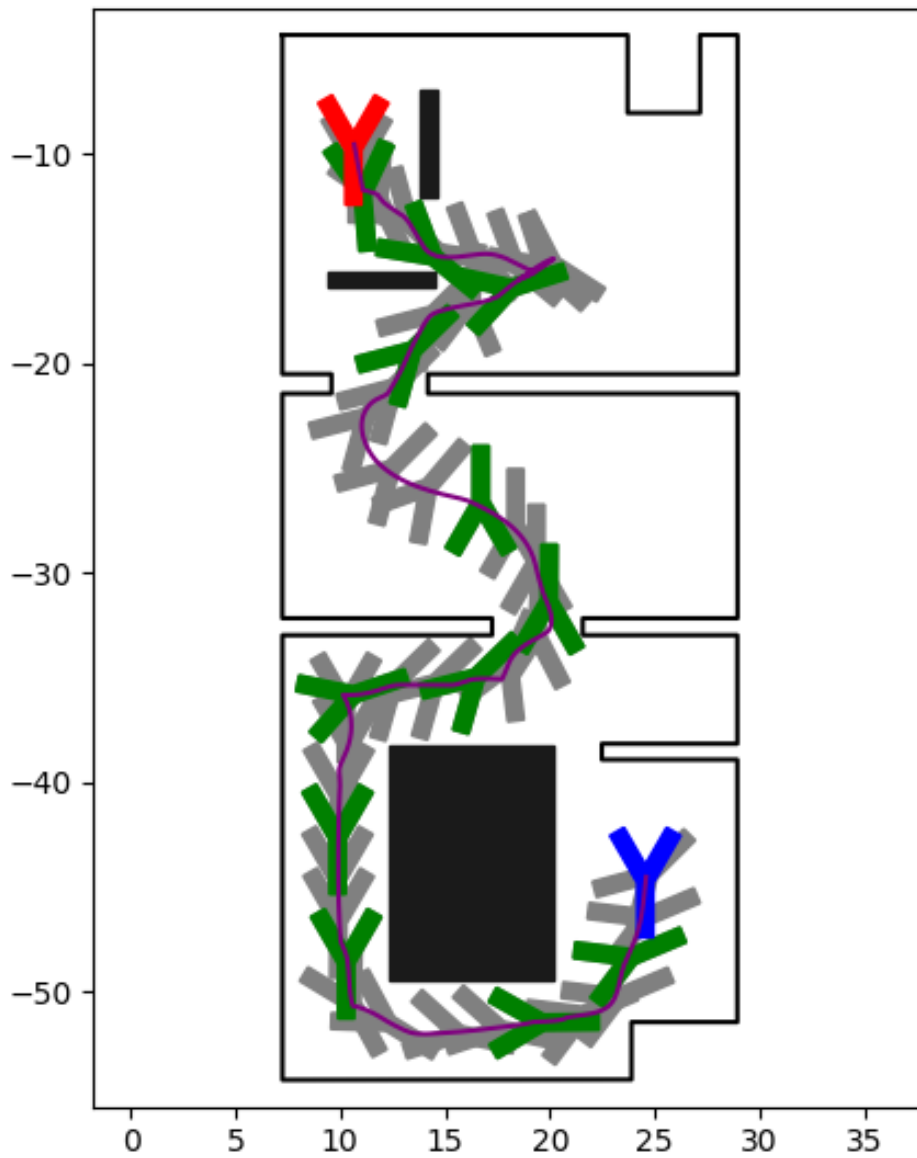


FIGURE 3.5: Motion profile executed by the robot. The purple line corresponds to the path traversed by the robot's reference point p . Blue and red poses correspond to robot's initial and desired configurations, respectively, whereas gray and green poses correspond to intermediate configurations captured at regular time intervals.

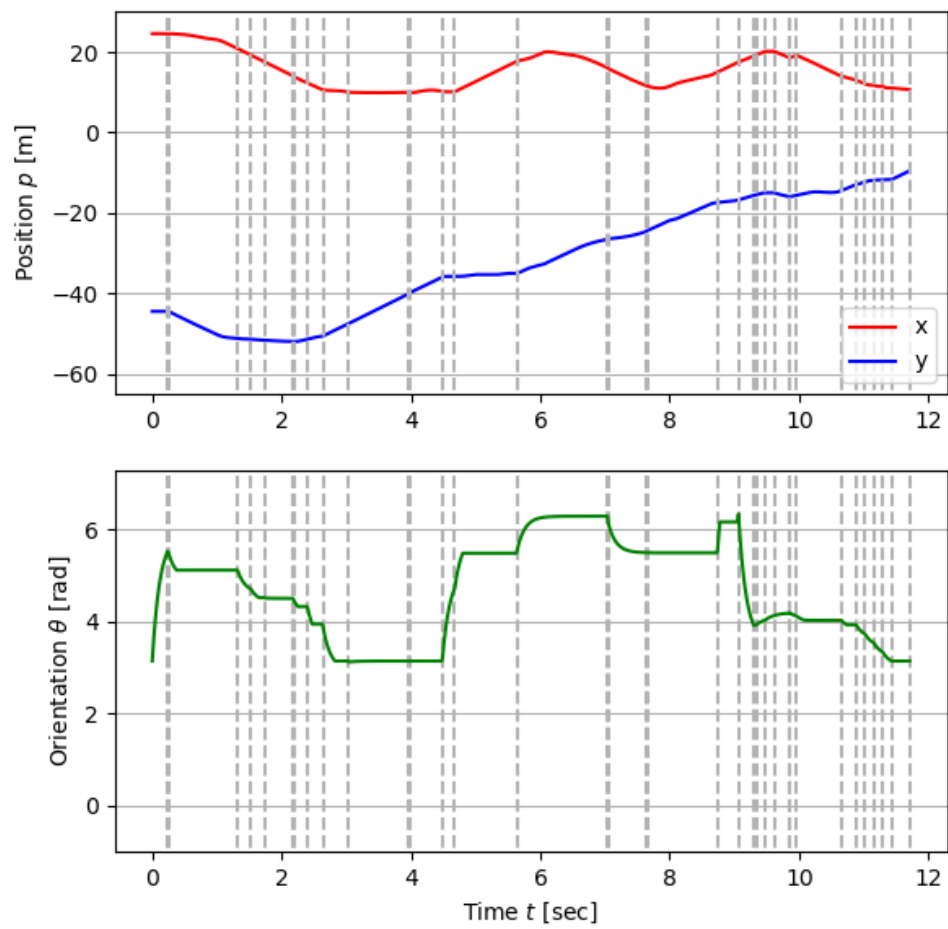


FIGURE 3.6: State of robot over time. Vertical gray lines indicate transitions of the hybrid controller.

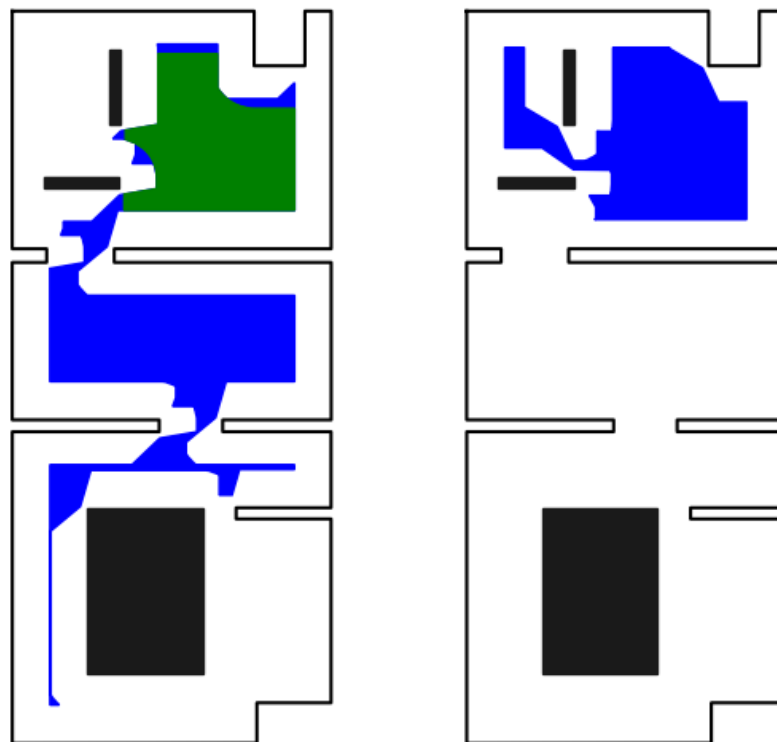


FIGURE 3.7: Examples of $\mathcal{F}(\mathcal{C}^I)$ (blue) and $\mathcal{G}(\mathcal{C}^I)$ (green) generated by the planner.

Chapter 4

Coordinating the Motion of Multiple Independent Disks

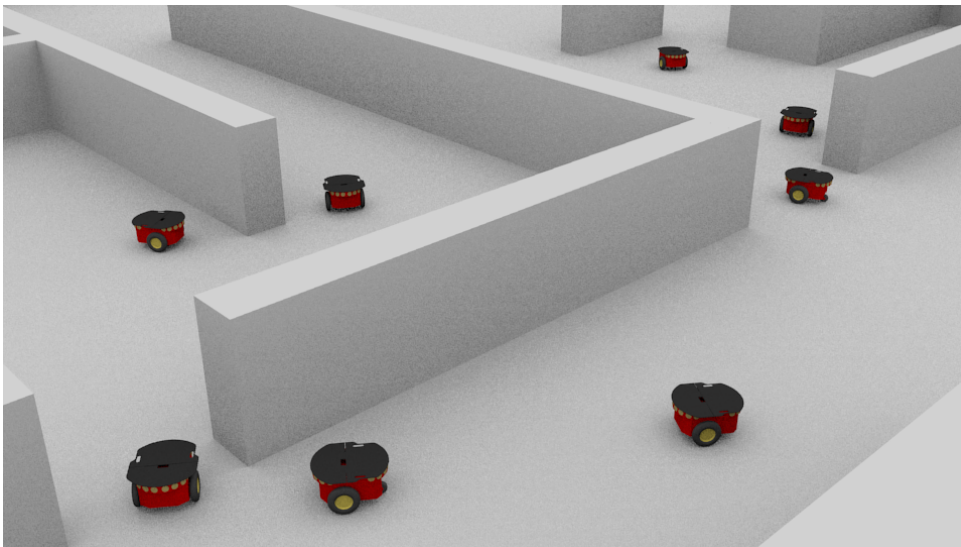


FIGURE 4.1: Multiple disk-shaped robots navigating within a complex, planar workspace.

4.1 Introduction

In this chapter, we present a hybrid control scheme to address the navigation problem for a team of disk-shaped robotic platforms operating within an obstacle cluttered, planar workspace. Given an initial and a desired configuration of the system, we devise a hierarchical cell decomposition methodology equipped with a suitable labeling mechanism which is able to determine which regions of the configuration space need to be further subdivided at each iteration, thus avoiding redundant cell expansions. Particularly, by computing appropriate over- and an under-approximations of each robot's footprint, our algorithm can determine which cells may contain feasible configurations of the system, while automatically discarding cells that are determined to contain none. Furthermore, given a solution to the given problem generated by the high-level planner, we employ harmonic transformations

and harmonic potential fields to accomplish safe transitions between adjacent cells, thus ensuring almost global convergence to the desired configuration. We remark that, since the distributed low-level control does not need explicit knowledge of the every robot's current position in order to compute control inputs, it requires very low communication resources as the robots only need to exchange information when they successfully complete a transition from one configuration cell to the next. Finally, we present simulation results that demonstrate the efficacy of the proposed control scheme.

The outline of this chapter is as follows. At the end of this section, we define some preliminary notions and notation used throughout this chapter. In [Section 4.2](#) we formulate the problem addressed in this work. In [Section 4.3](#) we elaborate on the proposed planner's design as well as the velocity control scheme employed for safely executing the computed plan. Finally, simulation results verifying the efficacy of the proposed control scheme are presented in [Section 4.4](#).

Notation: Throughout this chapter, we shall use $\mathfrak{I}_N \triangleq \{1, 2, \dots, N\}$ (resp. $\mathfrak{I}_N^* \triangleq \{0\} \cup \mathfrak{I}_N$) to denote the set consisting of all natural numbers up to N , starting from 1 (resp. 0). Additionally, given sets A and B , we use ∂A , $\text{int}(A)$, $\text{cl}(A)$ to denote the boundary, interior and closure of A , respectively, and $A \setminus B$ to denote the complement of B w.r.t. A .

4.2 Problem Formulation

We consider a team of $N_{\mathcal{R}}$ robots operating within a compact planar workspace $\mathcal{W} \subseteq \mathbb{R}^2$ occupied by a set of N_o disjoint, fixed inner obstacles $\mathcal{O}_i, i \in \mathfrak{I}_{N_o}$. We assume that each robot i has a disk-shaped body $\mathcal{R}_i \subset \mathbb{R}^2$ with radius $r_i > 0$. Let \mathcal{F}_w and $\mathcal{F}_i, i \in \mathfrak{I}_{N_{\mathcal{R}}}$ be the coordinate frames arbitrarily embedded in \mathcal{W} and $\mathcal{R}_i, i \in \mathfrak{I}_{N_{\mathcal{R}}}$, respectively. We shall refer to the origin of each $\mathcal{F}_i, i \in \mathfrak{I}_{N_{\mathcal{R}}}$ as the reference point of the corresponding robot. Moreover, without loss of generality, we assume that the reference point of each robot coincides with the center of its body. Let $p_i \triangleq [x_i, y_i]^T \in \mathbb{R}^2$ denote the relative position of i -th robot's reference point w.r.t. the workspace's coordinate frame \mathcal{F}_w , and let $\mathcal{R}_i(p)$ to denote its footprint, i.e., the space occupied by \mathcal{R}_i when placed at position p . Throughout this chapter, we shall use $\mathfrak{C} \subseteq \mathbb{R}^{2N_{\mathcal{R}}}$ to denote the robotic system's configuration space and $P \triangleq [p_1^T, p_2^T, \dots, p_{N_{\mathcal{R}}}^T]^T \in \mathfrak{C}$ to denote the stacked vector of robot positions. For the sake of brevity, we shall also use $P[i]$ to denote the i -th component of P , i.e., $P[i] = p_i$. Let \mathcal{W}^o denote the complement of \mathcal{W} , i.e., $\mathcal{W}^o \triangleq \mathbb{R}^2 \setminus \mathcal{W}$. We also define a configuration P as feasible iff the following conditions hold:

$$\begin{aligned} \mathcal{R}_i(P[i]) \cap \mathcal{R}_j(P[j]) &= \emptyset, \quad \forall i \neq j \in \mathfrak{I}_{N_{\mathcal{R}}} \\ \mathcal{R}_i(P[i]) \cap \mathcal{W}^o &= \emptyset, \quad \forall i \in \mathfrak{I}_{N_{\mathcal{R}}} \end{aligned} \tag{4.1}$$

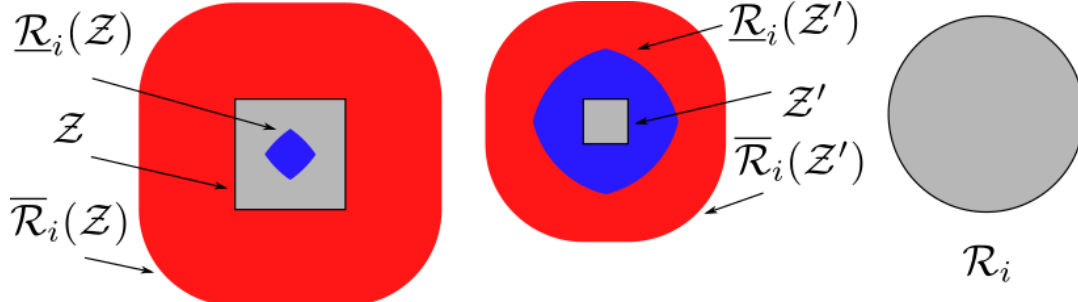


FIGURE 4.2: Over-approximation $\mathcal{R}_i(\mathcal{Z})$ (resp. $\mathcal{R}_i(\mathcal{Z}')$) and under-approximation $\underline{\mathcal{R}}_i(\mathcal{Z})$ (resp. $\underline{\mathcal{R}}_i(\mathcal{Z}')$) of the footprint of robot i when swept over $\mathcal{Z} \subset \mathbb{R}^2$ (resp. \mathcal{Z}).

and we shall use $\mathcal{C}^f \subset \mathcal{C}$ to denote the set of all feasible configurations of the robotic system, whereas its complement $\mathcal{C}^o \triangleq \mathcal{C} \setminus \mathcal{C}^f$ corresponds to the set all infeasible configurations. Furthermore, we assume that the motion of each robot i obeys the single-integrator kinematic model:

$$\dot{p}_i = u_i, \quad i \in \mathcal{I}_{N_{\mathcal{R}}} \quad (4.2)$$

where u_i denotes the control input.

Let P_{init} and P_{des} be two given feasible configurations of the multi-robot system. Our goal is to design a control scheme that drives any robot i , initialized at $p_{\text{init},i} = P_{\text{init}}[i]$, to the specified desired position $p_{\text{des},i} = P_{\text{des}}[i]$, while avoiding inter-robot and robot-workspace collisions, i.e., $P(t) \in \mathcal{C}^f$ for all $t \geq 0$.

4.3 Control Design

To address the aforementioned problem, first, we employ a hierarchical cell decomposition scheme for partitioning the configuration space of the multi-robot system \mathcal{C} into cells, as described in [Subsection 4.3.1](#). Then, we design a high-level planner, in [Subsection 4.3.2](#), which recursively expands the aforementioned structure until a sequence of adjacent cells connecting P_{init} and P_{des} is found. Finally, the low-level control scheme employed to ensure safe transition between cells until the goal configuration is reached, is presented in [Subsection 4.3.3](#).

4.3.1 Configuration Space Decomposition

In this subsection, we present the hierarchical cell decomposition scheme that will be employed in our approach. We begin with disregarding inter-robot collisions and considering the configuration space of each individual robot. Particularly, the configuration space of robot i , denoted herein by $\mathcal{A}_i(\mathcal{W})$, corresponds to the largest subset of \mathcal{W} where the reference point of robot i can be placed such that $\mathcal{R}_i(p_i) \cap \mathcal{W}^o = \emptyset$, for all $p_i \in \mathcal{A}_i(\mathcal{W})$. Also, given a subset \mathcal{Z} of \mathcal{W} , we shall use $\mathcal{A}_i(\mathcal{Z})$ to

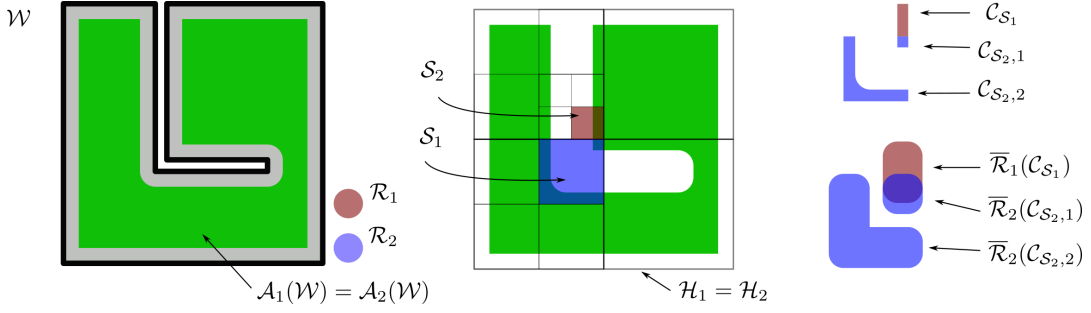


FIGURE 4.3: Example of a hierarchical configuration space decomposition for a system of two identical robots. The green area corresponds to the configuration space $\mathcal{A}_i(\mathcal{W})$ of each robot. For the sake of simplicity, we assume $\mathcal{H}_1 = \mathcal{H}_2$. The workspace slice \mathcal{S}_1 corresponding to robot 1 consists of a single simple cell ($\mathcal{C}_{\mathcal{S}_1}$) whereas slice \mathcal{S}_2 corresponding to robot 2 consists of two cells ($\mathcal{C}_{\mathcal{S}_2,1}, \mathcal{C}_{\mathcal{S}_2,2}$). The compound cell ($\mathcal{C}_{\mathcal{S}_1}, \mathcal{C}_{\mathcal{S}_2,1}$) is labeled as mixed because \mathcal{R}_1 and \mathcal{R}_2 may intersect when $p_1 \in \mathcal{C}_{\mathcal{S}_1}$ and $p_2 \in \mathcal{C}_{\mathcal{S}_2,1}$ since $\overline{\mathcal{R}_1}(\mathcal{C}_{\mathcal{S}_1}) \cap \overline{\mathcal{R}_2}(\mathcal{C}_{\mathcal{S}_2,1}) \neq \emptyset$, whereas ($\mathcal{C}_{\mathcal{S}_1}, \mathcal{C}_{\mathcal{S}_2,2}$) is marked as admissible.

denote the set of feasible positions of robot i which belong to \mathcal{Z} , i.e.:

$$\mathcal{A}_i(\mathcal{Z}) \triangleq \{ p \mid p \in \mathcal{Z} \text{ and } \mathcal{R}_i(p) \cap \mathcal{W}^o = \emptyset \}, \forall i \in \mathcal{I}_{N_R}. \quad (4.3)$$

In addition to $\mathcal{A}_i(\cdot)$, which corresponds to the set of feasible positions of robot i , we also consider two estimations of the area that is potentially occupied by \mathcal{R}_i when p_i is restricted in a subset \mathcal{Z} of \mathcal{W} . Particularly, given a robot $i \in \mathcal{I}_{N_R}$ and a set $\mathcal{Z} \subseteq \mathcal{A}_i(\mathcal{W})$, let $\overline{\mathcal{R}_i}(\mathcal{Z})$ and $\underline{\mathcal{R}_i}(\mathcal{Z})$ be an over-approximation and under-approximation, respectively, of the footprint of \mathcal{R}_i when robot i is swept over \mathcal{Z} such that:

$$\begin{aligned} \overline{\mathcal{R}_i}(\mathcal{Z}) &\supseteq \bigcup_{p \in \mathcal{Z}} \mathcal{R}_i(p) \\ \underline{\mathcal{R}_i}(\mathcal{Z}) &\subseteq \bigcap_{p \in \mathcal{Z}} \mathcal{R}_i(p) \end{aligned} \quad (4.4)$$

and

$$\begin{aligned} \overline{\mathcal{R}_i}(\mathcal{Z}) &\subseteq \overline{\mathcal{R}_i}(\mathcal{Z}'), \quad \forall \mathcal{Z} \subseteq \mathcal{Z}' \\ \underline{\mathcal{R}_i}(\mathcal{Z}) &\supseteq \underline{\mathcal{R}_i}(\mathcal{Z}'), \quad \forall \mathcal{Z} \subseteq \mathcal{Z}'. \end{aligned} \quad (4.5)$$

An example of such approximations can be seen in [Figure 4.2](#).¹

We now consider a set $\mathcal{S} \subseteq \mathbb{R}^2$ that has the form $[x_1, x_2] \times [y_1, y_2]$. We shall refer to such a set as a simple slice of \mathbb{R}^2 . Given a simple slice \mathcal{S} and a robot $i \in \mathcal{I}_{N_R}$, we will use $\mathcal{W}_S^i \triangleq \mathcal{A}_i(\mathcal{W}) \cap \mathcal{S}$ to denote the set of feasible positions of robot i (neglecting inter-robot collisions) that are contained in \mathcal{S} . A set $\mathcal{S} = \{ \mathcal{S}_i \mid i \in \mathcal{I}_{N_S} \}$ of N_S simple

¹For disk shaped robots, a valid over-approximation $\overline{\mathcal{R}_i}(\mathcal{Z})$ can be computed by offsetting \mathcal{Z} by r_i , whereas $\underline{\mathcal{R}_i}(\mathcal{Z})$ can be calculated by $\bigcap_{p \in \partial \mathcal{Z}} \mathcal{R}_i(p)$.

slices shall be called a cover of \mathcal{W} iff

$$\mathcal{W} = \bigcup_{j \in \mathcal{I}_{N_S}} \mathcal{S}_j \cap \mathcal{W}. \quad (4.6)$$

We note that a cover \mathfrak{S} partitions $\mathcal{A}_i(\mathcal{W})$ into a set of regions \mathcal{W}_S^i , $\mathcal{S} \in \mathfrak{S}$ each of which consists of zero or more individually connected but pairwise disjoint subsets $\mathcal{C}_{S,j}^i$, $j \in \mathcal{I}_{N_S(\mathcal{W}_S^i)}$, that shall be referred to as workspace (or simple) cells. A cover $\widehat{\mathfrak{S}} = \{(i, \mathcal{S}_i) \mid i \in \mathcal{I}_{N_R}\}$ of the configuration space \mathfrak{C} is respectively defined by assigning a cover to each robot. Accordingly, a configuration space (or compound) slice $\widehat{\mathcal{S}}$ is defined as $\widehat{\mathcal{S}} = \{(i, \mathcal{S}_i) \mid i \in \mathcal{I}_{N_R}\}$, where \mathcal{S}_i , $i \in \mathcal{I}_{N_R}$ is a set of simple slices. Likewise, a configuration space cover $\widehat{\mathfrak{S}} = \{(i, \mathcal{S}_i) \mid i \in \mathcal{I}_{N_R}\}$ induces a partitioning of \mathfrak{C} into regions $\widehat{\mathcal{W}}_{\widehat{\mathcal{S}}} \triangleq \mathcal{W}_{S_1}^1 \times \mathcal{W}_{S_2}^2 \times \dots \times \mathcal{W}_{S_{N_R}}^{N_R}$, where $\widehat{\mathcal{S}} = \{(i, \mathcal{S}_i) \mid i \in \mathcal{I}_{N_R}\}$ is an element of $\widehat{\mathfrak{S}}$. We note that each of these regions may consists of zero or more individually connected but pairwise disjoint subsets $\widehat{\mathcal{C}}_{\widehat{\mathcal{S}},i}$, $i \in \mathcal{I}_{N_S(\widehat{\mathcal{C}}_{\widehat{\mathcal{S}},i})}$, which shall be referred to herein as configuration space (or compound) cells. Given the compound cell $\widehat{\mathcal{C}}_{\widehat{\mathcal{S}},i}$, we will use $\widehat{\mathcal{C}}_{\widehat{\mathcal{S}},i}^{[j]}$ to denote its j -th component, i.e., $\widehat{\mathcal{C}}_{\widehat{\mathcal{S}},i}^{[j]} \triangleq \mathcal{C}_{S_j,i}^j$, for all $j \in \mathcal{I}_{N_R}$. We remark that, unlike hierarchical decomposition schemes commonly encountered in the literature, which use cells of simple geometries (e.g., hypercubes or hyperrectangles), the configuration space cells considered in this chapter have, in general, arbitrary geometries since their components do not possess a pre-specified shape (see [Figure 4.3](#)). Although this choice renders navigation within a cell $\widehat{\mathcal{C}}$ more complicated, it generally results in coarser partitions since each component $\widehat{\mathcal{C}}^{[i]}$ of $\widehat{\mathcal{C}}$ belongs in $\mathcal{A}_i(\mathcal{W})$ by construction, thus the subdivision scheme has to accommodate only for potential inter-robot collisions.

Regarding now the transition between configuration space cells, we introduce some required notions of connectedness. We begin with considering two distinct simple slices \mathcal{S}_i and \mathcal{S}_j which shall be called adjacent iff their intersection $\mathcal{S}_i \cap \mathcal{S}_j$ is not empty. Moreover, let $\mathcal{C}_{S_m,i}$ and $\mathcal{C}_{S_n,j}$ be two distinct workspace cells. We define these simple cells as adjacent iff $\mathcal{C}_{S_m,i} \cap \mathcal{C}_{S_n,j} \neq \emptyset$. Apparently, $\mathcal{C}_{S_m,i}$ and $\mathcal{C}_{S_n,j}$ being adjacent implies that \mathcal{S}_m and \mathcal{S}_n are also adjacent. The aforementioned definitions can be naturally extended to compound slices and cells, as well. Particularly, two compound slices $\widehat{\mathcal{S}}_m = \{(i, \mathcal{S}_{m,i}) \mid i \in \mathcal{I}_{N_R}\}$ and $\widehat{\mathcal{S}}_n = \{(i, \mathcal{S}_{n,i}) \mid i \in \mathcal{I}_{N_R}\}$ are adjacent iff $\mathcal{S}_{m,i}$, $\mathcal{S}_{n,i}$ are adjacent, for all $i \in \mathcal{I}_{N_R}$, whereas, two compound cells $\widehat{\mathcal{C}}_i$ and $\widehat{\mathcal{C}}_j$ are adjacent iff $\widehat{\mathcal{C}}_i^{[k]}$ and $\widehat{\mathcal{C}}_j^{[k]}$ are adjacent, for all $k \in \mathcal{I}_{N_R}$. A path Π of configuration space cells is defined as any finite string of sequentially adjacent compound cells. Obviously, a path Π consisting of cells that lie entirely in \mathfrak{C}^f and contain both P_{init} and P_{des} is a valid solution to our path finding sub-problem. In order to discover such a path, we build a hierarchical decomposition $\mathfrak{H} = \{(i, \mathcal{H}) \mid i \in \mathcal{I}_{N_R}\}$ of the configuration space \mathfrak{C} by assigning to each robot i a hierarchical partitioning of the workspace \mathcal{W} , represented as a connected, directed tree $\mathcal{H} \triangleq (\mathcal{N}_{\mathcal{H}}, \mathcal{E}_{\mathcal{H}})$ such that:

- Each node $\mathcal{S} \in \mathcal{N}_{\mathcal{H}}$ is a simple slice.
- Every child \mathcal{S}_j of a given node \mathcal{S}_i (i.e., $(\mathcal{S}_i, \mathcal{S}_j) \in \mathcal{E}_{\mathcal{H}}$) is a strict subset of \mathcal{S}_i .
- The set of leaf nodes must form a cover of \mathcal{W} .

Finally, an algorithm was devised for appropriately expanding \mathfrak{H} until a solution is found, as described in the following subsection.

4.3.2 High-Level Planner

In this subsection, we present a high-level planner for finding a sequence Π of adjacent cells in \mathcal{C}^f connecting the initial P_{init} and goal P_{des} configurations. One of the main advantages of the proposed algorithm is the use of a suitable labeling scheme, which allows it to recursively subdivide, at each iteration, configuration space cells that lie on the boundary between \mathcal{C}^f and \mathcal{C}^o , while ignoring cells that lie completely inside \mathcal{C}^f or \mathcal{C}^o . To do so, this labeling scheme exploits the over- and under-approximations $\overline{\mathcal{R}}_i$ and $\underline{\mathcal{R}}_i$ of each robot's footprint, defined in [Subsection 4.3.1](#), to determine whether a robot may collide with another one while each robot navigates independently within its respective workspace cell. More specifically, given a compound cell $\hat{\mathcal{C}}$, the employed cell labeling scheme works as follows:

- If the intersection of all $\overline{\mathcal{R}}_i(\hat{\mathcal{C}}^{[i]})$, $i \in \mathfrak{I}_{N_{\mathcal{R}}}$ is empty, then, by virtue of (4.4), no robot may come across another while $P \in \hat{\mathcal{C}}$, thus $\hat{\mathcal{C}}$ is entirely contained in \mathcal{C}^f . Such a compound cell is marked as *admissible*.
- If the intersection of all $\underline{\mathcal{R}}_i(\hat{\mathcal{C}}^{[i]})$, $i \in \mathfrak{I}_{N_{\mathcal{R}}}$ is non-empty, then, by virtue of (4.4), for every $P \in \hat{\mathcal{C}}$ there exists at least one pair of intersecting robots, thus $\hat{\mathcal{C}}$ is entirely contained in \mathcal{C}^o . Such a compound cell is marked as *inadmissible*.
- If $\hat{\mathcal{C}}$ is neither admissible nor inadmissible, it is marked as *mixed*.

In general, mixed cells encapsulate both feasible and infeasible configurations and expanding them (recursively) should yield admissible and inadmissible sub-cells. On the other hand, by virtue of (60), subdivision of admissible (resp. inadmissible) cells yields only admissible (resp. inadmissible) cells, without contributing any further in the configuration space's exploration.

The planner's main search algorithm is described in [Algorithm 4](#), which initially constructs a coarse compound slice hierarchical partitioning made of each robot's feasible set $\mathcal{A}_i(\mathcal{W})$, thus enclosing all \mathcal{C}^f (functions INITIALIZEHIERARCHY and INITIALIZECELLS). Then, the initially computed compound cells get expanded until admissible ones, containing P_{init} and P_{des} , are found (function FINDENCLOSINGACCELL), whereas inability to find such compound cells indicates infeasibility of the given problem and the algorithm terminates. Next, an initial path Π connecting $\hat{\mathcal{C}}_{\text{init}}$ or $\hat{\mathcal{C}}_{\text{goal}}$ made of compound cells belonging to the exploration's frontier set $S_{\hat{\mathcal{C}},F}$ (i.e.,

the set of unexpanded admissible and mixed cells), is built (function CONNECTSTRINGS). At each iteration, the first mixed compound cell of Π (function GETFIRSTMIXEDCCELL) is removed from the frontier and is expanded (function EXPANDCCELL) by subdividing the widest simple cell \mathcal{C} whose over-approximation $\overline{\mathcal{R}_i}(\mathcal{C})$ intersects with another (functions GETCONFLICTINGSCELLS and SELECTSCELLWITHWIDESTSSSLICE) into smaller ones, as seen in [Algorithm 5](#). Finally, a new path is constructed using standard back-tracking techniques, until either Π consists only of admissible cells or no new path of mixed and admissible cells leading to P_{des} can be found.

Algorithm 4 Planner's Main Algorithm

```

function FINDAPATH(  $P_{\text{init}}, P_{\text{des}}$  )
   $\mathfrak{H} \leftarrow \text{INITIALIZEHIERARCHY}$ 
   $S_{\hat{\mathcal{C}}} \leftarrow \text{INITIALIZECCELLS}(\mathfrak{H})$ 
   $S_{\hat{\mathcal{C}},F} \leftarrow S_{\hat{\mathcal{C}}}$ 
   $\hat{\mathcal{C}}_{\text{init}}, \mathfrak{H}, S_{\hat{\mathcal{C}}}, S_{\hat{\mathcal{C}},F} \leftarrow$ 
    FINDENCLOSINGACCELL(  $P_{\text{init}}, \mathfrak{H}, S_{\hat{\mathcal{C}}}, S_{\hat{\mathcal{C}},F}$  )
   $\hat{\mathcal{C}}_{\text{goal}}, \mathfrak{H}, S_{\hat{\mathcal{C}}}, S_{\hat{\mathcal{C}},F} \leftarrow$ 
    FINDENCLOSINGACCELL(  $P_{\text{des}}, \mathfrak{H}, S_{\hat{\mathcal{C}}}, S_{\hat{\mathcal{C}},F}$  )
  if  $\hat{\mathcal{C}}_{\text{init}}$  is null or  $\hat{\mathcal{C}}_{\text{goal}}$  is null then
    return null
  end if
   $\Pi \leftarrow \text{CONNECTSTRINGS}([\hat{\mathcal{C}}_{\text{init}}], [\hat{\mathcal{C}}_{\text{goal}}], S_{\hat{\mathcal{C}},F})$ 
  while not (( $\Pi$  is null) or ISADMISSIBLEP( $\Pi$ )) do
     $\Pi, \mathfrak{H}, S_{\hat{\mathcal{C}}}, S_{\hat{\mathcal{C}},F} \leftarrow$ 
      EXPANDPATH(  $\Pi, \mathfrak{H}, S_{\hat{\mathcal{C}}}, S_{\hat{\mathcal{C}},F}$  )
  end while
  return  $\Pi$ 
end function

```

Algorithm 5 Path Expansion at Mixed Compound Cell

```

function EXPANDPATH(  $\Pi, \mathfrak{H}, S_{\hat{\mathcal{C}}}, S_{\hat{\mathcal{C}},F}$  )
   $\hat{\mathcal{C}} \leftarrow \text{GETFIRSTMIXEDCCELL}(\Pi)$ 
   $\mathcal{L}_{\text{pre}}, \mathcal{L}_{\text{suf}} \leftarrow \text{SPLITSTRING}(\Pi, \hat{\mathcal{C}})$ 
   $S_{\mathcal{C}} \leftarrow \text{GETCONFLICTINGSCELLS}(\hat{\mathcal{C}})$ 
   $i, \mathcal{S}, \mathcal{C} \leftarrow \text{SELECTSCELLWITHWIDESTSSSLICE}(S_{\mathcal{C}})$ 
   $S_{\hat{\mathcal{C}}_R}, \mathfrak{H}, S_{\hat{\mathcal{C}}}, S_{\hat{\mathcal{C}},F} \leftarrow$ 
    EXPANDCCELL(  $\hat{\mathcal{C}}, i, \mathcal{S}, \mathcal{C}, \mathfrak{H}, S_{\hat{\mathcal{C}}}, S_{\hat{\mathcal{C}},F}$  )
   $\Pi \leftarrow \text{CONNECTSTRINGS}(\mathcal{L}_{\text{pref}}, \mathcal{L}_{\text{suf}}, S_{\hat{\mathcal{C}},F})$ 
  return  $\Pi, \mathfrak{H}, S_{\hat{\mathcal{C}}}, S_{\hat{\mathcal{C}},F}$ 
end function

```

4.3.3 Velocity Control Law

Given now a path Π consisting of N_{Π} admissible configuration space cells, we present a distributed control law for safely navigating from one cell to the next until the goal

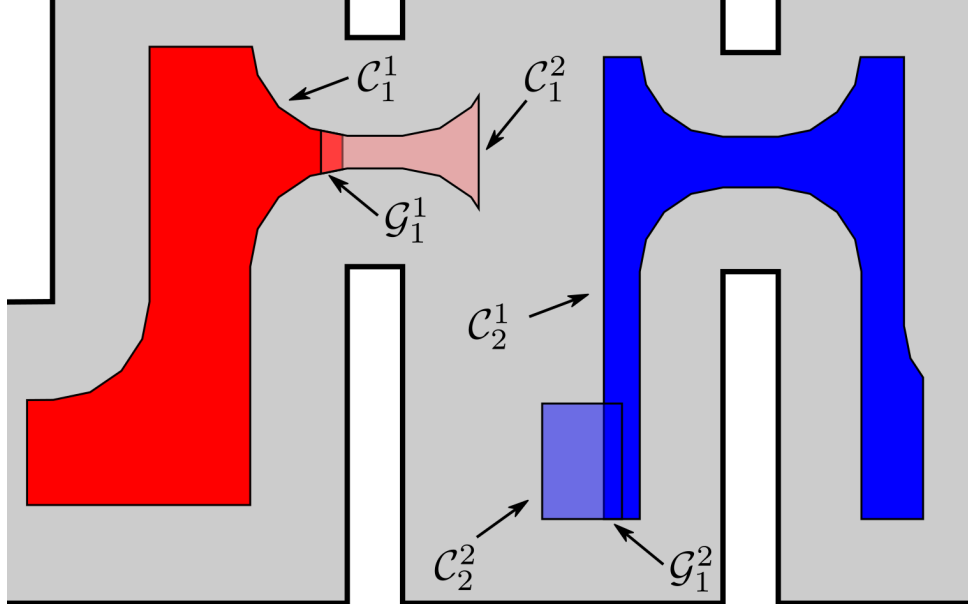


FIGURE 4.4: Two adjacent compound cells $\widehat{C}_1 = (C_1^1, C_2^1)$ and $\widehat{C}_2 = (C_2^1, C_2^2)$. In order robot 1 (resp. robot 2) to successfully move from C_1^1 to C_2^1 (resp. from C_2^1 to C_2^2) it has to reach any point of $\mathcal{G}_{1,1}$ (resp. $\mathcal{G}_{2,1}$).

configuration P_{des} is reached. First, we consider two consecutive compound cells \widehat{C}_ℓ and $\widehat{C}_{\ell+1}$ in Π , for which we compute the goal set $\mathcal{G}_{i,\ell} \triangleq \widehat{C}_\ell^{[i]} \cap \widehat{C}_{\ell+1}^{[i]}$ of each robot i , that contains feasible configurations in both $\widehat{C}_\ell^{[i]}$ and $\widehat{C}_{\ell+1}^{[i]}$ and is non-empty by construction. Respectively, the goal set corresponding to the last cell of Π consists of just the desired configuration P_{des} , i.e., $\widehat{C}_{N_\Pi} = \{P_{\text{des}}\}$ (see Figure 4.4). Furthermore, let $\mathcal{F}_{\ell,i} \triangleq \widehat{C}_\ell^{[i]} \setminus \text{int}(\mathcal{G}_{\ell,i})$ and $\mathcal{G}'_{\ell,i} \triangleq \mathcal{G}_{\ell,i} \cap \mathcal{F}_{\ell,i}$, for all $k \in \mathfrak{I}_{N_\Pi}$. Notice that $\mathcal{G}'_{\ell,i}$ is generally made of one or more pairwise disjoint subsets of arbitrary connectedness as well as that robot i should navigate to any of these regions² without escaping $\widehat{C}_\ell^{[i]}$ in order to successfully traverse to the next specified workspace cell. Respectively, the transition from \widehat{C}_ℓ to $\widehat{C}_{\ell+1}$ is considered complete after every robot i reaches $\widehat{C}_{\ell+1}^{[i]}$. We also remark that when $\widehat{C}_\ell^{[i]} \subseteq \widehat{C}_{\ell+1}^{[i]}$, robot i simply needs to retain its current position during step ℓ .

In order to fulfill the aforementioned specifications, we equip each robot i with a controller u_i based on suitable workspace transformations and adaptive artificial potential fields, which were originally presented in [113] and possess guaranteed domain invariance and almost global convergence properties. More specifically, we build a diffeomorphic transformation $q_i^\ell = T_i^\ell(p_i)$ that maps $\mathcal{F}_{\ell,i}$ to the unit disk \mathcal{D} , the outer boundary of $\mathcal{F}_{\ell,i}$ to the unit circle $\partial\mathcal{D}$ and collapses all inner boundaries to distinct points $q_{i,j}^\ell$, $j \in \mathfrak{I}_{N_i^\ell}$, where N_i^ℓ is the genus of $\mathcal{F}_{\ell,i}$. We now distinguish the following two cases of possible goal sets: a) $\mathcal{G}'_{\ell,i}$ being an inner boundary of $\mathcal{F}_{\ell,i}$, and

²When more than one of such disjoint goal subsets are reachable from the connected component of $\mathcal{F}_{\ell,i}$ that contains p_i , one of them is arbitrarily (though, deterministically) selected and assigned as the goal set. A more sophisticated approach for choosing goal regions could be employed but it exceeds the scope of the current work.

b) $\mathcal{G}'_{\ell,i}$ being part of the outer boundary of $\mathcal{F}_{\ell,i}$. Depending on the case, T_i^ℓ must be appropriately adapted to simplify the subsequent potential field's design. Particularly, case (a) can be accommodated by modifying T_i^ℓ such that $\mathcal{G}_{\ell,i}$ collapses to an inner point $q_{\text{des},i}^\ell$ of \mathcal{D} , whereas case (b) is addressed by designing T_i^ℓ such that $\mathcal{G}'_{\ell,i}$ collapses to a single point $q_{\text{des},i}^\ell$ on $\partial\mathcal{D}$. Next, we define the harmonic potential field ϕ_i^ℓ used by robot i during step ℓ by placing point harmonic sources upon the corresponding goal configuration $q_{\text{des},i}^\ell$ and the transformed inner obstacles $q_{i,j}^\ell$, given by:

$$\phi_i^\ell = k_{i,d}^\ell \ln \left(\frac{\|q_i^\ell - q_{\text{des},i}^\ell\|}{2} \right) - \sum_{j \in \mathcal{J}_{N_i}^\ell} k_{i,j}^\ell \ln \left(\frac{\|q_i^\ell - q_{i,j}^\ell\|}{2} \right) \quad (4.7)$$

where $k_{i,d}^\ell > 0$ and $k_{i,j}^\ell \geq 0$ are adaptively varying parameters. Finally, the control law u_i^ℓ of robot i during step ℓ is given by

$$u_i^\ell = -Ks(q_i^\ell, k_{v,i}^\ell) \left(J_i^\ell(q_i^\ell) \right)^{-1} \nabla_{q_i^\ell} \psi_i^\ell(q_i^\ell, k_{v,i}^\ell) \quad (4.8)$$

where K is a positive control gain, J_i^ℓ is the Jacobian matrix of T_i^ℓ , s is a factor ensuring collision avoidance with the outer boundary and $\psi_i^\ell = 1 + \tanh(w\phi_i^\ell)/2$, with w a positive constant.

4.4 Simulation Results

In this section, we present simulation results demonstrating the efficacy of the proposed methodology. Particularly, we consider 5 scenarios where a system consisting of 2, 4, 6, 8 and 10 robots, respectively, initialized within the workspace depicted in [Figure 4.5](#), is required to reach a specified final configuration. The time required by the proposed planner, as well as the total amount of compound cells generated during the solution of each case is shown in [Table 4.1](#). We remark that the planner expanded mixed compound cells by subdividing the corresponding conflicting simple slice into four identical overlapping sub-slices. The motion profiles executed by the robots in each corresponding case can be seen in [Figure 4.5](#), [Figure 4.6](#), [Figure 4.7](#), [Figure 4.8](#), [Figure 4.9](#). Additionally, [Figure 4.10](#) and [Figure 4.11](#) depict the initial and goal configurations as well as the computed enclosing cells, respectively, for the 8-robot scenario. As one can verify from the figures that the robots can successfully navigate to their individual goals.

Amount of Robots	2	4	6	8	10
Time (sec)	0.088	0.240	0.845	1.36	31.1
Compound Cells	51	348	823	1014	3363

TABLE 4.1: Execution time and amount of generated compound cells required by the high-level planner for solving each scenario.

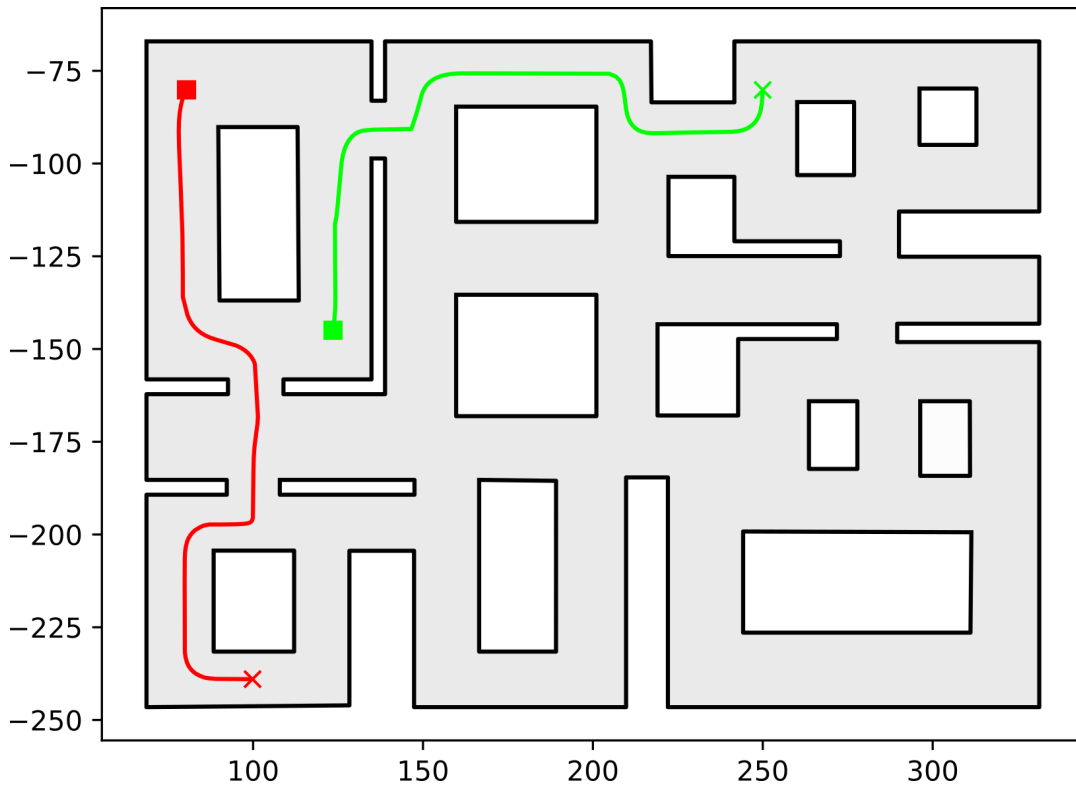


FIGURE 4.5: Executed trajectories of the 2 robot case. Squares indicate initial positions whereas the corresponding goal positions are depicted using crosses.

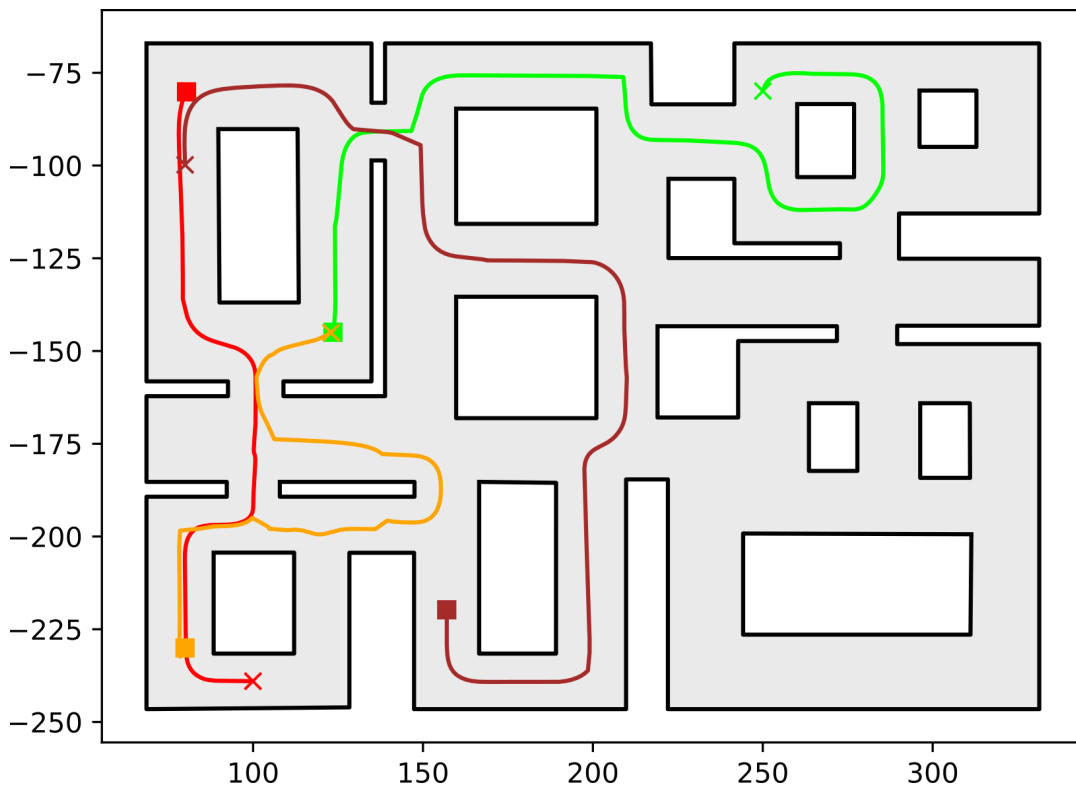


FIGURE 4.6: Executed trajectories of the 4 robot case. Squares indicate initial positions whereas the corresponding goal positions are depicted using crosses.

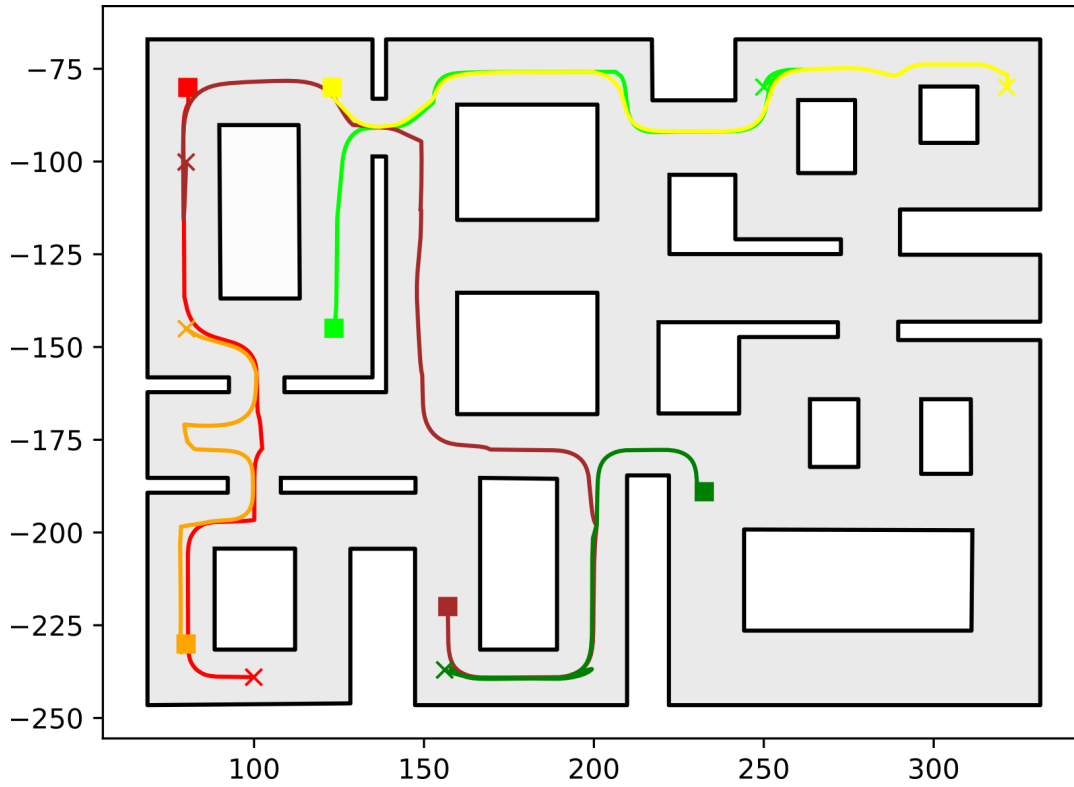


FIGURE 4.7: Executed trajectories of the 6 robot case. Squares indicate initial positions whereas the corresponding goal positions are depicted using crosses.

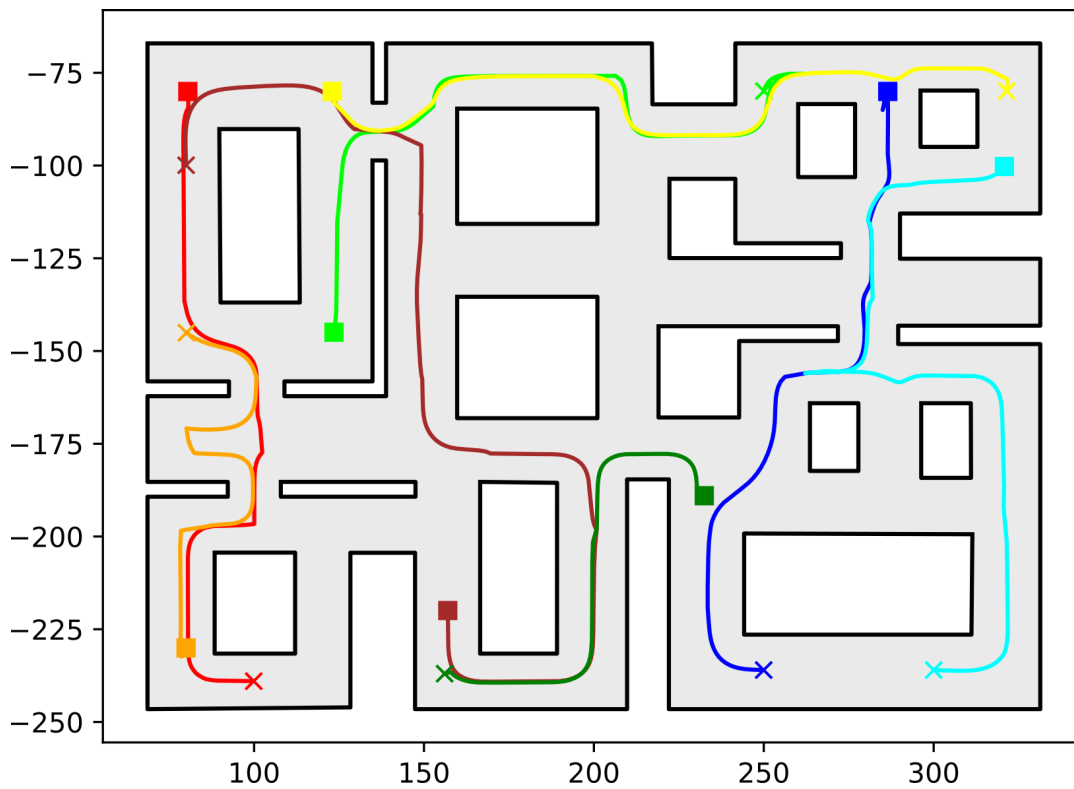


FIGURE 4.8: Executed trajectories of the 8 robot case. Squares indicate initial positions whereas the corresponding goal positions are depicted using crosses.

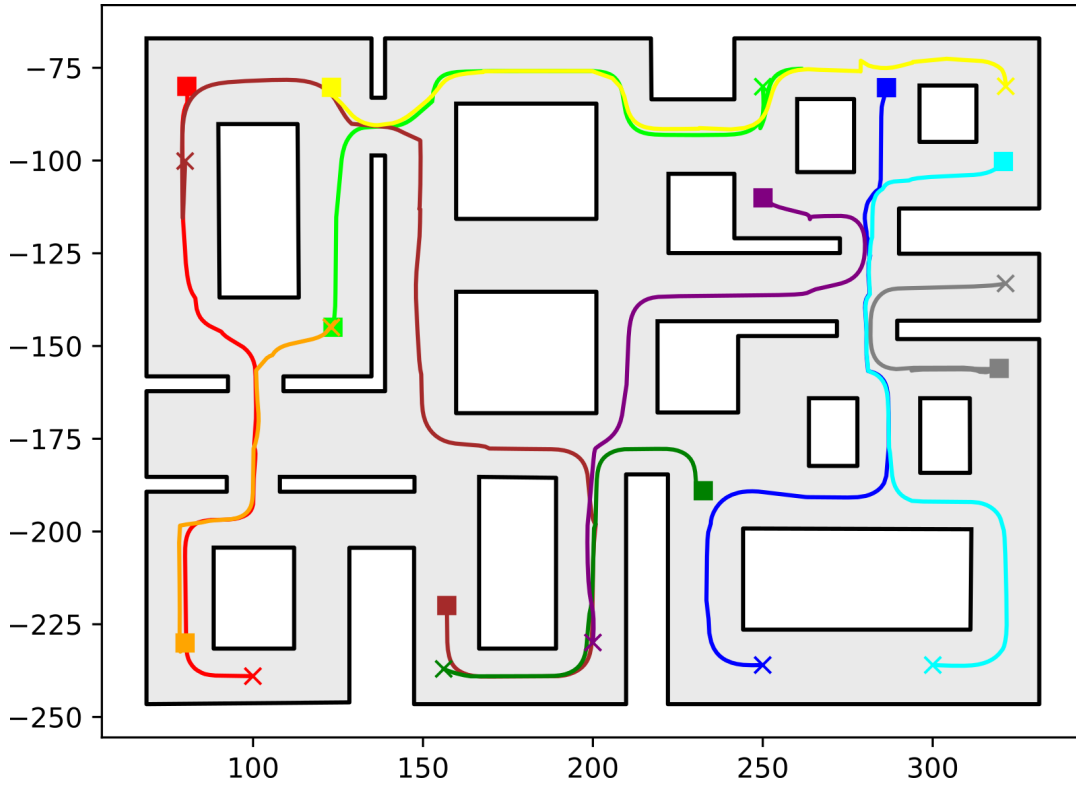


FIGURE 4.9: Executed trajectories of the 10 robot case. Squares indicate initial positions whereas the corresponding goal positions are depicted using crosses.

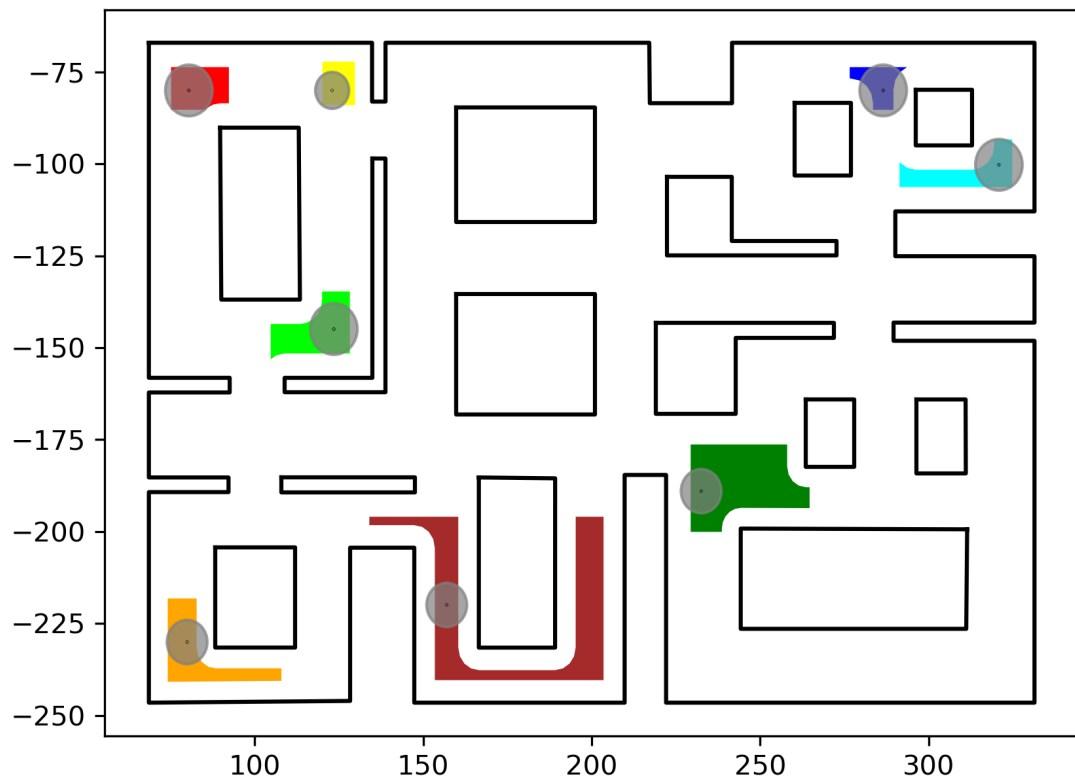


FIGURE 4.10: The initial robot positions $p_{init,i}$, $i \in \mathcal{I}_8$ and calculated initial compound cell \hat{C}_{init} .

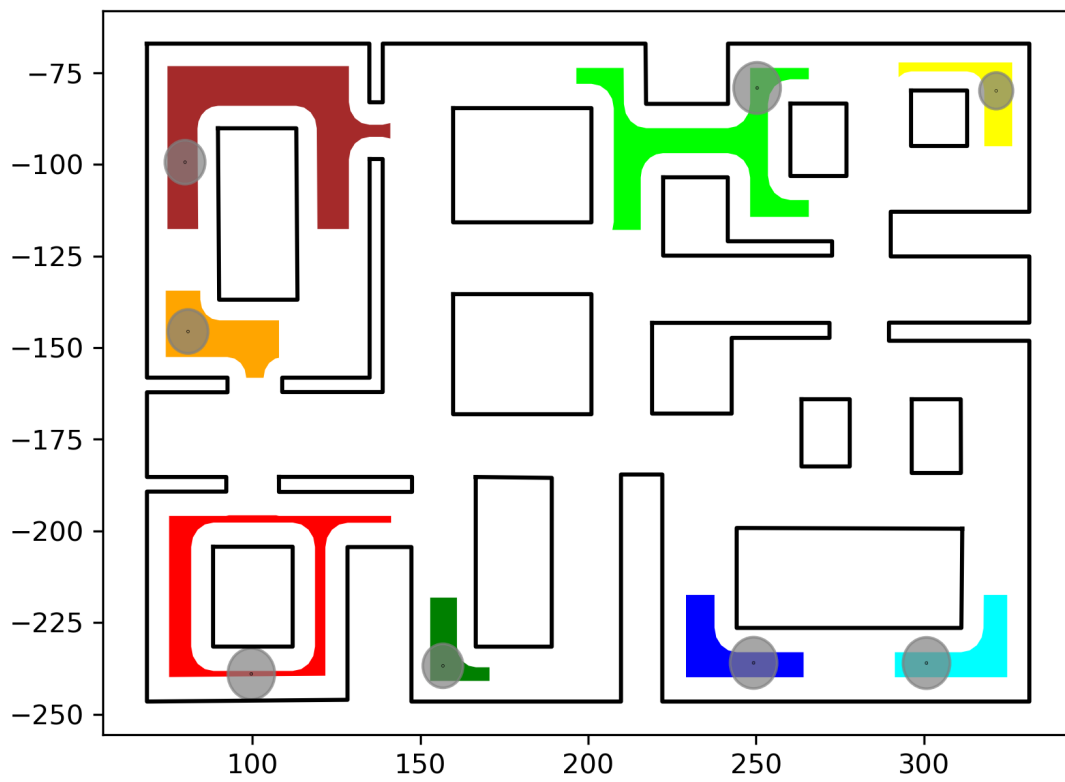


FIGURE 4.11: The desired robot positions $p_{\text{des},i}$, $i \in \mathcal{J}_8$ and calculated goal compound cell \hat{C}_{goal} .

Chapter 5

Cooperative Load Transportation



FIGURE 5.1: Cooperative object transportation by two robots.

5.1 Introduction

In this chapter, we present a methodology for coordinating transportation of an object carried by a team of mobile manipulators which operate within a compact planar workspace. Owing to the object rotation and the robot's manipulators, our scheme must take into consideration the varying shape of the robotic system in order to build a plan that can safely drive the robotic system to the goal configuration. To this end, we devise a high-level planner which is tasked with building a sequence of adjacent configuration space cells connecting the system's initial and desired configurations, each of which defines an allowable set of configurations for the object, as well as explicit intervals for each manipulator's states. Furthermore, appropriately

designed under- and over-approximations of the free configuration space are used in order to guide the configuration space's exploration by selecting the cells that need further subdivision. In addition, methodologies based on Reference Governors and Prescribed Performance Control are coupled with harmonic maps in order to design a distributed control law for realizing the specified cell transitions, which possesses guaranteed invariance and almost global convergence properties. Finally, we remark that the proposed low-level control law does not require continuous information exchange between the robots, which rely only on measurements of the object's current configuration and their own states in order to compute their respective control inputs.

The outline of this chapter is as follows. Firstly, we present some preliminary notation and definitions in [Section 5.2](#). Next, the problem tackled in this chapter is formulated in [Section 5.3](#). In [Section 5.4](#), we present the control scheme employed for driving the robotic system to the specified goal configuration while ensuring collision avoidance with the workspace boundary, and we elaborate on the closed-loop system's properties in [Section 5.5](#). Finally, we provide simulation results verifying the efficacy of our approach in [Section 5.6](#).

5.2 Preliminaries

Throughout this chapter, we shall use \mathbb{R} to denote the set of real numbers and \mathbb{N} to denote the set of natural numbers starting from zero. Moreover, we shall use $\mathcal{I}_N \triangleq \{1, 2, \dots, N\}$ (resp. $\mathcal{I}_N^* \triangleq \{0, 1, 2, \dots, N\}$) to denote the set consisting of all natural numbers up to N , starting from 1 (resp. 0). Additionally, given sets A and B , we use ∂A , $\text{int}(A)$, $\text{cl}(A)$ to denote the boundary, interior, closure respectively, and $A \setminus B$ to denote the complement of B w.r.t. A .

Given a coordinate frame \mathcal{F}_O in \mathbb{R}^2 and two points $P_A, P_B \in \mathbb{R}^2$, we will use ${}^{(O)}_{P_A}P_B$ to denote the position of point P_B relative to point P_A , whose coordinates are expressed w.r.t. \mathcal{F}_O . Given frames $\mathcal{F}_A, \mathcal{F}_B, \mathcal{F}_C$, we will use ${}^{\{A\}}_{\{B\}}P_{\{C\}} \in \mathbb{R}^2$ to denote the position of the origin of frame \mathcal{F}_C relative to the origin of frame \mathcal{F}_B , whose coordinates are expressed w.r.t. \mathcal{F}_A . Accordingly, given frames $\mathcal{F}_A, \mathcal{F}_B$, we will use ${}^{\{A\}}_{\{B\}}R \in \mathbb{R}^{2 \times 2}$ to denote the rotation matrix corresponding to the relative orientation of \mathcal{F}_B w.r.t. \mathcal{F}_A .

Given a rotation angle θ , let $R(\theta)$ be the rotation matrix defined as

$$R(\theta) \triangleq \begin{bmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{bmatrix}.$$

For two given coordinate frames $\mathcal{F}_A, \mathcal{F}_B$, we define ${}_{\{B\}}^{\{A\}}\mathcal{T}$ as the homogeneous transformation from frame \mathcal{F}_B to \mathcal{F}_A , defined as

$${}_{\{B\}}^{\{A\}}\mathcal{T} \triangleq \begin{bmatrix} {}_{\{B\}}^{\{A\}}R & {}_{\{A\}}P_{\{B\}} \\ 0 & 1 \end{bmatrix}.$$

We recall that the following equation holds for any given point P :

$$\begin{bmatrix} {}_{\{A\}}P \\ {}_{\{A\}}1 \end{bmatrix} = {}_{\{B\}}^{\{A\}}\mathcal{T} \cdot \begin{bmatrix} {}_{\{B\}}P \\ {}_{\{B\}}1 \end{bmatrix}$$

where ${}_{\{A\}}P$ is the position of P w.r.t. frame \mathcal{F}_A and ${}_{\{B\}}P$ is the position of P w.r.t. frame \mathcal{F}_B . For brevity's sake, we shall abuse notation slightly and write ${}_{\{A\}}P = {}_{\{B\}}^{\{A\}}\mathcal{T} \cdot {}_{\{B\}}P$ instead of the above when convenient.

5.3 Problem Formulation

We consider a compact workspace $\mathcal{W} \subseteq \mathbb{R}^2$ enclosed by a static outer boundary $\partial\mathcal{W}_0$ and N_o inner static boundaries $\partial\mathcal{W}_i$, $i \in \mathcal{I}_{N_o}$, with $N_o \in \mathbb{N}$. More specifically, we assume that \mathcal{W} can be written as follows:

$$\mathcal{W} \triangleq \overline{\mathcal{O}}_0 \setminus \bigcup_{i \in \mathcal{I}_{N_o}} \mathcal{O}_i. \quad (5.1)$$

where \mathcal{O}_0 denotes the area that lies outside of $\partial\mathcal{W}_0$ with $\overline{\mathcal{O}}_0 \triangleq \mathbb{R}^2 \setminus \mathcal{O}_0$, and \mathcal{O}_i denotes the area enclosed by $\partial\mathcal{W}_i$, for all $i \in \mathcal{I}_{N_o}$ (see [Figure 5.2](#)). We shall also use $\overline{\mathcal{W}}$ to denote the complement of \mathcal{W} w.r.t. \mathbb{R}^2 , i.e., $\overline{\mathcal{W}} \triangleq \mathbb{R}^2 \setminus \mathcal{W}$, which is assumed to be closed. In addition, the workspace outer boundary $\partial\mathcal{W}_0 \triangleq \partial\mathcal{O}_0$ and its inner boundaries $\partial\mathcal{W}_i \triangleq \partial\mathcal{O}_i$, $i \in \mathcal{I}_{N_o}$ are considered to be disjoint Jordan curves. Without loss of generality, we assume that \mathcal{W} is embedded with the arbitrarily positioned and oriented inertial frame $\mathcal{F}_{\mathcal{W}}$.

We now consider an object $\mathcal{L} \subset \mathbb{R}^2$ whose body is a compact, closed, polygonal 2-manifold, able to translate and rotate freely within \mathcal{W} as long as it is not in contact with the workspace boundary. Let $\mathcal{F}_{\mathcal{L}}$ be a fixed coordinate frame arbitrarily embedded in \mathcal{L} . We shall use $p_{\mathcal{L}}$ and $\theta_{\mathcal{L}}$ to denote the current position and orientation of \mathcal{L} w.r.t. $\mathcal{F}_{\mathcal{W}}$, i.e.:

$$p_{\mathcal{L}} \triangleq {}_{\{\mathcal{W}\}}^{\{\mathcal{L}\}}P_{\{\mathcal{L}\}} \quad R(\theta_{\mathcal{L}}) \triangleq {}_{\{\mathcal{L}\}}^{\{\mathcal{W}\}}R.$$

Object \mathcal{L} is considered a rigid body and let $M_{\mathcal{L}}, P_{\mathcal{L},\text{com}}, I_{\mathcal{L}}$ denote the object's mass, its center of mass, and its moment of inertia about $P_{\mathcal{L},\text{com}}$, respectively, expressed w.r.t. frame $\mathcal{F}_{\mathcal{L}}$. Assuming that $P_{\mathcal{L},\text{com}}$ coincides with the origin of $\mathcal{F}_{\mathcal{L}}$, the dynamics of \mathcal{L} is given by:

$$\begin{aligned} M_{\mathcal{L}} \cdot \ddot{p}_{\mathcal{L}} &= \tau_{\mathcal{L},p} \\ I_{\mathcal{L}} \cdot \ddot{\theta}_{\mathcal{L}} &= \tau_{\mathcal{L},\theta} \end{aligned}$$

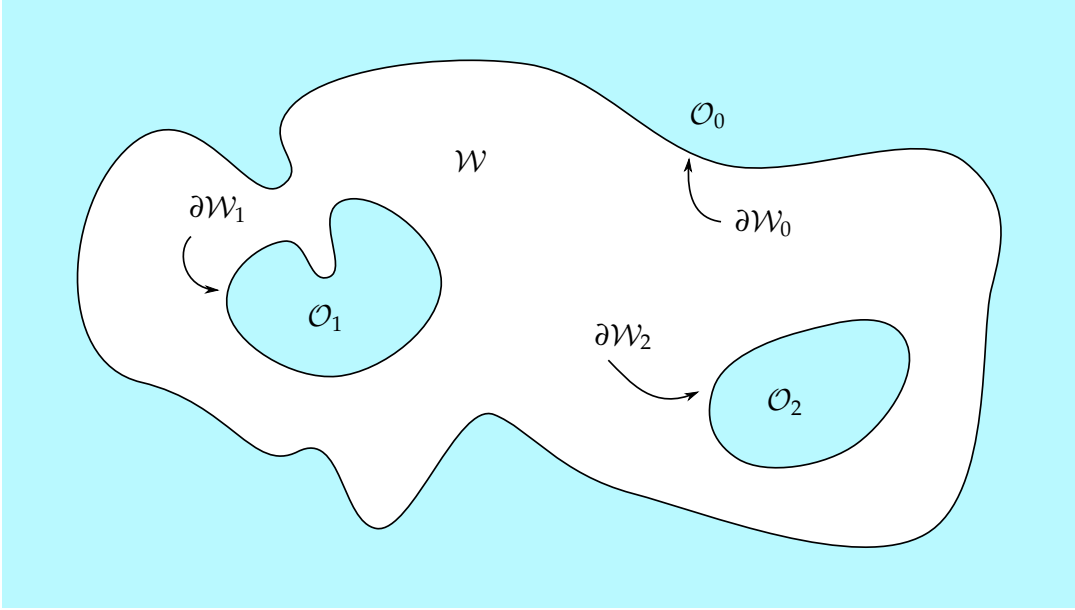


FIGURE 5.2: Typical workspace.

where $\tau_{\mathcal{L},p} \in \mathbb{R}^2$ and $\tau_{\mathcal{L},\theta} \in \mathbb{R}$ are the force and torque applied externally to the object. Lastly, we define $\mathcal{L}(p, \theta)$ as the footprint of \mathcal{L} , i.e., the space of \mathcal{W} that the body of \mathcal{L} occupies when $p_{\mathcal{L}} = p$ and $\theta_{\mathcal{L}} = \theta$.

In order to transport object \mathcal{L} from an initial configuration to a desired one, a team of $N_{\mathcal{R}} \geq 2$ cooperating mobile manipulators is employed. More specifically, each robot \mathcal{R}_i , $i \in \mathcal{I}_{N_{\mathcal{R}}}$ consists of a holonomic base platform \mathcal{B}_i and a manipulator \mathcal{A}_i which is attached to the base and is equipped with an end-effector $\mathcal{A}_{i,E}$ that rigidly grasps object \mathcal{L} at a specified point, and is thus able to exert a wrench onto it. The kinematics and dynamics of each mobile manipulator \mathcal{R}_i , $i \in \mathcal{I}_{N_{\mathcal{R}}}$ is described in detail in section 5.3.1 and section 5.3.1, respectively. It is also assumed that the bodies of \mathcal{B}_i and \mathcal{A}_i can be described by compact, closed and connected 2-manifolds, for all $i \in \mathcal{I}_{N_{\mathcal{R}}}$.

Thus, given an initial configuration $q_{\mathcal{L},\text{init}} = [p_{\mathcal{L},\text{init}}^T, \theta_{\mathcal{L},\text{init}}]^T$ and a desired configuration $q_{\mathcal{L},\text{goal}} = [p_{\mathcal{L},\text{goal}}^T, \theta_{\mathcal{L},\text{goal}}]^T$ for the object \mathcal{L} , our goal is to design a control scheme for the mobile manipulators \mathcal{R}_i , $i \in \mathcal{I}_{N_{\mathcal{R}}}$ which can drive the object to its destination, if a path between the two configurations exists, while ensuring that neither the object nor the robots will collide with the workspace boundary $\partial\mathcal{W}$. In addition, if the given problem is infeasible (i.e., no collision-free path connecting the given configurations exists) our control scheme should be able to conclude so in finite time.

5.3.1 Mobile Manipulator Kinematics

For each $i \in \mathcal{I}_{N_{\mathcal{R}}}$, let $\mathcal{F}_{\mathcal{B}_i}$ be a body-fixed frame arbitrarily embedded in \mathcal{B}_i . Without loss of generality, we assume that the origin of $\mathcal{F}_{\mathcal{B}_i}$ coincides with the center of

rotation of the base platform \mathcal{B}_i . For brevity's sake, let p_i and θ_i denote the current position and orientation of $\mathcal{F}_{\mathcal{B}_i}$ w.r.t. $\mathcal{F}_{\mathcal{W}}$, i.e.:

$$p_i \triangleq \begin{matrix} \{W\} \\ \{W\} \end{matrix} P_{\{\mathcal{B}_i\}} \quad R(\theta_i) \triangleq \begin{matrix} \{W\} \\ \{\mathcal{B}_i\} \end{matrix} R.$$

Furthermore, we will use $\mathcal{B}_i(p, \theta)$ to denote the footprint of the base platform of robot \mathcal{R}_i when it is centered at p with orientation θ .

Regarding the manipulator \mathcal{A}_i affixed to robot \mathcal{R}_i , we assume that it consists of one or more links $\mathcal{A}_{i,j}$, $j \in N_{\mathcal{A}_i}$ which are connected such that they form an open chain. Furthermore, the first link $\mathcal{A}_{i,1}$ is rigidly affixed to the base platform \mathcal{B}_i , whereas the end-effector is rigidly affixed to the last link $\mathcal{A}_{i,N_{\mathcal{A}_i}}$, for all $i \in \mathcal{I}_{N_{\mathcal{R}}}$. The indexing of the remaining links of each manipulator is such that the body of link $\mathcal{A}_{i,j+1}$ is able to either rotate or slide about the joint it shares with link $\mathcal{A}_{i,j}$. For each manipulator \mathcal{A}_i , we shall use $q_{i,j}$ and $\mathcal{D}_{q_{i,j}}$ to denote the state and domain, respectively, of the j -th degree of freedom, corresponding to the joint between links $\mathcal{A}_{i,j}$ and $\mathcal{A}_{i,j+1}$, for all $j \in \mathcal{I}_{N_{\mathcal{A}_i}-1}$ and $i \in \mathcal{I}_{N_{\mathcal{R}}}$. We remark that each domain $\mathcal{D}_{q_{i,j}}$ is a subset of either \mathbb{R} or \mathbb{S}^1 depending on whether the joint is prismatic or revolute, respectively. The augmented state vector z_i of robot \mathcal{R}_i as follows:

$$z_i \triangleq [p_i^T, \theta_i, q_i^T]^T$$

where q_i is the stacked vector of joint states of manipulator \mathcal{A}_i , for all $i \in \mathcal{I}_{N_{\mathcal{R}}}$. Similarly, for each $i \in \mathcal{I}_{N_{\mathcal{R}}}$ and $j \in \mathcal{I}_{N_{\mathcal{A}_i}}$, let $\mathcal{F}_{\mathcal{A}_{i,j}}$ be a body-fixed frame arbitrarily embedded in $\mathcal{A}_{i,j}$. Additionally, we affix an arbitrary coordinate frame \mathcal{F}_{E_i} at the point of contact between the end-effector of manipulator \mathcal{A}_i and the object \mathcal{L} . For the sake of simplicity and without harming generality, we assume herein that a) the origin of frame $\mathcal{F}_{\mathcal{A}_{i,j+1}}$ lies on the axis of rotation or sliding of the j -th joint, and b) the origin of frame \mathcal{F}_{E_i} coincides with the corresponding contact point (see Figure 5.3).

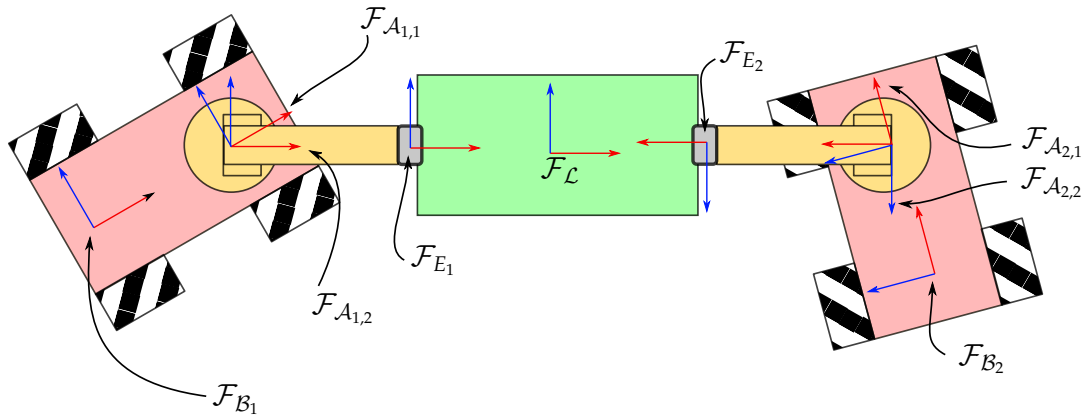


FIGURE 5.3: Example of robotic system consisting of two mobile manipulators carrying a rectangular object. Each platform is equipped with a 2-link manipulator, which is able to rotate about the joint with the base.

Regarding each robot's forward kinematics, we shall use $\mathcal{T}_{\mathcal{B}_i}(p, \theta)$ to denote the rigid transformation from $\mathcal{F}_{\mathcal{B}_i}$ to $\mathcal{F}_{\mathcal{W}}$ when the robot's is placed at p with orientation θ , i.e., ${}^{\{\mathcal{W}\}}P_{\{\mathcal{B}_i\}} = p$ and ${}^{\{\mathcal{W}\}}R_{\{\mathcal{B}_i\}} = R(\theta)$. Additionally, let $\mathcal{T}_{\mathcal{A}_i}(q_i)$ be the forward kinematics of manipulator \mathcal{A}_i , i.e., $\mathcal{T}_{\mathcal{A}_i}(q_i) \triangleq {}^{\{\mathcal{A}_{i,1}\}}\mathcal{T}_{\{\mathcal{E}_i\}}$. Since the manipulator of each robot is rigidly attached to its base, there exists a fixed homogeneous transformation, denoted by $\mathcal{T}_{\mathcal{B}_i, \mathcal{A}_i}$, between the base \mathcal{B}_i and the manipulator's first link $\mathcal{A}_{i,1}$, i.e.: $\mathcal{T}_{\mathcal{B}_i, \mathcal{A}_i} \triangleq {}^{\{\mathcal{B}_i\}}\mathcal{T}_{\{\mathcal{A}_{i,1}\}}$. The forward kinematics $\mathcal{T}_{\mathcal{R}_i}(p, \theta, q)$ of robot \mathcal{R}_i is given by:

$$\mathcal{T}_{\mathcal{R}_i}(p, \theta, q) \triangleq \mathcal{T}_{\mathcal{B}_i}(p, \theta) \cdot \mathcal{T}_{\mathcal{B}_i, \mathcal{A}_i} \cdot \mathcal{T}_{\mathcal{A}_i}(q).$$

Lastly, for each $i \in \mathfrak{I}_{N_{\mathcal{R}}}$, we shall use $\mathcal{J}_{\mathcal{R}_i}$ and $\mathcal{J}_{\mathcal{A}_i}$ to denote the Jacobian matrices of robot \mathcal{R}_i and its manipulator \mathcal{A}_i , i.e.:

$$\begin{bmatrix} {}^{\{\mathcal{W}\}}\dot{P}_{\{\mathcal{E}_i\}} \\ {}^{\{\mathcal{W}\}}\Omega_{\{\mathcal{E}_i\}} \end{bmatrix} = \mathcal{J}_{\mathcal{R}_i}(z_i) \cdot z_i$$

and

$$\begin{bmatrix} {}^{\{\mathcal{B}_i\}}\dot{P}_{\{\mathcal{E}_i\}} \\ {}^{\{\mathcal{B}_i\}}\Omega_{\{\mathcal{E}_i\}} \end{bmatrix} = \mathcal{J}_{\mathcal{A}_i}(q_i) \cdot \dot{q}_i$$

where $\Omega_{\{\mathcal{E}_i\}}$ is the angular velocity of end-effector $\mathcal{A}_{i,E}$.

5.3.2 Mobile Manipulator Dynamics

The dynamics of each robot \mathcal{R}_i , $i \in \mathfrak{I}_{N_{\mathcal{R}}}$ is assumed to obey the standard Euler-Lagrange model, i.e.:

$$M_{\mathcal{R}_i}(z_i) \cdot \ddot{z}_i + C_{\mathcal{R}_i}(z_i, \dot{z}_i) \cdot \dot{z}_i + G_{\mathcal{R}_i}(z_i) = \tau_{m,i} - (\mathcal{J}_{\mathcal{R}_i}(z_i))^T \cdot \tau_{e,i} \quad (5.2)$$

where $M_{\mathcal{R}_i}$, $C_{\mathcal{R}_i}$, $G_{\mathcal{R}_i} \in \mathbb{R}^{(3+N_{\mathcal{A}_i}) \times (3+N_{\mathcal{A}_i})}$ are the corresponding mass, coriolis and gravity matrices, $\tau_{m,i} \in \mathbb{R}^{(3+N_{\mathcal{A}_i})}$ is the wrench applied by the robot's actuators to the robot, and $\tau_{e,i} \in \mathbb{R}^3$ is the wrench applied by the robot to the object \mathcal{L} via its end-effector.

5.4 Control Design

To address the aforementioned problem, we design a hybrid control scheme which consists of:

- a) a high-level controller that given an initial configuration $q_{\mathcal{L}, \text{init}}$ and a final configuration $q_{\mathcal{L}, \text{goal}}$, can compute a sequence of reachable intermediate goals for the robotic system, if a solution to the above problem exists, or determine its infeasibility otherwise, and

- b) a low-level controller which utilizes appropriate workspace transformations in order to drive the object and the mobile manipulators from each goal to the next while avoiding collisions with the workspace boundary.

More specifically, the high-level controller, presented in Section 5.4.1, constructs a partitioning of the system's configuration space into cells by adaptively subdividing the domain of the robotic system's degrees of freedom until a sequence of connected cells containing $q_{\mathcal{L},\text{init}}$ and $q_{\mathcal{L},\text{goal}}$ is found (if one exists). Then, for each cell, intermediate goals for the object's position $p_{\mathcal{L}}$ and orientation $\theta_{\mathcal{L}}$ are computed, as described in Section 5.4.2, and a suitable low-level control law is employed for driving the system to the corresponding goal configuration while ensuring forward invariance of the current configuration space cell.

5.4.1 Configuration Space Decomposition

In this subsection, we present the hierarchical cell decomposition scheme that shall be employed for designing a sequence of high-level, feasible instructions that define a "path" leading to the desired configuration. Before doing so, we shall first take a closer look at the configuration space \mathcal{C} of the aforementioned robotic system. Throughout this subsection, we shall model this system as one virtual robot \mathfrak{R} consisting of $N_{\mathfrak{R}} = 1 + \sum_{i \in \mathcal{I}_{N_{\mathcal{R}}}} N_{\mathcal{A}_i}$ connected components, which correspond to the object \mathcal{L} , the base platform \mathcal{B}_i and the links $\mathcal{A}_{i,j}$ of each mobile manipulator \mathcal{R}_i , for all $j \in \mathcal{I}_{N_{\mathcal{A}_i}}$ and $i \in \mathcal{I}_{N_{\mathcal{R}}}$.

One can readily see that the components of \mathfrak{R} form an undirected tree $\mathfrak{T}(\mathfrak{n}, \mathfrak{e})$, where \mathfrak{n} is the set of components and $\mathfrak{e} \subset \mathfrak{n} \times \mathfrak{n}$ is the set of connections between the nodes. We shall use \mathfrak{R}_i to denote the i -th component of \mathfrak{R} . A connection $(i, j) \in \mathfrak{e}$ implies that the j -th component is able to move (rotate, translate, slice) relative to the i -th component about a pivot point $\mathfrak{P}_{i,j}$. Furthermore, given $i \in \mathcal{I}_{N_{\mathfrak{R}}}$, we will use \mathfrak{n}_c^i to denote the children of component \mathfrak{R}_i , i.e. the set of components \mathfrak{R}_j such that $(i, j) \in \mathfrak{e}$, for all $j \in \mathcal{I}_{N_{\mathfrak{R}}}$. Moreover, \mathfrak{n}_p^i will be used to denote the parent \mathfrak{R}_j of component \mathfrak{R}_i , i.e. the sole component such that $(j, i) \in \mathfrak{e}$, if one exists. Accordingly, we define \mathfrak{n}_d^i and \mathfrak{n}_a^i as the set of descendants and ancestors, respectively, of component \mathfrak{R}_i . Without loss of generality, we can choose the indexing of the components such that the first component of \mathfrak{R} is the root of \mathfrak{T} , corresponding to the object \mathcal{L} . For simplicity's sake, we will use $\mathcal{F}_{\mathfrak{R}_i}$, $i \in \mathcal{I}_{N_{\mathfrak{R}}}$ to denote the coordinate frames embedded in each component of \mathfrak{R} and we shall refer to their origins as the reference point of the corresponding component, respectively. Also, let $p_{\mathfrak{R}} \triangleq [x_{\mathfrak{R}}, y_{\mathfrak{R}}]^T \in \mathbb{R}^2$ and $\theta_{\mathfrak{R}} \in \mathcal{D}_{\theta} \subseteq \mathbb{S}^1$ denote the relative position of the robotic system's reference point and the relative orientation of its coordinate frame $\mathcal{F}_{\mathfrak{R}_0}$ w.r.t. the workspace's coordinate frame $\mathcal{F}_{\mathcal{W}}$, respectively.

Regarding the coupling between components, we will refer to the joint between two connected components as prismatic (resp. revolute) if the child is able to slide

(resp. rotate) about the corresponding pivot point. We will use q_i and \mathcal{D}_{q_i} , $i \in \mathcal{I}_{N_q}$, to denote the degree of freedom and its domain, respectively, corresponding to the joint between the i -th component and its parent, where $N_q \triangleq \sum_{j \in \mathcal{I}_{N_R}} (N_{A_j} - 1)$. Without loss of generality, since each component other than the root has exactly one parent, we assume that each pivot $\mathfrak{P}_{i,j}$ coincides with the origin of frame $\mathcal{F}_{\mathfrak{R}_j}$. Furthermore, by treating the orientation $\theta_{\mathcal{L}}$ of the object as a virtual joint state, the state z of the virtual robotic system \mathfrak{R} is defined as follows:

$$z \triangleq [p_{\mathfrak{R}}^T, \theta_{\mathfrak{R}}, q^T]^T = [p_{\mathfrak{R}}^T, q^T]^T$$

where $q \triangleq [q_i]_{i \in \mathcal{I}_{N_q}^*}$, is the stacked vector of virtual joint parameters with $q_0 \triangleq \theta_{\mathfrak{R}}$ and $q_i \triangleq q_i$ for all $i \in \mathcal{I}_{N_q}$.

Let us now consider the footprint of the robotic system while it moves within the workspace. We notice that, for each $i \in \mathcal{I}_{N_{\mathfrak{R}}}$, the footprint of the individual component \mathfrak{R}_i , i.e., the area occupied by it at a given configuration, is defined by the position of its pivot point and the current value of its (virtual) joint parameter. We shall use $\mathfrak{R}_i(p, q)$ to denote the footprint when the pivot point is placed at p and the joint parameter value is q . We also remark that, although each component may move freely w.r.t. its pivot point, any motion of theirs propagates directly to their children, thus potentially inducing a translation and/or rotation onto every one of its descendants n_d^i . Thus, the footprint of component \mathfrak{R}_i can also be defined in terms of the current position $p_{\mathfrak{R}}$ of the robotic system and the (virtual) joint parameters of every component \mathfrak{R}_j belonging to n_d^i . By remarking that the footprint $\mathfrak{R}(z)$ of the robotic system at a given configuration z is simply the union of the footprints of its individual components, i.e.:

$$\mathfrak{R}(p_{\mathfrak{R}}, q) \triangleq \bigcup_{i \in \mathcal{I}_{N_{\mathfrak{R}}}} \mathfrak{R}_i(p_{\mathfrak{R}}, [q_j]_{j \in n_d^i \cup i}^T) \quad (5.3)$$

we are now ready to formally define the set of admissible configurations to our problem. For brevity, $\mathfrak{R}(q)$ will be used instead of $\mathfrak{R}(0, q)$ where is deemed preferable. By noticing that the configuration space \mathcal{C} of this robotic system is a manifold diffeomorphic to $\mathbb{R}^2 \times \mathbb{S}^1 \times \mathcal{D}_{q_1} \times \dots \times \mathcal{D}_{q_{N_q}}$ and recalling that neither the object \mathcal{L} nor any of the robots \mathcal{R}_i , $i \in \mathcal{I}_{N_R}$ are allowed to collide with the workspace boundary $\partial\mathcal{W}$, the set \mathcal{C}_f of collision free configurations of \mathfrak{R} is given by:

$$\mathcal{C}_f = \{z \mid \mathcal{W} \cap \mathfrak{R}(z) = \emptyset \text{ and } z \in \mathcal{C}\}. \quad (5.4)$$

Finally, let $\mathcal{C}_o \triangleq \mathcal{C} \setminus \mathcal{C}_f$.

Now, in order to design a continuous “path” inside \mathcal{C}_f connecting the two given configurations $q_{\mathcal{L}, \text{init}}$ and $q_{\mathcal{L}, \text{goal}}$, we extend the methodology presented in Chapter 3. More specifically, by designing a suitable cover of the free configuration space via

recursive subdivision of the domain of \mathbf{q} , our goal is to obtain a hierarchical partitioning of \mathcal{C}_f . For each implicitly defined cell, we compute over- and under-approximations, whose shape is much simpler than the shape of the corresponding exact cell, which are used for both guiding the configuration space's exploration, as well as designing a high-level plan will drive the robotic system to its goal. To do so, we first consider the domain \mathcal{D}_{q_i} of the joint state q_i , for $i \in \mathcal{I}_{N_q}^*$. Furthermore, we shall refer to a set of the form $\mathcal{S}_{[q_{i1}, q_{i2}]}^{q_i}$ as a simple slice of the parameter q_i , where $q_{i1}, q_{i2} \in \mathcal{D}_{q_i}$. Also, a set $\mathfrak{S}_i = \{\mathcal{S}_j^{q_i} \mid j \in \mathcal{I}_{N_{\mathfrak{S}_i}}\}$ consisting of $N_{\mathfrak{S}_i}$ simple slices of q_i shall be called a cover of \mathcal{D}_{q_i} iff

$$\mathcal{D}_{q_i} = \bigcup_{j \in \mathcal{I}_{N_{\mathfrak{S}_i}}} \mathcal{S}_j^{q_i}$$

for all $k, \ell \in \mathcal{I}_{N_{\mathfrak{S}_i}}$ with $k \neq \ell$. A compound slice $\widehat{\mathcal{S}}$ is defined as a set of simple slices of the form $\widehat{\mathcal{S}} = \{\mathcal{S}^{q_i} \mid i \in \mathcal{I}_{N_q}^*\}$. Respectively, a set $\widehat{\mathfrak{S}} = \{\mathfrak{S}_i \mid i \in \mathcal{I}_{N_q}^*\}$ is called a cover of the free configuration space \mathcal{C}_f iff each \mathfrak{S}_i is a cover of \mathcal{D}_{q_i} . We note that a cover $\widehat{\mathfrak{S}}$ induces a partitioning of \mathcal{C}_f into regions

$$\mathcal{C}_{\widehat{\mathfrak{S}}} = \{[p_{\mathfrak{R}}^T, \mathbf{q}^T]^T \mid p_{\mathfrak{R}} \in \mathcal{W} \text{ and } \mathbf{q} \in \widehat{\mathcal{S}}\} \cap \mathcal{C}_f, \quad \widehat{\mathcal{S}} \in \widehat{\mathfrak{S}} \quad (5.5)$$

each of which consists of $N_{\widehat{\mathcal{S}}} \geq 0$ individually connected but pairwise disjoint subsets $\widehat{\mathcal{C}}_{\widehat{\mathcal{S}}, i}$, $i \in \mathcal{I}_{N_{\widehat{\mathcal{S}}}}$. Taking a closer look at the connectedness of these cells, one can readily see that two configuration space cells $\widehat{\mathcal{C}}_{\widehat{\mathcal{S}}, i}$ and $\widehat{\mathcal{C}}_{\widehat{\mathcal{S}}, j}$ are adjacent and the projections of $\widehat{\mathcal{C}}_{\widehat{\mathcal{S}}, i}$, $\widehat{\mathcal{C}}_{\widehat{\mathcal{S}}, j}$ onto the plane intersect. We recall that two distinct simple slices $\mathcal{S}_i^{q_k}$ and $\mathcal{S}_j^{q_k}$ are called adjacent iff their intersection $\mathcal{S}_i^{q_k} \cap \mathcal{S}_j^{q_k}$ is not empty, whereas two compound slices $\widehat{\mathcal{S}}_i = \{\mathcal{S}_i^{q_k} \mid k \in \mathcal{I}_{N_q}^*\}$, and $\widehat{\mathcal{S}}_j = \{\mathcal{S}_j^{q_k} \mid k \in \mathcal{I}_{N_q}^*\}$, are called adjacent iff $\mathcal{S}_i^{q_k}$, $\mathcal{S}_j^{q_k}$ are adjacent, for all $k \in \mathcal{I}_{N_q}^*$.

Similarly to the method employed in Chapter 3, in order to avoid explicitly computing the shape of a given configuration space cell, we shall define suitable over- and under-approximations of it, which, in addition, shall be used for guiding the configuration space exploration in a similar manner. In order to build these approximations of the set of free configurations corresponding to the compound slice $\widehat{\mathcal{S}} = \{\mathcal{S}^{q_i} \mid i \in \mathcal{I}_{N_q}^*\}$, we first compute an over-approximation $\overline{\mathfrak{R}}(\widehat{\mathcal{S}})$ and an under-approximation $\underline{\mathfrak{R}}(\widehat{\mathcal{S}})$ of the robotic system's footprint as follows:

$$\begin{aligned} \overline{\mathfrak{R}}(\widehat{\mathcal{S}}) &= \bigcup_{\mathbf{q} \in \mathcal{I}_{\widehat{\mathcal{S}}}} \mathfrak{R}(\mathbf{q}) \\ \underline{\mathfrak{R}}(\widehat{\mathcal{S}}) &= \bigcap_{\mathbf{q} \in \mathcal{I}_{\widehat{\mathcal{S}}}} \mathfrak{R}(\mathbf{q}) \end{aligned} \quad (5.6)$$

where

$$\mathcal{I}_{\widehat{\mathcal{S}}} = \mathcal{S}^{q_0} \times \mathcal{S}^{q_1} \times \dots \times \mathcal{S}^q. \quad (5.7)$$

We remark that, although seemingly daunting at first sight, the computation of $\overline{\mathfrak{R}}(\widehat{\mathcal{S}})$ and $\underline{\mathfrak{R}}(\widehat{\mathcal{S}})$ can be significantly simplified by recalling that the footprint of each component does not necessarily depends on every component of \mathfrak{q} but only on those of its ancestors, by virtue of the robotic system's tree-like structure. An example of such over- and under-approximation for a robotic system consisting of two connected components can be seen in [Figure 5.4](#). Following these definitions, the over-approximation $\overline{\mathcal{C}}_{\widehat{\mathcal{S}}}$ and the under-approximation $\underline{\mathcal{C}}_{\widehat{\mathcal{S}}}$ of a given partition $\mathcal{C}_{\widehat{\mathcal{S}}}$, can be computed as follows:

$$\begin{aligned}\overline{\mathcal{C}}_{\widehat{\mathcal{S}}} &\triangleq \{[p^T, q^T]^T \mid p \in \overline{\mathcal{W}}_{\widehat{\mathcal{S}}} \text{ and } q \in \mathcal{I}_{\widehat{\mathcal{S}}}\} \\ \underline{\mathcal{C}}_{\widehat{\mathcal{S}}} &\triangleq \{[p^T, q^T]^T \mid p \in \underline{\mathcal{W}}_{\widehat{\mathcal{S}}} \text{ and } q \in \mathcal{I}_{\widehat{\mathcal{S}}}\}\end{aligned}\tag{5.8}$$

where

$$\begin{aligned}\overline{\mathcal{W}}_{\widehat{\mathcal{S}}} &\triangleq \mathcal{W} \ominus \underline{\mathfrak{R}}(\widehat{\mathcal{S}}) \\ \underline{\mathcal{W}}_{\widehat{\mathcal{S}}} &\triangleq \mathcal{W} \ominus \overline{\mathfrak{R}}(\widehat{\mathcal{S}})\end{aligned}\tag{5.9}$$

with $A \ominus B$ denoting the Minkowski difference of sets A and B , and $\mathcal{I}_{\widehat{\mathcal{S}}} = \mathcal{S}^{q_0} \times \mathcal{S}^{q_1} \times \dots \times \mathcal{S}^{q_l}$. Obviously, each of $\overline{\mathcal{C}}_{\widehat{\mathcal{S}}}$ and $\underline{\mathcal{C}}_{\widehat{\mathcal{S}}}$ consists of individually connected but pairwise disjoint cells $\overline{\mathcal{C}}_{\widehat{\mathcal{S}},i'}$, $i' \in \mathcal{I}_{N_{\overline{\mathcal{C}}_{\widehat{\mathcal{S}}}}}$ and $\underline{\mathcal{C}}_{\widehat{\mathcal{S}},i'}$, $i' \in \mathcal{I}_{N_{\underline{\mathcal{C}}_{\widehat{\mathcal{S}}}}}$, respectively, which enclose or are enclosed by the cells of $\mathcal{C}_{\widehat{\mathcal{S}}}$.

At this point, we remark that the approximation of \mathcal{C}_f improves as one subdivides the configuration space into more and finer slices. Thus, choosing a sufficiently fine partitioning of \mathcal{C} , a sequence of adjacent under-approximation cells connecting $q_{\mathcal{L},\text{init}}$ and $q_{\mathcal{L},\text{goal}}$ will appear, as long as one exists in the first place. Instead of choosing such a fine partitioning arbitrary, similarly to the methodology proposed in [chapter 2](#), we design an adaptive subdivision scheme which makes also use of the space's over-approximations for choosing which slice to subdivide at each iteration. More specifically, we design an algorithm which given compound cover $\widehat{\mathcal{S}}$, it tries to find a sequence $\underline{\Pi}$ of adjacent under-approximation cells connecting the initial and goal configurations. If no such path can be found, then our algorithm tries to connect the two given configurations with a sequence $\overline{\Pi}$ made of adjacent over-approximation cells instead. If such a path exists, then a slice corresponding to a cell of $\overline{\Pi}$ is selected according to a suitable heuristic and gets subdivided, producing a new partitioning of \mathcal{C}_f . Otherwise, if no such path can be found, then this obviously indicates that the problem at hand is infeasible (i.e., the two given configurations exist in disjoint components of the robotic system's configuration space) and our algorithm terminates. In short, one can readily verify that the following statements hold:

1. If there exists a path of adjacent under-approximation cells for a given cover $\widehat{\mathcal{S}}$ containing $q_{\mathcal{L},\text{init}}$ and $q_{\mathcal{L},\text{goal}}$, then a solution to our problem exists.
2. If there exists a path of adjacent over-approximation cells for a given cover $\widehat{\mathcal{S}}$ containing $q_{\mathcal{L},\text{init}}$ and $q_{\mathcal{L},\text{goal}}$, then whether our problem has a solution is unknown and further expansion of $\widehat{\mathcal{S}}$ is in order.

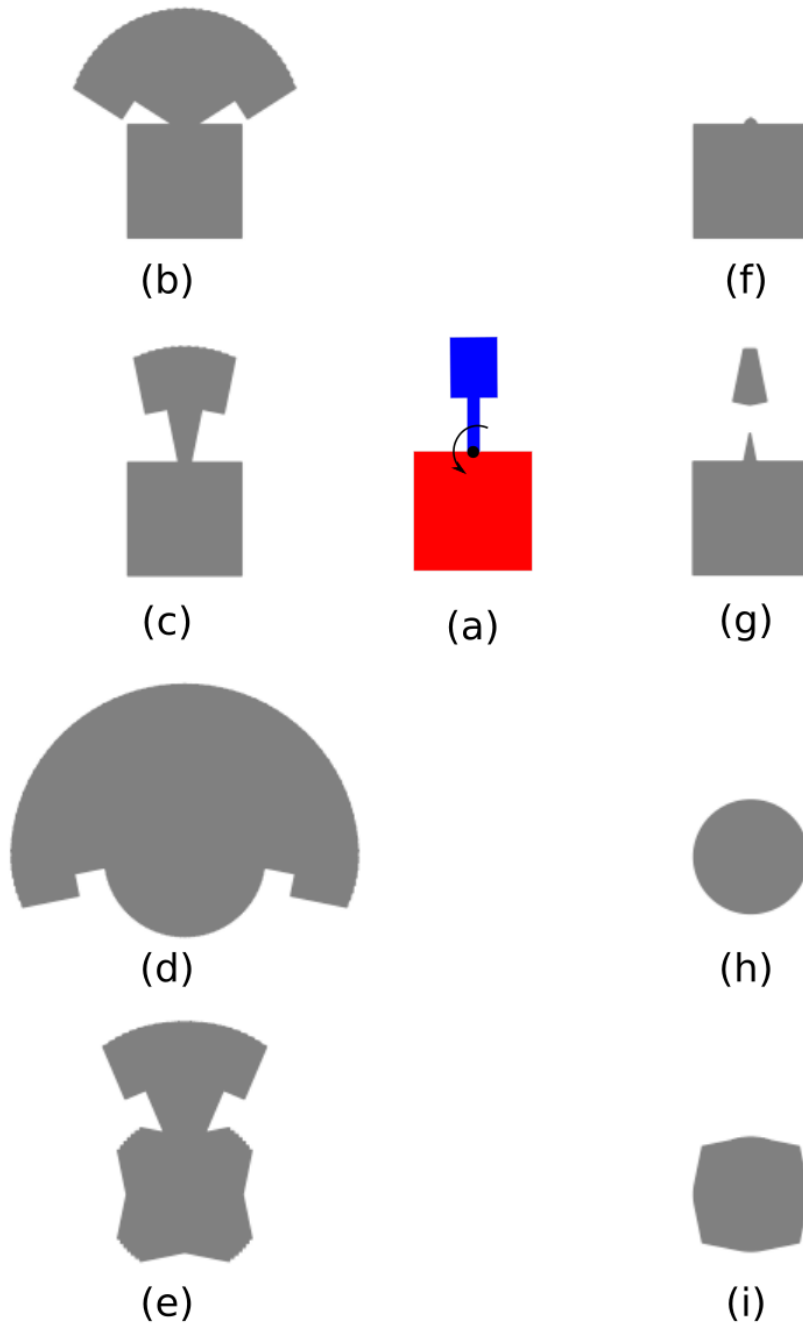


FIGURE 5.4: Over- and under-approximations of a robotic system (a) consisting of two components corresponding to slices: $\{0\} \times [2\pi - 1, 1]$ (b and f), $\{0\} \times [2\pi - 0.2, 0.2]$ (c and g), $[3\pi/2, \pi/2] \times [2\pi - 0.2, 0.2]$ (d and h), $[2\pi - 0.2, 0.2] \times [2\pi - 0.2, 0.2]$ (e and i).

3. If there is no path of adjacent over-approximation cells for a given cover $\widehat{\mathcal{S}}$ containing $q_{\mathcal{L},\text{init}}$ and $q_{\mathcal{L},\text{goal}}$, then our problem is infeasible.

Algorithm 6 Configuration space exploration algorithm.

```

function CONNECTCONFIGS( $q_{\mathcal{L},\text{init}}, q_{\mathcal{L},\text{goal}}$ )
   $\widehat{\mathcal{S}} \leftarrow \{\mathcal{D}_{q_i} \mid i \in \mathcal{I}_{N_q}^*\}$ 
  loop
     $\widehat{\mathcal{S}}, \underline{\mathcal{C}}_{\text{init}}, \overline{\mathcal{C}}_{\text{init}} \leftarrow \text{FINDENCLOSINGCELLS}(q_{\mathcal{L},\text{init}}, \widehat{\mathcal{S}})$ 
     $\widehat{\mathcal{S}}, \underline{\mathcal{C}}_{\text{goal}}, \overline{\mathcal{C}}_{\text{goal}} \leftarrow \text{FINDENCLOSINGCELLS}(q_{\mathcal{L},\text{goal}}, \widehat{\mathcal{S}})$ 

    if  $\overline{\mathcal{C}}_{\text{init}}$  is Nil or  $\overline{\mathcal{C}}_{\text{goal}}$  is Nil then
      return Nil
    end if
     $\underline{\Pi} \leftarrow \text{CONNECTUACELLS}(\underline{\mathcal{C}}_{\text{init}}, \underline{\mathcal{C}}_{\text{goal}})$ 
    if  $\underline{\Pi}$  is Nil then
       $\overline{\Pi} \leftarrow \text{CONNECTOACELLS}(\overline{\mathcal{C}}_{\text{init}}, \overline{\mathcal{C}}_{\text{goal}})$ 
      if  $\overline{\Pi}$  is empty then
        return Nil
      else
         $\widehat{\mathcal{S}} \leftarrow \text{REFINE}(\widehat{\mathcal{S}}, \overline{\Pi}, \{\underline{\mathcal{C}}_{\text{init}}\}, \{\underline{\mathcal{C}}_{\text{goal}}\})$ 
      end if
    else
      return  $\underline{\Pi}$ 
    end if
  end loop
end function

```

The proposed algorithm can be seen in [Algorithm 6](#). More specifically, we begin the configuration space exploration with a rough partitioning of \mathcal{C} induced by a compound slice covering the entire domain of virtual joint parameters q . Then, we search for cells $\underline{\mathcal{C}}_{\text{init}}$ and $\underline{\mathcal{C}}_{\text{goal}}$ containing the robot's initial and final configurations, respectively, by subdividing $\widehat{\mathcal{S}}$. If no such pair of cells exists, our initial problem is obviously infeasible and the algorithm terminates. Otherwise, we try to connect $\underline{\mathcal{C}}_{\text{init}}$ and $\underline{\mathcal{C}}_{\text{goal}}$ using the available under-approximation cells corresponding to $\widehat{\mathcal{S}}$. If this attempt fails, then we try instead to find a path of over-approximation cells connecting $\overline{\mathcal{C}}_{\text{init}}$ and $\overline{\mathcal{C}}_{\text{goal}}$. If such a path cannot be found, this also implies that no solution exists and the algorithm terminates. Otherwise, a heuristic is utilized for selecting a compound slice in $\widehat{\mathcal{S}}$ to be expanded and the process starts anew. The heuristic used, which can be seen in [Algorithm 7](#), selects which slice of $\widehat{\mathcal{S}}$ to expand as follows. Given a path $\overline{\Pi}$ of over-approximation cells, it essentially tries to construct a path made of the under-approximation cells that belong in the same compound slices as the elements of $\overline{\Pi}$. Failing to connect under-approximation cells belonging in two adjacent compound slices $\widehat{\mathcal{S}}_i$ and $\widehat{\mathcal{S}}_j$ indicates that the connectedness of the over- and under-approximation cells in this slices is not the same, which means that these slices need to be further expanded. Thus, the largest simple slice of these compound slices gets subdivided and the function returns. Finally, we remark that

the functions `CONNECTUACELLS` and `CONNECTOACELLS` employ standard graph search algorithms for constructing the corresponding paths based on a heuristic that penalizes cells with smaller slices (i.e., cells corresponding to larger slices are preferred).

Algorithm 7 Heuristic choosing next simple slice for subdivision.

```

function REFINE( $\widehat{\mathcal{G}}, \overline{\Pi}, src, dst$ )
  if  $len(\overline{\Pi}) = 1$  then
    return SUBDIVIDE( $\widehat{\mathcal{G}}, \overline{\Pi}[0]$ )
  else
     $cells \leftarrow \{\}$ 
     $connected \leftarrow \{\}$ 
    if  $len(\overline{\Pi}) == 2$  then
       $cells \leftarrow dst$ 
    else
       $\overline{\mathcal{C}}_{\widehat{S}_j} \leftarrow \overline{\Pi}[1]$ 
      for all  $\underline{\mathcal{C}}_{\widehat{S}_j}$  in  $\mathcal{C}_{\widehat{S}_j}$  do
         $cells \leftarrow cells \cup \{\underline{\mathcal{C}}_{\widehat{S}_j}\}$ 
      end for
    end if
    for all  $\underline{\mathcal{C}}_{\widehat{S}_i}$  in  $src$  do
      for all  $\underline{\mathcal{C}}_{\widehat{S}_j}$  in  $cells$  do
        if  $\underline{\mathcal{C}}_{\widehat{S}_i} \cap \underline{\mathcal{C}}_{\widehat{S}_j} \neq \emptyset$  then
           $connected \leftarrow connected \cup \{\underline{\mathcal{C}}_{\widehat{S}_j}\}$ 
        end if
      end for
    end for
    if  $len(connected) > 0$  then
      return REFINE( $\widehat{\mathcal{G}}, \overline{\Pi}[1:], connected, dst$ )
    else
      return SUBDIVIDELONGEST( $\widehat{\mathcal{G}}, \widehat{S}_i, \widehat{S}_j$ )
    end if
  end if
end function

```

5.4.2 Distributed Control Law

Given now a path $\overline{\Pi}$ of cells obtained by the high-level planner described in the previous sub-section, we shall now design a distributed control scheme for the mobile manipulators that ensures safe transitions from one cell to the next till the goal configuration $q_{\mathcal{L},goal}$ is reached. Let $\underline{\mathcal{C}}_{\widehat{S}}$ be a cell in $\overline{\Pi}$ and let $\underline{\mathcal{C}}_{\widehat{S}}$ denote its the projection on the plane. We recall that $\underline{\mathcal{C}}_{\widehat{S}}$ is an under-approximation of the actual free configuration space, constructed by extruding $\mathcal{W}_{\widehat{S}}$, which implies that, as long as $q \in \widehat{S}$, then $p_{\mathcal{L}}$ can safely occupy any position of $\mathcal{P}_{p_{\mathcal{L}}}(\underline{\mathcal{C}}_{\widehat{S}})$. We also note that $\underline{\mathcal{C}}_{\widehat{S}}$ is a non-empty, compact region of \mathbb{R}^2 with arbitrary connectedness and shape. Exploiting this fact, we can decouple the low-level control laws for: a) the object's position $p_{\mathcal{L}}$,

b) the object's orientation $\theta_{\mathcal{L}}$, and c) the joints q_i of each manipulator \mathcal{R}_i , $i \in \mathcal{I}_{N_{\mathcal{R}}}$, as explained in the following.

For each intermediate cell of $\underline{\Pi}$, we can obtain goal sets corresponding to $p_{\mathcal{L}}$, $\theta_{\mathcal{L}}$ and q separately, by computing its intersection with the next one (which is non-empty by construction of $\underline{\Pi}$), Let us consider a pair of consecutive cells $\underline{\mathcal{C}}_{\widehat{\mathcal{S}}_i}$ and $\underline{\mathcal{C}}_{\widehat{\mathcal{S}}_j}$ in $\underline{\Pi}$. Regarding the object's position, in order to safely traverse from $\underline{\mathcal{C}}_{\widehat{\mathcal{S}}_i}$ to $\underline{\mathcal{C}}_{\widehat{\mathcal{S}}_j}$, it is sufficient that $p_{\mathcal{L}}$ reaches the set

$$\mathcal{G}_{p_{\mathcal{L}}}(\underline{\mathcal{C}}_{\widehat{\mathcal{S}}_i}, \underline{\mathcal{C}}_{\widehat{\mathcal{S}}_j}) \triangleq \mathcal{P}_{p_{\mathcal{L}}}(\underline{\mathcal{C}}_{\widehat{\mathcal{S}}_i}) \cap \mathcal{P}_{p_{\mathcal{L}}}(\underline{\mathcal{C}}_{\widehat{\mathcal{S}}_j}). \quad (5.10)$$

We also note that $\mathcal{G}_{p_{\mathcal{L}}}(\underline{\mathcal{C}}_{\widehat{\mathcal{S}}_i}, \underline{\mathcal{C}}_{\widehat{\mathcal{S}}_j})$ is generally made of one or more disjoint subsets of arbitrary connectedness and that, as long as the object's position reaches either of these, the system can cross to the next cell. Respectively, a goal set corresponding to the object's orientation can be obtained by computing the intersection of the corresponding simple slices of $\widehat{\mathcal{S}}_i$ and $\widehat{\mathcal{S}}_j$, i.e.:

$$\mathcal{G}_{\theta_{\mathcal{L}}}(\underline{\mathcal{C}}_{\widehat{\mathcal{S}}_i}, \underline{\mathcal{C}}_{\widehat{\mathcal{S}}_j}) \triangleq \mathcal{S}_i^{q_0} \cap \mathcal{S}_j^{q_0}. \quad (5.11)$$

Goal sets for the joints of each mobile manipulator can be computed in a similar manner. Particularly, let $\mathcal{P}_{A_k}(\underline{\mathcal{C}}_{\widehat{\mathcal{S}}})$ denote the projection $\underline{\mathcal{C}}_{\widehat{\mathcal{S}}}$ along the dimensions corresponding to the degrees of freedom of A_k . Obviously, $\mathcal{P}_{A_k}(\underline{\mathcal{C}}_{\widehat{\mathcal{S}}})$ is equal to the product of the simple slices of $\widehat{\mathcal{S}}$ corresponding to q_k . Then, the corresponding goal set of q_k is given by

$$\mathcal{G}_{A_k}(\underline{\mathcal{C}}_{\widehat{\mathcal{S}}_i}, \underline{\mathcal{C}}_{\widehat{\mathcal{S}}_j}) \triangleq \mathcal{P}_{A_k}(\underline{\mathcal{C}}_{\widehat{\mathcal{S}}_i}) \cap \mathcal{P}_{A_k}(\underline{\mathcal{C}}_{\widehat{\mathcal{S}}_j}), \quad \forall k \in \mathcal{I}_{N_{\mathcal{R}}}. \quad (5.12)$$

Thus, for successfully driving the robotic system from $\underline{\mathcal{C}}_{\widehat{\mathcal{S}}_i}$ to $\underline{\mathcal{C}}_{\widehat{\mathcal{S}}_j}$, we need to design decoupled control laws for the mobile manipulators which:

- ensure invariance of the current cell, i.e. $p_{\mathcal{L}} \in \mathcal{P}_{p_{\mathcal{L}}}(\underline{\mathcal{C}}_{\widehat{\mathcal{S}}_i})$, $\theta_{\mathcal{L}} \in \mathcal{S}_i^{q_0}$ and $q_k \in \mathcal{P}_{A_k}(\underline{\mathcal{C}}_{\widehat{\mathcal{S}}_i})$, $\forall k \in \mathcal{I}_{N_{\mathcal{R}}}$, until the transition is complete, and
- ensure convergence of the system's states to the corresponding goals sets $\mathcal{G}_{p_{\mathcal{L}}}(\underline{\mathcal{C}}_{\widehat{\mathcal{S}}_i}, \underline{\mathcal{C}}_{\widehat{\mathcal{S}}_j})$, $\mathcal{G}_{\theta_{\mathcal{L}}}(\underline{\mathcal{C}}_{\widehat{\mathcal{S}}_i}, \underline{\mathcal{C}}_{\widehat{\mathcal{S}}_j})$ and $\mathcal{G}_{A_k}(\underline{\mathcal{C}}_{\widehat{\mathcal{S}}_i}, \underline{\mathcal{C}}_{\widehat{\mathcal{S}}_j})$, $k \in \mathcal{I}_{N_{\mathcal{R}}}$.

Finally, the transition is considered complete after all states have reached the corresponding goal sets. We remark that, regarding the last cell of $\underline{\Pi}$, the goal sets corresponding to the object's position and orientation can taken equal to $\{p_{\mathcal{L}, \text{goal}}\}$ and $\{\theta_{\mathcal{L}, \text{goal}}\}$, respectively, while the joints of the manipulators need only to remain within the bounds imposed by the last cell.

Before we proceed with formulating the corresponding control laws, we must first formally state the following assumptions about our system.

Assumption 1. Each robot \mathcal{R}_k , $k \in \mathfrak{I}_{N_{\mathcal{R}}}$ has exact knowledge of the object's and its own dynamic model, i.e., $M_{\mathcal{L}}$, $I_{\mathcal{L}}$, $P_{\mathcal{L},\text{com}}$ and $M_{\mathcal{R}_k}$, $C_{\mathcal{R}_k}$, $D_{\mathcal{R}_k}$, $G_{\mathcal{R}_k}$ are known.

Assumption 2. Each robot \mathcal{R}_k , $k \in \mathfrak{I}_{N_{\mathcal{R}}}$ has full knowledge of its own state z_k and the current configuration $q_{\mathcal{L}}$ of the object \mathcal{L} .

Assumption 3. The plan generated by the high-level planner is available to all robots. Furthermore, each robot \mathcal{R}_k , $k \in \mathfrak{I}_{N_{\mathcal{R}}}$ is able to communicate with the others only for announcing that it is ready to transition to the next cell, i.e., that $p_{\mathcal{L}} \in \mathcal{G}_{p_{\mathcal{L}}}(\underline{\mathcal{C}}_{\hat{\mathcal{S}}_i}, \underline{\mathcal{C}}_{\hat{\mathcal{S}}_j})$, $\theta_{\mathcal{L}} \in \mathcal{G}_{\theta_{\mathcal{L}}}(\underline{\mathcal{C}}_{\hat{\mathcal{S}}_i}, \underline{\mathcal{C}}_{\hat{\mathcal{S}}_j})$ and $q_k \in \mathcal{G}_{\mathcal{A}_k}(\underline{\mathcal{C}}_{\hat{\mathcal{S}}_i}, \underline{\mathcal{C}}_{\hat{\mathcal{S}}_j})$.

Assumption 4. Each mobile manipulator \mathcal{R}_k , $k \in \mathfrak{I}_{N_{\mathcal{R}}}$ is sufficiently redundant, i.e. it can independently apply a desired wrench to its end-effector while keeping q_k in $\mathcal{P}_{\mathcal{A}_k}(\underline{\mathcal{C}}_{\hat{\mathcal{S}}_i})$. Additionally, the lower diagonal $N_{\mathcal{A}_k} - 1 \times N_{\mathcal{A}_k} - 1$ block of $(M_{\mathcal{R}_k})^{-1} (I - (\mathcal{J}_{\mathcal{R}_k}(z_k))^{\dagger} \cdot \mathcal{J}_{\mathcal{R}_k}(z_k))$ is non-singular.

Object's Position

First, we shall design a suitable vector field for safely driving the object's position $p_{\mathcal{L}}$ to $\mathcal{G}_{p_{\mathcal{L}}}(\underline{\mathcal{C}}_{\hat{\mathcal{S}}_i}, \underline{\mathcal{C}}_{\hat{\mathcal{S}}_j})$. To do so, we construct a transformation T_i of $\mathcal{F}_{p_{\mathcal{L}}}(\underline{\mathcal{C}}_{\hat{\mathcal{S}}_i}, \underline{\mathcal{C}}_{\hat{\mathcal{S}}_j}) \triangleq \mathcal{P}_{p_{\mathcal{L}}}(\underline{\mathcal{C}}_{\hat{\mathcal{S}}_i}) \setminus \mathcal{G}_{p_{\mathcal{L}}}(\underline{\mathcal{C}}_{\hat{\mathcal{S}}_i}, \underline{\mathcal{C}}_{\hat{\mathcal{S}}_j})$ to the unit disk and collapse the selected component of $\mathcal{G}_{p_{\mathcal{L}}}(\underline{\mathcal{C}}_{\hat{\mathcal{S}}_i}, \underline{\mathcal{C}}_{\hat{\mathcal{S}}_j})$ to a point, using the procedure described in [Subsection 3.3.3](#). By recalling that T_i is a diffeomorphism that collapses all inner obstacles of to isolated points, one can readily verify that the chance of a line connecting the image $q_{\mathcal{L}}^{[i]} \triangleq T_i(p_{\mathcal{L}})$ of the object's current position to the image $q_{\mathcal{L},d}^{[i]}$ of the current cell's goal is zero [[67](#), [65](#)]. Therefore, the following velocity control law would safely drive the object's position to the goal set for almost all initial configurations:

$$v_{p_{\mathcal{L}}}^{[i]}(p_{\mathcal{L}}) \triangleq (J_{T_i}(p_{\mathcal{L}}))^{-1} \cdot (q_{\mathcal{L},d}^{[i]} - q_{\mathcal{L}}^{[i]}) \quad (5.13)$$

where J_{T_i} is the Jacobian matrix of T_i . In order to design a law for the desired force to be applied to the object \mathcal{L} by the robots, we employ a novel methodology presented in [[2](#)] which allows us to extend the vector field [Equation 5.13](#) to second-order dynamics. The corresponding control law for the desired force applied to the object is given by

$$\begin{aligned} \tau_{\mathcal{L},p}^{des} &= -M_{\mathcal{L}} \cdot (K_{p_{\mathcal{L}}}^{[i]} \cdot (p_{\mathcal{L}} - p_{\mathcal{L},G}^{[i]}) + \zeta_{p_{\mathcal{L}}}^{[i]} \cdot \dot{p}_{\mathcal{L}}) \\ \dot{p}_{\mathcal{L},G}^{[i]} &= K_{p_{\mathcal{L},G}^{[i]}}^{[i]} \cdot \frac{v_{p_{\mathcal{L}}}^{[i]}(p_{\mathcal{L},G}^{[i]})}{\|v_{p_{\mathcal{L}}}^{[i]}(p_{\mathcal{L},G}^{[i]})\|} \cdot \min \left(\|v_{p_{\mathcal{L}}}^{[i]}(p_{\mathcal{L},G}^{[i]})\|, \sqrt{\Delta E_{p_{\mathcal{L}}}^{[i]}(p_{\mathcal{L}}, p_{\mathcal{L},G}^{[i]}) / K_{p_{\mathcal{L}}}^{[i]}} \right) \end{aligned} \quad (5.14)$$

where

$$\Delta E_{p_{\mathcal{L}}}^{[i]}(p_{\mathcal{L}}, p_{\mathcal{L},G}^{[i]}) \triangleq K_{p_{\mathcal{L}}}^{[i]} \cdot d \left(p_{\mathcal{L},G}^{[i]}, \partial \mathcal{F}_{p_{\mathcal{L}}}(\underline{\mathcal{C}}_{\hat{\mathcal{S}}_i}, \underline{\mathcal{C}}_{\hat{\mathcal{S}}_j}) \right) - E_{p_{\mathcal{L}}}^{[i]}(p_{\mathcal{L}}, p_{\mathcal{L},G}^{[i]}) \quad (5.15)$$

$$E_{p_{\mathcal{L}}}^{[i]}(p_{\mathcal{L}}, p_{\mathcal{L},G}^{[i]}) \triangleq \frac{1}{2} \cdot \left(\|\dot{p}_{\mathcal{L}}\|^2 + K_{p_{\mathcal{L}}}^{[i]} \cdot \|p_{\mathcal{L}} - p_{\mathcal{L},G}^{[i]}\|^2 \right) \quad (5.16)$$

$p_{\mathcal{L},G}^{[i]}$ is the (virtual) state of the reference governor, $d(x, \mathcal{X})$ is the distance of x from the set \mathcal{X} , $K_{p_{\mathcal{L}}}^{[i]}$, $K_{p_{\mathcal{L},G}}^{[i]}$ are fixed, positive gains and $\zeta_{p_{\mathcal{L}}}^{[i]}$ is a virtual damping.

Object's Orientation

To drive the orientation $\theta_{\mathcal{L}}$ of the object to the specified goal set $\mathcal{G}_{\theta_{\mathcal{L}}}(\underline{\mathcal{C}}_{\hat{\mathcal{S}}_i}, \underline{\mathcal{C}}_{\hat{\mathcal{S}}_j})$ while ensuring that it remains within $\mathcal{S}_i^{q_0} = [\theta_l^{[i]}, \theta_l^{[i]}]$, we design the desired torque $\tau_{\mathcal{L},\theta}$ applied to the object based on the Prescribe Performance Control (PPC) methodology. We assume that $\mathcal{G}_{\theta_{\mathcal{L}}}(\underline{\mathcal{C}}_{\hat{\mathcal{S}}_i}, \underline{\mathcal{C}}_{\hat{\mathcal{S}}_j})$ is of the form $[\theta_{l,G}^{[i]}, \theta_{u,G}^{[i]}]$, which can be ensured by designing the partitioning scheme of the configuration space planner described in [Subsection 5.4.1](#) such that the compound slices which form a valid cover are overlapping. We now define the following two performance functions

$$\begin{aligned} \underline{\rho}_{\theta_{\mathcal{L}}}^{[i]}(t) &\triangleq \theta_{l,G}^{[i]} + (\theta_l^{[i]} - \theta_{l,G}^{[i]}) \cdot e^{-\lambda_{\theta_{\mathcal{L}}} \cdot t} \\ \bar{\rho}_{\theta_{\mathcal{L}}}^{[i]}(t) &\triangleq \theta_{u,G}^{[i]} + (\theta_l^{[i]} - \theta_{u,G}^{[i]}) \cdot e^{-\lambda_{\theta_{\mathcal{L}}} \cdot t} \end{aligned} \quad (5.17)$$

where t is time and $\lambda_{\theta_{\mathcal{L}}}$ is a positive constant. The corresponding control law is given by

$$\tau_{\mathcal{L},\theta}^{des} = I_{\mathcal{L}} \cdot \left(-K_{\theta_{\mathcal{L},2}}^{[i]} \cdot (\dot{\theta}_{\mathcal{L}} - v_{\theta_{\mathcal{L}}}^{[i]}) + \dot{v}_{\theta_{\mathcal{L}}}^{[i]} - a_{\theta_{\mathcal{L}}}^{[i]} \cdot \ln \left(\frac{\theta_{\mathcal{L}} - \underline{\rho}_{\theta_{\mathcal{L}}}^{[i]}}{\bar{\rho}_{\theta_{\mathcal{L}}}^{[i]} - \theta_{\mathcal{L}}} \right) \right) \quad (5.18)$$

$$v_{\theta_{\mathcal{L}}}^{[i]} \triangleq \frac{b_{\theta_{\mathcal{L}}}^{[i]} - K_{\theta_{\mathcal{L},1}}^{[i]} \cdot \ln \left(\frac{\theta_{\mathcal{L}} - \underline{\rho}_{\theta_{\mathcal{L}}}^{[i]}}{\bar{\rho}_{\theta_{\mathcal{L}}}^{[i]} - \theta_{\mathcal{L}}} \right)}{a_{\theta_{\mathcal{L}}}^{[i]}} \quad (5.19)$$

$$\begin{aligned} a_{\theta_{\mathcal{L}}}^{[i]} &\triangleq \frac{1}{\theta_{\mathcal{L}} - \underline{\rho}_{\theta_{\mathcal{L}}}^{[i]}} + \frac{1}{\bar{\rho}_{\theta_{\mathcal{L}}}^{[i]} - \theta_{\mathcal{L}}} \\ b_{\theta_{\mathcal{L}}}^{[i]} &\triangleq \frac{\dot{\underline{\rho}}_{\theta_{\mathcal{L}}}^{[i]}}{\theta_{\mathcal{L}} - \underline{\rho}_{\theta_{\mathcal{L}}}^{[i]}} + \frac{\dot{\bar{\rho}}_{\theta_{\mathcal{L}}}^{[i]}}{\bar{\rho}_{\theta_{\mathcal{L}}}^{[i]} - \theta_{\mathcal{L}}} \end{aligned} \quad (5.20)$$

with $K_{\theta_{\mathcal{L},1}}^{[i]}$ and $K_{\theta_{\mathcal{L},2}}^{[i]}$ being positive gains.

Manipulators

Considering now the control scheme for the mobile manipulators, we remark that, by virtue of [Assumption 1](#) and [Assumption 2](#) and assuming a common initialization policy for the virtual states of the reference governors corresponding to the object's position and orientation, respectively, each robot is able to compute the desired total force $\tau_{\mathcal{L},p}$ and torque $\tau_{\mathcal{L},\theta}$ that should be applied to the object. Thus, the wrench $\tau_{e,k}$ that each robot \mathcal{R}_k , $k \in \mathcal{I}_{N_{\mathcal{R}}}$ should apply to the object via its end-effector is given

by

$$\tau_{e,k} = \frac{1}{N_{\mathcal{R}}} \cdot \begin{bmatrix} \tau_{\mathcal{L},p} \\ \tau_{\mathcal{L},\theta} - (\tau_{\mathcal{L},p})^T \cdot {}_{\{E_k\}}P_{\mathcal{L},com}^\perp \end{bmatrix} \quad (5.21)$$

where ${}_{\{E_k\}}P_{\mathcal{L},com}^\perp = R(\frac{\pi}{2}) \cdot R(\theta_{\mathcal{L}}) \cdot {}_{\{E_k\}}P_{\mathcal{L},com}$ with ${}_{\{E_k\}}P_{\mathcal{L},com}$ being the position of the object's center of mass relative to the contact point of manipulator \mathcal{A}_k . Furthermore, each robot must also ensure that $q_k \in \mathcal{P}_{\mathcal{A}_k}(\underline{\mathcal{C}}_{\hat{\mathcal{S}}_i})$ while driving q_k to $\mathcal{G}_{\mathcal{A}_k}(\underline{\mathcal{C}}_{\hat{\mathcal{S}}_i}, \underline{\mathcal{C}}_{\hat{\mathcal{S}}_j})$. To do so, we shall exploit the redundancy of each robot to design a force in the null-space of $\mathcal{J}_{\mathcal{R}_k}$ which can ensure that the aforementioned specifications are met without affecting the force applied to the object. We now recall the dynamics of mobile manipulator \mathcal{R}_k :

$$M_{\mathcal{R}_k}(z_k) \cdot \ddot{z}_k + C_{\mathcal{R}_k}(z_k, \dot{z}_k) \cdot \dot{z}_k + G_{\mathcal{R}_k}(z_k) = \tau_{m,k} - (\mathcal{J}_{\mathcal{R}_k}(z_k))^T \cdot \tau_{e,k} \quad (5.22)$$

Assuming known dynamic parameters and state, we can design

$$\tau_{m,k} = C_{\mathcal{R}_k}(z_k, \dot{z}_k) \cdot \dot{z}_k + G_{\mathcal{R}_k}(z_k) + \tau_{m,k,1} + \left(I - (\mathcal{J}_{\mathcal{R}_k}(z_k))^\dagger \cdot \mathcal{J}_{\mathcal{R}_k}(z_k) \right) \cdot \tau_{m,k,2} \quad (5.23)$$

where $\tau_{m,k,1}$ and $\tau_{m,k,2}$ are new virtual inputs to be defined later and $(\mathcal{J}_{\mathcal{R}_k})^\dagger$ denotes the pseudo-inverse of $\mathcal{J}_{\mathcal{R}_k}$. Substituting the above in [Equation 5.22](#) yields:

$$M_{\mathcal{R}_k}(z_k) \cdot \ddot{z}_k = \tau_{m,k,1} + \left(I - (\mathcal{J}_{\mathcal{R}_k}(z_k))^\dagger \cdot \mathcal{J}_{\mathcal{R}_k}(z_k) \right) \cdot \tau_{m,k,2} - (\mathcal{J}_{\mathcal{R}_k}(z_k))^T \cdot \tau_{e,k} \quad (5.24)$$

We now consider the above dynamical model in the robot's task-space:

$$M'_{\mathcal{R}_k} \cdot \begin{bmatrix} \dot{p}_{E_k} \\ \dot{\theta}_{E_k} \end{bmatrix} + C'_{\mathcal{R}_k} \cdot \begin{bmatrix} \dot{p}_{E_k} \\ \dot{\theta}_{E_k} \end{bmatrix} = (\mathcal{J}_{\mathcal{R}_k}^T)^\dagger \cdot \tau_{m,k,1} - \tau_{e,k} \quad (5.25)$$

where \dot{p}_{E_k} and $\dot{\theta}_{E_k}$ are the position and orientation of the corresponding end-effector's contact point and

$$\begin{aligned} M'_{\mathcal{R}_k} &= (\mathcal{J}_{\mathcal{R}_k}^T)^\dagger \cdot M_{\mathcal{R}_k} \cdot (\mathcal{J}_{\mathcal{R}_k})^\dagger \\ C'_{\mathcal{R}_k} &= -(\mathcal{J}_{\mathcal{R}_k}^T)^\dagger \cdot M_{\mathcal{R}_k} \cdot (\mathcal{J}_{\mathcal{R}_k})^\dagger \cdot \dot{\mathcal{J}}_{\mathcal{R}_k} \cdot (\mathcal{J}_{\mathcal{R}_k})^\dagger. \end{aligned} \quad (5.26)$$

Let $\begin{bmatrix} p_{E_k}^T & \theta_{E_k} \end{bmatrix}^T = \mathcal{T}_{\mathcal{L},E_k}(p_{\mathcal{L}}, \theta_{\mathcal{L}})$ be the rigid transformation between the positions and orientations of the corresponding points. It holds that

$$\begin{bmatrix} \dot{p}_{E_k} \\ \dot{\theta}_{E_k} \end{bmatrix} = J_{\mathcal{L},E_k}(p_{\mathcal{L}}, \theta_{\mathcal{L}}) \cdot \begin{bmatrix} \dot{p}_{\mathcal{L}} \\ \dot{\theta}_{\mathcal{L}} \end{bmatrix} \quad (5.27)$$

$$\begin{bmatrix} \ddot{p}_{E_k} \\ \ddot{\theta}_{E_k} \end{bmatrix} = J_{\mathcal{L},E_k}(p_{\mathcal{L}}, \theta_{\mathcal{L}}) \cdot \begin{bmatrix} \ddot{p}_{\mathcal{L}} \\ \ddot{\theta}_{\mathcal{L}} \end{bmatrix} + \dot{J}_{\mathcal{L},E_k}(p_{\mathcal{L}}, \theta_{\mathcal{L}}) \cdot \begin{bmatrix} \dot{p}_{\mathcal{L}} \\ \dot{\theta}_{\mathcal{L}} \end{bmatrix} \quad (5.28)$$

with $J_{\mathcal{L},E_k}$ denoting the Jacobian matrix of this rigid transformation. Therefore, [Equation 5.25](#) can be re-written w.r.t. to the object's state as follows

$$M''_{\mathcal{R}_k} \cdot \begin{bmatrix} \ddot{p}_{\mathcal{L}} \\ \ddot{\theta}_{\mathcal{L}} \end{bmatrix} + C''_{\mathcal{R}_k} \cdot \begin{bmatrix} \dot{p}_{\mathcal{L}} \\ \dot{\theta}_{\mathcal{L}} \end{bmatrix} = (\mathcal{J}_{\mathcal{R}_k}^T)^\dagger \cdot \tau_{m,k,1} - \tau_{e,k} \quad (5.29)$$

with

$$\begin{aligned} M''_{\mathcal{R}_k} &= M'_{\mathcal{R}_k} \cdot J_{\mathcal{L},E_k} \\ C''_{\mathcal{R}_k} &= C'_{\mathcal{R}_k} \cdot J_{\mathcal{L},E_k} + M'_{\mathcal{R}_k} \cdot \dot{J}_{\mathcal{L},E_k}. \end{aligned} \quad (5.30)$$

We notice that achieving our indented behavior, i.e., the object obeying the dynamics imposed by [Equation 5.14](#) and [Equation 5.18](#) while distributing the load equally between the robots, is equivalent to

$$\tau_{e,k} = \frac{1}{N_{\mathcal{R}}} \left[I_{\mathcal{L}} \cdot \ddot{\theta}_{\mathcal{L}} - M_{\mathcal{L}} \cdot \ddot{p}_{\mathcal{L}} \right] \cdot \{E_k\} P_{\mathcal{L},com}^\perp = \frac{1}{N_{\mathcal{R}}} M_{\mathcal{L},E_k} \cdot \begin{bmatrix} \ddot{p}_{\mathcal{L}} \\ \ddot{\theta}_{\mathcal{L}} \end{bmatrix} \quad (5.31)$$

where $M_{\mathcal{L},E_k}$ is the fragment of the object's inertia, as perceived by the manipulator \mathcal{R}_k . Substituting the above into [Equation 5.25](#) yields

$$\left(M''_{\mathcal{R}_k} + \frac{1}{N_{\mathcal{R}}} \cdot M_{\mathcal{L},E_k} \right) \cdot \begin{bmatrix} \ddot{p}_{\mathcal{L}} \\ \ddot{\theta}_{\mathcal{L}} \end{bmatrix} + C''_{\mathcal{R}_k} \cdot \begin{bmatrix} \dot{p}_{\mathcal{L}} \\ \dot{\theta}_{\mathcal{L}} \end{bmatrix} = (\mathcal{J}_{\mathcal{R}_k}^T)^\dagger \cdot \tau_{m,k,1}. \quad (5.32)$$

As such, we can see that selecting

$$\tau_{m,k,1} \triangleq \mathcal{J}_{\mathcal{R}_k} \cdot \left(\left(M''_{\mathcal{R}_k} + \frac{1}{N_{\mathcal{R}}} \cdot M_{\mathcal{L},E_k} \right) \cdot \begin{bmatrix} \tau_{\mathcal{L},p}^{des}/M_{\mathcal{L}} \\ \tau_{\mathcal{L},\theta}^{des}/I_{\mathcal{L}} \end{bmatrix} + C''_{\mathcal{R}_k} \cdot \begin{bmatrix} \dot{p}_{\mathcal{L}} \\ \dot{\theta}_{\mathcal{L}} \end{bmatrix} \right) \quad (5.33)$$

will achieve the desired behavior, assuming all $N_{\mathcal{R}}$ robots execute the same control law.

Considering again the [Equation 5.24](#), we shall now design $\tau_{m,k,2}$ appropriately in order to satisfy the manipulator joint limit specifications. We recall that the projection of $\tau_{m,k,2}$ w.r.t. $I - (\mathcal{J}_{\mathcal{R}_k}(z_k))^\dagger \cdot \mathcal{J}_{\mathcal{R}_k}(z_k)$ has no effect on the wrench applied to the attached object. Now, let $L_{A,k}, L_{B,k}, L_{C,k}, L_{D,k}$ be matrices such that $L_{A,k} \in \mathbb{R}^{3 \times 3}$ and

$$L_k \triangleq M_{\mathcal{R}_k}^{-1} \cdot \left(I - (\mathcal{J}_{\mathcal{R}_k}(z_k))^\dagger \cdot \mathcal{J}_{\mathcal{R}_k}(z_k) \right) = \begin{bmatrix} L_{A,k} & L_{B,k} \\ L_{C,k} & L_{D,k} \end{bmatrix}. \quad (5.34)$$

By recalling that $L_{D,k}$ is assumed to be invertible according to [Assumption 4](#), we employ the Prescribed Performance Control method along with back-stepping to design $\tau_{m,k,2}$ as follows:

$$\tau_{m,k,2} \triangleq \begin{bmatrix} 0 \\ 0 \\ 0 \\ (L_{D,k})^{-1} \cdot v_{k,B} \end{bmatrix} \quad (5.35)$$

where

$$v_{k,B} \triangleq \begin{bmatrix} -(\dot{q}_{k,1} - v_{k,b,1}) + \dot{v}_{k,b,1} - a_{q_{k,1}} \cdot \ln \left(\frac{q_{k,1} - \underline{\rho}_{q_{k,1}}}{\bar{\rho}_{q_{k,1}} - q_{k,1}} \right) \\ -(\dot{q}_{k,1} - v_{k,b,1}) + \dot{v}_{k,b,2} - a_{q_{k,2}} \cdot \ln \left(\frac{q_{k,2} - \underline{\rho}_{q_{k,2}}}{\bar{\rho}_{q_{k,2}} - q_{k,2}} \right) \\ \vdots \\ -(\dot{q}_{k,N_{\mathcal{A}_k}-1} - v_{k,b,N_{\mathcal{A}_k}-1}) + \dot{v}_{k,b,N_{\mathcal{A}_k}-1} - a_{q_{k,N_{\mathcal{A}_k}-1}} \cdot \ln \left(\frac{q_{k,N_{\mathcal{A}_k}-1} - \underline{\rho}_{q_{k,N_{\mathcal{A}_k}-1}}}{\bar{\rho}_{q_{k,N_{\mathcal{A}_k}-1}} - q_{k,N_{\mathcal{A}_k}-1}} \right) \end{bmatrix} \quad (5.36)$$

$$v_{k,b,\ell} \triangleq \frac{b_{q_{k,\ell}} - \ln \left(\frac{q_{k,\ell} - \underline{\rho}_{q_{k,\ell}}}{\bar{\rho}_{q_{k,\ell}} - q_{k,\ell}} \right)}{a_{q_{k,\ell}}} \quad (5.37)$$

$$a_{q_{k,l}} \triangleq \frac{1}{q_{k,l} - \underline{\rho}_{q_{k,l}}} + \frac{1}{\bar{\rho}_{q_{k,l}} - q_{k,l}} \quad (5.38)$$

$$b_{q_{k,l}} \triangleq \frac{\dot{\underline{\rho}}_{q_{k,l}}}{q_{k,l} - \underline{\rho}_{q_{k,l}}} + \frac{\dot{\bar{\rho}}_{q_{k,l}}}{\bar{\rho}_{q_{k,l}} - q_{k,l}}$$

is the reference velocity control law, $\underline{\rho}_{q_{k,\ell}}$ and $\bar{\rho}_{q_{k,\ell}}$ are performance functions which smoothly “shrink” $\mathcal{P}_{\mathcal{A}_k}(\underline{\mathcal{C}}_{\hat{\mathcal{S}}_i})$ to $\mathcal{G}_{\mathcal{A}_k}(\underline{\mathcal{C}}_{\hat{\mathcal{S}}_i}, \underline{\mathcal{C}}_{\hat{\mathcal{S}}_j})$, given by

$$\begin{aligned} \underline{\rho}_{q_{k,\ell}}(t) &\triangleq \bar{q}_{G,k,\ell} + (\bar{q}_{k,\ell} - \bar{q}_{G,k,\ell}) \cdot e^{-\lambda_q \cdot t} \\ \bar{\rho}_{q_{k,\ell}}(t) &\triangleq \underline{q}_{G,k,\ell} + (\underline{q}_{k,\ell} - \underline{q}_{G,k,\ell}) \cdot e^{-\lambda_q \cdot t} \end{aligned} \quad (5.39)$$

with $\bar{q}_{k,\ell}$, $\underline{q}_{k,\ell}$ and $\bar{q}_{G,k,\ell}$, $\underline{q}_{G,k,\ell}$ being the lower and upper bounds of the joint parameters of \mathcal{A}_k corresponding to $\mathcal{P}_{\mathcal{A}_k}(\underline{\mathcal{C}}_{\hat{\mathcal{S}}_i})$ and $\mathcal{G}_{\mathcal{A}_k}(\underline{\mathcal{C}}_{\hat{\mathcal{S}}_i}, \underline{\mathcal{C}}_{\hat{\mathcal{S}}_j})$ respectively.

5.5 Stability Analysis

In this section, we provide an analysis of the robotic system’s stability properties under the proposed control scheme.

Proposition 9. *Given two adjacent under-approximation cells $\underline{\mathcal{C}}_{\hat{\mathcal{S}}_i}$ and $\underline{\mathcal{C}}_{\hat{\mathcal{S}}_j}$, the object’s configuration will asymptotically converge to $\mathcal{G}_{p_{\mathcal{L}}}(\underline{\mathcal{C}}_{\hat{\mathcal{S}}_i}, \underline{\mathcal{C}}_{\hat{\mathcal{S}}_j}) \times \mathcal{G}_{\theta_{\mathcal{L}}}(\underline{\mathcal{C}}_{\hat{\mathcal{S}}_i}, \underline{\mathcal{C}}_{\hat{\mathcal{S}}_j})$ for almost all initial configurations under control laws [Equation 5.14](#) and [Equation 5.18](#). Furthermore, the set $\mathcal{F}_{p_{\mathcal{L}}}(\underline{\mathcal{C}}_{\hat{\mathcal{S}}_i}, \underline{\mathcal{C}}_{\hat{\mathcal{S}}_j}) \times \mathcal{S}_i^{q_0}$ is invariant.*

Proof. We begin this proof by first recalling that, as long as object’s orientation and robot joints remain within the bounds imposed by $\underline{\mathcal{C}}_{\hat{\mathcal{S}}_i}$, control of the object’s position and orientation can be safely decoupled. Regarding the object’s position, one can readily verify that since T_i is a diffeomorphism in $\mathcal{F}_{p_{\mathcal{L}}}(\underline{\mathcal{C}}_{\hat{\mathcal{S}}_i}, \underline{\mathcal{C}}_{\hat{\mathcal{S}}_j})$ (see [chapter 2](#)), the reference velocity control law i is Lipschitz, has exactly one critical point which is located at the transformed goal configuration and is inward pointing at the outer

boundary of $\mathcal{F}_{p_{\mathcal{L}}}(\underline{\mathcal{C}}_{\hat{\mathcal{S}}_i}, \underline{\mathcal{C}}_{\hat{\mathcal{S}}_j})$. Then, by invocation of Theorem 2 in [2], the control law [Equation 5.14](#) ensures invariance of cell and convergence to the goal set of $p_{\mathcal{L}}$ for almost all initial configurations.

Regarding the object's orientation, we define the following coordinate transformation:

$$\begin{aligned} z_{1,i} &= \ln \left(\frac{\theta_{\mathcal{L}} - \rho_{\theta_{\mathcal{L}}}^{[i]}}{\bar{\rho}_{\theta_{\mathcal{L}}}^{[i]} - \theta_{\mathcal{L}}} \right) \\ z_{2,i} &= \dot{\theta}_{\mathcal{L}} - v_{\theta_{\mathcal{L}}}^{[i]} \end{aligned} \quad (5.40)$$

and consider the following Lyapunov candidate:

$$V = \frac{1}{2} \cdot z_{1,i}^2 + \frac{1}{2} \cdot z_{2,i}^2. \quad (5.41)$$

The time derivatives of $z_{1,i}$ and $z_{2,i}$ are given by

$$\begin{aligned} \dot{z}_{1,i} &= a_{\theta_{\mathcal{L}}}^{[i]} \cdot z_{2,i} + a_{\theta_{\mathcal{L}}}^{[i]} \cdot v_{\theta_{\mathcal{L}}}^{[i]} - b_{\theta_{\mathcal{L}}}^{[i]} \\ \dot{z}_{2,i} &= \frac{\tau_{\mathcal{L},\theta}}{I_{\mathcal{L}}} - \dot{v}_{\theta_{\mathcal{L}}}^{[i]} \end{aligned} \quad (5.42)$$

Thus, computing the derivative of V w.r.t. time yields

$$\begin{aligned} \dot{V} &= z_{1,i} \cdot \dot{z}_{1,i} + z_{2,i} \cdot \dot{z}_{2,i} \\ &= a_{\theta_{\mathcal{L}}}^{[i]} \cdot z_{1,i} \cdot z_{2,i} + z_{1,i} \cdot \left(a_{\theta_{\mathcal{L}}}^{[i]} \cdot v_{\theta_{\mathcal{L}}}^{[i]} - b_{\theta_{\mathcal{L}}}^{[i]} \right) + z_{2,i} \cdot \left(\frac{\tau_{\mathcal{L},\theta}}{I_{\mathcal{L}}} - \dot{v}_{\theta_{\mathcal{L}}}^{[i]} \right) \end{aligned} \quad (5.43)$$

Noting that $a_{\theta_{\mathcal{L}}}^{[i]} \cdot v_{\theta_{\mathcal{L}}}^{[i]} - b_{\theta_{\mathcal{L}}}^{[i]} = -K_{\theta_{\mathcal{L},1}}^{[i]} z_{1,i}^2$ and substituting the control law for $\tau_{\mathcal{L},\theta}$ to the above, we obtain

$$\dot{V} = - \left(K_{\theta_{\mathcal{L},1}}^{[i]} \cdot z_{1,i}^2 + K_{\theta_{\mathcal{L},2}}^{[i]} \cdot z_{2,i}^2 \right). \quad (5.44)$$

Since \dot{V} is negative definite, assuming that the initial value of $\theta_{\mathcal{L}}$ lies within the specified bounds, the proposed control law ensures that $\mathcal{S}_i^{\text{q0}}$ remains invariant and that $\theta_{\mathcal{L}}$ will asymptotically converge to $(\theta_{l,G}^{[i]} + \theta_{u,G}^{[i]}) / 2$. \square

Proposition 10. *Given two adjacent under-approximation cells $\underline{\mathcal{C}}_{\hat{\mathcal{S}}_i}$ and $\underline{\mathcal{C}}_{\hat{\mathcal{S}}_j}$, under the control law [Equation 5.23](#), the joint states q_k of mobile manipulator \mathcal{R}_k will converge to $\mathcal{G}_{\mathcal{A}_k}(\underline{\mathcal{C}}_{\hat{\mathcal{S}}_i}, \underline{\mathcal{C}}_{\hat{\mathcal{S}}_j})$. Furthermore, the set $\mathcal{P}_{\mathcal{A}_k}(\underline{\mathcal{C}}_{\hat{\mathcal{S}}_i})$ is invariant.*

Proof. We consider once again [Equation 5.24](#). Since $M_{\mathcal{R}_k}$ is an inertia matrix, we

know that its inverse exists, thus multiplying both sides with $(M_{\mathcal{R}_k})^{-1}$ and substituting [Equation 98](#) yields:

$$\begin{aligned} \ddot{z}_k &= (M_{\mathcal{R}_k}(z_k))^{-1} \cdot \left(I - (\mathcal{J}_{\mathcal{R}_k}(z_k))^\dagger \cdot \mathcal{J}_{\mathcal{R}_k}(z_k) \right) \cdot \tau_{m,k,2} + \\ &\quad (M_{\mathcal{R}_k}(z_k))^{-1} \cdot \tau_{m,k,1} - (M_{\mathcal{R}_k}(z_k))^{-1} \cdot (\mathcal{J}_{\mathcal{R}_k}(z_k))^T \cdot \tau_{e,k} \\ &= \begin{bmatrix} L_{B,k} \cdot (L_{D,k})^{-1} \\ I \end{bmatrix} \cdot v_{k,B} + (M_{\mathcal{R}_k}(z_k))^{-1} \cdot \tau_{m,k,1} - (M_{\mathcal{R}_k}(z_k))^{-1} \cdot (\mathcal{J}_{\mathcal{R}_k}(z_k))^T \cdot \tau_{e,k} \end{aligned} \quad (5.45)$$

We note that the term $\tau_{m,k,2}$, which is designed to ensure satisfaction of joint parameter specifications, has no effect on the stability properties involving the object's position $p_{\mathcal{L}}$ and orientation $\theta_{\mathcal{L}}$ by virtue of $I - (\mathcal{J}_{\mathcal{R}_k}(z_k))^\dagger \cdot \mathcal{J}_{\mathcal{R}_k}(z_k)$. Therefore, the last two r.h.s. terms of [Equation 5.45](#) are bounded by design and vanish as the object approaches the specified configuration corresponding to the current cell. As such, the dynamics of the joint parameters can be written as:

$$\ddot{q}_k = v_{k,B} + w_B \quad (5.46)$$

where the term w_B corresponding to $\tau_{m,k,1}$ and $\tau_{e,k}$ and can be viewed as a bounded and vanishing disturbance. We now define the following coordinate transformation for each joint value $q_{k,\ell}$, $\ell \in \mathfrak{I}_{N_{A_k}-1}$:

$$\begin{aligned} z_{q,1,\ell} &= \ln \left(\frac{q_{k,\ell} - \underline{\rho}_{q_{k,\ell}}}{\bar{\rho}_{q_{k,\ell}} - q_{k,\ell}} \right) \\ z_{q,2,\ell} &= \dot{q}_{k,\ell} - v_{k,b,\ell} \end{aligned} \quad (5.47)$$

and consider the Lyapunov candidate

$$V_\ell = \frac{1}{2} \cdot z_{q,1,\ell}^2 + \frac{1}{2} \cdot z_{q,2,\ell}^2. \quad (5.48)$$

Following the same procedure as above, we derive that

$$\begin{aligned} \dot{V}_\ell &= - \left(z_{q,1,\ell}^2 + z_{q,2,\ell}^2 \right) + z_{q,2,\ell} \cdot w_{B,\ell} \\ &\leq -z_{q,1,\ell}^2 - z_{q,2,\ell}^2 + |z_{q,2,\ell}| \cdot |w_{B,\ell}| \\ &\leq -z_{q,1,\ell}^2 - |z_{q,2,\ell}| \cdot |w_{B,\ell}| + (w_{B,\ell})^2 \end{aligned} \quad (5.49)$$

which implies that $z_{q,1,\ell}$, $z_{q,2,\ell}$ and the control law are globally uniformly bounded ([Lemma 2.28 \[51\]](#)), and, thus, concludes the proof. \square

Theorem 7. *The robotic system under the distributed control law [Equation 5.23](#) will successfully drive the object \mathcal{L} to its goal configuration $q_{\mathcal{L},\text{init}}$, from almost all initial configurations.*

Proof. First, we note that, by virtue of [Assumption 4](#) and the design of [Equation 5.23](#),

the total force and torque applied to the object's center of mass by the robotic system is equal to the desired ones specified by Equation 5.14 and Equation 5.18, since the remaining terms either cancel the robot's dynamics or are projected along the kernel space of $\mathcal{J}_{\mathcal{R}_k}$, respectively. As such, according to Proposition 9, the object is guaranteed to traverse from one cell to another until it arrives at the desired configuration $q_{\mathcal{L},\text{goal}}$, starting from almost any initial configuration $q_{\mathcal{L},\text{init}}$, as long as the robots do not collide with the workspace boundary. But, according to Proposition 10, the configurations of the mobile manipulators remain within the bounds specified by the under-approximation cell $\underline{\mathcal{C}}_{\hat{\mathcal{S}}}$, which, by design of the high-level planner, implies that the robotic system's footprint cannot intersect the workspace's boundary. \square

5.6 Simulation Results

To demonstrate the efficacy of the proposed control scheme, we consider a robotic system consisting of two mobile manipulators holding a rectangular object, as depicted in Figure 5.3 operating within the workspace seen in Figure 5.5. The robotic system was initialized at $q_{\mathcal{L},\text{init}} = [0.9, 2, 1.57]^T$ and $q_{1,1} = q_{2,1} = 0$ whereas the desired configuration of the object was set to $q_{\mathcal{L},\text{goal}} = [5, 8, 4.663]^T$. The intervals for the object's orientation and robot joints generated by the high-level planner can be seen in Table 5.1, whereas the control parameters selected during this simulation are given in Table 5.2. Figure 5.6 shows the trajectory executed by the robotic system under the proposed control law, whereas plots of the object's position, orientation and corresponding rates can be seen in Figure 5.7, Figure 5.8, Figure 5.9, Figure 5.10, respectively. Accordingly, Figure 5.11 and Figure 5.12 show the evolution of each manipulator's state, as well as the computed lower and upper bounds corresponding to each cell. The total force and torque applied to the object is also displayed in Figure 5.11 and Figure 5.12, respectively. As one can verify from the aforementioned figures, the robotic system successfully reaches the goal configuration while satisfying the specifications corresponding to $\theta_{\mathcal{L}}, q_{1,1}, q_{2,1}$.

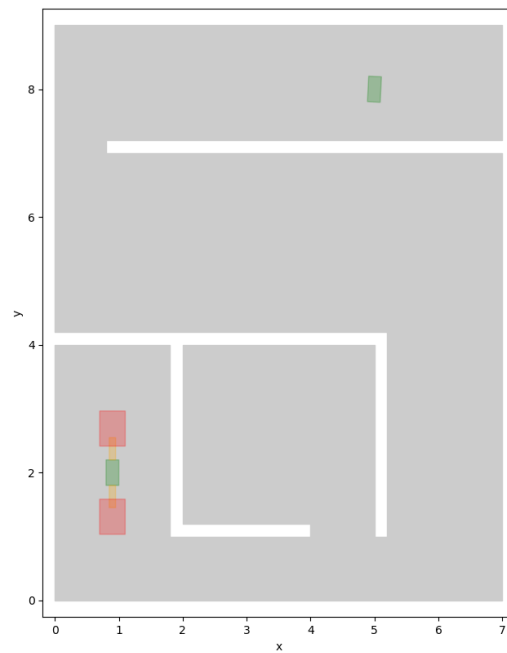


FIGURE 5.5: Initial and goal configuration of the robotic system and object, respectively.

Cell ID	θ_l	θ_u	$\bar{q}_{1,1}$	$\underline{q}_{1,1}$	$\bar{q}_{2,1}$	$\underline{q}_{2,1}$
1	1.470	1.867	-0.100	0.567	-0.100	0.567
2	1.667	2.065	-0.100	0.567	-0.567	0.100
3	1.863	2.456	-0.100	0.567	-1.035	0.367
4	1.863	2.456	-0.100	0.333	-1.870	-0.835
5	2.256	2.652	-0.100	0.333	-1.503	-0.835
6	2.452	2.849	-0.100	0.333	-1.035	-0.368
7	2.649	2.947	-0.100	0.217	-0.567	0.100
8	2.747	3.045	-0.050	0.108	-0.050	0.108
9	2.895	3.191	-0.050	0.108	-0.050	0.108
10	3.091	3.388	-0.050	0.158	-0.158	0.050
11	3.287	3.584	0.067	0.284	-0.284	-0.067
12	3.484	3.781	0.184	0.518	-0.518	-0.184
13	3.681	3.977	0.418	0.985	-0.985	-0.418
14	3.877	4.172	0.651	0.984	-0.984	-0.651
15	4.072	4.370	0.418	0.985	-0.985	-0.418
16	4.270	4.576	0.184	0.518	-0.518	-0.184
17	4.476	4.762	0.067	0.284	-0.284	-0.067
18	4.564	4.762	0.067	0.284	-0.284	-0.067
19	4.564	4.762	-0.050	0.108	-0.108	0.050
20	4.564	4.762	-0.050	0.108	-0.108	0.050
21	4.564	4.762	-0.985	0.050	-0.108	0.050
22	4.564	4.762	-1.453	-0.885	-0.108	0.050
23	4.564	4.762	-1.687	-1.353	-0.108	0.050
24	4.564	4.762	-1.687	-1.528	-0.108	-0.050
25	4.564	4.762	-1.687	-1.528	-0.168	-0.008
26	4.564	4.762	-1.687	-1.528	-0.284	-0.067
27	4.564	4.762	-1.687	-1.528	-0.518	-0.184
28	4.564	4.762	-1.687	-1.528	-0.985	-0.418
29	4.564	4.762	-1.687	-1.528	-1.453	-0.885
30	4.564	4.762	-1.687	-1.528	-1.687	-1.353

TABLE 5.1: Lower and upper bounds of the intervals corresponding to each cell, as generated by the planner.

Parameter	Value
$M_{\mathcal{L}}$	1 kg
$I_{\mathcal{L}}$	1 kg m ²
$K_{p_{\mathcal{L}}}$	50
$\zeta_{p_{\mathcal{L}}}$	5
$K_{p_{\mathcal{L}},G}$	5
$\lambda_{\theta_{\mathcal{L}}}$	2
$K_{\theta_{\mathcal{L}},1}$	1
$K_{\theta_{\mathcal{L}},2}$	1
λ_q	2

TABLE 5.2: Simulation parameters.

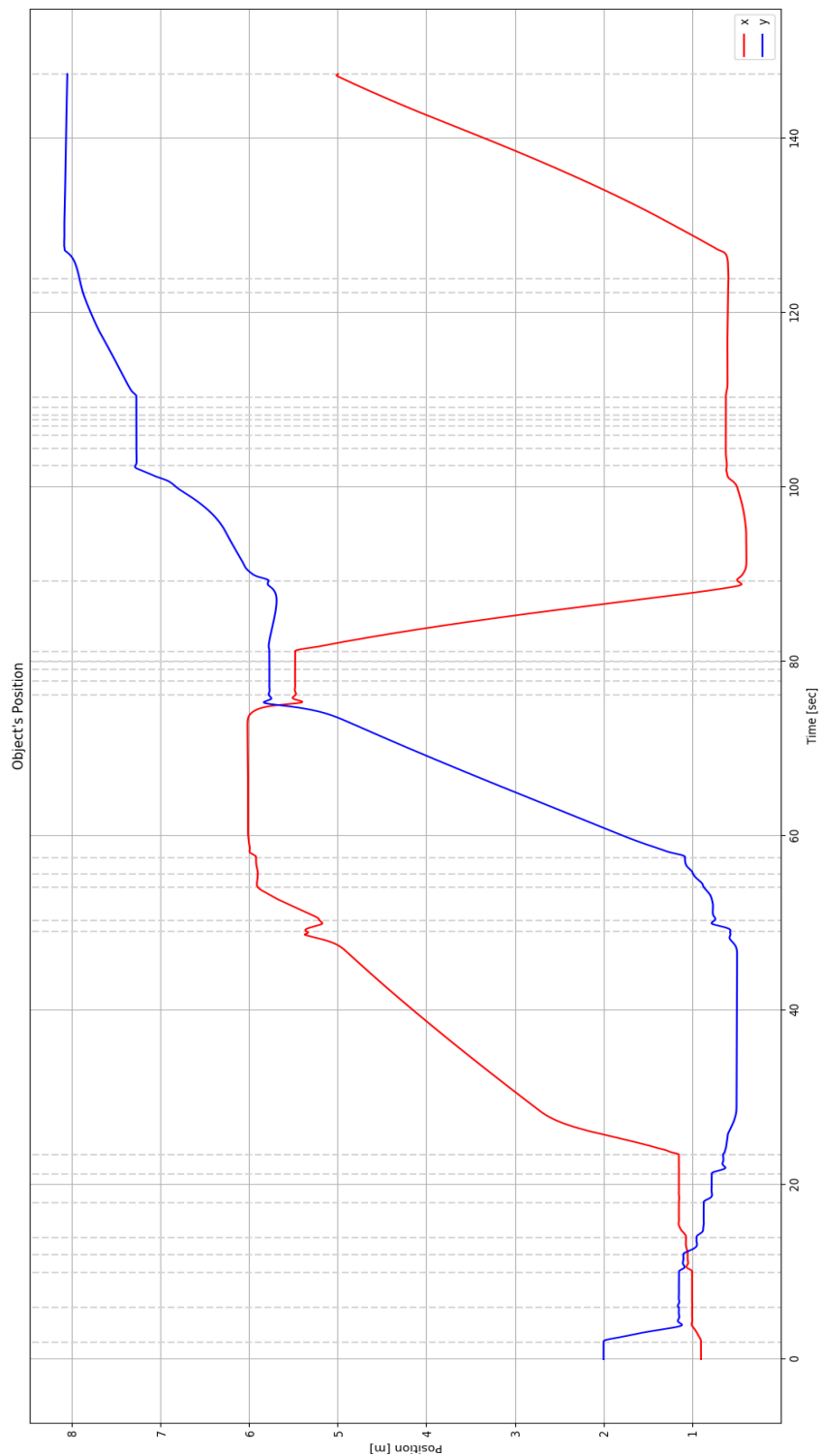


FIGURE 5.7: Evolution of the object's position $p_{\mathcal{L}}$ over time. The vertical dashed lines indicate transitions between consecutive cells.

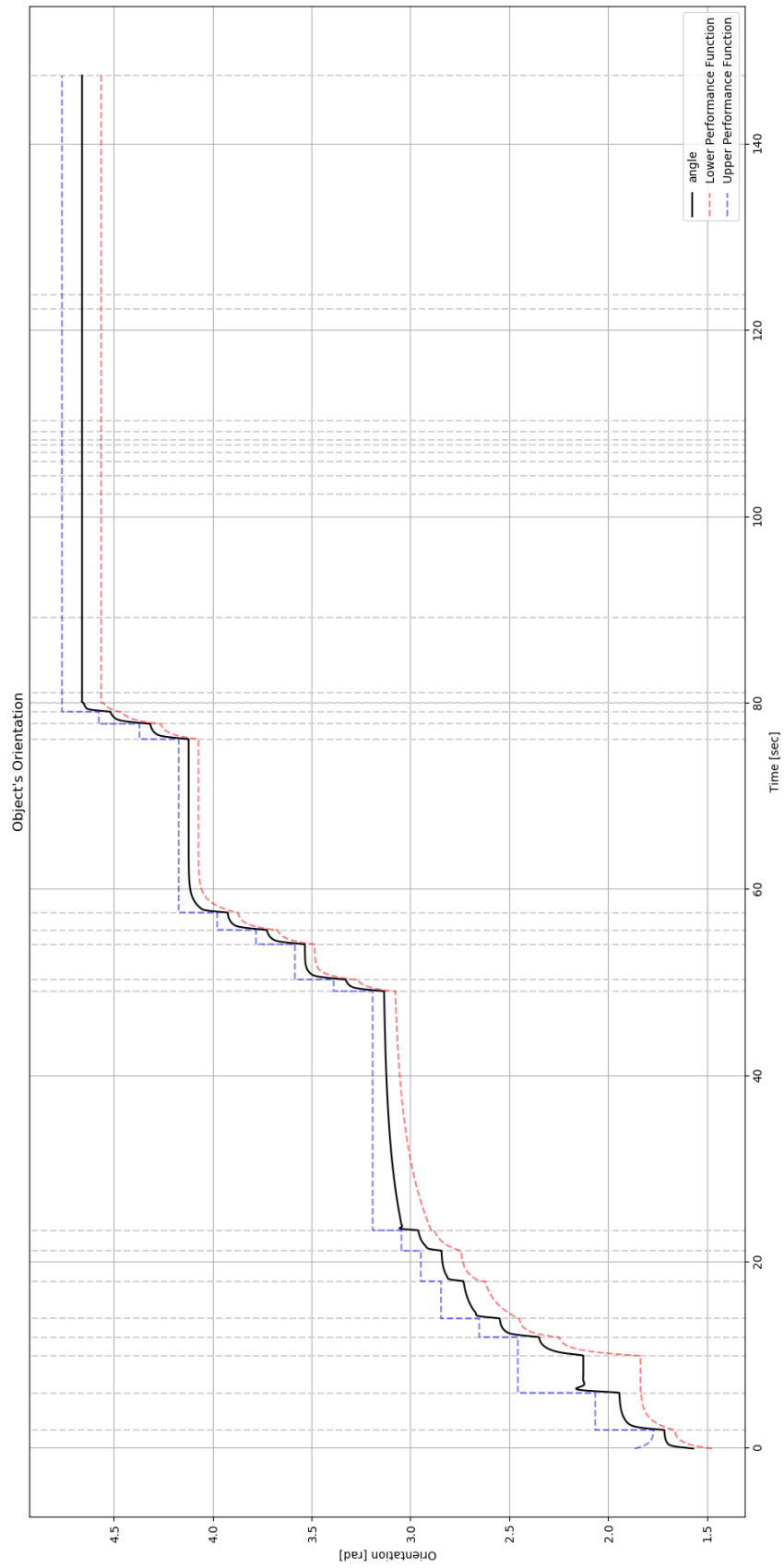
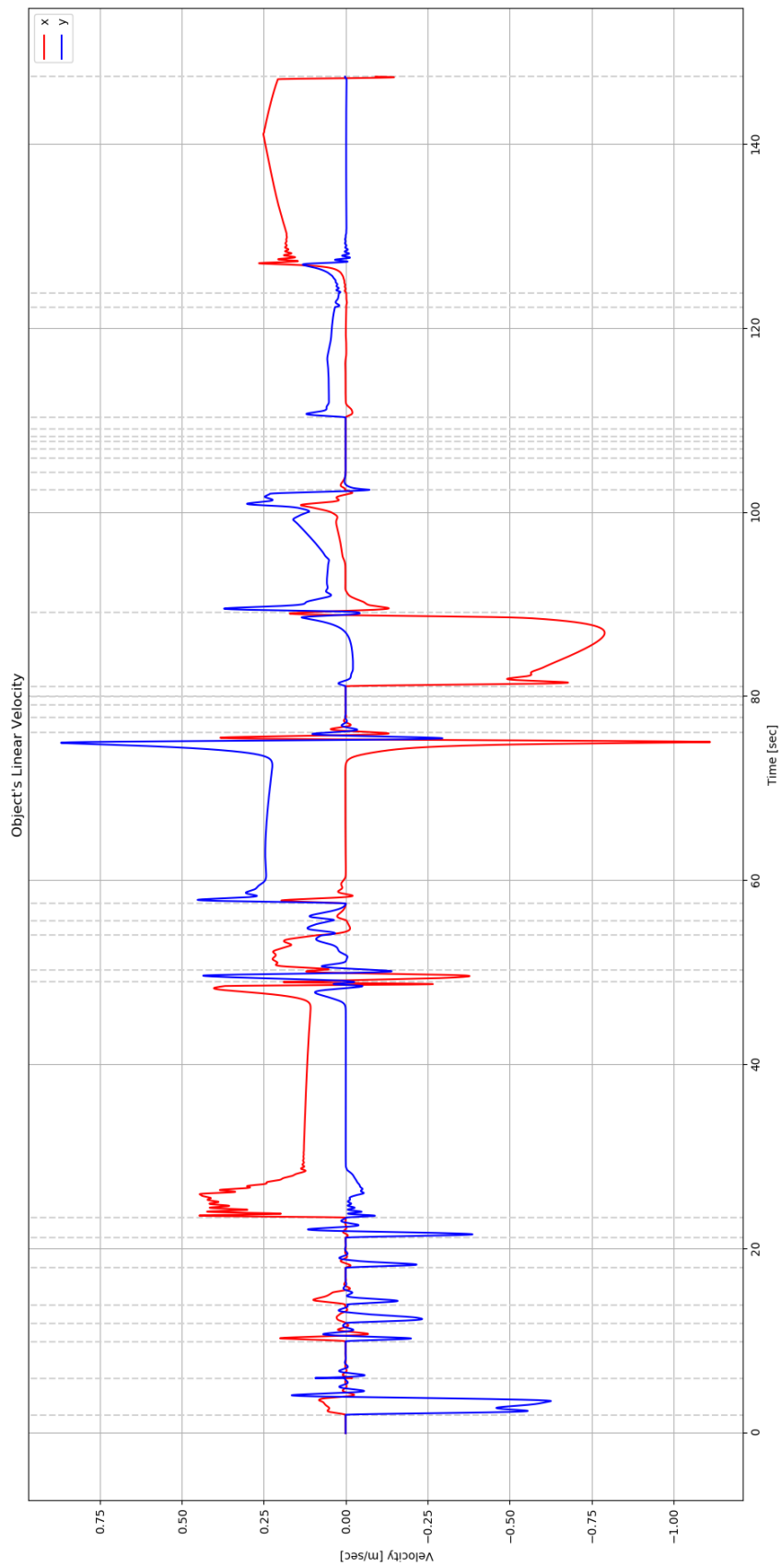
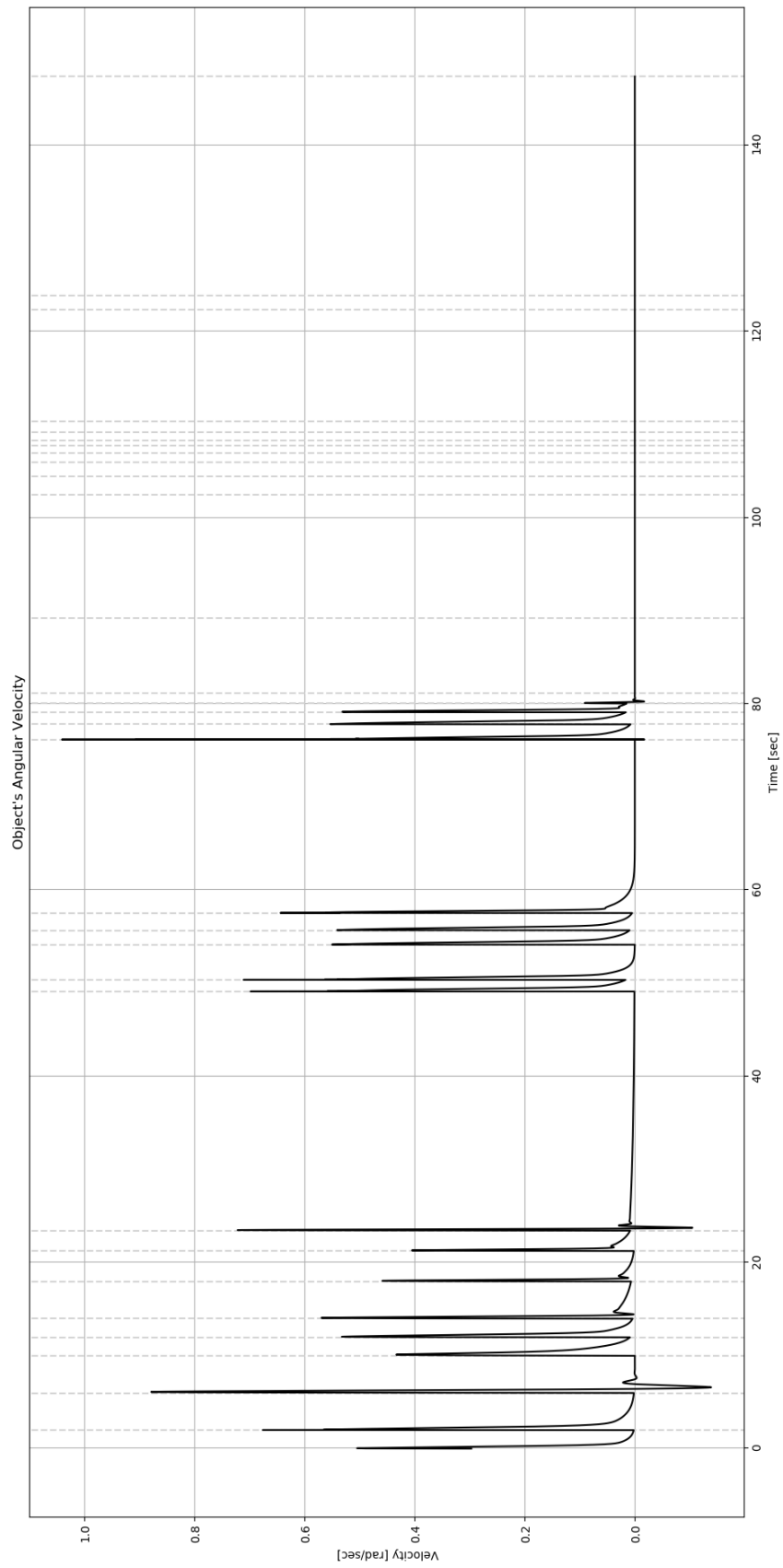


FIGURE 5.8: Evolution of the object's orientation $\theta_{\mathcal{L}}$ over time (solid line), as well as the corresponding performance functions $\rho_{\theta_{\mathcal{L}}}$ and $\bar{\rho}_{\theta_{\mathcal{L}}}$.

FIGURE 5.9: Object's linear velocity $\dot{p}_{\mathcal{L}}$.

FIGURE 5.10: Object's angular velocity $\dot{\theta}_{\mathcal{L}}$.

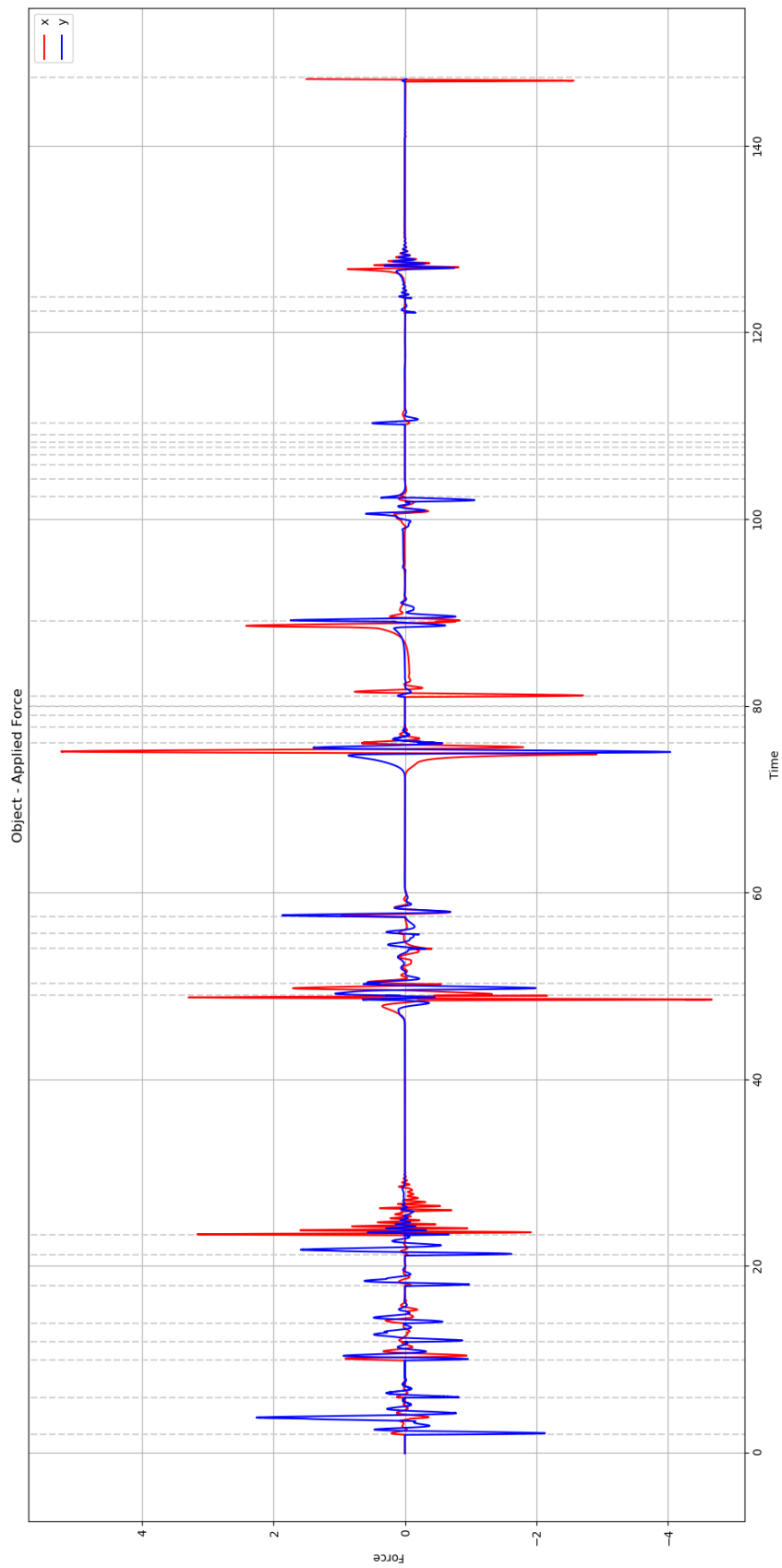


FIGURE 5.11: Total force $\tau_{\mathcal{L},p}^{des}$ applied to the object by the robots.

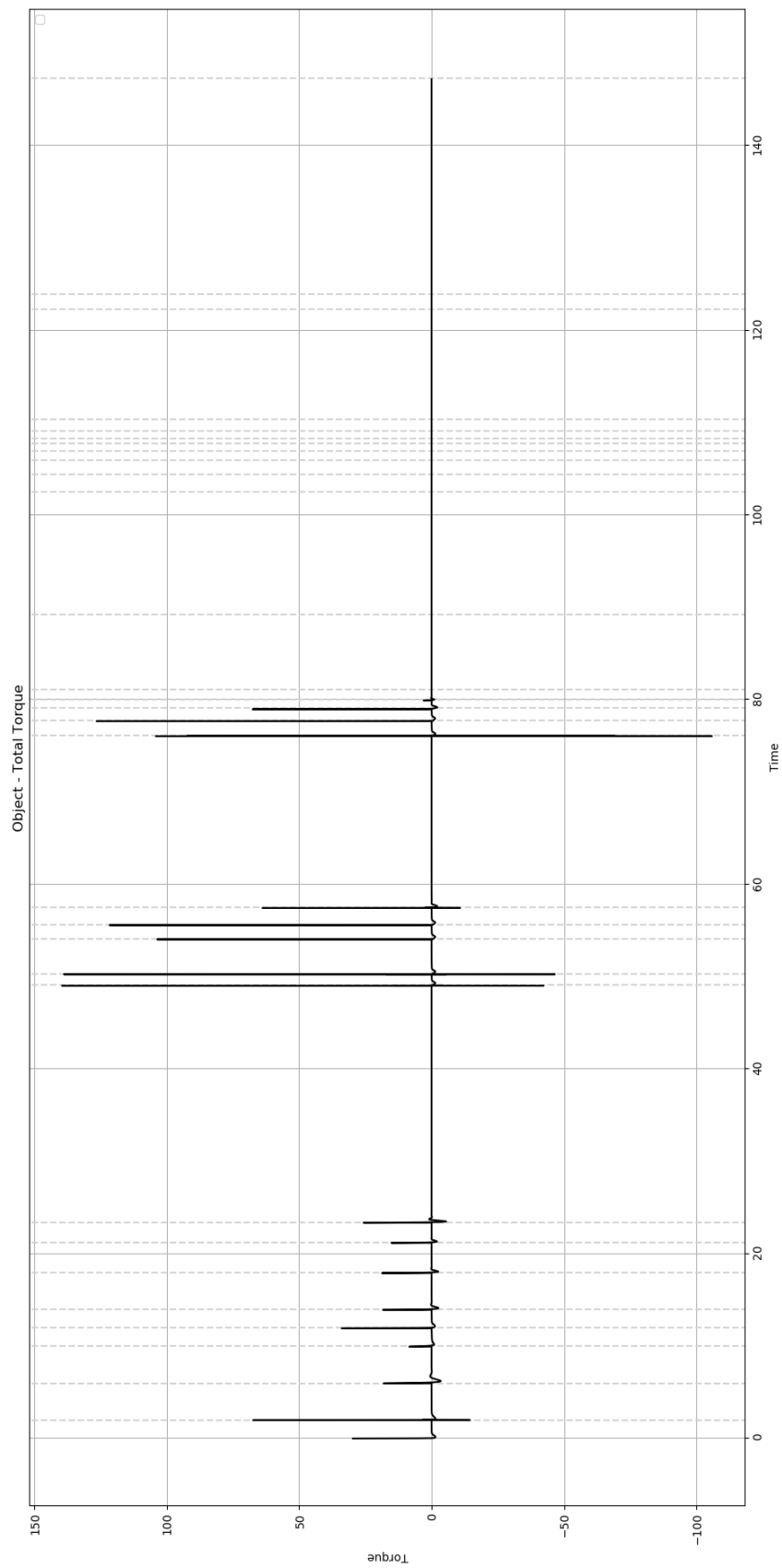


FIGURE 5.12: Total torque $\tau_{L,\theta}^{des}$ applied to the object by the robots.

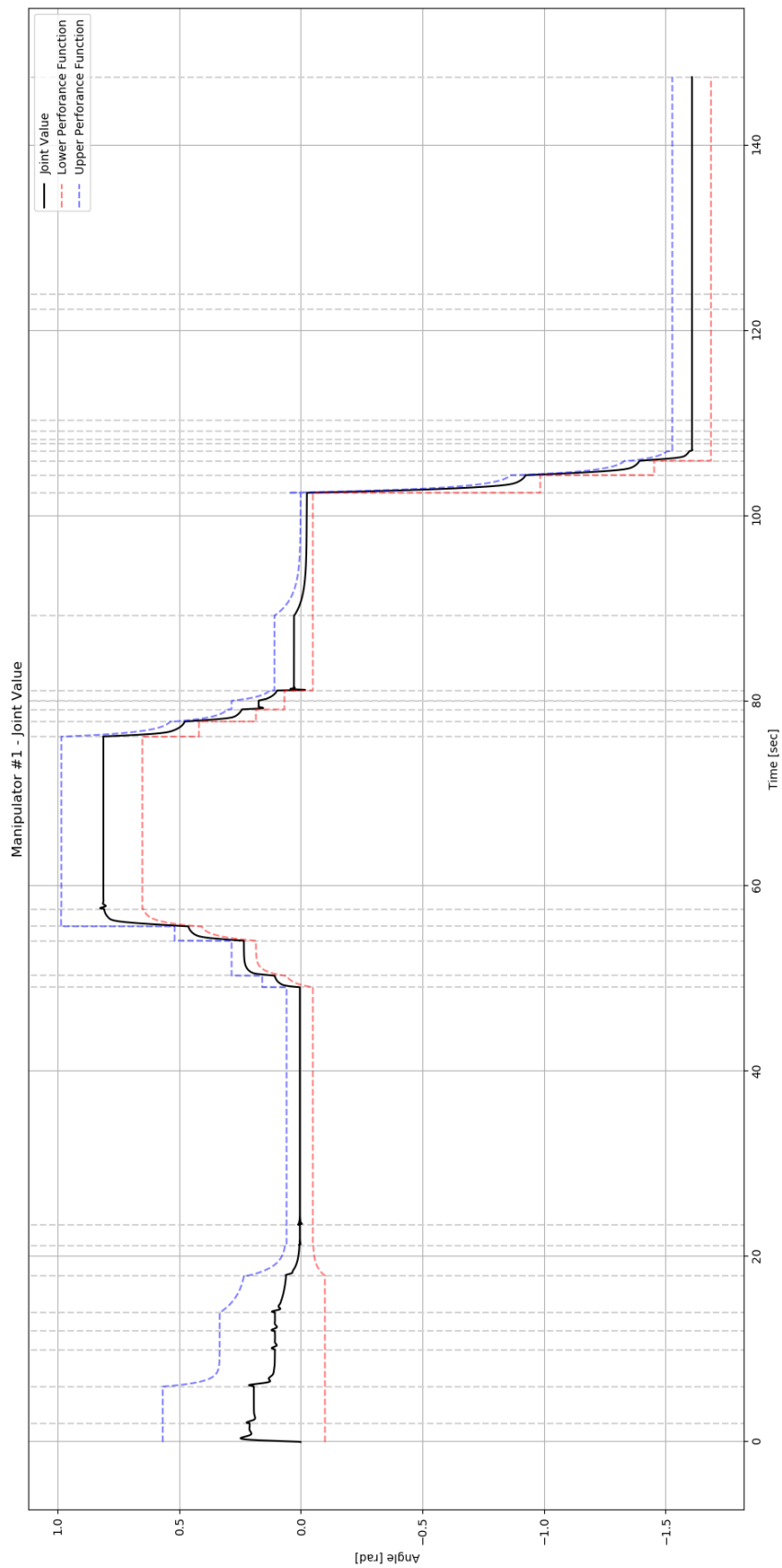


FIGURE 5.13: Evolution of joint value $q_{1,1}$ with corresponding lower and upper bounds.

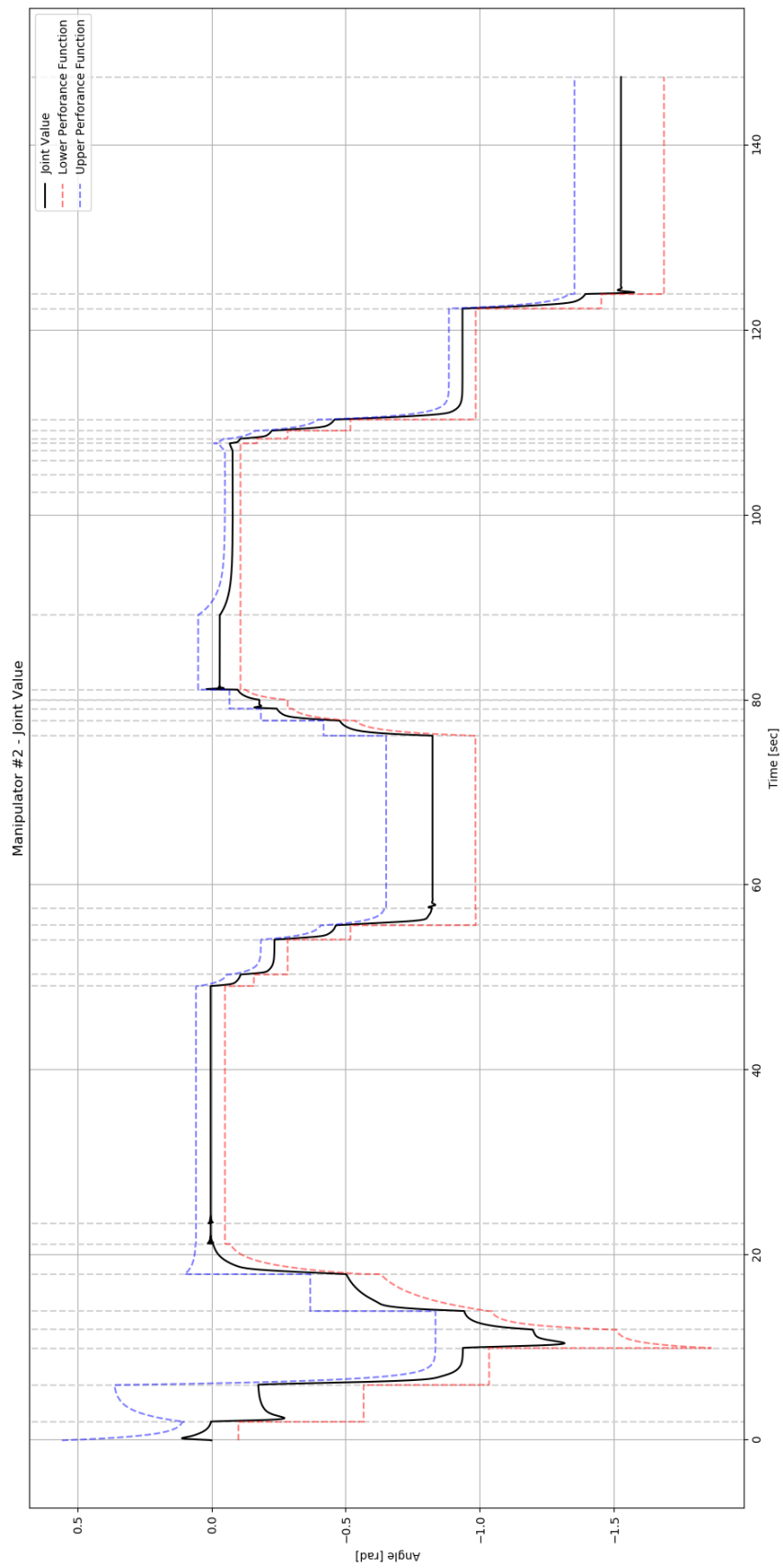


FIGURE 5.14: Evolution of joint value $q_{2,1}$ with corresponding lower and upper bounds.

Chapter 6

Conclusions and Future Work

In this dissertation, we successfully employed hierarchical decomposition techniques coupled with suitably designed adaptive configuration space exploration schemes for building versatile and complete planners, capable of computing “paths” connecting two given configurations, or determine the absence of feasible solutions, after a finite amount of steps. Moreover, distributed control laws based on artificial potential fields, which could realize the generated high-level plans, were devised for addressing the navigation problem within complex, planar workspaces with guaranteed convergence properties. Simulation and experimental results validating the efficacy of the aforementioned schemes were also provided.

More specifically, we employed harmonic map theory to build a transformation of complex planar workspaces directly to point worlds that are proper for robot navigation. In addition, we presented a novel motion planning control scheme based on closed form harmonic potential fields coupled with appropriate adaptive laws for their parameters that can safely navigate a robot to its goal state from almost all initial configurations. This control scheme was also extended to accommodate the navigation problem of non-holonomic robots and keep the numeric computations tractable for large workspaces. Subsequently, we presented a methodology for designing a hybrid control scheme to tackle the navigation problem for a robotic platform of arbitrary shape. Given an initial and desired robot configuration, we designed an slice decomposition scheme which adaptively refined the configuration space representation until either a path connecting these two configuration could be found or recognize that no solution exists.

Furthermore, we adapted the aforementioned schemes in order to address the navigation problem involving a team of disk-shaped robots operating within an obstacle cluttered planar workspace, by designing a high-level planner capable of adaptively subdividing the system’s configuration space exploiting over- and under-approximations of each robot’s free space to select the most promising cells for subdivision. Finally, we presented a hybrid control scheme for addressing the cooperative object transportation problem involving a team of mobile manipulators. Particularly, the planner originally designed for the navigation of a planar, polygonal

robot, was extended for exploring configuration spaces of higher dimensions, resulting from both rotation of the object as well as reconfiguration of the robotic team. In addition, a distributed low-level control law was employed for realizing the given plan with guaranteed collision avoidance and convergence properties.

Considering future directions, a challenging one is extending the transformations and artificial potential fields devised in [chapter 2](#) to higher dimensional spaces since building harmonic transformations for domains other than the plane is not a straight-forward process. Particularly, one hindering factor is that convexity of the transformation's image is not sufficient for rendering such a transformation a diffeomorphism in higher dimensions, such as \mathbb{R}^3 . Furthermore, to the best of the authors' knowledge, the classification of domains with dimension greater than 3 w.r.t. their genus has no general solution, which renders even the problem of selecting the image of the corresponding transformation unclear. Regarding the configuration space exploration methodologies introduced in this work, a direction involves combining the planner presented in [chapter 4](#) with the ones presented in [chapter 3](#) and [chapter 5](#) in order to enable coordination of multiple polygonal-shaped or articulated robotic platforms operating within the same workspace. In addition, by exploiting the fact that most communication specifications can be readily expressed as regions about each robot where its corresponding neighbors must reside at all times, the aforementioned planner can be also extended in order to accommodate for connectivity constraints between cooperating robots. Moreover, due to the generality of the high-level planner presented in [chapter 5](#), extending the proposed scheme for addressing the cooperative transportation problem involving non-rigid objects (i.e., deformable objects such as sheets) should be straight-forward. Also, it is possible to enable this planner to take inter-robot collisions into consideration by employing a scheme based on mixed cells, similar to the one presented in [chapter 4](#). Finally, on the topic of cooperative load transportation, we aim to reduce the unnecessarily high control effort required by the employed Prescribed Performance Control laws and smooth the transitions between consecutive cells in terms of the forces applied by the mobile manipulators, as well as relax certain requirements about the object's and robots' dynamics, such as exact knowledge of the model and commonly available exact knowledge of the object's configuration.

Appendices

Appendix A

Proofs of Chapter 2

A.1 Proof of Proposition 1

By construction, it holds that $1 - (\tanh(\phi/w))^2 > 0$ for all $q \in \partial\mathcal{D} \setminus \{q_d\}$. The gradient of ϕ w.r.t. q is given by

$$\nabla_q \phi = k_d \frac{q - q_d}{\|q - q_d\|^2} - \sum_{i=1}^N k_i \frac{q - q_i}{\|q - q_i\|^2}. \quad (\text{A.1})$$

Computing the inner product of $\nabla_q \phi$ and q yields:

$$\begin{aligned} (\nabla_q \phi)^T q &= k_d \frac{\|q\|^2 - q_d^T q}{\|q - q_d\|^2} - \sum_{i=1}^N k_i \frac{(q - q_i)^T q}{\|q - q_i\|^2} \\ &\geq k_d \frac{1 - q_d^T q}{\|q - q_d\|^2} - \max_i(k_i) \sum_{i=1}^N \frac{1}{\|q - q_i\|}. \end{aligned} \quad (\text{A.2})$$

Given that all q_i lie within $\text{int}(\mathcal{D})$, the second term in the r.h.s. of (A.2) is finite for all $q \in \partial\mathcal{D}$. Similarly, the first term in the r.h.s. of (A.2) is positive for all $q \neq q_d$. Let $q' \in \partial\mathcal{D} \setminus \{q_d\}$. Additionally, the continuity of $(1 - q_d^T q)/\|q - q_d\|^2$ and $(1 - \tanh(\phi/w)^2)/(2w)$ implies that there exists a closed neighborhood $\mathcal{F}(q')$ of q' , not containing q_d , where both are positive. Hence, selecting

$$k' = k_d \min_{q \in \mathcal{F}(q')} \left(\frac{1 - q_d^T q}{\|q - q_d\|^2} \frac{1}{\sum_{i=1}^N \frac{1}{\|q - q_i\|}} \right) \quad (\text{A.3})$$

ensures that $(\nabla_q \phi)^T q > 0$ for all $q \in \mathcal{F}(q')$. Moreover, computing the derivative of $d = 1 - \|q\|^2$ w.r.t. time for all $q \in \mathcal{F}(q')$ and assuming $k_i < k'$, $\forall i \in \mathcal{I}_N$, yields $\dot{d} = 2\mathcal{K}_u s \nabla_q \psi^T q > 0$, thus the distance from the workspace boundary increases, which concludes the proof.

A.2 Proof of Proposition 2

Similarly to the proof of Proposition 3 in [62], we proceed by defining $\tilde{q}_d \triangleq q - q_d$, $\tilde{q}_i \triangleq q - q_i$ for all $i \in \mathfrak{I}_N$. Let also $\hat{q}_d \triangleq \tilde{q}_d / \|\tilde{q}_d\|$ and $\hat{q}_i \triangleq \tilde{q}_i / \|\tilde{q}_i\|$. Accordingly, the hessian of ϕ can be computed by:

$$\nabla_{\tilde{q}}^2 \phi = \frac{k_d}{\|\tilde{q}_d\|^2} (\mathcal{I}_2 - 2\hat{q}_d \hat{q}_d^T) - \sum_{i \in \mathfrak{I}_N} \frac{k_i}{\|\tilde{q}_i\|^2} (\mathcal{I}_2 - 2\hat{q}_i \hat{q}_i^T). \quad (\text{A.4})$$

Notice that at a critical point of ϕ it holds:

$$\begin{aligned} k_d \frac{\hat{q}_d}{\|\tilde{q}_d\|} &= \sum_{i \in \mathfrak{I}_N} k_i \frac{\hat{q}_i}{\|\tilde{q}_i\|} \implies \\ \frac{k_d^2}{\|\tilde{q}_d\|^2} \hat{q}_d \hat{q}_d^T &= \sum_{i \in \mathfrak{I}_N} \frac{k_i^2}{\|\tilde{q}_i\|^2} \hat{q}_i \hat{q}_i^T + \\ &\quad \sum_{i \in \mathfrak{I}_N} \sum_{j \in \mathfrak{I}_N \setminus \{i\}} \frac{k_i k_j}{\|\tilde{q}_i\| \|\tilde{q}_j\|} (\hat{q}_i \hat{q}_j^T + \hat{q}_j \hat{q}_i^T). \end{aligned} \quad (\text{A.5})$$

Substituting (A.5) into (A.4) and re-arranging the terms yields:

$$\begin{aligned} \nabla_{\tilde{q}}^2 \phi &= \left(\frac{k_d}{\|\tilde{q}_d\|^2} - \sum_{i \in \mathfrak{I}_N} \frac{k_i}{\|\tilde{q}_i\|^2} \right) \mathcal{I} + \\ &\quad 2 \left(\sum_{i \in \mathfrak{I}_N} \frac{k_i (k_d - k_i)}{k_d} \frac{1}{\|\tilde{q}_i\|^2} \hat{q}_i \hat{q}_i^T - \right. \\ &\quad \left. \frac{1}{k_d} \sum_{i \in \mathfrak{I}_N} \sum_{j \in \mathfrak{I}_N \setminus \{i\}} \frac{k_i k_j}{\|\tilde{q}_i\| \|\tilde{q}_j\|} (\hat{q}_i \hat{q}_j^T + \hat{q}_j \hat{q}_i^T) \right). \end{aligned} \quad (\text{A.6})$$

Next, we argue that for any given set of radii $\rho_i > 0$ such that $\mathcal{D}_{\rho_i}(q_i)$, $i \in \mathfrak{I}_N$ are disjoint disks that lie entirely within our domain, there exists $k'_d > 0$ such that no critical point of ϕ exists within $\mathcal{D} \setminus \bigcup_{i \in \mathfrak{I}_N} \mathcal{D}_{\rho_i}(q_i)$, for all $k_d > k'_d$. This implies that, by choosing k_d sufficiently large, each critical point of ϕ belongs to a single $\mathcal{D}_{\rho_i}(q_i)$. Let q^* be a critical point and $\ell = \operatorname{argmin}_{i \in \mathfrak{I}_N} \|q^* - q_i\|$. To show that $\nabla_{\tilde{q}}^2 \phi(q^*)$ is not degenerate, it suffices to show that its eigenvalue $\lambda(q^*)$ is positive. We recall that λ is lower bounded by the quadratic form $\hat{x}^T \nabla_{\tilde{q}}^2 \phi \hat{x}$, for all $\|\hat{x}\| = 1$. By considering the

direction of \tilde{q}_ℓ and after some tedious calculations, we obtain:

$$\begin{aligned}
\hat{q}_\ell^T \nabla_q^2 \phi(q^*) \hat{q}_\ell &= \frac{k_d}{\|\tilde{q}_d\|^2} \\
&+ \frac{k_\ell}{\|\tilde{q}_\ell\|^2} \left(\frac{k_d - 2k_\ell}{k_d} - 4 \frac{\|\tilde{q}_\ell\|}{k_d} \sum_{i \in \mathcal{I}_N \setminus \{\ell\}} \frac{k_i}{\|\tilde{q}_i\|} 2(\hat{q}_\ell^T \hat{q}_i) \right) \\
&+ 2 \sum_{i \in \mathcal{I}_N \setminus \{\ell\}} \frac{k_i(k_d - k_i)}{k_d} \frac{1}{\|\tilde{q}_i\|^2} (\hat{q}_\ell^T \hat{q}_i)^2 \\
&- \sum_{i \in \mathcal{I}_N \setminus \{\ell\}} \frac{k_i}{\|\tilde{q}_i\|^2} \\
&- \frac{2}{k_d} \sum_{i \in \mathcal{I}_N \setminus \{\ell\}} \sum_{j \in \mathcal{I}_N \setminus \{i, \ell\}} \frac{k_i k_j}{\|\tilde{q}_i\| \|\tilde{q}_j\|} 2(\hat{q}_\ell^T \hat{q}_i)(\hat{q}_\ell^T \hat{q}_j).
\end{aligned} \tag{A.7}$$

The first r.h.s. term of (A.7) is strictly positive. Since all k_i are bounded and non-negative, choosing k_d sufficiently large renders the second and third r.h.s. terms non-negative. Furthermore, note that the fourth and fifth r.h.s. terms are bounded for all $q^* \in \mathcal{D}_{\rho_\ell}(q_\ell)$. Thus, by choosing k_d sufficiently large, the first three terms of (A.7) can be made dominant, thus rendering $\hat{q}_\ell^T \nabla_q^2 \phi \hat{q}_\ell$ positive at q^* , which concludes the proof.

A.3 Proof of Proposition 3

Firstly, we will show that the robot cannot escape through the workspace's outer boundary. Let us assume that $q \rightarrow q' \in \partial \mathcal{D} \setminus \{q_d\}$. Then, $\dot{q} \rightarrow 0$ by virtue of (14), since $s(q, k) = 0$ for all $\|q\| = 1$ with $(\nabla_q \phi)^T q \leq 0$. Additionally, $w_0 \rightarrow 1$ and $w_i \rightarrow 0$, for all $i \in \mathcal{I}_N$. Thus, $\dot{k}_i < 0$ holds within a neighborhood of $\partial \mathcal{D}$, while $k_i > 0$, which implies that $k_i \rightarrow 0$, for all $i \in \mathcal{I}_N$. Moreover, Proposition 1 dictates that there exists $k' > 0$ for which any point in $\partial \mathcal{D} \setminus \{q_d\}$ is repulsive under ψ . Since (2.20) dictates that all k_i become less than k' in finite time, this contradicts our supposition.

Next, we consider collision avoidance between the robot and the inner obstacles. Let us assume that the robot approaches obstacle i . By construction, $w_i \rightarrow 1$ while $\nabla_q \psi \rightarrow 0$ and $\bar{w}_j \rightarrow 0$, for all $j \in \mathcal{I}_N^* \setminus \{i\}$. Notice that there exists a neighborhood \mathcal{N}_i of q_i such that $w_0 = 0$ for all $q \in \mathcal{N}_i$ due to continuity of \bar{w}_0 and $\xi_2(\bar{w}_0; \epsilon_3)$. Additionally, since the robot is assumed to approach q_i , $\dot{q}^T(q - q_i)$ cannot be identically zero inside \mathcal{N}_i . As such, as long as $k_i < \bar{k}_i$, $\dot{k}_i \geq 0$ inside \mathcal{N}_i without $\dot{k}_i = 0$ for all $q \in \mathcal{N}_i$. This implies that $k_i \nearrow 0$ as $q \rightarrow q_i$, thus rendering q_i a local maximum of ψ . Thus, there exists a neighborhood of q_i inside which $(\nabla_q \psi)^T(q - q_i) > 0$, which contradicts with our assumption.

A.4 Proof of Proposition 4

Let $V \triangleq \psi(q, k)$ be a candidate Lyapunov function, which is non-negative and vanishes only when $q = q_d$. Differentiating V along the system's trajectories yields:

$$\begin{aligned} \dot{V} = & \frac{1 - \tanh(\phi/w)^2}{2w} \left((\nabla_q \phi)^T \dot{q} \right. \\ & \left. + \ln\left(\frac{\|q - q_d\|}{2}\right) \dot{k}_d - \sum_{i \in \mathcal{J}_N} \ln\left(\frac{\|q - q_i\|}{2}\right) \dot{k}_i \right). \end{aligned} \quad (\text{A.8})$$

Given that $\dot{q} = J\dot{p}$, the first term of (A.8) can be further expanded as follows

$$(\nabla_q \phi)^T \dot{q} = -\mathcal{K}_{us} \frac{1 - \tanh(\phi/w)^2}{2w} \|\nabla_q \phi\|^2, \quad (\text{A.9})$$

which is non-positive for all $q \in \Omega$ and becomes zero only on the critical points of ψ . The second term of (A.8) is non-positive since $\dot{k}_d \geq 0$ by construction and invariance of \mathcal{W} (see Proposition 3) implies $\|q - q_d\| \leq 2$ which, in turn, implies $\ln\left(\frac{\|q - q_d\|}{2}\right) \leq 0$. Similarly, the sign of each term of the sum is determined solely by the sign of the corresponding \dot{k}_i . Substituting (2.20) yields:

$$\begin{aligned} & - \sum_{i \in \mathcal{J}_N} \ln\left(\frac{\|q - q_i\|}{2}\right) \dot{k}_i \\ & \leq - \sum_{i \in \mathcal{J}_N} \ln\left(\frac{\|q - q_i\|}{2}\right) (\bar{k}_i - k_i) w_i \ell_i g_i. \end{aligned} \quad (\text{A.10})$$

Given that $g_i \leq \|\nabla_q \psi\|^2$ and $\sum_{i \in \mathcal{J}_N} \bar{k}_i w_i \leq 1$ by construction, expanding ℓ_i into the right hand side of (A.10) leads to:

$$\begin{aligned} & - \sum_{i \in \mathcal{J}_N} \ln\left(\frac{\|q - q_i\|}{2}\right) (\bar{k}_i - k_i) w_i \ell_i g_i \\ & \leq \mathcal{K}_{us} \|\nabla_q \psi\|^2 \sum_{i \in \mathcal{J}_N} (\bar{k}_i - k_i) w_i \\ & \leq \mathcal{K}_{us} \frac{1 - \tanh(\phi/w)^2}{2w} \|\nabla_q \phi\|^2. \end{aligned} \quad (\text{A.11})$$

Thus, (A.8) is non-positive. Therefore, by invoking Lyapunov's Stability Theorem (Theorem 3.1 [45]) we may conclude that q_d is stable. Finally, LaSalle's Theorem (Theorem 3.4 [45]) dictates that the system will converge to the largest invariant set, which, in our case, consists of the critical points of ψ , thus concluding the proof.

A.5 Proof of Proposition 5

At the critical point z^* of system (2.32) the hessian $\nabla_q^2 \phi$ of ϕ is non-degenerate, since otherwise $\dot{k}_d \neq 0$. Additionally, $q^* \in \Omega \setminus \{q_d\}$ implies that $1 - (\tanh(\phi/w))^2 \neq 0$.

These two facts mean that $\nabla_q^2 \psi$ has two eigenvalues at z^* , namely λ and $-\lambda$, with $\lambda > 0$. Computing the Jacobian of f_q w.r.t. q at z^* yields:

$$\begin{aligned}\nabla_q f_q &= -\mathcal{K}_u (\nabla_q \psi)^T \nabla_q s - \mathcal{K}_u s \nabla_q^2 \psi \\ &= -\mathcal{K}_u s \nabla_q^2 \psi\end{aligned}\tag{A.12}$$

since $\nabla_q \psi(q^*) = 0$. Furthermore, by construction of the adaptive law (2.20), the Jacobian of f_k w.r.t. z at z^* is $\mathbf{0}_{(1+N) \times (3+N)}$. Thus, linearization of the system f_z at z^* yields

$$\begin{aligned}\nabla_z f_z(z^*) &= -\mathcal{K}_u s \frac{1 - \tanh(\phi/w)^2}{2w} \\ &\quad \begin{bmatrix} \nabla_q^2 \phi & \frac{\partial \phi}{\partial k_d} & \frac{\partial \phi}{\partial k_1} & \cdots & \frac{\partial \phi}{\partial k_N} \\ \mathbf{0}_{(1+N) \times (3+N)} \end{bmatrix}.\end{aligned}\tag{A.13}$$

Since the top-left block $\nabla_q^2 \phi$ is invertible at z^* , using the well known property of block matrix determinants we can see that $\nabla_{z^*} f_z$ has two non-zero eigenvalues, particularly the eigenvalues of $\nabla_q^2 \psi$ and a zero eigenvalue with multiplicity $1 + N$. Thus, $\nabla_z f_z(z^*)$ has exactly one positive eigenvalue, rendering z^* a saddle point of (2.32) (Theorem 3.7 [45]).

A.6 Proof of **Theorem 2**

In **Proposition 4**, we have proved that $\dot{\psi} < 0$ for all $q \in \Omega \setminus \{q_d\}$, except for the critical points of ϕ that lie in it. Lasalle's Invariance Theorem (Theorem 3.4 [45]) dictates that system (2.32) will converge to: a) either the desired configuration q_d , or b) the obstacles q_i , or c) a critical point $z^* = (q^*, k^*)$ with $q^* \in \Omega \setminus \{q_d\}$. We know from **Proposition 3** that critical points of case (b) are repulsive; therefore no trajectory of the system may converge to them. Regarding the critical point z^* corresponding to case (c), **Proposition 5** dictates that it must be a non-isolated, degenerate equilibrium of the whole system (2.32), since $\nabla_z f_z$ has one positive, one negative and several zero eigenvalues. Let \bar{k}_d be the upper bound of k_d that the closed-loop system can possibly attain, as indicated by **Proposition 2**. In order to prove that the set of initial conditions leading to these points has zero Lebesgue measure, we will study the properties of the gradient-like system¹ $(\psi(z), F_{z,\tau}(z))$ in the domain \mathcal{S}_z , where the scalar potential $\psi(z)$ is treated as a function to be minimized, the map $F_{z,\tau}(z) : \mathcal{S}_z \mapsto \mathbb{R}^{N+3}$ is given by

$$F_{z,\tau}(z(t)) \triangleq z(t + \tau) = z(t) + \int_t^{t+\tau} f_z(z(s)) ds\tag{A.14}$$

¹By definition, a gradient-like system is a pair of a scalar cost function and a dynamical system for which each non-equilibrium initial condition moves the state towards a new one whose cost is less than that of the initial state.

for any $\tau > 0$, and $\mathcal{S}_z \triangleq \mathcal{D} \times [1, \bar{k}_d] \times [0, \bar{k}_1] \times \dots \times [0, \bar{k}_N]$. Notice that \mathcal{S}_z is convex and closed. Additionally, the map $F_{z,\tau}(z)$ is a locally Lipschitz diffeomorphism in \mathcal{S}_z and \mathcal{S}_z is forward invariant under $F_{z,\tau}(z)$ (by virtue of [Proposition 3](#) and design of adaptive law [\(2.20\)](#)), for all $\tau > 0$. Furthermore, the unwanted equilibria of $F_{z,\tau}$ are strict saddles. Thus, following similar arguments with the proof of [Theorem 3](#) in [\[82\]](#), we conclude that the set of all initial conditions that converge to these saddles has zero Lebesgue measure, which implies that almost every trajectory of the system converges to q_d , i.e., the only stable equilibrium of [\(2.32\)](#), thus completing the proof.

A.7 Proof of [Theorem 3](#)

We begin by noting that, by virtue of [\(2.39\)](#), we only need to study the trajectories of [\(2.34\)](#) in the workspace's image since that motion is traced exactly by our robot. Considering the first part of the theorem, we note that by following the same arguments as in the proof of [Proposition 3](#) we may conclude that the robot cannot escape through the workspace's outer boundary. Likewise, assuming that $q \rightarrow q_i$, for some $i \in \mathcal{I}_N$, implies that $(\hat{n}_\theta^T \nabla_q \psi)^T \hat{n}_\theta^T (q - q_i)$ cannot be identically zero in a neighborhood of q_i . As such, since $\dot{k}_i \geq 0$ in the neighborhood of q_i , k_i cannot vanish as the robot approaches q_i which contradicts our original supposition.

To prove the second part of the theorem, first we show that the only equilibria of the closed-loop system coincide with the critical points of ψ . Assuming that $s_v \neq 0$, it is readily seen that both inner products in [\(2.36\)](#) vanish simultaneously only when $\nabla_q \psi = 0$. Considering now the case when $s_v = 0$, we note that this can only happen when $q \in \partial\mathcal{D}$ and \hat{n}_θ is tangent to $\partial\mathcal{D}$. For $\tilde{\omega}$ to also vanish when $s_v = 0$, the gradient $\nabla_q \psi$ should also be tangent to $\partial\mathcal{D}$. Recalling that the adaptive laws for k ensure that $\nabla_q \psi$ will eventually point inwards, we conclude that no equilibria other than the critical points of ψ exist.

Next, we consider ψ as a Lyapunov candidate function, whose derivative along the system's trajectories is given by [\(A.8\)](#) (note that ψ does not depend on θ). Substituting [\(2.36\)](#) in the first term of [\(A.8\)](#) yields:

$$(\nabla_q \psi)^T \dot{q} = -\mathcal{K}_v s_v \frac{1 - \tanh(\phi/w)^2}{2w} \left((\hat{n}_\theta)^T \nabla_q \psi \right)^2 \quad (\text{A.15})$$

Regarding the remaining terms of [\(A.8\)](#), given that $g_{v,i} \leq (\hat{n}_\theta^T \nabla_q \psi)^2$ one can readily verify that:

$$\begin{aligned} & - \sum_{i \in \mathcal{I}_N} \ln \left(\frac{\|q - q_i\|}{2} \right) \dot{k}_i \\ & \leq -\mathcal{K}_v s_v \frac{1 - \tanh(\phi/w)^2}{2w} \left((\hat{n}_\theta)^T \nabla_q \psi \right)^2. \end{aligned} \quad (\text{A.16})$$

Thus, invoking Lyapunov's Stability Theorem ([Theorem 3.1 \[45\]](#)) and LaSalle's Theorem ([Theorem 3.4 \[45\]](#)) concludes the proof similarly to [Proposition 4](#).

A.8 Proof of *Theorem 4*

chapter the robot's safety under the closed loop system, we note that when the $p \in \mathcal{P}_{s_\ell} \setminus \mathcal{P}_{s_{\ell+1}}$ or $p \in \mathcal{P}_{s_n}$, for all $\ell < n$, the individual control laws render every point on the corresponding partition's boundaries repulsive. When $p \in \mathcal{C}_{s_\ell, s_{\ell+1}}$, we note that, by construction, both $u^{[s_\ell]}$ and $u^{[s_{\ell+1}]}$ vanish when the robot approaches any point of $\partial\mathcal{P}_{s_\ell} \cap \partial\mathcal{P}_{s_{\ell+1}}$, preventing the robot from escaping. Additionally, the adaptive laws of each individual potential field will eventually render both $\nabla_p \psi_{s_\ell}$ and $\nabla_p \psi_{s_{\ell+1}}$ inward looking w.r.t. \mathcal{W} , rendering $\partial\mathcal{P}_{s_\ell} \cap \partial\mathcal{P}_{s_{\ell+1}}$ repulsive.

While $p \in \mathcal{P}_{s_\ell}$, we consider $V \triangleq \psi_{s_\ell}$ as a Lyapunov function candidate and we examine its time derivative along the system's trajectories when $p \in \mathcal{C}_{s_\ell, s_{\ell+1}}$:

$$\begin{aligned}
 \dot{V} &= (\nabla_p \psi_{s_\ell})^T \dot{p} + (\nabla_{k^{[s_\ell]}} \psi_{s_\ell})^T \dot{k}^{[s_\ell]} \\
 &= (\nabla_p \psi_{s_\ell})^T (u^{[s_\ell]} + \eta_{c, s_\ell, s_{\ell+1}} \eta_{t, s_\ell, s_{\ell+1}} u^{[s_{\ell+1}]}) + \\
 &\quad (\nabla_{k^{[s_\ell]}} \psi_{s_\ell})^T k^{[s_\ell]} \\
 &= (\nabla_p \psi_{s_\ell})^T u^{[s_\ell]} + (\nabla_{k^{[s_\ell]}} \psi_{s_\ell})^T k^{[s_\ell]} + \\
 &\quad \eta_{c, s_\ell, s_{\ell+1}} \eta_{t, s_\ell, s_{\ell+1}} (\nabla_p \psi_{s_\ell})^T u^{[s_{\ell+1}]}
 \end{aligned} \tag{A.17}$$

We recall that the first two r.h.s. terms of (A.17) are non-positive as shown in *Proposition 4*. Likewise, the last term is rendered non-positive by virtue of $\eta_{c, s_\ell, s_{\ell+1}}$. Additionally, we note that the equilibrium points of the system in $p \in \mathcal{C}_{s_\ell, s_{\ell+1}}$ correspond only to critical points of ψ_{s_ℓ} . By virtue of $\eta_{c, s_\ell, s_{\ell+1}}$, which vanishes at a critical point of ψ_{s_ℓ} along with its derivative, one can easily verify that the Jacobian of (2.41) is equal to the one of $u^{[s_\ell]}$, whose properties were studied in *Proposition 2*. Finally, following a similar procedure as in the proof of *Theorem 2*, we conclude that the system will converge to the specified goal configuration for almost all initial configurations.

Appendix B

Proofs of Chapter 3

B.1 Proof of Proposition 6

We note that since the workspace \mathcal{W} is an open subset of \mathbb{R}^2 , its complement must be closed, which implies that \mathcal{O}_i , $i \in \mathcal{I}_o$ are closed subsets of \mathbb{R}^2 . By recalling that \mathcal{R} is assumed to be closed, \mathcal{O}_i^c , $i \in \mathcal{I}_o$ must be closed subsets of \mathcal{CS} . Thus, $\mathcal{CS}_{\text{free}}$ is open, being the complement of \mathcal{O}_i^c , $i \in \mathcal{I}_o$ w.r.t. \mathcal{CS} .

B.2 Proof of Proposition 7

Let $p \in \mathcal{R}(\theta_o)$. We now consider the open disk $\mathcal{B}_{\varepsilon/2}(p)$ with radius $\varepsilon/2$ about p and note that the intersection of $\mathcal{B}_{\varepsilon/2}(p)$ and the interior of $\mathcal{R}(\theta_o)$ is also a non-empty union of one or more open 2-manifolds. In addition, the distance between any two points of $\mathcal{B}_{\varepsilon/2}(p)$ is at most ε . Let $\mathcal{C}(p)$ be the connected component of $\text{int}\mathcal{R}(\theta) \cap \mathcal{B}_{\varepsilon/2}(p)$ whose distance from p is zero; $p \in \mathcal{C}(p)$ if and only if $p \in \text{int}\mathcal{R}(\theta)$, otherwise p lies on the boundary of $\mathcal{C}(p)$. We now consider the image $r(p, \theta)$ of p which is obtained by rotating p about the pivot point p^* of \mathcal{R} by an angle $d\theta = \theta - \theta_o$. It is easy to see that the co-domain of $r(p, \theta)$ is the circle centered at p^* with radius $\text{dist}(p^*, p)$ and that $r(p, \theta)$ is a continuous function of θ . Since $r(p, \theta_o) \in \mathcal{B}_{\varepsilon/2}(p)$, then there exist $\theta_l(p)$ and $\theta_u(p)$ such that $\theta_l(p) < \theta_o < \theta_u(p)$ and

$$r(p, \theta) \in \mathcal{B}_{\varepsilon}(p), \quad \forall \theta \in (\theta_l(p), \theta_u(p)).$$

Thus, by construction, it holds that

$$\text{dist}(\mathcal{C}(p), r(p, \theta)) < \varepsilon, \quad \forall \theta \in (\theta_l(p), \theta_u(p)).$$

Noting that

$$\overline{\mathcal{R}}([\theta_o, \hat{\theta}_u]) = \bigcup_{p \in \mathcal{R}(\theta_o)} \bigcup_{\theta \in [\theta_o, \hat{\theta}_u]} r(p, \theta)$$

and

$$\overline{\mathcal{R}}([\hat{\theta}_l, \theta_o]) = \bigcup_{p \in \mathcal{R}(\theta_o)} \bigcup_{\theta \in [\hat{\theta}_l, \theta_o]} r(p, \theta)$$

concludes the proof.

B.3 Proof of **Proposition 8**

We note that if a point $p \in \text{int}\mathcal{R}\theta_o$ does not belong to $\underline{\mathcal{R}}(\theta_o, \theta), \theta \neq \theta_o$, there has to exist $\theta' \in [\theta_o, \theta]$ such that $p \in \partial\mathcal{R}\theta'$. As such, we consider a point \hat{p} that lies on the boundary of $\mathcal{R}\theta_o$. Let $\mathcal{C}(\hat{p}, d)$ be the set of point inside $\text{int}\mathcal{R}\theta_o$ at exactly distance $d \leq \varepsilon$ away from \hat{p} . Given that the robot's body is a 2-manifold, by selecting sufficiently small d , we can ensure that $\mathcal{C}(\hat{p}, d)$ is non-empty, for all $\hat{p} \in \partial\mathcal{R}\theta_o$. Now, let $\tilde{p}(\hat{p})$ be the point of $\mathcal{C}(\hat{p}, d)$ that is farthest away from $\partial\mathcal{R}\theta_o$ and let $\tilde{d}(\hat{p})$ be that distance. Since $\tilde{p}(\hat{p})$ is an interior point of $\mathcal{R}\theta_o$, then $\tilde{d}(\hat{p}) > 0$. Similarly to the proof of **Proposition 7**, we can show that either a) there exists $\theta(\hat{p}) \neq \theta_o$ such that $\tilde{p}(\hat{p}) \in \mathcal{R}(\theta')$, for all $\theta' \in [\theta_o, \theta(\hat{p})]$, or $\tilde{p}(\hat{p}) \in \mathcal{R}(\theta')$, for all $\theta' \in \mathbb{S}^1$. As such, selecting $\theta = \min_{\hat{p}} \theta(\hat{p})$ concludes the proof.

B.4 Proof of **Theorem 5**

We first consider the case that the navigation problem presented to our planner has at least one solution. Then, since $\mathcal{CS}_{\text{free}}$ is open, as shown in **Proposition 6**, one can find a continuous curve \mathcal{P} of configurations that lies entirely in $\mathcal{CS}_{\text{free}}$ with the initial and goal configurations being its end-points. As such, each configuration $z = [p^T, \theta]^T \in \mathcal{P}$ is well separated from the free configuration space's boundary, which implies that $d_z = \text{dist}(p, \mathcal{W}^c(\theta)) > 0$, where

$$\mathcal{W}^c(\theta) \triangleq \{q \in \mathbb{R}^2 \mid \mathcal{W} \cap \mathcal{R}(q, \theta) \neq \emptyset\}.$$

But, since $\overline{\mathcal{R}}(\mathcal{I})$ and $\underline{\mathcal{R}}(\mathcal{I})$ are continuous for sufficiently small \mathcal{I} , as shown in **Proposition 7** and **Proposition 8**, respectively, one can find $\theta_l(z)$ and $\theta_u(z)$ sufficiently close to θ , for each $z = [p^T, \theta]^T \in \mathcal{P}$, such that

$$\bar{d}_z = \text{dist}(p, \overline{\mathcal{W}}^c(\theta)) > 0$$

and

$$\underline{d}_z = \text{dist}(p, \underline{\mathcal{W}}^c(\theta)) > 0$$

where

$$\begin{aligned} \overline{\mathcal{W}}^c(\theta) &\triangleq \{q \in \mathbb{R}^2 \mid \mathcal{W} \cap \overline{\mathcal{R}}(q, [\theta_l(z), \theta_u(z)]) \neq \emptyset\} \\ \underline{\mathcal{W}}^c(\theta) &\triangleq \{q \in \mathbb{R}^2 \mid \mathcal{W} \cap \underline{\mathcal{R}}(q, [\theta_l(z), \theta_u(z)]) \neq \emptyset\} \end{aligned}$$

with $\overline{\mathcal{R}}(q, [\theta_l(z), \theta_u(z)])$ and $\underline{\mathcal{R}}(q, [\theta_l(z), \theta_u(z)])$ denoting $\overline{\mathcal{R}}([\theta_l(z), \theta_u(z)])$ and $\underline{\mathcal{R}}([\theta_l(z), \theta_u(z)])$ translated to q , respectively. By recalling that our algorithm prioritizes subdivision of the largest slice in a sequence of cells (**Algorithm 3**), it is bound to obtain a solution after a finite number of iterations.

We shall now consider the case where the problem given to our planner is infeasible. This would imply that the initial and goal configurations lie within two disjoint configuration space cells. Furthermore, since $\mathcal{CS}_{\text{free}}$ is open, these two cells have a positive distance from one another. By following a procedure similar to the above, we can show that after a sufficient amount of subdivisions, the distance between each cell and its over-approximation, as computed by the planner, can be rendered arbitrary small after a finite number of iterations, thus resulting in the over-approximating cells corresponding to each disjoint component of $\mathcal{CS}_{\text{free}}$ becoming also disjoint. As this would imply that no path made of over-approximating cells can be found after that point, the proposed algorithm would automatically terminate, which concludes the proof.

Appendix C

List of Publications

My research has resulted to 6 peer-reviewed papers for international conferences world-wide and 5 journal papers. The list of publications up to date is as follows:

Journal Papers

[J1] Constantinos Vrohidis, Panagiotis Vlantis, Charalampos P. Bechlioulis, Kostas J. Kyriakopoulos, "*Reconfigurable multi-robot coordination with guaranteed convergence in obstacle cluttered environments under local communication*", *Auton Robot* 42, 853–873 (2018) doi:10.1007/s10514-017-9660-y

[J2] Constantinos Vrohidis, Panagiotis Vlantis, Charalampos P. Bechlioulis, Kostas J. Kyriakopoulos, "*Prescribed time scale robot navigation*", in *IEEE Robotics and Automation Letters*, vol. 3, no. 2, pp. 1191-1198, April 2018.
doi: 10.1109/LRA.2018.2794616

[J3] Panagiotis Grontas, Panagiotis Vlantis, Charalampos P. Bechlioulis, Kostas J. Kyriakopoulos, "*Computationally Efficient Harmonic-based Reactive Exploration*", *IEEE Robotics and Automation Letters (RA-L)* (accepted for publication).

[J4] Panagiotis Vlantis, Constantinos Vrohidis, Charalampos P. Bechlioulis, Kostas J. Kyriakopoulos, "*Robot Navigation in Complex Workspaces Using Harmonic Maps and Adaptive Artificial Potential Fields*" (under preparation)

[J5] Panagiotis Vlantis, Charalampos P. Bechlioulis, Kostas J. Kyriakopoulos, “*Cooperative Load Transportation using Hierarchical Cell Decomposition and Harmonic Maps*” (under preparation)

Conference Papers

[C1] Panagiotis Vlantis, Constantinos Vrohidis, Charalampos P. Bechlioulis, Kostas J. Kyriakopoulos, “*Robot navigation in complex workspaces using harmonic maps*”, 2018 IEEE International Conference on Robotics and Automation (ICRA), Brisbane, QLD, 2018, pp. 1726-1731.
doi: 10.1109/ICRA.2018.8460695

[C2] Constantinos Vrohidis, Panagiotis Vlantis, Charalampos P. Bechlioulis, Kostas J. Kyriakopoulos, “*Prescribed Time Scale Robot Navigation in Dynamic Environments*”, 2018 European Control Conference (ECC), Limassol, 2018, pp. 1803-1808.
doi: 10.23919/ECC.2018.8550556

[C3] Michalis Logothetis, George C. Karras, Shahab Heshmati-Alamdari, Panagiotis Vlantis, Kostas J. Kyriakopoulos, “*A Model Predictive Control Approach for Vision-Based Object Grasping via Mobile Manipulator*”, 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Madrid, 2018, pp. 1-6.
doi: 10.1109/IROS.2018.8593759

[C4] Panagiotis Vlantis, Constantinos Vrohidis, Charalampos P. Bechlioulis, Kostas J. Kyriakopoulos, “*Orientation-Aware Motion Planning in Complex Workspaces using Adaptive Harmonic Potential Fields*” 2019 International Conference on Robotics and Automation (ICRA), Montreal, QC, Canada, 2019, pp. 8592-8598.
doi: 10.1109/ICRA.2019.8794053

[C5] Michalis Logothetis, Panagiotis Vlantis, Constantinos Vrohidis, George C. Karras, Kostas J. Kyriakopoulos, “*A Motion Planning Scheme for Cooperative Loading Using Heterogeneous Robotic Agents*”, 2019 International Conference on Robotics and Automation (ICRA), Montreal, QC, Canada, 2019, pp. 9660-9666.
doi: 10.1109/ICRA.2019.8794323

[C6] Panagiotis Vlantis, Charalampos P. Bechlioulis, Kostas J. Kyriakopoulos, *“Multi-Robot Navigation using Hierarchical Cell Decomposition and Harmonic Maps”*, 2020 International Conference on Robotics and Automation (ICRA) (submitted).

Bibliography

- [1] O. Arslan and D. E. Koditschek. “Exact robot navigation using power diagrams”. In: *2016 IEEE International Conference on Robotics and Automation (ICRA)*. 2016, pp. 1–8. DOI: [10.1109/ICRA.2016.7487090](https://doi.org/10.1109/ICRA.2016.7487090).
- [2] O. Arslan and D. E. Koditschek. “Smooth extensions of feedback motion planners via reference governors”. In: *2017 IEEE International Conference on Robotics and Automation (ICRA)*. 2017, pp. 4414–4421. DOI: [10.1109/ICRA.2017.7989510](https://doi.org/10.1109/ICRA.2017.7989510).
- [3] O. Arslan, V. Pacelli, and D. E. Koditschek. “Sensory steering for sampling-based motion planning”. In: *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 2017, pp. 3708–3715.
- [4] Omur Arslan, Dan P. Guralnik, and Daniel E. Koditschek. “Hierarchically clustered navigation of distinct Euclidean particles”. In: *2012 50th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*. Oct. 2012, nil. DOI: [10.1109/allerton.2012.6483320](https://doi.org/10.1109/allerton.2012.6483320). URL: <https://doi.org/10.1109/allerton.2012.6483320>.
- [5] Omur Arslan and Daniel E Koditschek. “Sensor-based reactive navigation in unknown convex sphere worlds”. In: *The International Journal of Robotics Research* 38.2-3 (2019), pp. 196–223. DOI: [10.1177/0278364918796267](https://doi.org/10.1177/0278364918796267).
- [6] Charalampos P. Bechlioulis and Kostas J. Kyriakopoulos. “Collaborative Multi-Robot Transportation in Obstacle-Cluttered Environments Via Implicit Communication”. In: *Frontiers in Robotics and AI* 5.nil (2018), nil. DOI: [10.3389/frobt.2018.00090](https://doi.org/10.3389/frobt.2018.00090). URL: <https://doi.org/10.3389/frobt.2018.00090>.
- [7] R. Bohlin and L. E. Kavraki. “Path planning using lazy PRM”. In: *Proceedings 2000 ICRA. Millennium Conference. IEEE International Conference on Robotics and Automation. Symposia Proceedings*. Vol. 1. 2000, 521–528 vol.1. DOI: [10.1109/ROBOT.2000.844107](https://doi.org/10.1109/ROBOT.2000.844107).
- [8] Jean-Daniel Boissonnat and Mariette Yvinec. *Algorithmic Geometry*. New York, NY, USA: Cambridge University Press, 1998. ISBN: 0-521-56529-4.
- [9] V. Boor, M. H. Overmars, and A. F. van der Stappen. “The Gaussian sampling strategy for probabilistic roadmap planners”. In: *Proceedings 1999 IEEE International Conference on Robotics and Automation*. Vol. 2. 1999, 1018–1023 vol.2. DOI: [10.1109/ROBOT.1999.772447](https://doi.org/10.1109/ROBOT.1999.772447).
- [10] R. A. Brooks and T. Lozano-Pérez. “A subdivision algorithm in configuration space for findpath with rotation”. In: *IEEE Transactions on Systems, Man,*

- and *Cybernetics* SMC-15.2 (1985), pp. 224–233. ISSN: 0018-9472. DOI: [10.1109/TSMC.1985.6313352](https://doi.org/10.1109/TSMC.1985.6313352).
- [11] C. Cai and S. Ferrari. “Information-Driven Sensor Path Planning by Approximate Cell Decomposition”. In: *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)* 39.3 (2009), pp. 672–689. ISSN: 1083-4419. DOI: [10.1109/TSMCB.2008.2008561](https://doi.org/10.1109/TSMCB.2008.2008561).
- [12] John F. Canny. *The Complexity of Robot Motion Planning*. Cambridge, MA, USA: MIT Press, 1988. ISBN: 0-262-03136-1.
- [13] C. I. Connolly. “Applications of harmonic functions to robotics”. In: *Proceedings of the 1992 IEEE International Symposium on Intelligent Control*. 1992, pp. 498–502. DOI: [10.1109/ISIC.1992.225141](https://doi.org/10.1109/ISIC.1992.225141).
- [14] Christopher Connolly, J B Burns, and R Weiss. “Path Planning Using Laplace’s Equation”. In: (June 1995).
- [15] N. J. Cowan, J. D. Weingarten, and D. E. Koditschek. “Visual servoing via navigation functions”. In: *IEEE Transactions on Robotics and Automation* 18.4 (2002), pp. 521–533. ISSN: 1042-296X.
- [16] Jun Deng and Zhijun Li. “Model predictive control for visual servo steering of nonholonomic mobile robots”. In: *Proceeding of the 11th World Congress on Intelligent Control and Automation*. June 2014, nil. DOI: [10.1109/wcica.2014.7052738](https://doi.org/10.1109/wcica.2014.7052738). URL: <https://doi.org/10.1109/wcica.2014.7052738>.
- [17] Satyan Devadoss and Joseph O’Rourke. *Discrete and Computational Geometry*. Apr. 2011. ISBN: 978-0-691-14553-2.
- [18] D. V. Dimarogonas. “Sufficient Conditions for Decentralized Potential Functions Based Controllers Using Canonical Vector Fields”. In: *IEEE Transactions on Automatic Control* 57.10 (2012), pp. 2621–2626. ISSN: 0018-9286. DOI: [10.1109/TAC.2012.2191319](https://doi.org/10.1109/TAC.2012.2191319).
- [19] K.D. Do. “Formation control of multiple elliptical agents with limited sensing ranges”. In: *Automatica* 48.7 (2012), pp. 1330–1338. ISSN: 0005-1098.
- [20] Peter Duren and Walter Hengartner. “Harmonic mappings of multiply connected domains”. In: *Pacific J. Math.* 180 (1997), pp. 201–220.
- [21] Farbod Fahimi, C. Nataraj, and Hashem Ashrafiuon. “Real-time obstacle avoidance for multiple mobile robots”. In: *Robotica* 27.2 (2009), 189–198. DOI: [10.1017/S0263574708004438](https://doi.org/10.1017/S0263574708004438).
- [22] H. J. S. Feder and J. J. E. Slotine. “Real-time path planning using harmonic potentials in dynamic environments”. In: *Proceedings of International Conference on Robotics and Automation*. Vol. 1. 1997, 874–881 vol.1. DOI: [10.1109/ROBOT.1997.620144](https://doi.org/10.1109/ROBOT.1997.620144).
- [23] I. Filippidis and K. J. Kyriakopoulos. “Adjustable navigation functions for unknown sphere worlds”. In: *2011 50th IEEE Conference on Decision and Control and European Control Conference*. 2011, pp. 4276–4281. DOI: [10.1109/CDC.2011.6161176](https://doi.org/10.1109/CDC.2011.6161176).

- [24] I. F. Filippidis and K. J. Kyriakopoulos. "Navigation Functions for everywhere partially sufficiently curved worlds". In: *2012 IEEE International Conference on Robotics and Automation*. 2012, pp. 2115–2120. DOI: [10.1109/ICRA.2012.6225105](https://doi.org/10.1109/ICRA.2012.6225105).
- [25] Ioannis Filippidis and Kostas J. Kyriakopoulos. "Navigation functions for focally admissible surfaces". In: *2013 American Control Conference*. June 2013, nil. DOI: [10.1109/acc.2013.6579966](https://doi.org/10.1109/acc.2013.6579966). URL: <https://doi.org/10.1109/acc.2013.6579966>.
- [26] Alexandros Filotheou, Alexandros Nikou, and Dimos V. Dimarogonas. "Decentralized Control of Uncertain Multi-Agent Systems with Connectivity Maintenance and Collision Avoidance". In: *2018 European Control Conference (ECC)*. June 2018, nil. DOI: [10.23919/ecc.2018.8550343](https://doi.org/10.23919/ecc.2018.8550343). URL: <https://doi.org/10.23919/ecc.2018.8550343>.
- [27] J. Fink, M. A. Hsieh, and V. Kumar. "Multi-robot manipulation via caging in environments with obstacles". In: *2008 IEEE International Conference on Robotics and Automation*. 2008, pp. 1471–1476. DOI: [10.1109/ROBOT.2008.4543409](https://doi.org/10.1109/ROBOT.2008.4543409).
- [28] Santiago Garrido et al. "Robotic Motion Using Harmonic Functions and Finite Elements". In: *Journal of Intelligent and Robotic Systems* 59.1 (2009), pp. 57–73. DOI: [10.1007/s10846-009-9381-3](https://doi.org/10.1007/s10846-009-9381-3). URL: <https://doi.org/10.1007/s10846-009-9381-3>.
- [29] Roland Geraerts and Mark H. Overmars. "Sampling and node adding in probabilistic roadmap planners". In: *Robotics and Autonomous Systems* 54.2 (2006), 165–173. ISSN: 0921-8890. DOI: [10.1016/j.robot.2005.09.026](https://doi.org/10.1016/j.robot.2005.09.026). URL: <http://dx.doi.org/10.1016/j.robot.2005.09.026>.
- [30] Yoav Golan et al. "Online Robot Navigation Using Continuously Updated Artificial Temperature Gradients". In: *IEEE Robotics and Automation Letters* 2.3 (2017), pp. 1280–1287. DOI: [10.1109/lra.2017.2665682](https://doi.org/10.1109/lra.2017.2665682). URL: <https://doi.org/10.1109/lra.2017.2665682>.
- [31] L. Guibas et al. "Linear time algorithms for visibility and shortest path problems inside simple polygons". In: *Proceedings of the second annual symposium on Computational geometry - SCG '86*. 1986, nil. DOI: [10.1145/10515.10516](https://doi.org/10.1145/10515.10516). URL: <https://doi.org/10.1145/10515.10516>.
- [32] Meng Guo, Michael M. Zavlanos, and Dimos V. Dimarogonas. "Controlling the Relative Agent Motion in Multi-Agent Formation Stabilization". In: *IEEE Transactions on Automatic Control* 59.3 (2014), pp. 820–826. DOI: [10.1109/tac.2013.2281480](https://doi.org/10.1109/tac.2013.2281480). URL: <https://doi.org/10.1109/tac.2013.2281480>.
- [33] S. Hutchinson G. A. Kantor W. Burgard L. E. Kavraki H. Choset K. M. Lynch and S. Thrun. "Principles of Robot Motion: Theory, Algorithms and Implementations". In: (2005). ISSN: 0263-5747.

- [34] Rashmi Hegde and Dimitra Panagou. "Multi-agent motion planning and coordination in polygonal environments using vector fields and model predictive control". In: *2016 European Control Conference (ECC)*. June 2016, nil. DOI: [10.1109/ecc.2016.7810561](https://doi.org/10.1109/ecc.2016.7810561). URL: <https://doi.org/10.1109/ecc.2016.7810561>.
- [35] Thomas Howard et al. "Model-Predictive Motion Planning: Several Key Developments for Autonomous Mobile Robots". In: *IEEE Robotics & Automation Magazine* 21.1 (2014), pp. 64–73. DOI: [10.1109/mra.2013.2294914](https://doi.org/10.1109/mra.2013.2294914).
- [36] D. Hsu et al. "The bridge test for sampling narrow passages with probabilistic roadmap planners". In: *2003 IEEE International Conference on Robotics and Automation ()*. DOI: [10.1109/robot.2003.1242285](https://doi.org/10.1109/robot.2003.1242285). URL: <http://dx.doi.org/10.1109/robot.2003.1242285>.
- [37] David Hsu et al. "On finding narrow passages with probabilistic roadmap planners". In: 1998.
- [38] Vitor A. M. Jorge et al. "Ouroboros: Using potential field in unexplored regions to close loops". In: *2015 IEEE International Conference on Robotics and Automation (ICRA)*. May 2015, nil. DOI: [10.1109/icra.2015.7139479](https://doi.org/10.1109/icra.2015.7139479). URL: <https://doi.org/10.1109/icra.2015.7139479>.
- [39] Sertac Karaman and Emilio Frazzoli. "Sampling-based algorithms for optimal motion planning". In: *The International Journal of Robotics Research* 30.7 (2011), pp. 846–894. DOI: [10.1177/0278364911406761](https://doi.org/10.1177/0278364911406761).
- [40] L. E. Kavraki et al. "Probabilistic roadmaps for path planning in high-dimensional configuration spaces". In: *IEEE Transactions on Robotics and Automation* 12.4 (1996), pp. 566–580. ISSN: 1042-296X. DOI: [10.1109/70.508439](https://doi.org/10.1109/70.508439).
- [41] Lydia Kavraki and Jean-claude Latombe. "Randomized Preprocessing of Configuration Space for Path Planning: Articulated Robots". In: (Sept. 2002).
- [42] Lydia E. Kavraki. "Random Networks in Configuration Space for Fast Path Planning". UMI Order No. GAX95-16854. PhD thesis. Stanford, CA, USA, 1995.
- [43] K. Kedem and M. Sharir. "An efficient algorithm for planning collision-free translational motion of a convex polygonal object in 2-dimensional space amidst polygonal obstacles". In: *Proceedings of the first annual symposium on Computational geometry - SCG '85*. 1985, nil. DOI: [10.1145/323233.323244](https://doi.org/10.1145/323233.323244). URL: <https://doi.org/10.1145/323233.323244>.
- [44] K. Kedem, M. Sharir, and S. Toledo. "On Critical Orientations in the Kedem-Sharir Motion Planning Algorithm". In: *Discrete & Computational Geometry* 17.2 (1997), pp. 227–239. DOI: [10.1007/bf02770875](https://doi.org/10.1007/bf02770875). URL: <https://doi.org/10.1007/bf02770875>.
- [45] Hassan K Khalil. "Nonlinear Systems". In: *Prentice-Hall, New Jersey* 2 (1996).
- [46] O. Khatib. "Real-time obstacle avoidance for manipulators and mobile robots". In: *Proceedings. 1985 IEEE International Conference on Robotics and Automation*. Vol. 2. 1985, pp. 500–505. DOI: [10.1109/ROBOT.1985.1087247](https://doi.org/10.1109/ROBOT.1985.1087247).

- [47] J. O. Kim and P. K. Khosla. "Real-time obstacle avoidance using harmonic potential functions". In: *IEEE Transactions on Robotics and Automation* 8.3 (1992), pp. 338–349. ISSN: 1042-296X. DOI: [10.1109/70.143352](https://doi.org/10.1109/70.143352).
- [48] Zachary Kingston, Mark Moll, and Lydia E. Kavraki. "Sampling-Based Methods for Motion Planning with Constraints". In: *Annual Review of Control, Robotics, and Autonomous Systems* 1.1 (2018), pp. 159–185. DOI: [10.1146/annurev-control-060117-105226](https://doi.org/10.1146/annurev-control-060117-105226).
- [49] D. Koditschek. "Exact robot navigation by means of potential functions: Some topological considerations". In: *Proceedings. 1987 IEEE International Conference on Robotics and Automation*. Vol. 4. 1987, pp. 1–6. DOI: [10.1109/ROBOT.1987.1088038](https://doi.org/10.1109/ROBOT.1987.1088038).
- [50] Shreyas Kousik et al. "Bridging the Gap Between Safety and Real-Time Performance in Receding-Horizon Trajectory Design for Mobile Robots". In: *CoRR* abs/1809.06746 (2018).
- [51] Miroslav Krstic, Petar V. Kokotovic, and Ioannis Kanellakopoulos. *Nonlinear and Adaptive Control Design*. 1st. USA: John Wiley & Sons, Inc., 1995. ISBN: 0471127329.
- [52] S. M. LaValle and J. J. Kuffner. "Randomized kinodynamic planning". In: *Proceedings 1999 IEEE International Conference on Robotics and Automation*. Vol. 1. 1999, 473–479 vol.1. DOI: [10.1109/ROBOT.1999.770022](https://doi.org/10.1109/ROBOT.1999.770022).
- [53] Steven M. LaValle. *Planning Algorithms*. New York, NY, USA: Cambridge University Press, 2006. ISBN: 0521862051.
- [54] Steven M. Lavalle. *Rapidly-Exploring Random Trees: A New Tool for Path Planning*. Tech. rep. 1998.
- [55] Steven M. LaValle and Jr. James J. Kuffner. "Randomized Kinodynamic Planning". In: *The International Journal of Robotics Research* 20.5 (2001), pp. 378–400. DOI: [10.1177/02783640122067453](https://doi.org/10.1177/02783640122067453). eprint: <https://doi.org/10.1177/02783640122067453>. URL: <https://doi.org/10.1177/02783640122067453>.
- [56] Steven M. Lavalle, James J. Kuffner, and Jr. "Rapidly-Exploring Random Trees: Progress and Prospects". In: *Algorithmic and Computational Robotics: New Directions*. 2000, pp. 293–308.
- [57] Daniel Leven and Micha Sharir. "On the Number of Critical Free Contacts of a Convex Polygonal Object Moving in Two-Dimensional Polygonal Space". In: *Discrete & Computational Geometry* 2.3 (1987), pp. 255–270. DOI: [10.1007/bf02187883](https://doi.org/10.1007/bf02187883). URL: <https://doi.org/10.1007/bf02187883>.
- [58] C. Li and H. G. Tanner. "Navigation Functions With Time-Varying Destination Manifolds in Star Worlds". In: *IEEE Transactions on Robotics* 35.1 (2019), pp. 35–48. ISSN: 1552-3098. DOI: [10.1109/TR0.2018.2875421](https://doi.org/10.1109/TR0.2018.2875421).
- [59] Zhijun Li et al. "Vision-Based Model Predictive Control for Steering of a Non-holonomic Mobile Robot". In: *IEEE Transactions on Control Systems Technology* nil.nil (2015), pp. 1–1. DOI: [10.1109/tcst.2015.2454484](https://doi.org/10.1109/tcst.2015.2454484).

- [60] Lars Lindemann and Dimos V. Dimarogonas. "Control Barrier Functions for Multi-Agent Systems Under Conflicting Local Signal Temporal Logic Tasks". In: *IEEE Control Systems Letters* 3.3 (2019), pp. 757–762. DOI: [10.1109/lcsys.2019.2917975](https://doi.org/10.1109/lcsys.2019.2917975). URL: <https://doi.org/10.1109/lcsys.2019.2917975>.
- [61] Stephen R. Lindemann and Steven M. LaValle. "Current Issues in Sampling-Based Motion Planning". In: *Robotics Research. The Eleventh International Symposium*. Ed. by Paolo Dario and Raja Chatila. Berlin, Heidelberg: Springer Berlin Heidelberg, 2005, pp. 36–54. ISBN: 978-3-540-31508-7.
- [62] S. G. Loizou. "Closed form Navigation Functions based on harmonic potentials". In: *2011 50th IEEE Conference on Decision and Control and European Control Conference*. 2011, pp. 6361–6366. DOI: [10.1109/CDC.2011.6161438](https://doi.org/10.1109/CDC.2011.6161438).
- [63] S. G. Loizou. "Navigation functions in topologically complex 3-D workspaces". In: *2012 American Control Conference (ACC)*. 2012, pp. 4861–4866. DOI: [10.1109/ACC.2012.6315464](https://doi.org/10.1109/ACC.2012.6315464).
- [64] S. G. Loizou. "The Multi-Agent Navigation Transformation: Tuning-Free Multi-Robot Navigation". In: *Robotics: Science and Systems*. July 2014. DOI: [10.15607/RSS.2014.X.017](https://doi.org/10.15607/RSS.2014.X.017).
- [65] S. G. Loizou. "The Navigation Transformation". In: *IEEE Transactions on Robotics* 33.6 (2017), pp. 1516–1523. ISSN: 1552-3098. DOI: [10.1109/TR0.2017.2725323](https://doi.org/10.1109/TR0.2017.2725323).
- [66] S. G. Loizou and A. Jadbabaie. "Density Functions for Navigation-Function-Based Systems". In: *IEEE Transactions on Automatic Control* 53.2 (2008), pp. 612–617. ISSN: 0018-9286. DOI: [10.1109/TAC.2008.917745](https://doi.org/10.1109/TAC.2008.917745).
- [67] Savvas G Loizou. "The navigation transformation: Point worlds, time abstractions and towards tuning-free navigation". In: *Control & Automation (MED), 2011 19th Mediterranean Conference on*. IEEE. 2011, pp. 303–308.
- [68] J. M. Lourenço and J. M. Lemos. "Predictive Adaptive Control of Plants With Online Structural Changes Based on Multiple Models". In: *International Journal of Adaptive Control and Signal Processing* 22.8 (2008), pp. 774–794. DOI: [10.1002/acs.1024](https://doi.org/10.1002/acs.1024). URL: <https://doi.org/10.1002/acs.1024>.
- [69] Lozano-Perez. "Spatial Planning: A Configuration Space Approach". In: *IEEE Transactions on Computers* C-32.2 (1983), pp. 108–120. ISSN: 0018-9340. DOI: [10.1109/TC.1983.1676196](https://doi.org/10.1109/TC.1983.1676196).
- [70] T. Lozano-Perez. "A simple motion-planning algorithm for general robot manipulators". In: *IEEE Journal on Robotics and Automation* 3.3 (1987), pp. 224–238. ISSN: 0882-4967. DOI: [10.1109/JRA.1987.1087095](https://doi.org/10.1109/JRA.1987.1087095).
- [71] Tomás Lozano-Pérez and Michael A. Wesley. "An Algorithm for Planning Collision-Free Paths Among Polyhedral Obstacles". In: *Communications of the ACM* 22.10 (1979), pp. 560–570. DOI: [10.1145/359156.359164](https://doi.org/10.1145/359156.359164). URL: <https://doi.org/10.1145/359156.359164>.
- [72] Spyros Maniatopoulos, Dimitra Panagou, and Kostas J. Kyriakopoulos. "Model Predictive Control for the navigation of a nonholonomic vehicle with field-of-view constraints". In: *2013 American Control Conference*. June 2013, nil. DOI:

- 10.1109/acc.2013.6580446. URL: <https://doi.org/10.1109/acc.2013.6580446>.
- [73] Iraj Mantegh, Michael R. M. Jenkin, and Andrew A. Goldenberg. "Path Planning for Autonomous Mobile Robots Using the Boundary Integral Equation Method". In: *Journal of Intelligent & Robotic Systems* 59.2 (2010), pp. 191–220. DOI: [10.1007/s10846-010-9394-y](https://doi.org/10.1007/s10846-010-9394-y).
- [74] Nancy M. Amato and Yan Wu. "A Randomized Roadmap Method for Path and Manipulation Planning". In: *IEEE International Conference on Robotics and Automation*. 1996, pp. 113–120.
- [75] Lucia K. Dale Christopher Jones Daniel Vallejo Nancy M. Amato O. Burchan Bayazit. "OBPRM: An Obstacle-Based PRM for 3D Workspaces". In: *In Proc. Int. Wkshp. on Alg. Found. of Rob. (WAFR)*. 1998, pp. 155–168.
- [76] Colm Ó'Dúnlaing, Micha Sharir, and Chee Yap. "Generalized Voronoi Diagrams for a Ladder: Ii. Efficient Construction of the Diagram". In: *Algorithmica* 2.1-4 (1987), pp. 27–59. DOI: [10.1007/bf01840348](https://doi.org/10.1007/bf01840348). URL: <https://doi.org/10.1007/bf01840348>.
- [77] Colm Ó'Dunlaing, Micha Sharir, and Chee K. Yap. "Generalized Voronoi Diagrams for Moving a Ladder. I: Topological Analysis". In: *Communications on Pure and Applied Mathematics* 39.4 (1986), pp. 423–483. DOI: [10.1002/cpa.3160390402](https://doi.org/10.1002/cpa.3160390402). URL: <https://doi.org/10.1002/cpa.3160390402>.
- [78] P. Ogren and N. E. Leonard. "A convergent dynamic window approach to obstacle avoidance". In: *IEEE Transactions on Robotics* 21.2 (2005), pp. 188–195. ISSN: 1552-3098. DOI: [10.1109/TR0.2004.838008](https://doi.org/10.1109/TR0.2004.838008).
- [79] Mark H. Overmars. *A Random Approach to Motion Planning*. Tech. rep. Dept. Computer Science, Utrecht University, 1992.
- [80] K Kozłowski P Szulczyński D Pazderski. "Real-time obstacle avoidance using harmonic potential functions". In: *Journal of Automation Mobile Robotics and Intelligent* (2011).
- [81] V. Pacelli, O. Arslan, and D. E. Koditschek. "Integration of Local Geometry and Metric Information in Sampling-Based Motion Planning". In: *2018 IEEE International Conference on Robotics and Automation (ICRA)*. 2018, pp. 3061–3068. DOI: [10.1109/ICRA.2018.8460739](https://doi.org/10.1109/ICRA.2018.8460739).
- [82] I. Panageas and G. Piliouras. "Gradient Descent Only Converges to Minimizers: Non-Isolated Critical Points and Invariant Regions". In: *ArXiv e-prints* (May 2016). arXiv: [1605.00405](https://arxiv.org/abs/1605.00405) [math.DS].
- [83] Dimitra Panagou, Dusan M. Stipanovic, and Petros G. Voulgaris. "Distributed Coordination Control for Multi-Robot Networks Using Lyapunov-Like Barrier Functions". In: *IEEE Transactions on Automatic Control* 61.3 (2016), pp. 617–632. DOI: [10.1109/tac.2015.2444131](https://doi.org/10.1109/tac.2015.2444131). URL: <https://doi.org/10.1109/tac.2015.2444131>.
- [84] Dimitra Panagou, Herbert G. Tanner, and Kostas J. Kyriakopoulos. "Control Design for a Class of Nonholonomic Systems Via Reference Vector Fields and

- Output Regulation". In: *Journal of Dynamic Systems, Measurement, and Control* 137.8 (2015), nil. DOI: [10.1115/1.4030335](https://doi.org/10.1115/1.4030335). URL: <https://doi.org/10.1115/1.4030335>.
- [85] Dimitra Panagou, Herbert G. Tanner, and Kostas J. Kyriakopoulos. "Control of nonholonomic systems using reference vector fields". In: *IEEE Conference on Decision and Control and European Control Conference*. Dec. 2011, nil. DOI: [10.1109/cdc.2011.6160922](https://doi.org/10.1109/cdc.2011.6160922). URL: <https://doi.org/10.1109/cdc.2011.6160922>.
- [86] Dimitra Panagou, Herbert G. Tanner, and Kostas J. Kyriakopoulos. "Dipole-like fields for stabilization of systems with Pfaffian constraints". In: *2010 IEEE International Conference on Robotics and Automation*. May 2010, nil. DOI: [10.1109/robot.2010.5509296](https://doi.org/10.1109/robot.2010.5509296). URL: <https://doi.org/10.1109/robot.2010.5509296>.
- [87] Byungjae Park and Wan Kyun Chung. "Adaptive node sampling method for probabilistic roadmap planners". In: *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems (2009)*. DOI: [10.1109/iros.2009.5354185](https://doi.org/10.1109/iros.2009.5354185). URL: <http://dx.doi.org/10.1109/iros.2009.5354185>.
- [88] S. Paternain, D. E. Koditschek, and A. Ribeiro. "Navigation Functions for Convex Potentials in a Space With Convex Obstacles". In: *IEEE Transactions on Automatic Control* 63.9 (2018), pp. 2944–2959. ISSN: 0018-9286. DOI: [10.1109/TAC.2017.2775046](https://doi.org/10.1109/TAC.2017.2775046).
- [89] M. Peasgood, C.M. Clark, and J. McPhee. "A Complete and Scalable Strategy for Coordinating Multiple Robots Within Roadmaps". In: *IEEE Transactions on Robotics* 24.2 (2008), pp. 283–292. DOI: [10.1109/tro.2008.918056](https://doi.org/10.1109/tro.2008.918056).
- [90] Mika T. Rantanen and Martti Juhola. "Speeding up probabilistic roadmap planners with locality-sensitive hashing". In: *Robotica* 33.7 (2014), 1491–1506. ISSN: 1469-8668. DOI: [10.1017/s0263574714000873](https://doi.org/10.1017/s0263574714000873). URL: <http://dx.doi.org/10.1017/s0263574714000873>.
- [91] E. Rimon and D.E. Koditschek. "Exact Robot Navigation Using Artificial Potential Fields". In: *IEEE Transactions on Robotics and Automation* 8.5 (1992), pp. 501–518.
- [92] G. P. Roussos, D. V. Dimarogonas, and K. J. Kyriakopoulos. "3D navigation and collision avoidance for a non-holonomic vehicle". In: *2008 American Control Conference*. 2008, pp. 3512–3517. DOI: [10.1109/ACC.2008.4587037](https://doi.org/10.1109/ACC.2008.4587037).
- [93] M. Saha and J.-C. Latombe. "Finding narrow passages with probabilistic roadmaps: the small step retraction method". In: *2005 IEEE/RSJ International Conference on Intelligent Robots and Systems (2005)*. DOI: [10.1109/iros.2005.1545606](https://doi.org/10.1109/iros.2005.1545606). URL: <http://dx.doi.org/10.1109/iros.2005.1545606>.
- [94] O. Salzman, M. Hemmer, and D. Halperin. "On the Power of Manifold Samples in Exploring Configuration Spaces and the Dimensionality of Narrow Passages". In: *IEEE Transactions on Automation Science and Engineering* 12.2 (2015), pp. 529–538. ISSN: 1545-5955.

- [95] Oren Salzman et al. "Motion Planning Via Manifold Samples". In: *Algorithmica* 67.4 (2013), pp. 547–565. DOI: [10.1007/s00453-012-9736-1](https://doi.org/10.1007/s00453-012-9736-1).
- [96] Jacob T Schwartz and Micha Sharir. "On the "Piano Movers" Problem. II. General Techniques for Computing Topological Properties of Real Algebraic Manifolds". In: *Adv. Appl. Math.* 4.3 (Sept. 1983), pp. 298–351. ISSN: 0196-8858. DOI: [10.1016/0196-8858\(83\)90014-3](https://doi.org/10.1016/0196-8858(83)90014-3). URL: [http://dx.doi.org/10.1016/0196-8858\(83\)90014-3](http://dx.doi.org/10.1016/0196-8858(83)90014-3).
- [97] Jacob T. Schwartz and Micha Sharir. "On the piano movers' problem: V. The case of a rod moving in three-dimensional space amidst polyhedral obstacles". In: *Communications on Pure and Applied Mathematics* 37.6 (1984), pp. 815–848. DOI: [10.1002/cpa.3160370605](https://doi.org/10.1002/cpa.3160370605). eprint: <https://onlinelibrary.wiley.com/doi/pdf/10.1002/cpa.3160370605>. URL: <https://onlinelibrary.wiley.com/doi/abs/10.1002/cpa.3160370605>.
- [98] Micha Sharir and Shmuel Sifrony. "Coordinated Motion Planning for Two Independent Robots". In: *Annals of Mathematics and Artificial Intelligence* 3.1 (1991), pp. 107–130. DOI: [10.1007/bf01530889](https://doi.org/10.1007/bf01530889). URL: <https://doi.org/10.1007/bf01530889>.
- [99] Rahul Shome et al. "Drrt*: Scalable and Informed Asymptotically-Optimal Multi-Robot Motion Planning". In: *Autonomous Robots* (2019). DOI: [10.1007/s10514-019-09832-9](https://doi.org/10.1007/s10514-019-09832-9).
- [100] Edson Prestes e Silva et al. "Exploration Method Using Harmonic Functions". In: *Robotics and Autonomous Systems* 40.1 (2002), pp. 25–42. DOI: [10.1016/s0921-8890\(02\)00209-9](https://doi.org/10.1016/s0921-8890(02)00209-9).
- [101] Kiril Solovey, Oren Salzman, and Dan Halperin. "Finding a Needle in an Exponential Haystack: Discrete RRT for Exploration of Implicit Roadmaps in Multi-robot Motion Planning". In: *Springer Tracts in Advanced Robotics*. Springer Tracts in Advanced Robotics. Springer International Publishing, 2015, pp. 591–607. DOI: [10.1007/978-3-319-16595-0_34](https://doi.org/10.1007/978-3-319-16595-0_34).
- [102] Paul Spirakis and Chee K. Yap. "Strong Np-Hardness of Moving Many Discs". In: *Information Processing Letters* 19.1 (1984), pp. 55–59. DOI: [10.1016/0020-0190\(84\)90130-3](https://doi.org/10.1016/0020-0190(84)90130-3). URL: [https://doi.org/10.1016/0020-0190\(84\)90130-3](https://doi.org/10.1016/0020-0190(84)90130-3).
- [103] Zheng Sun et al. "Narrow passage sampling for probabilistic roadmap planning". In: *IEEE Transactions on Robotics* 21.6 (2005), 1105–1115. ISSN: 1552-3098. DOI: [10.1109/tro.2005.853485](https://doi.org/10.1109/tro.2005.853485). URL: <http://dx.doi.org/10.1109/tro.2005.853485>.
- [104] Petr Svestka and Mark H. Overmars. "Coordinated Path Planning for Multiple Robots". In: *Robotics and Autonomous Systems* 23.3 (1998), pp. 125–152. DOI: [10.1016/s0921-8890\(97\)00033-x](https://doi.org/10.1016/s0921-8890(97)00033-x).
- [105] A. Swingler and S. Ferrari. "A cell decomposition approach to cooperative path planning and collision avoidance via disjunctive programming". In: *49th IEEE Conference on Decision and Control (CDC)*. 2010, pp. 6329–6336. DOI: [10.1109/CDC.2010.5717137](https://doi.org/10.1109/CDC.2010.5717137).

- [106] Jacob T. Schwartz and Micha Sharir. "On the Piano Movers' Problem: III. Coordinating the Motion of Several Independent Bodies: The Special Case of Circular Bodies Moving Amidst Polygonal Barriers". In: *International Journal of Robotic Research - IJRR* 2 (Sept. 1983), pp. 46–75. DOI: [10.1177/027836498300200304](https://doi.org/10.1177/027836498300200304).
- [107] Jacob T. Schwartz and Micha Sharir. "On the "Piano Movers" problem. I: The case of a two-dimensional rigid polygonal body moving amidst polygonal barriers". In: *Communications on Pure and Applied Mathematics* 36 (May 1983), pp. 345–398. DOI: [10.1002/cpa.3160360305](https://doi.org/10.1002/cpa.3160360305).
- [108] Herbert G. Tanner and Amit Kumar. "Formation Stabilization of Multiple Agents Using Decentralized Navigation Functions". In: *Robotics: Science and Systems I*. June 2005, nil. DOI: [10.15607/rss.2005.i.007](https://doi.org/10.15607/rss.2005.i.007). URL: <https://doi.org/10.15607/rss.2005.i.007>.
- [109] H.G. Tanner and A. Kumar. "Towards Decentralization of Multi-robot Navigation Functions". In: *Proceedings of the 2005 IEEE International Conference on Robotics and Automation*. nil, nil. DOI: [10.1109/robot.2005.1570754](https://doi.org/10.1109/robot.2005.1570754). URL: <https://doi.org/10.1109/robot.2005.1570754>.
- [110] P Velagapudi, K Sycara, and P Scerri. "Decentralized prioritized planning in large multirobot teams". In: *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*. 2010, pp. 4603–4609. DOI: [10.1109/iros.2010.5649438](https://doi.org/10.1109/iros.2010.5649438).
- [111] Christos K. Verginis and Dimos V. Dimarogonas. "Closed-Form Barrier Functions for Multi-Agent Ellipsoidal Systems With Uncertain Lagrangian Dynamics". In: *IEEE Control Systems Letters* 3.3 (2019), pp. 727–732. DOI: [10.1109/lcsys.2019.2917822](https://doi.org/10.1109/lcsys.2019.2917822). URL: <https://doi.org/10.1109/lcsys.2019.2917822>.
- [112] Christos K. Verginis et al. "Robust Distributed Control Protocols for Large Vehicular Platoons With Prescribed Transient and Steady-State Performance". In: *IEEE Transactions on Control Systems Technology* 26.1 (2018), pp. 299–304. DOI: [10.1109/tcst.2017.2658180](https://doi.org/10.1109/tcst.2017.2658180). URL: <https://doi.org/10.1109/tcst.2017.2658180>.
- [113] P. Vlantis et al. "Robot Navigation in Complex Workspaces Using Harmonic Maps". In: *2018 IEEE International Conference on Robotics and Automation (ICRA)*. 2018, pp. 1726–1731. DOI: [10.1109/ICRA.2018.8460695](https://doi.org/10.1109/ICRA.2018.8460695).
- [114] Hari Voruganti, Bhaskar Dasgupta, and Unter Hommel. "A novel potential field based domain mapping method". In: (Jan. 2006).
- [115] Constantinos Vrohidis et al. "Prescribed Time Scale Robot Navigation in Dynamic Environments". In: *2018 European Control Conference (ECC)*. June 2018, nil. DOI: [10.23919/ecc.2018.8550556](https://doi.org/10.23919/ecc.2018.8550556). URL: <https://doi.org/10.23919/ecc.2018.8550556>.
- [116] Constantinos Vrohidis et al. "Reconfigurable Multi-Robot Coordination With Guaranteed Convergence in Obstacle Cluttered Environments Under Local Communication". In: *Autonomous Robots* 42.4 (2017), pp. 853–873. DOI: [10.1007/s105470170100](https://doi.org/10.1007/s105470170100).

- 1007/s10514-017-9660-y. URL: <https://doi.org/10.1007/s10514-017-9660-y>.
- [117] Glenn Wagner and Howie Choset. "M*: A complete multirobot path planning algorithm with performance bounds". In: *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*. 2011, pp. 3260–3267. DOI: [10.1109/iroso.2011.6095022](https://doi.org/10.1109/iroso.2011.6095022).
- [118] Glenn Wagner, Minsu Kang, and Howie Choset. "Probabilistic path planning for multiple robots with subdimensional expansion". In: *2012 IEEE International Conference on Robotics and Automation*. 2012, pp. 2886–2892. DOI: [10.1109/icra.2012.6225297](https://doi.org/10.1109/icra.2012.6225297).
- [119] Adam Wiktor et al. "Decentralized and complete multi-robot motion planning in confined spaces". In: *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*. 2014, pp. 1168–1175. DOI: [10.1109/iroso.2014.6942705](https://doi.org/10.1109/iroso.2014.6942705).
- [120] Hsin-Yi Yeh et al. "UOBPRM: A uniformly distributed obstacle-based PRM". In: *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems (2012)*. DOI: [10.1109/iroso.2012.6385875](https://doi.org/10.1109/iroso.2012.6385875). URL: <http://dx.doi.org/10.1109/IR0S.2012.6385875>.
- [121] Jiandong Zhong and Jianbo Su. "Robot Path Planning in Narrow Passages based on Probabilistic Roadmaps". In: *International Journal of Robotics and Automation* 28.3 (2013). ISSN: 1925-7090. DOI: [10.2316/journal.206.2013.3.206-3598](https://doi.org/10.2316/journal.206.2013.3.206-3598). URL: <http://dx.doi.org/10.2316/journal.206.2013.3.206-3598>.
- [122] D. J. Zhu and J. . Latombe. "New heuristic algorithms for efficient hierarchical path planning". In: *IEEE Transactions on Robotics and Automation* 7.1 (1991), pp. 9–20. ISSN: 1042-296X. DOI: [10.1109/70.68066](https://doi.org/10.1109/70.68066).

Εκτενής Ελληνική Περίληψη

Εισαγωγή

Κίνητρο

Η αυτόνομη λειτουργία ρομποτικών πλατφορμών μέσα περιβάλλοντα γεμάτα με εμπόδια αποτελεί ένα ενεργό ερευνητικό αντικείμενο με την αυτόνομη πλοήγηση να αποτελεί αναμφισβήτητα μια θεμελιώδη πτυχή της. Επιπλέον, καθώς τα καθήκοντα που τα ρομπότ καλούνται να εκπληρώσουν αυξάνονται σε πολυπλοκότητα μέρα με τη μέρα, η χρήση πολύ-ρομποτικών συστημάτων, τα οποία εμφανίζουν γενικά υψηλότερη ευρωστία και ευελιξία από μονό-ρομποτικά συστήματα, προοδευτικά αυξάνεται. Έτσι, τα προβλήματα αυτόνομης πλοήγησης που πρέπει να αντιμετωπιστούν γίνονται όλο και πιο απαιτητικά, αυξάνοντας την ανάγκη για πιο αποτελεσματικά και σθεναρά σχήματα σχεδιασμού πορείας και προγραμματισμού κίνησης. Επίσης, με σκοπό το σχεδιασμό αποτελεσματικών και αποδεδειγμένα σωστών συστημάτων πλοήγησης, αναπόφευκτα απαιτείται η μοντελοποίηση και κατανόηση της περίπλοκης τοπολογίας των χώρων διαμόρφωσης υψηλών διαστάσεων που προκύπτουν λόγω χρήσης ρομποτικών συστημάτων με πολλούς βαθμούς ελευθερίας. ζονφίγυρατιον

Στη βιβλιογραφία κανείς μπορεί να βρει ένα ευρύ φάσμα μεθοδολογιών για σχεδιασμό συστημάτων ελέγχου που επιτρέπουν σε ρομπότ να πλοηγηθούν εντός σύνθετων χώρων, με τον σχεδιασμό κίνησης στον χώρο διαμόρφωσης [33] να είναι από τις παλαιότερες. Δεδομένης μιας ρητής περιγραφής του χώρου διαμόρφωσης του συστήματος, σχήματα σχεδιασμού και ελέγχου, όπως αυτά που βασίζονται σε μεθόδους ‘αποσύνθεσης σε κελιά’ [53, 8] και κατασκευής ‘οδικού χάρτη’ [12], μπορούν να χρησιμοποιηθεί για την ασφαλή οδήγηση ενός ρομπότ προς τον στόχο του πληρώντας παράλληλα τις προβλεπόμενες προδιαγραφές εργασίας. Ωστόσο, η ακριβής περιγραφή του χώρου διαμόρφωσης ακόμη και για απλά συστήματα, από απόψεως γεωμετρίας και τοπολογίας, μπορεί να αποδειχθεί αρκετά δύσκολη, αν όχι αδύνατη. Από την άλλη πλευρά, προσεγγιστικές τεχνικές αποσύνθεσης, όπως ‘προβολή τεμαχίων’ [69, 70] και ‘ιεραρχική και προσεγγιστική αποσύνθεση σε κελιά’ [10], υπερφαλαγγίζουν το παραπάνω ζήτημα κατασκευάζοντας μια προσέγγιση του χώρου διαμόρφωσης ως ένωση πεπερασμένων, απλούστερων γεωμετρικά σχημάτων, όπως υπερ-ορθογώνια και πολύτοπα, εις βάρος ακρίβειας και πληρότητας. Επιπλέον, η πολυπλοκότητα του σχεδιασμού κίνησης είναι γνωστό ότι αυξάνεται εκθετικά ως προς την διάσταση του χώρου διαμόρφωσης [12]. Ως εκ τούτου, οι τεχνικές πλοήγησης χώρο διαμόρφωσης γενικά περιορίζεται σε πολύ χαμηλή διαστάσεων και απλές ρυθμίσεις.

Ο σχεδιασμός κίνησης μέσω δειγματοληψίας [54, 40] αποτελεί μια άλλη σύγχρονη εναλλακτική λύση που καταφέρνει να παρακάμψει τα προαναφερθέντα θέματα, καθώς ουσιαστικά απαιτεί έναν έμμεσο ορισμό του χώρου διαμόρφωσης, ο οποίος είναι γενικά πολύ πιο εύκολο να κατασκευαστεί. Από την άλλη, αυτές οι μεθοδολογίες αντιμετωπίζουν μόνο το υπό-πρόβλημα σχεδιασμού πορείας και παρέχουν μια λύση ανοικτού βρόχου που μια άλλη κατηγορία ελεγκτών πρέπει να ακολουθήσει όσο το δυνατόν ακριβέστερα, προκειμένου να εξασφαλιστεί παραμονή εντός του ελεύθερου χώρου διαμόρφωσης. Επίσης, αυτή η κατηγορία σχεδιαστών είναι γνωστό ότι δυσκολεύονται κατά την εύρεση λύσης σε προβλήματα που υπόκεινται σε σχετικώς αυστηρούς περιορισμούς (π.χ., στενούς διαδρόμους). Ένα άλλο μειονέκτημα των σχεδιαστών βάσει δειγματοληψίας είναι επίσης ότι δεν δύνανται να αντιληφθούν ανυπαρξία λύσεως ενός δεδομένου προβλήματος, με αποτέλεσμα η αδυναμία απάντησης ερωτημάτων συνδεσιμότητα εντός συγκεκριμένου χρονικού διαστήματος να μη παρέχει καμία χρήσιμη πληροφορία.

Τέλος, μία κατηγορία ελεγκτών που αντιμετωπίζουν τα προβλήματα σχεδιασμού διαδρομής και κινήσεων ως ένα βασίζεται στον σχεδιασμό εντολών κίνησης σύμφωνα με ένα Τεχνητό Πεδίο Δυναμικού. Τα Τεχνητά Πεδία Δυναμικού, τα οποία αρχικά προτάθηκαν από [46], γενικά παρέχουν ένα απλό μέσο για την κωδικοποίηση προδιαγραφών αποφυγής συγκρούσεων, με την κλίση τους να λειτουργεί ως διεύθυνση αναφοράς που κατευθύνει το ρομπότ προς την επιθυμητή διαμόρφωση. Παρά την προαισθηματική φύση τους, όπως φαίνεται στο [49], αυτή η κατηγορία ελεγκτών υποφέρει αναπόφευκτα από την παρουσία ανεπιθύμητων σημείων ισοροπίας που επάγεται από την τοπολογία του χώρου εργασίας και των οποίων η περιοχή έλξης μπορεί να μην είναι ασήμαντη. Επίσης, τεχνητά δυναμικά πεδία που εκφράζονται σε κλειστή μορφή, γενικά, περιορίζονται σε γεωμετρικά απλούς χώρους, με αποτέλεσμα την ανάγκη σχεδιασμού περίτεχνων μετασχηματισμών που απεικονίζουν πραγματική χώρους εργασίας σε απλούστερους ιδεατούς κόσμους. Συν τοις άλλοις, τα περισσότερα τεχνητά δυναμικά πεδία και μετασχηματισμοί απαιτούν επίπονη ρύθμιση παραμέτρων σχεδιασμού προκειμένου να εξαλειφθούν τα ανεπιθύμητα τοπικά ελάχιστα ή να καταστήσουν τους μετασχηματισμούς διφαιομορφισμούς.

Εν συντομία, τα σχήματα σχεδιασμού πορείας και κινήσεως για την πλοήγηση ρομπότ που αναφέρθηκαν παραπάνω διαφέρουν ουσιαστικά το ένα από το άλλο. Επιπλέον, κάθε οικογένεια παρουσιάζει ένα διαφορετικό σύνολο πλεονεκτημάτων και μειονεκτημάτων: μέθοδοι βάσει δειγματοληψίας μπορούν να αντιμετωπίσουν περίπλοκα προβλήματα αλλά δεν μπορούν να εκμεταλλευτούν τοπικές ιδιότητες του χώρου διαμόρφωσης, μέθοδοι αποσύνθεσης χώρων διαμόρφωσης χρησιμοποιούν μια ρητή αναπαράσταση που διασφαλίζει πληρότητα σε βάρος υψηλού υπολογιστικό κόστος, και τεχνητά πεδία δυναμικού παράγουν νόμους ελέγχου κλειστού βρόχου με εγγυημένες ιδιότητες ασφαλείας και σύγκλισης για τοπολογικά και γεωμετρικά απλούς χώρους εργασίας. Ως εκ τούτου, το ερώτημα που φυσικά προκύπτει είναι κατά πόσον είναι δυνατόν να συνδυαστούν τα πλεονεκτήματα αυτών των μεθόδων με σκοπό το σχεδιασμό πιο ευέλικτων αυτόνομων συστημάτων πλοήγησης ρομπότ. Σε αυτή την διατριβή, απαντάμε αυτό το ερώτημα καταφατικά καθώς χρησιμοποιούμε μεθόδους ιεραρχικής αποσύνθεσης, σε συνδυασμό με

κατάλληλα σχεδιασμένα προσαρμοστικά σχήματα εξερεύνησης, για την κατασκευή ευέλικτων και ολοκληρωμένων σχημάτων σχεδίασης πλοήγησης που μπορούν να απαντήσουν σε ερωτήματα συνδεσιμότητας, ή να προσδιορίσουν την απουσία λύσης, μετά από ένα πεπερασμένο αριθμό βημάτων. Επιπλέον, εφοδιάζουμε ρομποτικά συστήματα με κατανομημένους νόμους ελέγχου που βασίζονται σε τεχνητά πεδία δυναμικού τα οποία μπορεί να πραγματοποιήσουν με ασφάλεια τα παραγόμενα σχέδια υψηλού επιπέδου, αντιμετωπίζοντας προβλήματα πλοήγησης εντός σύνθετων χώρων εργασίας με εγγυημένη ιδιότητες σύγκλισης. Επίσης, σε αυτή την εργασία, σχεδιάζουμε πιο εξελιγμένους αλγορίθμους υψηλού επιπέδου λαμβάνοντας επίσης τις ιδιότητες των προτεινόμενων νόμων ελέγχου χαμηλού επιπέδου, για να αντιμετωπίσουμε ευκολότερα περιβάλλοντα αυθαίρετης συνδεσιμότητας και σχήματος, εξαλείφοντας έτσι την ανάγκη για κατακερματισμό του χώρου διαμόρφωσης σε απλούστερα τμήματα. Τέλος, επιδεικνύουμε την αποτελεσματικότητα των προτεινόμενων μεθοδολογιών αντιμετωπίζοντας πρακτικά προβλήματα όπως συντονισμό της κίνησης πολλών ρομπότ που λειτουργούν εντός του ίδιου χώρου, και συντονισμένης μεταφοράς αντικειμένου από μια ομάδα συνεργαζόμενων κινούμενων ρομποτικών βραχιόνων.

Πλοήγηση Δισκοειδών Ρομπότ στο Επίπεδο

Εισαγωγή

Σε αυτό το κεφάλαιο, αντιμετωπίζουμε το πρόβλημα πλοήγησης για ένα ρομπότ που λειτουργεί εντός ενός στατικού, συμπαγούς, επίπεδου χώρου εργασίας με αυθαίρετη συνδεσιμότητα και σχήμα. Για να διαχειριστούμε την τοπολογία του χώρου εργασίας, κατασκευάζουμε μια αρμονική απεικόνιση του χώρου εργασίας σε διάτρητο δίσκο, η οποία, σε αντίθεση με τους μετασχηματισμούς που προτείνονται σε [91] και [64], δεν απαιτεί την αποσύνθεση του χώρου εργασίας σε δέντρα αστεριών. Επιπλέον, σε αντίθεση με [91, 64, 65] όπου απαιτείται ρητή αναπαράσταση των ορίων του χώρου εργασίας, η προσέγγισή μας χρειάζεται μια επαρκώς ακριβής πολυγωνική προσέγγιση του χώρου εργασίας, η οποία μπορεί εύκολα να αποκτήσει στην πράξη μέσω SLAM. Επιπλέον, χρησιμοποιούμε τεχνητά αρμονικά πεδία κλειστής μορφής για την πλοήγηση, παρόμοια με αυτά σε [62], αλλά τροποποιημένα καταλλήλως με στόχο την θέσπιση προσαρμοστικών νόμων για τις παραμέτρους τους για να αντιμετωπίσουμε αποτελεσματικά συμπαγείς χώρους εργασίας. Επίσης, παρουσιάζουμε λεπτομερώς την διαδικασία κατασκευής του προτεινόμενου μετασχηματισμού χρησιμοποιώντας αριθμητικές τεχνικές, και ειδικότερα τη Μέθοδο Επιφανειακών Στοιχείων. Τέλος, προσαρμόζουμε τη μεθοδολογία μας σε δίκυκλα ρομπότ, που απαντώνται συνήθως σε πραγματικές εφαρμογές, και προτείνουμε έναν αλγόριθμο για να αντιμετωπίσουμε αποδοτικά το υπολογιστικό κόστος που επιφέρουν μεγάλοι χώροι εργασίας.

Σημειογραφία: Χρησιμοποιούμε $\mathcal{D}_r(x)$ για να δηλώσουμε τον ανοιχτό δίσκο με ακτίνα $r > 0$ και κέντρο $x \in \mathbb{R}^2$. Επιπροσθέτως, \mathcal{D} και $\partial\mathcal{D}$ δηλώνουν το κλειστό δίσκο και κύκλο με μοναδιαία ακτίνα και κέντρο την αρχή των αξόνων, αντίστοιχα. Επίσης, ορίζουμε $\mathcal{I}_N \triangleq \{1, 2, \dots, N\}$ και $\mathcal{I}_N^* \triangleq \{0\} \cup \mathcal{I}_N$. Δεδομένων συνόλων $A, B, \subseteq \mathbb{R}^n$, χρησιμοποιούμε $\text{cl}(A)$, ∂A , $\text{int}(A)$, \bar{A} για να δηλώσουμε την κλειστότητα, σύνορο, εσωτερικό και συμπλήρωμα του A ως προς το \mathbb{R}^n , αντίστοιχα, και $A \setminus B$ για να δηλώσουμε το συμπλήρωμα του B ω.ρ.τ. A . Επιπλέον, χρησιμοποιούμε $\mathbf{0}_N$ και $\mathbf{1}_N$ για να δηλώσουμε τους πίνακες-στήλη μήκους N με στοιχεία μηδέν ή ένα, αντίστοιχα, και $\mathbf{0}_{N \times M}$ για να δηλώσουμε το $N \times M$ μηδενικό πίνακα. Επίσης ορίζουμε το $\mathbf{1}_{N \times M}^k$, $k \in \mathcal{I}_M$ ως τον $N \times M$ πίνακα του οποίου η k -οστή στήλη είναι ίση με $\mathbf{1}_N$ και κάθε άλλη στήλη είναι ίση με $\mathbf{0}_N$. Δεδομένης μίας διανυσματικής συνάρτησης $f(x)$, χρησιμοποιούμε $\nabla_x f$ για να δηλώσουμε την Ιακωβιανή της $f(x)$. Επίσης, για δεδομένη καμπύλη C , χρησιμοποιούμε

$\text{len}(C)$ για να δηλώσουμε το μήκος του. Τέλος, θα λέμε ότι ένα σύνολο A είναι ελκτικό (αντίστοιχα. απωστικό) υπό ένα δυναμικό πεδίο ψ , όταν υπάρχει ένα σημείο $P_0 \notin \text{cl}(A)$ τέτοιο ώστε αν αρχικοποιούμε το ρομπότ στο P_0 και κινούμαστε αντίθετα προς την κλίση του ψ , το ρομπότ θα συγκλίνει (αντίστοιχα. δεν θα συγκλίνει) στο ∂A .

Ορισμός Προβλήματος

Θεωρούμε ένα ρομπότ που λειτουργεί μέσα σε ένα συμπαγή χώρο εργασίας $\mathcal{W} \subset \mathbb{R}^2$, οριοθετημένο από ένα ενιαίο εξωτερικό και ένα πεπερασμένο σύνολο εσωτερικών καμπυλών Jordan^1 , που αντιστοιχούν στα περιγράμματα των στατικών εμπόδιων. Θεωρούμε ότι το \mathcal{W} μπορεί να γραφτεί ως

$$\mathcal{W} = \overline{\mathcal{W}_0} \setminus \bigcup_{i \in \mathcal{I}_N} \mathcal{W}_i \quad (1)$$

όπου \mathcal{W}_i , $i \in \mathcal{I}_N^*$ δηλώνει περιοχές του \mathbb{R}^2 που το ρομπότ μπορεί να καταλαμβάνει. Ιδιαίτερα, το συμπλήρωμα του \mathcal{W}_0 θεωρείται ότι είναι μια απλά συνδεδεμένη περιοχή που μπορεί επίσης να περιλαμβάνει ένα αυστηρό υποσύνολο του ορίου της² και $\mathcal{W}_1, \mathcal{W}_2, \dots, \mathcal{W}_N$ θεωρούνται κλειστά, απλά συνδεδεμένα, συμπαγή σύνολα εντός του $\overline{\mathcal{W}_0}$ τα οποία δεν τέμνονται μεταξύ τους. Ορίζουμε $p = [x, y]^T \in \mathbb{R}^2$ ως την θέση του ρομπότ και υποθέσουμε ότι η κίνηση του ρομπότ περιγράφεται από το μοντέλο απλού ολοκληρωτή:

$$\dot{p} = u \quad (2)$$

όπου $u \in \mathbb{R}^2$ είναι το αντίστοιχο διάνυσμα εισόδου ελέγχου.

Πρόβλημα 1. Στόχος μας είναι να σχεδιάσουμε ένα νόμο ελέγχου κίνησης που μπορεί να οδηγήσει επιτυχώς ένα ρομπότ με κινηματική (2) προς μια δεδομένη διαμόρφωση γκολ $p_d \in \mathcal{W}$, από σχεδόν κάθε εφικτή αρχική διαμόρφωση $p_{\text{init}} \in \mathcal{W}$, διασφαλίζοντας παράλληλα αποφυγή συγκρούσεων, δηλαδή, $p(t) \in \mathcal{W}$ για κάθε $t \geq 0$.

Σχόλιο 1. Τα αποτελέσματα που παρουσιάζονται σε αυτό το κεφάλαιο μπορούν να χρησιμοποιηθούν εύκολα για την πλοήγηση ρομπότ δίσκου με ακτίνα $R > 0$ αυξάνοντας κατάλληλα τα σύνορα του χώρου εργασίας με την διάσταση του ρομπότ.

Αρμονικοί Μετασχηματισμοί για Πλοήγηση στο Επίπεδο

Σε αυτήν την ενότητα, παρουσιάζουμε μια μεθοδολογία που απεικονίζει τον χώρο εργασίας του ρομπότ σε ένα διάτρητο δίσκο, στον οποίο θα σχεδιαστεί παρακάτω ο νόμος ελέγχου. Συγκεκριμένα, στόχος μας είναι ο σχεδιασμός ενός μετασχηματισμού T :

¹Μία καμπύλη Jordan είναι μία συνεχής επίπεδη καμπύλη που δεν τέμνει τον εαυτό της.

²Αυτό αντιστοιχεί σε περιπτώσεις όπου θέλουμε να τοποθετήσουμε τον προορισμό του ρομπότ στο εξωτερικό σύνορο του χώρου εργασίας, η οποία δεν καταλαμβάνεται από κάποιο πραγματικό εμπόδιο, π.χ., μία πόρτα ενός διαμερίσματος.

$\text{cl}(\mathcal{W}) \mapsto \mathcal{D}$, από την κλειστότητα του χώρου διαμόρφωσης του ρομπότ $\text{cl}(\mathcal{W})$ στον μοναδιαίο δίσκο \mathcal{D} , με τις παρακάτω ιδιότητες:

1. $T(\cdot)$ απεικονίζει το εξωτερικό σύνορο $\partial\mathcal{W}_0$ στον μοναδιαίο κύκλο $\partial\mathcal{D}$,
2. $T(\cdot)$ απεικονίζει το σύνορο $\partial\mathcal{W}_i, i \in \mathcal{I}_N$ κάθε εμποδίου σε ένα ξεχωριστό σημείο $q_i = [\mathbf{u}_i, \mathbf{v}_i]^T \in \text{int}(\mathcal{D})$,
3. $T(\cdot)$ είναι διφεομορφισμός για κάθε $p \in \text{int}(\mathcal{W})$.

Για το σκοπό αυτό, προτείνουμε έναν αρμονικό μετασχηματισμό $T(p) = [\mathbf{u}(p), \mathbf{v}(p)]^T$ (δηλαδή, $\mathbf{u}(p)$ και $\mathbf{v}(p)$ είναι αρμονικές συναρτήσεις). Η ύπαρξη αυτού του μετασχηματισμού αποδείχθηκε σε [20] και επαρκείς συνθήκες που τον καθιστούν διφεομορφισμός παρουσιάστηκαν ως μέρος της απόδειξης.

Θεώρημα 1. (Θεώρημα 2 σε [20]) Έστω $\mathcal{W} \subset \mathbb{R}^2$ ένας πεπερασμένα συνδεδεμένος ανοιχτός χώρος που οριοθετείται από καμπύλες *Jordan* $\partial\mathcal{W}_0, \partial\mathcal{W}_1, \dots, \partial\mathcal{W}_N$, όπου $\partial\mathcal{W}_0$ αντιστοιχεί στο εξωτερικό σύνορο. Έστω, επίσης, $\Omega \in \mathbb{R}^2$ ένας συμπαγής κυρτός χώρος. Υποθέτουμε ότι \bar{f} είναι ένας συνεχής, ασθενής ομοιομορφισμός που διατηρεί τον προσανατολισμό από $\partial\mathcal{W}_0$ στο $\partial\Omega$. Τότε, υπάρχει αρμονικός μετασχηματισμός $f: \mathcal{W} \mapsto \Omega$ τέτοιος ώστε:

$$f(p) = \bar{f}(p), \quad \forall p \in \partial\mathcal{W}_0, \quad (3)$$

$$\int_{\partial\mathcal{W}_i} \frac{\partial f}{\partial \mathbf{n}} ds = 0, \quad \forall i \in \mathcal{I}_N \quad (4)$$

όπου $\mathbf{n} = \mathbf{n}(p)$ είναι το μοναδιαίο διάνυσμα κάθετο στο σύνορο επί του σημείου $p \in \partial\mathcal{W}_i$, $i \in \mathcal{I}_N$, ο οποίος απεικονίζει το \mathcal{W} στο Ω , πλην N σημείων.

Ως εκ τούτου, για να σχεδιάσουμε τον εν λόγω μετασχηματισμό από τον χώρο εργασίας $\text{cl}(\mathcal{W})$ στο μοναδιαίο δίσκο \mathcal{D} , πρέπει πρώτα να σχεδιάσουμε μία απεικόνιση $\bar{T}: \partial\mathcal{W}_0 \mapsto \partial\mathcal{D}$ για το σύνορο του χώρου. (δλδ. $\bar{T}(p) \triangleq [\bar{\mathbf{u}}(p), \bar{\mathbf{v}}(p)]^T$), και μετά να λύσουμε δύο προβλήματα συνοριακών τιμών $\mathbf{u}(p)$ ανδ $\mathbf{v}(p)$ υπό τους περιορισμούς (3) και (4). Για να λύσουμε αυτά τα προβλήματα, χρησιμοποιούμε αριθμητικές μεθόδους και, πιο συγκεκριμένα, την Μέθοδο Επιφανειακών Στοιχείων, προσεγγίζοντας κάθε συνιστώσα του $T(p) = [\mathbf{u}(p), \mathbf{v}(p)]^T$ από ένα άθροισμα της μορφής:

$$\mathbf{u}(p) \approx \sum_{i=0}^N \sum_{j=1}^{\tilde{M}_i} \sum_{l=1}^{\tilde{L}} C_{ijl}^x H_{ijl}(p) \quad (5)$$

$$\mathbf{v}(p) \approx \sum_{i=0}^N \sum_{j=1}^{\tilde{M}_i} \sum_{l=1}^{\tilde{L}} C_{ijl}^y H_{ijl}(p)$$

$$H_{ijl}(p) = \int_{E_{ij}} G_{ijl}(s) \ln(\|p - \tilde{p}_{i,j}(s)\|) ds \quad (6)$$

που προκύπτουν υποδιαιρώντας τα σύνορα σε $\tilde{M}_0, \tilde{M}_1, \dots, \tilde{M}_N$ πλήθος στοιχεία όπου \tilde{L} είναι το πλήθος των σημείων ελέγχου ανά στοιχείο, E_{ij} δηλώνει το j στοιχείο του

i ορίου, $\tilde{p}_{i,j}(s) \in E_{ij}$ είναι μια παραμετροποίηση του E_{ij} ως προς το μήκος του (δλδ. $\tilde{p}_{i,j} : [0, \text{len}(E_{ij})] \mapsto E_{ij}$), $G_{ijl} : [0, \text{len}(E_{ij})] \mapsto \mathbb{R}$ είναι η συνάρτηση μορφής που αντιστοιχεί στο l σημείο ελέγχου του E_{ij} , και $C_{ijl}^x, C_{ijl}^y \in \mathbb{R}$ είναι παράμετροι προς επιλογή τέτοιοι ώστε ο T να ικανοποιεί τις ιδιότητες 1-3, για κάθε $l \in \mathcal{I}_{\tilde{L}}$, $j \in \mathcal{I}_{\tilde{M}_i}$ και $i \in \mathcal{I}_N^*$. Οι εν λόγω παράμετροι προκύπτουν από την λύση των γραμμικών συστημάτων

$$AX = B_x, \quad AY = B_y \quad (7)$$

ως προς τα άγνωστα διανύσματα

$$\begin{aligned} X &= [C_{0,1,1}^x, \dots, C_{1,1,1}^x, \dots, C_{N, \tilde{M}_N, \tilde{L}}^x, \mathbf{u}_1, \dots, \mathbf{u}_N]^T \\ Y &= [C_{0,1,1}^y, \dots, C_{1,1,1}^y, \dots, C_{N, \tilde{M}_N, \tilde{L}}^y, \mathbf{v}_1, \dots, \mathbf{v}_N]^T. \end{aligned} \quad (8)$$

Ο πίνακας A και τα διανύσματα B_x, B_y προκύπτουν επιλέγοντας ένα σύνολο αυθαίρετων σημείων $\tilde{p}_{i,j}^*$ τέτοια ώστε: α) $\tilde{p}_{i,j}^* \in \partial W_i$ για κάθε $j \in \mathcal{I}_{\tilde{N}_i}$ και $i \in \mathcal{I}_N^*$, όπου $\tilde{N}_i, i \in \mathcal{I}_N^*$ δηλώνει το πλήθος σημείων επί του αντίστοιχου ορίου του \mathcal{W} , και β) $\sum_{i \in \mathcal{I}_N^*} \tilde{N}_i$ είναι ίσο με το πλήθος των άγνωστων παραμέτρων (δλδ. $\sum_{i \in \mathcal{I}_N^*} \tilde{M}_i \tilde{L}$) όπου:

$$A = \begin{bmatrix} A_0, & \mathbf{0}_{\tilde{N}_0 \times N} \\ A_1, & -\mathbb{1}_{\tilde{N}_1 \times N}^1 \\ \vdots & \vdots \\ A_N, & -\mathbb{1}_{\tilde{N}_N \times N}^N \\ A_n, & \mathbf{0}_{N \times N} \end{bmatrix}, \quad B_x = \begin{bmatrix} B_{x0} \\ \mathbf{0}_{\tilde{N}_1} \\ \vdots \\ \mathbf{0}_{\tilde{N}_N} \\ \mathbf{0}_N \end{bmatrix}, \quad B_y = \begin{bmatrix} B_{y0} \\ \mathbf{0}_{\tilde{N}_1} \\ \vdots \\ \mathbf{0}_{\tilde{N}_N} \\ \mathbf{0}_N \end{bmatrix} \quad (9)$$

$$A_k = \begin{bmatrix} H_{0,1,1}(\tilde{p}_{k,1}^*) & \cdots & H_{N, \tilde{M}_N, \tilde{L}}(\tilde{p}_{k,1}^*) \\ H_{0,1,1}(\tilde{p}_{k,2}^*) & \cdots & H_{N, \tilde{M}_N, \tilde{L}}(\tilde{p}_{k,2}^*) \\ \vdots & \vdots & \vdots \\ H_{0,1,1}(\tilde{p}_{k, \tilde{N}_k}^*) & \cdots & H_{N, \tilde{M}_N, \tilde{L}}(\tilde{p}_{k, \tilde{N}_k}^*) \end{bmatrix}, \quad \forall k \in \mathcal{I}_N^* \quad (10)$$

$$A_n = \begin{bmatrix} \sum_k^{\tilde{N}_1} \frac{\partial H_{0,1,1}}{\partial n}(\tilde{p}_{1,k}^*) & \cdots & \sum_k^{\tilde{N}_1} \frac{\partial H_{N, \tilde{M}_N, \tilde{L}}}{\partial n}(\tilde{p}_{1,k}^*) \\ \sum_k^{\tilde{N}_2} \frac{\partial H_{0,1,1}}{\partial n}(\tilde{p}_{2,k}^*) & \cdots & \sum_k^{\tilde{N}_2} \frac{\partial H_{N, \tilde{M}_N, \tilde{L}}}{\partial n}(\tilde{p}_{2,k}^*) \\ \vdots & \vdots & \vdots \\ \sum_k^{\tilde{N}_N} \frac{\partial H_{0,1,1}}{\partial n}(\tilde{p}_{N,k}^*) & \cdots & \sum_k^{\tilde{N}_N} \frac{\partial H_{N, \tilde{M}_N, \tilde{L}}}{\partial n}(\tilde{p}_{N,k}^*) \end{bmatrix} \quad (11)$$

$$B_{x0} = \begin{bmatrix} \bar{\mathbf{u}}(\tilde{p}_{0,1}^*) \\ \bar{\mathbf{u}}(\tilde{p}_{0,2}^*) \\ \vdots \\ \bar{\mathbf{u}}(\tilde{p}_{0, \tilde{N}_0}^*) \end{bmatrix}, \quad B_{y0} = \begin{bmatrix} \bar{\mathbf{v}}(\tilde{p}_{0,1}^*) \\ \bar{\mathbf{v}}(\tilde{p}_{0,2}^*) \\ \vdots \\ \bar{\mathbf{v}}(\tilde{p}_{0, \tilde{N}_0}^*) \end{bmatrix} \quad (12)$$

Σχεδιασμός Νόμου Ελέγχου

Για να επιλύσουμε το πρόβλημα **Πρόβλημα 1**, εξοπλίζουμε το ρομπότ με μία αρμονικό απεικόνιση $q = T(p)$ από την κλειστότητα του χώρου εργασίας \mathcal{W} στον μοναδιαίο δίσκο \mathcal{D} και ένα τεχνητό δυναμικό πεδίο $\psi(q, k)$ μαζί με έναν προσαρμοστικό νόμο $\dot{k} = f_k(q, k)$ για τις παραμέτρους $k = [k_d, k_1, k_2, \dots, k_N]^T$. Η ταχύτητα του ρομπότ υπολογίζεται από τον παρακάτω τύπο:

$$u = -\mathcal{K}_u s(q, k) J^{-1}(p) \nabla_q \psi(q, k) \quad (13)$$

όπου $J(p)$ δηλώνει την Ιακωβιανή του $T(p)$, $s(p, k) \geq 0$ είναι μια συνεχώς παραγωγίσιμη συνάρτηση κέρδους που δίνεται από

$$s(q, k) = \gamma \sigma_p \left(\frac{1 - \|q\|}{\epsilon_p} \right) + (1 - \gamma) \sigma_v \left(\frac{(\nabla_q \psi)^T q}{\epsilon_v + \|\nabla_q \psi\| \|q\|} \right) \quad (14)$$

με

$$\sigma_p(x) = \begin{cases} x^2(3 - 2x), & \text{if } x \leq 1 \\ 1, & \text{if } x > 1 \end{cases}, \quad (15)$$

$$\sigma_v(x) = \begin{cases} x^2, & \text{if } x \geq 0 \\ 0, & \text{if } x < 0 \end{cases} \quad (16)$$

και \mathcal{K}_u , γ , ϵ_p , ϵ_v είναι σταθερές τέτοιες ώστε $\mathcal{K}_u, \epsilon_v > 0$ και $\gamma, \epsilon_p \in (0, 1)$. Πιο συγκεκριμένα, το ψ είναι ένα αρμονικό πεδίο που ορίζεται στην εικόνα $T(\mathcal{W})$ του χώρου εργασίας \mathcal{W} και η κλίση του $-\nabla_q \psi(q, k)$ ορίζει την διεύθυνση κίνησης στον πραγματικό χώρο μέσω του αντιστρόφου της Ιακωβιανής $J^{-1}(p)$. Εκ κατασκευής, το πεδίο που προκύπτει αποκλείει τις συγκρούσεις μεταξύ του ρομπότ και εσωτερικών εμποδίων ενώ εξασφαλίζει σχεδόν καθολική σύγκλιση στον επιθυμητό προορισμό. Ωστόσο, καθώς το \mathcal{W}_0 μπορεί να μην είναι απωστικό για μία αυθαίρετη, σταθερή επιλογή των k , ο προσαρμοστικός νόμος $f_k(p, k)$, μαζί με την συνάρτηση $s(p, k)$, εξασφαλίζουν παραμονή του ρομπότ εντός του χώρου εργασίας χωρίς να χωρίς να διακυβεύονται οι ιδιότητες σύγκλισης του συνολικού συστήματος. Οι ακόλουθες υποενότητες επικεντρώνονται σε κάθε στοιχείο του προτεινόμενου νόμου ελέγχου ξεχωριστά.

Τεχνητά Αρμονικά Δυναμικά Πεδία

Ακολουθώντας μια διαδικασία παρόμοια με εκείνη που προτείνεται στην εργασία [62], κατασκευάζουμε ένα τεχνητό αρμονικό δυναμικό πεδίο στον μετασχηματισμένο χώρο \mathcal{D} από σημειακές πηγές που τοποθετούνται στην επιθυμητή διαμόρφωση $q_d = T(p_d)$ καθώς και στα σημεία $q_i = T(\partial \mathcal{W}_i)$, $\forall i \in \mathcal{I}_N$ που αντιστοιχούν σε εσωτερικά εμπόδια, ως εξής:

$$\phi = k_d \ln \left(\frac{\|q - q_d\|}{2} \right) - \sum_{i=1}^N k_i \ln \left(\frac{\|q - q_i\|}{2} \right) \quad (17)$$

όπου $k_d > 0$ και $k_i \geq 0$ δηλώνουν βάρη αρμονικών πηγών που αλλάζουν σύμφωνα με του προσαρμοστικού νόμους που παρουσιάζονται στη συνέχεια. Μια ενδιαφέρουσα ιδιότητα αυτού του δυναμικού πεδίου, η οποία απορρέει από την αρχή του μεγίστου για αρμονικές συναρτήσεις, είναι ότι, για σταθερά k , το μόνο ελάχιστο βρίσκονται στο q_d και, ενδεχομένως, στο άπειρο. Ως άμεση συνέπεια, ο Εσσιανός πίνακας $\nabla_q^2 \phi$ μη-εκφυλισμένου κρίσιμου σημείου της ϕ στο εσωτερικό του χώρου, έχει μία θετική και μία αρνητική ιδιοτιμή με το ίδιο μέγεθος, δηλ. λ και $-\lambda$ με $\lambda > 0$.

Στην συνέχεια, ορίζουμε το πεδίο ψ βάσει του ϕ , που δίνεται από

$$\psi = \frac{1 + \tanh(\phi/w)}{2} \quad (18)$$

όπου w είναι μια θετική σταθερά. Καθώς το $\tanh(\phi/w)$ είναι αύξουσα συνάρτηση, τα μοναδικά κρίσιμα σημεία του ψ είναι αυτά που κληρονομεί από το ϕ . Επιπλέον, η κλίση του ψ ως προς το q , που δίνεται από:

$$\nabla_q \psi = \frac{1 - (\tanh(\phi/w))^2}{2w} \nabla_q \phi \quad (19)$$

είναι καλά ορισμένη για κάθε $q \in \mathcal{D}$. Επιπλέον, για να μπορέσουμε να τοποθετήσουμε το q_d σε περιοχές του $\partial\mathcal{D}$ οι οποίες δεν καταλαμβάνονται από φυσικά εμπόδια (για λόγους που θα γίνουν κατανοητοί στην συνέχεια), σχεδιάζουμε προσαρμοστικούς νόμους για τις παραμέτρους k του ϕ , που καθιστούν το εξωτερικό σύνορο απωστικό.

Πριν προχωρήσουμε με τον ορισμό των προσαρμοστικών νόμων, παρουσιάζουμε δύο προτάσεις που θα χρησιμοποιηθούν στην μετέπειτα ανάλυση.

Πρόταση 1. Έστω $k_d > 0$ και $q' \in \partial\mathcal{D} \setminus \{q_d\}$. Υπάρχει $k' > 0$ τέτοιο ώστε εάν $k_i < k'$, $\forall i \in \mathcal{I}_N$, τότε q' είναι απωστικό υπό του ψ .

Πρόταση 2. Εάν k_i είναι μη αρνητικά και πεπερασμένα, υπάρχει $k'_d > 0$ τέτοιο ώστε ψ είναι Morse για κάθε $k_d \geq k'_d$.

Προσαρμοστικοί Νόμοι Ελέγχου

Εδώ παρουσιάζουμε τον νόμο $\dot{k} = f_k(q, k)$ που προσαρμόζει τις παραμέτρους του ψ . Ο σκοπός του είναι να καταστήσει: α) το εξωτερικό σύνορο απωστικό και β) οποιοδήποτε κρίσιμο σημείο του ϕ στην γειτονιά του ρομπότ μη-εκφυλισμένο. Πιο συγκεκριμένα, θεωρούμε f_k με μορφή:

$$\begin{aligned} \dot{k}_d &= \xi_1(\lambda + \|\nabla_q \phi\|; \epsilon_1) \\ \dot{k}_i &= (\bar{k}_i - k_i) w_i \ell_i g_i - \mathcal{K}_k k_i h_i w_0 (g_0 + \xi_1(s; \epsilon_2)), \quad \forall i \in \mathcal{I}_N \end{aligned} \quad (20)$$

όπου τα w_i , g_i και h_i , $i \in \mathcal{I}_N$, δηλώνουν συναρτήσεις που θα οριστούν παρακάτω, \bar{k}_i , $i \in \mathcal{I}_N$, είναι επιθυμητά άνω όρια των k_i , λ δηλώνει την μη-αρνητική ιδιοτιμή του $\nabla_q^2 \phi$, \mathcal{K}_k είναι ένα θετικό κέρδος, και ϵ_1 , ϵ_2 είναι μικρές θετικές σταθερές. Η συνεχώς

παραγωγίσιμη συνάρτηση $\xi_1(x; \epsilon)$ και οι $\ell_i(q)$ δίνονται από:

$$\xi_1(x; \epsilon) = 1 - \sigma_p(x/\epsilon) \quad (21)$$

$$\ell_i(q) = -\frac{\mathcal{K}_{us}(q, k)}{\ln\left(\frac{\|q - q_i\|}{2}\right)}. \quad (22)$$

Σύμφωνα με **Πρόταση 1**, η πρώτη προδιαγραφή μας μπορεί να επιτευχθεί σχεδιάζοντας τα f_k ώστε να μειώνουν τα k_i καθώς το ρομπότ πλησιάζει το $\partial\mathcal{D}$. Επίσης, θα πρέπει να εξασφαλίσουμε ότι τα k_i δεν μηδενίζονται σε μία γειτονιά του q_i , για κάθε $i \in \mathcal{I}_N$. Ωε εκ τούτου, ορίζουμε τα g_i ως εξής:

$$g_i = \sigma_v(\bar{g}_i), \quad \forall i \in \mathcal{I}_N^* \quad (23)$$

με

$$\begin{aligned} \bar{g}_0 &= \frac{1}{4} \left(\alpha \|\nabla_q \psi\| \|q - q_d\| - (\nabla_q \psi)^T (q - q_d) \right) \\ \bar{g}_i &= \frac{1}{2} (\nabla_q \psi)^T (q - q_i), \quad \forall i \in \mathcal{I}_N \end{aligned} \quad (24)$$

όπου $\alpha \in (0, 1]$ σταθερά. Επίσης, ορίζουμε συνοδευτικά βάρη w_i για να εξασφαλίσουμε ότι μόνο ένας όρος (20) κυριαρχεί καθώς το ρομπότ πλησιάζει κάποιο σύνορο του \mathcal{W} :

$$\begin{aligned} w_0 &= \frac{\xi_2(\bar{w}_0; \epsilon_3)}{\bar{w}_0 + \sum_{j=1}^N (\bar{k}_j \bar{w}_j)} \\ w_i &= \frac{\bar{w}_i}{\bar{w}_0 + \sum_{j=1}^N (\bar{k}_j \bar{w}_j)}, \quad \forall i \in \mathcal{I}_N \end{aligned} \quad (25)$$

με

$$\bar{w}_i = \bar{r}_i / (r_i + \bar{r}_i), \quad \forall i \in \mathcal{I}_N^*, \quad (26)$$

$$r_i = \begin{cases} (1 - \|q\|)^2, & \text{εάν } i = 0, \\ \|q - q_i\|^2, & \text{εάν } i \in \mathcal{I}_N \end{cases} \quad (27)$$

$$\bar{r}_i = \sqrt[m]{\sum_{j \neq i} (r_j)^m}, \quad \forall i \in \mathcal{I}_N \quad (28)$$

$$\xi_2(x; \epsilon) = \begin{cases} 0, & \text{εάν } x < \epsilon \\ \left(\frac{x-\epsilon}{1-\epsilon}\right)^2 \left(3 - 2\frac{x-\epsilon}{1-\epsilon}\right), & \text{εάν } \epsilon \leq x \leq 1 \\ 1, & \text{οτηερwise} \end{cases} \quad (29)$$

για μία σταθερά $\epsilon_3 \in (0, 1)$ και ακέραιο $m < -1$ που λειτουργεί ως μία ομαλή εκτίμηση του $\min_{j \neq i} (r_j)$, $i \in \mathcal{I}_N$. Τέλος, τα βάρη h_i , $i \in \mathcal{I}_N$ ορίζονται ως:

$$h_i = 1 + \frac{\sigma_v(\bar{h}_i)}{1 + \sum_{j \in \mathcal{I}_N} \sigma_v(\bar{h}_j)} \quad (30)$$

με

$$\bar{h}_i = k_i \left(1 - (\tanh(\phi/w))^2\right)^2 \left(\frac{q_d - q}{\|q_d - q\|^2}\right)^T \frac{q_i - q}{\|q_i - q\|^2} \quad (31)$$

των οποίων ο στόχος είναι να επιταχύνουν την μείωση των k_i που συνεισφέρουν περισσότερο στις συνιστώσες του $\nabla_q \psi$ που ωθούν το ρομπότ προς το σύνορο.

Σχετικά με την δεύτερη απαίτηση, όπως φαίνεται από **Πρόταση 1**, επιλογή του k_d πάνω από ένα όριο αρκεί για να καταστήσει το ϕ απαλλαγμένο από εκφυλισμένα σημεία ισορροπίας. Από την άλλη πλευρά, καθώς το ρομπότ μπορεί να μην πλησιάσει ποτέ κάποιο εκφυλισμένο σημείο, αντί να θέσουμε το k_d επαρκώς μεγάλο εκ τω προτέρων, ο αντίστοιχος προσαρμοστικός νόμος αυξάνει το k_d μόνο όταν πραγματικά χρειάζεται.

Ανάλυση Ευστάθειας

Θεωρούμε το ακόλουθο σύστημα:

$$\dot{z} = f_z(z) \quad (32)$$

όπου $z = (q, k)$, $f_z(z) = (f_q, f_k)$ με $f_q = Ju$. Έστω Ω η εικόνα του \mathcal{W} μέσω του T , δλδ. $\Omega = T(\mathcal{W})$. Παρατηρούμε ότι Ω αποτελείται από $\text{int}(\mathcal{D})$, και ίσως από ένα μέρος του $\partial\mathcal{D}$, πλην των σημείων q_i . Στην συνέχεια, διατυπώνουμε τα χαρακτηριστικά ασφαλείας του συστήματος κλειστού βρόχου, το οποίο εγγυάται ότι το ρομπότ δεν θα συγκρουστεί με κάποιο εμπόδιο.

Πρόταση 3. Το ρομπότ θα παραμένει εντός του \mathcal{W} υπό την δυναμική (32) με νόμους ελέγχου (13) και (20), δλδ. $p(t) \in \mathcal{W}$, για κάθε $t \geq 0$.

Αφού αποκλείστηκε το ενδεχόμενο της σύγκρουσης με τα όρια του χώρου εργασίας, προχωρήσουμε με την απόδειξη ότι όλα τα κρίσιμα σημεία του ψ , όπου (32) μπορεί να συγκλίνει, είναι μη-εκφυλισμένα σαγματικά σημεία ή το q_d .

Πρόταση 4. Το δυναμικό ψ μειώνεται επί της τροχιάς του ρομπότ και η χρονική του παράγωγος μηδενίζεται μόνο στα κρίσιμα σημεία. Επιπλέον, η προ-εικόνα του q_d είναι ένα σύνολο ευσταθών σημείων ισορροπίας του (2).

Πρόταση 5. Έστω $z^* = (q^*, k^*)$ ένα κρίσιμο σημείο του συστήματος κλειστού βρόχου με $q^* \in \Omega \setminus \{q_d\}$. Τότε, το q^* είναι μη εκφυλισμένο σαγματικό σημείο του ψ .

Τέλος, κλείνουμε την συγκεκριμένη ενότητα με το κεντρικό θεωρητικό αποτέλεσμα.

Θεώρημα 2. Οι νόμοι ελέγχου (13) και (20) αποτελούν λύση του προβλήματος **Πρόβλημα 1**, δλδ. το σύστημα (2) υπό τον παραπάνω νόμο ελέγχου συγκλίνει ασφαλώς στο q_d , για σχεδόν κάθε αρχική συνθήκη.

Επεκτάσεις

Σε αυτή την ενότητα παρουσιάζουμε ορισμένες επεκτάσεις της προτεινόμενης μεθοδολογίας με σκοπό να αντιμετωπίσουμε: α) το πρόβλημα της ασφαλούς πλοήγησης για

δίτροχα ρομπότ και β) την υπολογιστική πολυπλοκότητα του μετασχηματισμού καθώς το μέγεθος του χώρου εργασίας μεγαλώνει.

Δίτροχα Ρομποτικά Οχήματα

Ξεκινάμε θεωρώντας ρομπότ του οποίου η κίνηση υπόκειται σε περιορισμούς της μορφής:

$$\begin{aligned}\dot{p} &= \hat{n}_\theta v \\ \dot{\theta} &= \omega\end{aligned}\quad (33)$$

όπου $\theta \in [0, 2\pi)$ δηλώνει τον προσανατολισμό του ρομπότ $\hat{n}_\theta = [\cos(\theta), \sin(\theta)]^T$, και $v, \omega \in \mathbb{R}$ είναι είσοδοι ελέγχου που αντιστοιχούν στην γραμμική και γωνιακή του ταχύτητα, αντίστοιχα. Πρώτα, ορίζουμε την κινηματική του ρομπότ στην εικόνα του χώρου διαμόρφωσης μέσω του προτεινόμενου μετασχηματισμού ως εξής:

$$\begin{aligned}\dot{q} &= \hat{n}_{\tilde{\theta}} \tilde{v} \\ \dot{\tilde{\theta}} &= \tilde{\omega}\end{aligned}\quad (34)$$

και παρατηρούμε ότι οι προσανατολισμοί θ και $\tilde{\theta}$ σχετίζονται από:

$$\hat{n}_{\tilde{\theta}} = \frac{J(p)\hat{n}_\theta}{\|J(p)\hat{n}_\theta\|}. \quad (35)$$

Ο νόμος έλεγχου ορίζεται ως εξής:

$$\begin{aligned}\tilde{v} &= -\mathcal{K}_v s_v(q, \tilde{\theta}, k) (\hat{n}_{\tilde{\theta}})^T \nabla_q \psi(q, k) \\ \tilde{\omega} &= -\mathcal{K}_\omega (\hat{n}_{\tilde{\theta}}^\perp)^T \nabla_q \psi(q, k)\end{aligned}\quad (36)$$

με $\mathcal{K}_v, \mathcal{K}_\omega \in \mathbb{R}$ σταθερά θετικά κέρδη, $\hat{n}_{\tilde{\theta}}^\perp = [-\sin(\theta), \cos(\theta)]^T$ και

$$\begin{aligned}s_v(q, \tilde{\theta}, k) &= \gamma \sigma_p \left(\frac{1 - \|q\|}{\epsilon_p} \right) + \\ & (1 - \gamma) \sigma_v \left(\frac{(\hat{n}_{\tilde{\theta}}^T \nabla_q \psi) \hat{n}_{\tilde{\theta}}^T q}{\epsilon_v + |\hat{n}_{\tilde{\theta}}^T \nabla_q \psi| \|q\|} \right).\end{aligned}\quad (37)$$

Επιπλέον, τροποποιούμε τους προσαρμοστικούς νόμους αντικαθιστώντας το s με s_v στις (20) ανδ (22) και \bar{g}_i , $i \in \mathcal{I}_N^*$, με

$$\begin{aligned}\bar{g}_{v,0} &= \frac{1}{4} \left(\alpha |\hat{n}_{\tilde{\theta}}^T \nabla_q \psi| \|q - q_d\| - (\hat{n}_{\tilde{\theta}}^T \nabla_q \psi) \hat{n}_{\tilde{\theta}}^T (q - q_d) \right) \\ \bar{g}_{v,i} &= \frac{1}{2} (\hat{n}_{\tilde{\theta}}^T \nabla_q \psi) \hat{n}_{\tilde{\theta}}^T (q - q_i), \quad \forall i \in \mathcal{I}_N\end{aligned}\quad (38)$$

αντίστοιχα, στην (23). Τέλος, εκφράζοντας τον παραπάνω νόμο ελέγχου στον πραγματικό χώρο, προκύπτει:

$$\begin{aligned} v &= v\tilde{v} \\ \omega &= \omega_{dq} + \omega_{d\bar{\theta}} \end{aligned} \quad (39)$$

όπου ω_{dq} και $\omega_{d\bar{\theta}}$ είναι όροι που αντιστοιχούν σε γωνιακές ταχύτητες λόγω γραμμικής και περιστροφικής κίνησης στην εικόνα του χώρου εργασίας:

$$\begin{aligned} \omega_{dq} &= -\tilde{v}^2 \left((J [\hat{n}_\theta \quad v\hat{n}_\theta^\perp])^{-1} \frac{\partial}{\partial \hat{n}_\theta} J \hat{n}_\theta \right)^T \begin{bmatrix} 0 \\ 1 \end{bmatrix} \\ \omega_{d\bar{\theta}} &= \tilde{\omega} \left((J [\hat{n}_\theta \quad v\hat{n}_\theta^\perp])^{-1} \hat{n}_\theta^\perp \right)^T \begin{bmatrix} 0 \\ 1 \end{bmatrix} \end{aligned} \quad (40)$$

με $v = \|J^{-1}\hat{n}_\theta\|$ και το $\frac{\partial}{\partial \hat{n}_\theta} J$ να δηλώνει την παράγωγο του J κατά την κατεύθυνση του \hat{n}_θ .

Για την ευστάθεια του παραπάνω συστήματος κλειστού βρόχου ισχύει το εξής:

Θεώρημα 3. Το ρομπότ θα παραμείνει εντός του \mathcal{W} υπό την δυναμική (33) εφοδιασμένο με τον παραπάνω νόμο ελέγχου. Επιπλέον, το ρομπότ θα συγκλίνει ασυμπτωτικά είτε σε κάποιο εσωτερικό κρίσιμο σημείο του ϕ είτε στην προ-εικόνα του q_d , η οποία είναι ευσταθής.

Σχόλιο 2. Το αποτέλεσμα του θεωρήματος **Θεώρημα 3** είναι λιγότερο ισχυρό σε σχέση με αυτό του **Θεώρημα 2** καθώς δεν εξασφαλίζει ότι το σύνολο των καταστάσεων που συγκλίνουν σε κάποιο κρίσιμο σημείο του ϕ (εκτός του q_d) έχει *Lebesgue* μηδενικό μέτρο.

Άτλας Αρμονικών Μετασχηματισμών

Καθώς το μέγεθος του χώρου εργασίας αυξάνεται, το πρόβλημα του υπολογισμού του μετασχηματισμού T αυξάνει σε πολυπλοκότητα, επειδή οι πόροι που απαιτούνται από αριθμητικές τεχνικές είναι πολυωνυμικοί ως προς τον αριθμό των στοιχείων που χρησιμοποιούνται για την αναπαράσταση \mathcal{W} . Εναλλακτικά, για να αντιμετωπίσουμε μεγάλου χώρους εργασίας, προτείνουμε την κατασκευή ενός άτλα $\mathcal{A} \triangleq \{(\mathcal{P}_i, T_i) \mid i \in \mathcal{I}_{N_A}\}$ που προκύπτει χωρίζοντας το \mathcal{W} σε N_A αλληλοεπικαλυπτόμενα υποσύνολα $\mathcal{P}_i \subset \mathcal{W}$, τέτοια ώστε $\bigcup_{i \in \mathcal{I}_{N_A}} \mathcal{P}_i = \mathcal{W}$ και κατασκευάζοντας έναν ξεχωριστό αρμονικό μετασχηματισμό T_i για κάθε \mathcal{P}_i . Αυτό μας επιτρέπει ουσιαστικά να λύσουμε πολλά μικρά (και υπολογιστικά λιγότερο απαιτητικά) προβλήματα αντί ενός μεγάλου, μειώνοντας έτσι τους συνολικούς πόρους που απαιτούνται για την αντιμετώπιση ενός δεδομένου χώρου εργασίας. Οπότε, δεδομένου ενός κατακερματισμού του χώρου \mathcal{W} , ορίζουμε τον γράφο $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ όπου $\mathcal{V} = \{\mathcal{P}_i \mid i \in \mathcal{I}_{N_A}\}$ δηλώνει το σύνολο των κόμβων (διαμερίσματα του χώρου εργασίας) και \mathcal{E} δηλώνει το σύνολο των ακμών μεταξύ στοιχείων του \mathcal{V} , με κάθε ακμή να αντιστοιχεί σε γειτονικά διαμερίσματα, δλδ. $(i, j) \in \mathcal{E}$ αν και μόνο αν $(\text{cl}(\mathcal{P}_i) \cap \text{cl}(\mathcal{P}_j)) \neq \emptyset$. Παρατηρούμε ότι για ένα δεδομένο άτλα \mathcal{A} , αρχική κατάσταση p_{init} και τελική κατάσταση p_d , μπορούμε να χρησιμοποιήσουμε ένα καθιερωμένο αλγόριθμου προσπέλασης γράφου

για την εύρεση μιας αλληλουχίας $\mathcal{S} = \{s_1, s_2, \dots, s_n\}$ που αντιστοιχούν σε διαμερίσματα που μπορεί το ρομπότ να προσπελάσει για να φτάσει στον στόχο του.

Τώρα θα επικεντρωθούμε στην μετάβαση μεταξύ δύο διαδοχικών κόμβων. Έστω $\mathcal{C}_{i,j} \triangleq \text{cl}(\mathcal{P}_i) \cap \text{cl}(\mathcal{P}_j)$ η κοινή περιοχή μεταξύ $\text{cl}(\mathcal{P}_i)$ και $\text{cl}(\mathcal{P}_j)$ και $\mathcal{B}_{i,j} \triangleq \partial\mathcal{P}_i \cap \mathcal{P}_j$ το σύνολο των σημείων του συνόρου του \mathcal{P}_i τα οποία ανήκουν στον \mathcal{P}_j και δεν καταλαμβάνονται από εμπόδια, για κάθε $i \in \mathcal{I}_{N_A}$ και κάθε j τέτοια ώστε $(i, j) \in \mathcal{E}$. Χωρίς βλάβη της γενικότητας, υποθέτουμε ότι ο \mathcal{A} κατασκευάζεται ώστε τα σύνολα $\mathcal{B}_{\ell,i} \cap \mathcal{B}_{\ell,j}$ είναι είτε κενά είτε αποτελούνται από μεμονωμένα σημεία. Σημειώνουμε ότι ώστε να ολοκληρωθεί επιτυχώς η μετάβαση μεταξύ δύο διαδοχικών κόμβων \mathcal{P}_i και \mathcal{P}_j , αρκεί το ρομπότ να φτάσει οποιοδήποτε σημείο του $\mathcal{B}_{i,j}$. Επίσης, κάθε $\mathcal{B}_{i,j}$ μπορεί να αποτελείται από ένα ή περισσότερα ξεχωριστά τμήματα $\mathcal{B}_{i,j}^\ell$, $\ell \in \mathcal{L}(i, j)$, με $\mathcal{L}(i, j)$ να δηλώνει κάποια αρίθμηση. Εκμεταλλευόμενοι το γεγονός ότι **Θεώρημα 1** απαιτεί έναν ασθενή ομοιομορφισμό του \bar{T}_i , μπορούμε να κατασκευάσουμε κάθε T_i έτσι ώστε τα τμήματα $\partial\mathcal{P}_i$ να καταρρεύσουν σε ξεχωριστά σημεία, δλδ. $T_i(\mathcal{B}_{i,j}^\ell) = q_{i,j}^\ell \in \partial\mathcal{D}$, το οποίο σημαίνει ότι επιλέγοντας $q_{i,j}^\ell$ ως ενδιαμέσο στόχο αρκεί να καταστήσει ολόκληρο το $\mathcal{B}_{i,j}^\ell$ ελκτικό. Οπότε, για κάθε ζεύγος \mathcal{P}_i και \mathcal{P}_j , επιλέγουμε (αυθαιρέτως) ένα $\mathcal{B}_{i,j}^\ell$ και κατασκευάζουμε έναν μετασχηματισμό $T_i : \mathcal{P}_i \mapsto \mathcal{D}$, με $q^{[i]} = T_i(p)$, και ένα πεδίο $\phi_i(q^{[i]}, k^{[i]})$ με προορισμό $q_d^{[i]} = q_{i,j}^\ell$. Επιπλέον, για να εξομαλύνουμε την μετάβαση, όταν υπάρχει αλληλοεπικάλυψη, προτείνουμε τον παρακάτω νόμο ελέγχου:

$$u = u^{[i]} + \eta_{c,i,j} \cdot \eta_{t,i,j} \cdot u^{[j]}, \quad \forall p \in \mathcal{C}_{i,j} \quad (41)$$

όπου $u^{[i]}$ και $u^{[j]}$ δηλώνουν εισόδους ελέγχου όπως ορίστηκαν στην (13) και υπολογίζονται χρησιμοποιώντας τα ψ_i, T_i και ψ_j, T_j , αντίστοιχα, $\eta_{t,i,j} : \mathcal{C}_{i,j} \mapsto [0, 1]$ είναι μία ομαλή συνάρτηση τέτοια ώστε

$$\eta_{t,i,j}(p) = \begin{cases} 0, & \text{εάν } p \in \mathcal{B}_{j,i} \\ 1, & \text{εάν } p \in \mathcal{B}_{i,j} \end{cases} \quad (42)$$

και

$$\eta_{c,i,j}(p, k^{[i]}, k^{[j]}) = \begin{cases} \frac{(\zeta_{i,j})^2}{\epsilon_4 + (\zeta_{i,j})^2}, & \text{εάν } \zeta_{i,j} \geq 0 \\ 0, & \text{εάν } \zeta_{i,j} < 0 \end{cases} \quad (43)$$

με $\zeta_{i,j} = (\nabla_p \phi_i)^T \cdot (\nabla_p \psi_j)$ και $\epsilon_4 > 0$ σταθερά. Αυτό που επιτυγχάνει η εν λόγω τροποποίηση είναι να εισάγει μια επιπλέον συνιστώσα επί της διεύθυνσης του $\nabla_p \psi_j$, στην ταχύτητα του ρομπότ όταν αυτή η συνιστώσα είναι ομόροπη με $\nabla_p \psi_i$. Ο αλγόριθμος του εν λόγω σχήματος ελέγχου φαίνεται στο **Αλγόριθμος 1**.

Αναφορικά με την ευστάθεια του τροποποιημένου συστήματος, είναι εύκολο κάποιος να επιβεβαιώσει την παρακάτω πρόταση:

Θεώρημα 4. Το σύστημα (2) με τον νόμο ελέγχου **Αλγόριθμος 1** συγκλίνει στο επιθυμητό προορισμό $p_d \in \mathcal{W}$ για σχεδόν κάθε αρχική συνθήκη $p_{\text{init}} \in \mathcal{W}$.

Αποτελέσματα

Για να δείξουμε την αποτελεσματικότητα του προτεινόμενου νόμου ελέγχου, παρουσιάζουμε τα αποτελέσματα προσομοιώσεων και πειραμάτων. Ο αλγόριθμος υπολογισμού του μετασχηματισμού υλοποιήθηκε σε C++ ενώ ο προτεινόμενος νόμος ελέγχου σε Python.

Προσομοιώσεις — Πλήρης Μετασχηματισμός

Σε αυτό το σενάριο κατασκευάστηκε μετασχηματισμός ολόκληρου του χώρου εργασίας (Εικόνα 2.4) και στο ρομπότ δόθηκε εντολή να πλοηγηθεί σε διαφορετικούς επιθυμητούς προορισμούς ξεκινώντας κάθε φορά από την ίδια αρχική θέση. Η αρχική διαμόρφωση και οι παράμετροι του ελεγκτή μας επιλέχθηκαν έτσι ώστε να επιδείξουν όσον το δυνατόν καλύτερα τις εγγυημένες ιδιότητες αποφυγής σύγκρουσης του συστήματός μας. Πιο συγκεκριμένα, οι αρχικές τιμές των παραμέτρων επιλέχθηκαν ως εξής: $k_d = 20$, $k_i = 1$ ανδ $\bar{k}_i = 20$ για κάθε $i \in \mathcal{I}_{10}$. Οι τιμές των υπόλοιπων παραμέτρων επιλέχθηκαν αντίστοιχα: $K_u = 100$, $w = 20$, $K_k = 100$, $\alpha = 1$, $\epsilon_p = 0.025$, $\epsilon_v = 0.1$, $\gamma = 0.7$, $\epsilon_1 = 0.01$, $\epsilon_2 = 0.1$, $\epsilon_3 = 0.1$. Οι προορισμοί και οι τροχιές που εκτέλεσε το ρομπότ, τόσο στον πραγματικό όσο και στον μετασχηματισμένο χώρο εργασίας, παρουσιάζονται στην Εικόνα 2.5. Επιπλέον, η Εικόνα 2.6 δείχνει την εξέλιξη του δυναμικού ψ για κάθε τροχιά. Οι προσομοιώσεις διεξήχθησαν χρησιμοποιώντας ολοκλήρωση κατά Euler με βήμα $10ms$. Όσον αφορά την υπολογιστική πολυπλοκότητα, η κατασκευή του αρμονικού μετασχηματισμού για αυτό το μεγάλο χώρο εργασίας χρειάστηκε $5.4s$ για να ολοκληρωθεί, δεδομένης μιας επαρκούς προσέγγισης του συνόρου από 3680 στοιχεία. Τέλος, ο υπολογισμός του μετασχηματισμού $T(p)$ και της Ιακωβιανής του $J(p)$ χρειαζόταν κατά μέσο όρο περίπου $6.0ms$ ανά βήμα.

Προσομοιώσεις — Άτλας Αρμονικών Μετασχηματισμών

Σε αυτή τη μελέτη περίπτωσης, χωρίσαμε τον προηγούμενο χώρο εργασίας σε ξεχωριστά δωμάτια (Εικόνα 2.4) και κατασκευάσαμε ένα αρμονικό μετασχηματισμός T_i για κάθε δωμάτιο. Το ρομπότ αρχικοποιήθηκε στην ίδια θέση με την προηγούμενη μελέτη και είχε εντολή να πλοηγηθεί προς το ίδιο σύνολο επιθυμητών προορισμών. Οι αρχικές τιμές των παραμέτρων ήταν $k^{[i]} = N^{[i]} + 3$, $k_j^{[i]} = 1$ και $\bar{k}_j^{[i]} = k^{[i]}$ για κάθε $j \in \mathcal{I}_{N^{[i]}}$ και $i \in \mathcal{I}_{N_A}$, όπου $N^{[i]}$ δηλώνει τον αριθμό των εμποδίων σε κάθε διαμέρισμα. Οι τιμές των υπόλοιπων παραμέτρων ήταν ίδιες με την προηγούμενη μελέτη. Οι τροχιές του ρομπότ φαίνονται στην Εικόνα 2.7. Η διάρκεια κατασκευής των μετασχηματισμών κυμαινόταν από $0.019s$ έως $0.211s$ (ανάλογα με το πλήθος των στοιχείων για να προσεγγιστεί επαρκώς κάθε δωμάτιο, το οποίο ήταν από 320 έως 1000 τμήματα) και ήταν αρκετά μικρότερη από αυτή της κατασκευής του ενιαίου χάρτη. Επιπλέον, ο υπολογισμός των $T_i(p)$ και $J_i(p)$ σε κάθε δωμάτιο ήταν χρειαζόταν από $1.0ms$ έως $2.2ms$ ανά βήμα, αντίστοιχα. Τέλος, πρέπει να σημειωθεί ότι σε αυτή την περίπτωση τα εσωτερικά εμπόδια του χώρου εργασίας απεικονίστηκαν σε σημεία πιο μακριά από τα σύνορα, το οποίο είναι ένα ενδιαφέρον αποτέλεσμα,

καθώς ελαττώνει πιθανά προβλήματα αριθμητικής φύσεως που μπορεί να προκύψουν κατά τον υπολογισμό του μετασχηματισμού κοντά στα εμπόδια.

Πειραματικά Αποτελέσματα

Για να επιβεβαιώσουμε τα αποτελέσματα της επέκτασης για μη-ολονομικά ρομπότ, πραγματοποιήσαμε πειράματα με την ρομποτική πλατφόρμα “Robotnik Summit-XL” (Εικόνα 2.8) η οποία λειτούργησε εντός του χώρου που φαίνεται στην Εικόνα 2.9. Τα σύνορα του χώρου εργασίας κατασκευάστηκαν άμεσα διαθέσιμους αλγόριθμους SLAM και επαυξήθηκαν με το σχήμα του ρομπότ (το οποίο προσεγγίστηκε από δίσκο). Ο χώρος εργασίας χωρίστηκε σε έξι επικαλυπτόμενα υποσύνολα και το ρομπότ είχε εντολή να επισκεφθείτε τρεις διαφορετικές θέσεις. Ένας έτοιμος αλγόριθμος χρησιμοποιήθηκε για εκτίμηση θέσεως και προσανατολισμού του ρομπότ χρησιμοποιώντας αισθητήρες επί της πλατφόρμας, ο οποίος ανανέωνε την εκτίμηση ανά περίπου 5Hz. Παρατηρούμε ότι ο αλγόριθμος μας κατάφερε με επιτυχία να οδηγήσει το ρομπότ με ασφάλεια στους προορισμούς του, όπως μπορεί κανείς να διαπιστώσει από τις τροχιές του ρομπότ στον πραγματικό χώρο (Εικόνα 2.10, Εικόνα 2.12, Εικόνα 2.14), αλλά και στην εικόνα αυτού (Εικόνα 2.11, Εικόνα 2.13, Εικόνα 2.15). Από την άλλη, παρατηρούμε μια ταλαντωτική συμπεριφορά στον μετασχηματισμένο χώρο — ειδικά στα διαμερίσματα $p1$ και $p2$ — η οποία πιθανότατα οφείλεται σε: α) την σχετικά αργή ανατροφοδότηση, και β) και την αντιστροφή της Ιακωβιανής του μετασχηματισμού, η οποία είναι σχεδόν ιδιόμορφη κοντά σε στενά περάσματα. Παρ’ όλα αυτά, τέτοια φαινόμενα μπορεί να μετριασθούν με καλύτερη επιλογή των κατατιμήσεων.

Πλοήγηση Πολυγωνικών Ρομπότ στο Επίπεδο

Εισαγωγή

Σε αυτό το κεφάλαιο, παρουσιάζουμε ένα υβριδικό σύστημα ελέγχου με σκοπό την αντιμετώπιση του προβλήματος πλοήγησης για μια ρομποτική πλατφόρμα αυθαίρετου σχήματος που κινείται μέσα σε ένα χώρο εργασίας με εσωτερικά εμπόδια. Δεδομένης μια αρχική και επιθυμητή διαμόρφωσης του ρομπότ, προτείνουμε μια μεθοδολογία που βασίζεται σε μεθόδους προσεγγιστικής αποσύνθεσης του χώρου σε κελιά που χρησιμοποιεί ευριστικούς κανόνες για να βελτιώσει προσαρμοστικά την διαμέριση του χώρου διαμόρφωσης σε μη-επικαλυπτόμενες, διαδοχικά τμήματα. Επιπλέον, χρησιμοποιούμε κατάλληλους μετασχηματισμούς του χώρου εργασίας και προσαρμοστικά δυναμικά πεδία για τον σχεδιασμό νόμων ελέγχου που ενσωματώνονται κομψά με το είδος της αναπαράστασης του χώρου διαμόρφωσης που χρησιμοποιείται, με στόχο την ασφαλή πλοήγηση μέσα σε δεδομένο κελί, για όλες σχεδόν τις αρχικές ρυθμίσεις, έως ότου επιτευχθεί η επιθυμητή διαμόρφωση. Τέλος, παρουσιάζουμε αποτελέσματα προσομοίωσης που επιδεικνύουν την αποτελεσματικότητα του προτεινόμενου συστήματος ελέγχου.

Σημειογραφία: Δεδομένου συνόλου \mathcal{S} , έστω $\partial\mathcal{S}$ το σύνορο του. Επίσης ορίζουμε \mathcal{D} και $\partial\mathcal{D}$ ως τον δίσκο και κύκλο με μοναδιαία ακτίνα με κέντρο την αρχή των αξόνων του \mathbb{R}^2 , αντίστοιχα. Δεδομένων συνόλων A και B , χρησιμοποιούμε $\text{cl}(A)$, $\text{int}(A)$, A/B και $A \ominus B$ για να δηλώσουμε την κλειστότητα του A , το εσωτερικό του A , το υποσύνολο του B που δεν ανήκει στο A και την Minkowski διαφορά του A από το B , αντίστοιχα. Τέλος, δεδομένης διανυσματικής συνάρτησης $f(x)$ χρησιμοποιούμε $\nabla_x f$ για να δηλώσουμε την Ιακωβιανή του.

Ορισμός του Προβλήματος

Θεωρούμε ένα ρομπότ που κινείται μέσα σε έναν συμπαγή χώρο $\mathcal{W} \subset \mathbb{R}^2$ που καταλαμβάνεται από σταθερά εμπόδια \mathcal{O}_i , $i \in \mathcal{I}_o \triangleq \{1, 2, \dots, N_o\}$. Υποθέτουμε ότι τα όρια του ρομπότ $\partial\mathcal{R}$, του χώρου εργασίας $\partial\mathcal{W}_0$ και των στατικών εμποδίων $\partial\mathcal{W}_i$ είναι πολυγωνικές καμπύλες Jordan, όπου $\partial\mathcal{W}_i \triangleq \partial\mathcal{O}_i, \forall i \in \mathcal{I}_o$. Επιπλέον, υποθέτουμε ότι το ρομπότ \mathcal{R} είναι μια κλειστή, 2-πολλαπλότητα του \mathbb{R}^2 , και ότι ο χώρος εργασίας \mathcal{W} είναι ένα κλειστό υποσύνολο του \mathbb{R}^2 . Θα χρησιμοποιήσουμε τα $\mathcal{F}_\mathcal{W}$ και $\mathcal{F}_\mathcal{R}$ για να δηλώσουμε δύο αυθαίρετως τοποθετημένα πλαίσια συντεταγμένων εντός του χώρου εργασίας και του

ρομπότ, αντίστοιχα, και θα αναφερόμαστε στην αρχή των αξόνων του \mathcal{F}_R ως το σημείο αναφοράς του ρομπότ. Έστω $z = [x, y, \theta]^T$ η διαμόρφωση του ρομπότ, που αποτελείται από την σχετική θέση $p = [x, y]^T \in \mathbb{R}^2$ και τον προσανατολισμό $n = [\cos \theta, \sin \theta]^T$, $\theta \in [0, 2\pi)$, του \mathcal{F}_R ως προς το \mathcal{F}_W . Η κίνηση του ρομπότ χαρακτηρίζεται από το μοντέλο ενός απλού ολοκληρωτή ως ακολούθως:

$$\begin{aligned}\dot{p} &= u \\ \dot{\theta} &= \omega\end{aligned}\tag{44}$$

όπου $u \in \mathbb{R}^2$ είναι η είσοδος ελέγχου της γραμμικής ταχύτητας και $\omega \in \mathbb{S}^1$ είναι η επιθυμητή γωνιακή ταχύτητα.

Ο χώρος διαμόρφωσης \mathcal{CS} του ρομπότ είναι μία πολλαπλότητα που μπορεί να απεικονιστεί στο $\mathbb{R}^2 \times \mathbb{S}^1$ και κάθε εμπόδιο \mathcal{O}_i αντιστοιχεί σε μία κλειστή περιοχή \mathcal{O}_i^c μέσα σε αυτόν που δίδεται από $\mathcal{O}_i^c \triangleq \{z | \mathcal{R}(z) \cap \mathcal{O}_i \neq \emptyset\}$, όπου $\mathcal{R}(z)$ δηλώνει το υποσύνολο του \mathcal{W} που καταλαμβάνεται από το ρομπότ στην διαμόρφωση z (για λόγους συντομίας, θα χρησιμοποιούμε $\mathcal{R}(\theta)$ αντί για $\mathcal{R}([0, 0, \theta])$). Ως εκ τούτου, ο ελεύθερος χώρος διαμόρφωσης $\mathcal{CS}_{\text{free}}$ του ρομπότ ορίζεται ως $\mathcal{CS}_{\text{free}} \triangleq \mathcal{CS} / \cup_{i \in \mathcal{I}_o} \mathcal{O}_i^c$.

Ο στόχος μας είναι ο σχεδιασμός σχήματος ελέγχου που μπορεί επιτυχώς να οδηγήσει το ρομπότ σε μια δεδομένη διαμόρφωση $z_{\text{final}} \in \mathcal{CS}_{\text{free}}$ από σχεδόν οποιαδήποτε αρχική κατάσταση $z_{\text{init}} \in \mathcal{CS}_{\text{free}}$, ενώ εξασφαλίζει ότι η προκύπτουσα τροχιά είναι ασφαλής, *i.e.*, $p(t) \in \mathcal{CS}_{\text{free}}$ για κάθε $t \geq 0$.

Σχήμα Ελέγχου

Για την αντιμετώπιση του παραπάνω προβλήματος, σχεδιάζουμε ένα υβριδικό σύστημα ελέγχου, το οποίο αποτελείται από: α) έναν ελεγκτή υψηλού επιπέδου που δεδομένης μίας αρχικής (z_{init}) και τελικής (z_{final}) διαμόρφωσης, μπορεί να υπολογίσει μια ακολουθία ενδιάμεσων στόχων για το ρομπότ, αν υπάρχει λύση στο παραπάνω πρόβλημα, ή να συμπεράνει ανυπαρξία αυτής, και β) έναν ελεγκτή χαμηλού επιπέδου ο οποίος χρησιμοποιεί αρμονικά τεχνητά πεδία και κατάλληλους μετασχηματισμούς του χώρου εργασίας για να οδηγήσει με ασφάλεια το ρομπότ διαδοχικά σε κάθε ένα από τους ενδιάμεσους στόχους. Πιο συγκεκριμένα, ο ελεγκτής υψηλού επιπέδου κατασκευάζει μία διαμέριση του $\mathcal{CS}_{\text{free}}$ μέσω προσαρμοστικής υποδιαίρεσης του \mathbb{S}^1 μέχρι να βρεθεί μια αλληλουχία Π από συνδεδεμένα κελιά που περιέχει τις z_{init} και z_{final} (εάν υπάρχει). Στην συνέχεια, για κάθε κελί σε Π , επιθυμητές θέσεις p και προσανατολισμοί θ υπολογίζονται ξεχωριστά και σχεδιάζονται κατάλληλα ζεύγη από ανεξάρτητους ελεγκτές χαμηλού επιπέδου.

Αποσύνθεση του Χώρου Διαμόρφωσης

Σε αυτή την ενότητα, παρέχουμε μια λεπτομερή περιγραφή της προτεινόμενης μεθοδολογίας για την κατασκευή μια ιεραρχικής προσέγγισης του χώρου διαμόρφωσης του ρομπότ.

Πιο συγκεκριμένα, προτείνουμε έναν αλγόριθμο για την διαμέριση του $\mathcal{CS}_{\text{free}}$ σε μη επικαλυπτόμενα τμήματα τα οποία προκύπτουν με προσαρμοστική υποδιαίρεση του συνόλου των δυνατών προσανατολισμών του ρομπότ. Ένα τμήμα \mathcal{S}^I που αντιστοιχεί στο διάστημα $\mathcal{I} = [a, b)$ γωνιών ορίζεται ως $\mathcal{S}^I \triangleq \{(x, y, \theta) | \theta \in \mathcal{I}\} \cap \mathcal{CS}_{\text{free}}$. Γενικά, \mathcal{S}^I αποτελείται από κανένα η περισσότερα μη-συνδεδεμένα σύνολα διαμορφώσεων με αυθαίρετη συνδεσιμότητα, τα οποία θα αναφέρονται στο εξής ως κελιά. Ως εκ τούτου, συσχετίζουμε κάθε τμήμα \mathcal{S}^I με μία αρίθμηση \mathcal{K}^I των κελιών του \mathcal{C}_k^I και θα χρησιμοποιούμε το $\mathcal{L}(\mathcal{S}^I)$ για να δηλώσουμε το σύνολο αυτών, δλδ. $\mathcal{C}_k^I \in \mathcal{L}(\mathcal{S}^I), \forall k \in \mathcal{K}^I$.

Δύο τμήματα \mathcal{S}^I και \mathcal{S}^J θα λέγονται γειτονικά εάν $\text{cl}(\mathcal{I}) \cap \text{cl}(\mathcal{J}) \neq \emptyset$. Επιπλέον, θα λέμε ότι δύο κελιά \mathcal{C}^I και \mathcal{C}^J , που ανήκουν σε γειτονικά τμήματα, είναι συνδεδεμένα εάν $\text{cl}(\mathcal{C}^I) \cap \text{cl}(\mathcal{C}^J) \neq \emptyset$. Για να αποφύγουμε τις επιπτώσεις που εισάγει η γεωμετρική πολυπλοκότητα του \mathcal{CS} , θα κατασκευάσουμε και θα χρησιμοποιήσουμε δύο επιπλέον, απλούστερες προσεγγίσεις του ελεύθερου χώρου διαμόρφωσης, δηλαδή μία υπερ-προσέγγιση $\overline{\mathcal{CS}}_{\text{free}}$ και μία υπό-προσέγγιση $\underline{\mathcal{CS}}_{\text{free}}$, οι οποίες κατασκευάζονται παρομοίως για τμήματα και κελιά και ο σκοπός του καθενός θα γίνει εμφανής παρακάτω.

Η μέθοδος που σχεδιάζεται σε αυτή την εργασία για την κατασκευή των εν λόγω προσεγγίσεων χρειάζεται την ύπαρξη δύο συναρτήσεων, \bar{f} και \underline{f} , για το υπολογισμό μίας υπερ-προσέγγισης $\overline{\mathcal{S}^I}$ και μίας υπό-προσέγγισης $\underline{\mathcal{S}^I}$ του ελεύθερου χώρου δεδομένου τμήματος \mathcal{S}^I . Οι παραπάνω συναρτήσεις είναι τέτοιες ώστε $\underline{\mathcal{S}^I} \subseteq \mathcal{S}^I \subseteq \overline{\mathcal{S}^I}$, $\bar{f}(\mathcal{S}^I) \triangleq \overline{\mathcal{S}^I} \rightarrow \mathcal{S}^I$ και $\underline{f}(\mathcal{S}^I) \triangleq \underline{\mathcal{S}^I} \rightarrow \mathcal{S}^I$ καθώς το μήκος του διαστήματος \mathcal{I} πηγαίνει στο μηδέν. Μια απλή προσέγγιση για την κατασκευή αυτών των συνόλων είναι υπολογίζοντας κατάλληλες προσεγγίσεις $\overline{\mathcal{W}^I}$ και $\underline{\mathcal{W}^I}$ της προβολής του ελεύθερου χώρου του \mathcal{S}^I πάνω στο xy -επίπεδο. Πιο συγκεκριμένα, τα σύνολα $\overline{\mathcal{S}^I}$ και $\underline{\mathcal{S}^I}$ δίνονται

$$\begin{aligned}\overline{\mathcal{S}^I} &= \{[x, y, \theta]^T | [x, y]^T \in \overline{\mathcal{W}^I}, \theta \in \mathcal{I}\} \\ \underline{\mathcal{S}^I} &= \{[x, y, \theta]^T | [x, y]^T \in \underline{\mathcal{W}^I}, \theta \in \mathcal{I}\}\end{aligned}\quad (45)$$

Για να υπολογιστούν τα $\overline{\mathcal{W}^I}$ και $\underline{\mathcal{W}^I}$, διάφοροι αλγόριθμοι μπορούν να χρησιμοποιηθούν, όπως η Μέθοδος Προβολής [122] είτε η Μέθοδος Σάρωσης [122]. Σε αυτό το κεφάλαιο, η δεύτερη προτιμήθηκε λόγω απλότητά και υπολογιστικής απόδοσης σε σχέση με την πρώτη. Ως εκ τούτου, τα $\overline{\mathcal{W}^I}$ και $\underline{\mathcal{W}^I}$ μπορούν να υπολογιστούν από

$$\begin{aligned}\overline{\mathcal{W}^I} &= \mathcal{W} \ominus \underline{\mathcal{R}}(\mathcal{I}), \\ \underline{\mathcal{W}^I} &= \mathcal{W} \ominus \overline{\mathcal{R}}(\mathcal{I}),\end{aligned}\quad (46)$$

όπου $\overline{\mathcal{R}}(\mathcal{I})$, $\underline{\mathcal{R}}(\mathcal{I})$ είναι σύνολα που δίνονται από

$$\begin{aligned}\overline{\mathcal{R}}(\mathcal{I}) &= \{p | \exists \theta \in \mathcal{I} : p \in \mathcal{R}(\theta)\}, \\ \underline{\mathcal{R}}(\mathcal{I}) &= \{p | \forall \theta \in \mathcal{I} : p \in \mathcal{R}(\theta)\},\end{aligned}\quad (47)$$

που αντιπροσωπεύουν τα υποσύνολα του \mathbb{R}^2 τα οποία το ρομπότ καταλαμβάνει είτε για τουλάχιστον μία είτε για κάθε προσανατολισμό εντός του \mathcal{I} , αντίστοιχα. (δες [Εικόνα 3.2](#)).

Οι έννοιες της γειτνίασης και συνδεσιμότητας που ορίστηκαν ανωτέρω για ακριβή τμήματα μπορούν εύκολα να επεκταθούν για τις υπερ- και υπο-προσεγγίσεις τους.

Σε αυτό το σημείο, πρέπει να σημειωθεί ότι καθώς ο χώρος διαμόρφωσης υποδιαιρείται σε όλο και περισσότερα τμήματα οι προσεγγίσεις $\overline{\mathcal{CS}}_{\text{free}}$ και $\underline{\mathcal{CS}}_{\text{free}}$ συγκλίνουν στο $\mathcal{CS}_{\text{free}}$, το οποίο διασφαλίζει την πληρότητα αυτής της προσέγγισης. Έτσι, υποδιαιρώντας επαρκώς τον $\mathcal{CS}_{\text{free}}$, θα βρεθεί μία αλληλουχία από κελιά υπο-προσεγγίσης που συνδέουν δύο δεδομένες διαμορφώσεις, εάν υπάρχει εξ αρχής. Αντί τυφλής υποδιαίρεσης του του $\mathcal{CS}_{\text{free}}$, προτείνουμε έναν εναλλακτικό αλγόριθμο, με βάση την προσέγγιση που παρουσιάζεται στο [122], η οποία κάνει επίσης χρήση της υπερ-προσεγγίσης του χώρου για να παρέχει έναν ευριστικό κανόνα για την επιλογή του επόμενου τμήματος προς υποδιαίρεση. Επιπροσθέτως, είναι επίσης σε θέση να αναγνωρίσει αν μια τέτοια αλληλουχία υπάρχει.

Έστω \mathcal{P} σύνολο που αποτελείται από N διαστήματα γωνιών $\mathcal{I}_1, \mathcal{I}_2, \dots, \mathcal{I}_N$. Θα λέμε ότι \mathcal{P} είναι μία διαμέριση του $\mathcal{CS}_{\text{free}}$ εάν $\mathcal{S}^{\mathcal{I}_1}, \mathcal{S}^{\mathcal{I}_2}, \dots, \mathcal{S}^{\mathcal{I}_N}$ είναι μη-επικαλυπτόμενα και $\mathcal{CS}_{\text{free}} = \cup_{k=1}^N \mathcal{S}^{\mathcal{I}_k}$. Επιπλέον, δεδομένης διαμέρισης \mathcal{P} , ένα μονοπάτι Π από κελιά ορίζεται ως μια πεπερασμένη αλληλουχία κελιών τέτοια ώστε $\mathcal{I}_k \in \mathcal{P}$ και τα κελιά κάθε διαδοχικού ζεύγους $(\mathcal{C}_k^{\mathcal{I}_k}, \mathcal{C}_{k+1}^{\mathcal{I}_{k+1}})$ είναι συνδεδεμένα για κάθε $k = 1, 2, \dots, N$. Είναι προφανές από τους ορισμούς των $\overline{\mathcal{S}}$ και $\underline{\mathcal{S}}$ ότι, για δεδομένα διαμερίσματα \mathcal{P} του \mathcal{CS} και ένα ζεύγος αρχικής z_{init} και τελικής z_{final} διαμόρφωσης, οι ακόλουθες προτάσεις ισχύουν:

1. Αν υπάρχει ένα μονοπάτι από υπο-προσεγγιστικά κελιά του \mathcal{P} που περιέχουν αυτές τις διαμορφώσεις, τότε υπάρχει λύση στο πρόβλημά.
2. Αν υπάρχει ένα μονοπάτι από υπερ-προσεγγιστικά κελιά του \mathcal{P} που περιέχουν αυτές τις δύο διαμορφώσεις, τότε ίσως υπάρχει λύση στο πρόβλημά μας.
3. Αν δεν υπάρχει μονοπάτι από υπερ-προσεγγιστικά κελιά του \mathcal{P} που περιέχουν αυτές τις δύο διαμορφώσεις, τότε το πρόβλημά μας δεν λύνεται.

Τώρα μπορούμε να παρουσιάσουμε τον **Αλγόριθμος 2** τον οποίο χρησιμοποιούμε για να βρούμε ένα μονοπάτι από υπο-προσεγγιστικά κελιά που συνδέουν δεδομένη αρχική και τελική διαμόρφωση του ρομπότ. Ξεκινάμε προσεγγίζοντας τους $\mathcal{CS}_{\text{free}}$, $\overline{\mathcal{CS}}_{\text{free}}$, $\underline{\mathcal{CS}}_{\text{free}}$ από τμήματα που καλύπτουν ολόκληρο το \mathbb{S}^1 . Μετά, ψάχνουμε για κελιά $\underline{\mathcal{C}}_{\text{init}}^{\mathcal{I}}$ και $\underline{\mathcal{C}}_{\text{final}}^{\mathcal{J}}$ του $\underline{\mathcal{CS}}_{\text{free}}$, πιθανότατα υποδιαιρώντας το \mathcal{P} , τα οποία περιέχουν τις δεδομένες διαμορφώσεις. Αν δεν υπάρχει τέτοιο ζεύγος, το οποίο υποδεικνύεται από το ότι είτε το $\overline{\mathcal{C}}_{\text{init}}^{\mathcal{I}}$ είτε το $\overline{\mathcal{C}}_{\text{final}}^{\mathcal{J}}$ είναι κενό, το αρχικό μας πρόβλημα δεν λύνεται. Διαφορετικά, προσπαθούμε να συνδέσουμε τα $\underline{\mathcal{C}}_{\text{init}}^{\mathcal{I}}$ και $\underline{\mathcal{C}}_{\text{final}}^{\mathcal{J}}$ χρησιμοποιώντας υπο-προσεγγιστικά κελιά της \mathcal{P} . Εάν αυτή η προσπάθεια αποτύχει, τότε προσπαθούμε να βρούμε ένα μονοπάτι από υπερ-προσεγγιστικά κελιά. Εάν ούτε ένα τέτοιο μονοπάτι υπάρχει, τότε το πρόβλημά μας δεν έχει λύση. Διαφορετικά, ένας ευριστικός κανόνας χρησιμοποιείται για την επιλογή του διαστήματος της διαμέρισης \mathcal{P} προς υποδιαίρεση και η διαδικασία ξεκινάει από την αρχή.

Ο ευριστικός κανόνας, που μπορεί κανείς να δει στον [Αλγόριθμος 3](#), επιλέγει πιο διάστημα της \mathcal{P} θα υποδιαιρείται προσπατώντας να κατασκευάσει ένα νέο μονοπάτι από κελιά που ανήκουν στο $\underline{\mathcal{CS}}_{\text{free}}$ αλλά και στα κελιά του $\overline{\mathcal{CS}}_{\text{free}}$ της αλληλουχίας που του δόθηκε ως όρισμα. Μετά την απομόνωση γειτονικών τμημάτων που περιέχουν κύτταρα που δεν μπορούν να συνδεθούν, το μεγαλύτερο αντίστοιχο διάστημα παίρνει υποδιαιρείται και η συνάρτηση επιστρέφει.

Υβριδικό Σχήμα Ελέγχου

Δεδομένου μονοπατιού Π , αυτή η υποενότητα περιγράφει πως να συνθέσουμε μία αλληλουχία από ελεγχτές χαμηλού επιπέδου οι οποίοι μπορούν ασφαλώς να οδηγήσουν το ρομπότ, από ένα κελί στο επόμενο, ξεκινώντας από την διαμόρφωση z_{init} , μέχρι να φτάσει την επιθυμητή διαμόρφωση z_{final} . Έστω \mathcal{C}^I κελί του Π και έστω $\mathcal{P}(\mathcal{C}^I)$ η προβολή του \mathcal{C}^I στο επίπεδο. Παρατηρούμε ότι $\mathcal{P}(\overline{\mathcal{C}^I}) = \overline{\mathcal{W}^I}$ και $\mathcal{P}(\underline{\mathcal{C}^I}) = \underline{\mathcal{W}^I}$. καθώς και, εκ κατασκευής, το \mathcal{C}^I είναι μια υπό-προσέγγιση του ελεύθερου χώρου διαμόρφωσης τέτοια ώστε

$$\mathcal{C}^I = \{[x, y, \theta]^T \mid [x, y]^T \in \mathcal{P}(\mathcal{C}^I) \text{ ανδ } \theta \in \mathcal{I}\}. \quad (48)$$

Αυτό συνεπάγεται ότι $\mathcal{P}(\mathcal{C}^I)$ είναι μη-κενή, συμπαγής περιοχή του \mathbb{R}^2 με αυθαίρετη συνδεσιμότητα και ότι η θέση και ο προσανατολισμός του ρομπότ μπορεί να αλλάξει ανεξάρτητα, εφόσον και οι δύο παραμένουν εντός του \mathcal{C}^I , χωρίς να διακυβεύεται η ασφάλεια του ρομπότ.

Εκμεταλλούμενοι αυτήν την ιδιότητα, μπορούμε να απεμπλέξουμε τους νόμους ελέγχου για την γραμμική και γωνιακή ταχύτητα του ρομπότ. Ως εκ τούτου, για κάθε ενδιάμεσο κελί Π , αρκεί να βρούμε ανεξαρτήτως επιθυμητές περιοχές θέσης και επιθυμητά εύρη γωνιών που εξασφαλίζουν μετάβαση στο επόμενο κελί. Ας θεωρήσουμε τώρα ένα ζεύγος διαδοχικών κελιών \mathcal{C}^I και \mathcal{C}^J του Π . Αναφορικά με την θέση του ρομπότ, για να ολοκληρωθεί επιτυχώς η μετάβαση από το \mathcal{C}^I στο \mathcal{C}^J , είναι αρκετό η θέση p να φτάσει το σύνολο $\mathcal{G}(\mathcal{C}^I)$ που ορίζεται ως $\mathcal{G}(\mathcal{C}^I) \triangleq \mathcal{P}(\mathcal{C}^I) \cap \mathcal{P}(\mathcal{C}^J)$, και είναι μη-κενό εκ κατασκευής. Επιπλέον, το $\mathcal{G}(\mathcal{C}^I)$ γενικά αποτελείται από ένα ή περισσότερα μη-συνδεδεμένα υποσύνολα. Επομένως, εφ'όσον φτάσει κάποια από αυτές τις περιοχές, το ρομπότ μπορεί να διασχίσει στο επόμενο κελί.

Έστω $\mathcal{F}(\mathcal{C}^I)$ όλες οι διαμορφώσεις του \mathcal{C}^I οι οποίες δεν ανήκουν στον σύνολο-στόχο $\mathcal{G}(\mathcal{C}^I)$, δλδ. $\mathcal{F}(\mathcal{C}^I) \triangleq \mathcal{P}(\mathcal{C}^I) \setminus \mathcal{P}(\mathcal{C}^J)$. Όταν $p \in \mathcal{F}(\mathcal{C}^I)$, πρέπει να σχεδιάσουμε κατάλληλο σχήμα ελέγχου u που μπορεί να οδηγήσει την θέση p στο υποσύνολο του ορίου του $\mathcal{G}(\mathcal{C}^I)$. Έστω $\mathcal{F}_C(\mathcal{C}^I)$ και $\mathcal{G}_p(\mathcal{C}^I)$ τα συνδεδεμένα μέρη του $\mathcal{F}(\mathcal{C}^I)$ που περιέχουν το p και το επιλεγμένο σύνολο-στόχος, αντίστοιχα. Θεωρούμε τις παρακάτω περιπτώσεις:

1. $\mathcal{G}_p(\mathcal{C}^I)$ είναι ένα σημείο του \mathcal{C}^I .
2. $\mathcal{G}_p(\mathcal{C}^I)$ είναι ένα εσωτερικό όριο του $\mathcal{F}_C(\mathcal{C}^I)$.
3. $\mathcal{G}_p(\mathcal{C}^I)$ είναι ένα συνεχές, μετρήσιμο υποσύνολο του εξωτερικού ορίου του $\mathcal{F}_C(\mathcal{C}^I)$.

4. $\mathcal{G}_p(\mathcal{C}^I)$ είναι ίσο με το εξωτερικό όριο του $\mathcal{F}_C(\mathcal{C}^I)$.

Όσον αφορά το σύνολο-στόχο που αντιστοιχεί στον προσανατολισμό του ρομπότ, σημειώνουμε ότι ταυτίζεται με $\text{cl}(\mathcal{I}) \cap \text{cl}(\mathcal{J})$. Δεδομένου συνδυασμού $\mathcal{F}_C(\mathcal{C}^I)$ και $\mathcal{G}_p(\mathcal{C}^I)$, μπορούμε να σχεδιάσουμε έναν νόμο ελέγχου $f_p(p, k)$ για την θέση και έναν απλό αναλογικό ελεγκτή $f_\theta(\theta)$ για τον προσανατολισμό, με εγγυημένες ιδιότητες ασφάλειας και σύγκλισης. Τέλος, όταν \mathcal{C}^I είναι το τελευταίο κελί του Π , οι παραπάνω νόμοι ελέγχου f_p και f_θ χρησιμοποιούνται για την οδήγηση του ρομπότ στην διαμόρφωση $z_{\text{final}} = (p_d, \theta_d)$ θέτοντας απλά $\mathcal{G}_p(\mathcal{C}^I) = \{p_d\}$ και $\mathcal{G}_\theta(\mathcal{C}^I) = \{\theta_d\}$.

Αρμονικές Συναρτήσεις Πλοήγησης και Μετασχηματισμοί

Στην ενότητα αυτή παρέχουμε μια σύντομη επισκόπηση του μετασχηματισμού και των προσαρμοστικών αρμονικών πεδίων τα οποία παρουσιάστηκαν πρώτα στο [113] και χρησιμοποιούνται στο σχεδιασμό ελεγκτών για την γραμμική ταχύτητα του ρομπότ.

Έστω \mathcal{F} ένα δεδομένο συμπαγές, πολλαπλώς συνδεδεμένος χώρος που οριοθετείται από $N_o^{\mathcal{F}} + 1$ καμπύλες Jordan, δηλαδή $\partial\mathcal{F}_0, \partial\mathcal{F}_1, \dots, \partial\mathcal{F}_{N_o^{\mathcal{F}}}$, όπου $\partial\mathcal{F}_0$ είναι το εξωτερικό σύνορο του \mathcal{F} . Ένας αρμονικός μετασχηματισμός $T(p) = [\mathbf{u}(p), \mathbf{v}(p)]^T$ (δλδ., $\mathbf{u}(p)$ και $\mathbf{v}(p)$ είναι αρμονικές συναρτήσεις), κατασκευάζεται για το \mathcal{F} , ικανοποιώντας τις παρακάτω ιδιότητες: α) απεικονίζει το εξωτερικό όριο $\partial\mathcal{F}_0$ του \mathcal{F} στον μοναδιαίο δίσκο $\partial\mathcal{D}$, β) απεικονίζει όλα τα εσωτερικά όρια $\partial\mathcal{F}_i$ σε ξεχωριστά σημεία $q_i = [\mathbf{u}_i, \mathbf{v}_i]^T \in \text{int}(\partial\mathcal{D})$, και γ) είναι διφεομορφισμός για κάθε $p \in \text{int}\mathcal{F}$. Ο υπολογισμός του T απαιτεί έναν ομοιομορφισμό που διατηρεί τον προσανατολισμό $\bar{T} : \partial\mathcal{F}_0 \mapsto \partial\mathcal{D}$ που μπορεί να κατασκευαστεί εύκολα. Έτσι, το αντίστοιχο πρόβλημα συντομικών συνθηκών μπορεί να λυθεί, π.χ., χρησιμοποιώντας αριθμητικές τεχνικές όπως Πεπερασμένα Στοιχεία ή Συνοριακά Στοιχεία, για την κατασκευή των $\mathbf{u}(p)$ και $\mathbf{v}(p)$.

Σε αυτή την υποενότητα, τέτοιοι μετασχηματισμοί χρησιμοποιούνται για την απεικόνιση κάθε υποσυνόλου $\mathcal{F}_C(\mathcal{C}^I)$ του κελιού \mathcal{C}^I . Έτσι, ανάλογα με την περίπτωση του $\mathcal{G}_p(\mathcal{C}^I)$, η εν λόγω μεθοδολογία πρέπει να προσαρμοστεί κατάλληλα. Πιο συγκεκριμένα, εάν $\mathcal{G}_p(\mathcal{C}^I)$ είναι ένα συνεχές υποσύνολο του εξωτερικού ορίου του $\mathcal{F}_C(\mathcal{C}^I)$, τότε το \bar{T} πρέπει να επιλεγεί έτσι ώστε όλες οι θέσεις $p \in \mathcal{G}_p(\mathcal{C}^I)$ να απεικονιστούν στο ίδιο σημείο q_d στο σύνορο. Όταν το $\mathcal{G}_p(\mathcal{C}^I)$ ταυτίζεται με ολόκληρο το εξωτερικό όριο του $\mathcal{F}_C(\mathcal{C}^I)$, ένας επιπλέον διφεομορφισμός T^* που απεικονίζει τον μοναδιαίο κύκλο στο κέντρο του και ανάποδα πρέπει να εφαρμοστεί στην εικόνα του T . Έτσι, η σύνθεση $T' = T^* \circ T$ πρέπει να χρησιμοποιηθεί αντί του T για την απεικόνιση του $\mathcal{F}_C(\mathcal{C}^I)$ στο \mathcal{D} . Σε αυτήν την περίπτωση, ο στόχος q_d απλά τοποθετείται στο κέντρο του \mathcal{D} . Μια απλή επιλογή για το T^* είναι:

$$T^*(x) = \frac{x}{c + \|x\|} \circ \frac{x}{\|x\|^2} \circ \frac{x}{1 - \|x\|} \quad (49)$$

όπου c θετική σταθερά. Τέλος, αναφορικά με την περίπτωση όπου το $\mathcal{G}_p(\mathcal{C}^I)$ ταυτίζεται με κάποιο εσωτερικό όριο του $\mathcal{F}_C(\mathcal{C}^I)$, δεν χρειάζεται επιπλέον δράση καθώς ο στόχος τοποθετείται απλώς πάνω στην εικόνα του εν λόγω συνόρου.

Δεδομένου τώρα του μετασχηματισμού (T ή T'), ορίζουμε ένα αρμονικό δυναμικό πεδίο ϕ τοποθετώντας πηγές στον στόχο q_d και εμπόδια $q_i = T(\partial\mathcal{F}_i)$, που ορίζεται από

$$\phi = k_d \cdot \ln\left(\frac{\|q - q_d\|}{2}\right) - \sum_{i=1}^{N_o^{\mathcal{F}}} k_i \cdot \ln\left(\frac{\|q - q_i\|}{2}\right) \quad (50)$$

όπου $k_d > 0$ και $k_i \geq 0$ είναι παράμετροι που προσαρμόζονται από κατάλληλο νόμο ελέγχου. Στην συνέχεια ορίζουμε ένα πεπερασμένο πεδίο αναφοράς ψ βάσει του ϕ , που δίνεται από

$$\psi = \frac{1 + \tanh(\omega \cdot \phi)}{2}, \quad (51)$$

όπου ω θετική σταθερά. Η γραμμική ταχύτητα του ρομπότ επιλέγεται ως:

$$u = f_p(p, k) \triangleq -\mathcal{K}_v \cdot s_\ell(q, k) \cdot J^{-1}(p) \cdot \nabla_q \psi(q, k) \quad (52)$$

όπου $q = T(p)$, \mathcal{K}_v είναι θετική σταθερά, το $J(p)$ δηλώνει την Ιακωβίνου του T , το ψ είναι το δυναμικό πεδίο εφοδιασμένο με τον προσαρμοστικό νόμο $\dot{k} = f_k(p, k)$ για τις παραμέτρους του $k = [k_d, k_1, k_2, \dots, k_{N_{\mathcal{F}_C(\mathcal{C}^I)}}]^T$, όπου $N_{\mathcal{F}_C(\mathcal{C}^I)}$ είναι το πλήθος των οπών $\mathcal{F}_C(\mathcal{C}^I)$, και $s_\ell(p, k) \geq 0$ είναι μία συνεχώς διαχωρίσιμη συνάρτηση που δίνεται από τον τύπο

$$s_\ell(q, k) = (1 - \|q\|) + \left(\max\left((\nabla_q \psi)^T \cdot q, 0\right)\right)^2. \quad (53)$$

Επιπλέον, ο προσαρμοστικός νόμος των παραμέτρων του ψ υπολογίζεται από

$$\begin{aligned} \dot{k}_d &= H_{e_1}(\lambda) \\ \dot{k}_i &= (\bar{k}_i - k_i) W_{i c_i} h_i - k_i (W_0 h_0 + H_{e_2}(s_\ell)), \forall i \in N_o^{\mathcal{F}} \end{aligned} \quad (54)$$

και ο σκοπός του είναι να εγγυηθεί το αναλλοίωτο του \mathcal{F} χωρίς να αλλοιώσει την ευστάθεια.

Τέλος, αναφορικά με τον προσανατολισμό του ρομπότ, χρησιμοποιούμε ένα απλό αναλογικό νόμο ελέγχου της μορφής

$$\omega = f_\theta(\theta) \triangleq -\mathcal{K}_\omega \cdot d_s(\theta, \theta_d) \quad (55)$$

όπου $d_s(\cdot, \cdot)$ είναι μια συνεχώς παραγωγίσιμη συνάρτηση που δίνει την μικρότερη προσημασμένη απόσταση μεταξύ δύο δεδομένων γωνιών και θ_d είναι η μέση τιμή του διαστήματος $\mathcal{G}_\theta(\mathcal{C}^I)$.

Προσομοιώσεις

Στην ενότητα αυτή παρουσιάζονται τα αποτελέσματα της προσομοίωσης που επιδεικνύει την αποτελεσματικότητα της μεθοδολογίας αυτού του κεφαλαίου. Πιο συγκεκριμένα, θεωρούμε ένα σενάριο που περιλαμβάνει ρομπότ σχήματος 'Υ' το οποίο πρέπει να πλοηγηθεί σε ένα πολύπλοκο χώρο εργασίας. ο προτεινόμενος αλγόριθμος χρησιμοποιήθηκε

για την απόκτηση ενός εφικτού μονοπατιού από κελιά που συνδέουν την αρχική και την επιθυμητή διαμόρφωση του ρομπότ. Συνολικά, 30 τμήματα προέκυψαν από την επίλυση του προβλήματος, που αντιστοιχούν σε διαστήματα γωνίας με μήκος από π έως $\pi/32$. Το προφίλ κίνησης του ρομπότ φαίνεται στην **Εικόνα 3.5**, που επιβεβαιώνει την σύγκλιση και ασφάλεια του σχήματος ελέγχου. Επιπλέον, η εξέλιξη της διαμόρφωσης z του ρομπότ καθώς και οι χρονικές στιγμές ολοκλήρωσης των μεταβάσεων μεταξύ γειτονικών κελιών φαίνονται στην **Εικόνα 3.6**. Τέλος, μερικά τυπικά παραδείγματα κελιών που παρήχθησαν από τον αλγόριθμο φαίνονται στην **Εικόνα 3.7**.

Συντονισμός Κίνησης Πολλών Δισκοειδών Ρομπότ

Εισαγωγή

Σε αυτό το κεφάλαιο, παρουσιάζουμε ένα υβριδικό σχήμα ελέγχου για να λύσουμε το πρόβλημα της πλοήγησης μιας ομάδας από δισκοειδή ρομπότ που λειτουργούν σε χώρο με εμπόδια. Δεδομένων μιας αρχικής και μιας επιθυμητής διαμόρφωσης του συστήματος, σχεδιάζουμε μια ιεραρχική μεθοδολογία αποσύνθεσης σε κελιά εφοδιασμένη με ένα κατάλληλο μηχανισμό σήμανσης που είναι σε θέση να καθορίσει ποιες περιοχές του χώρου διαμόρφωσης πρέπει να υποδιαιρευθούν περαιτέρω σε κάθε επανάληψη, αποφεύγοντας έτσι περιττές υποδιαιρέσεις. Ιδιαίτερα, υπολογίζοντας κατάλληλα υπερ- και υπο-προσεγγίσεις του αποτυπώματος κάθε ρομπότ, ο αλγόριθμος μας μπορεί να καθορίσει ποια κελιά μπορούν να περιέχουν εφικτές διαμορφώσεις του συστήματος, ενώ αυτόματα απορρίπτει κελιά που είναι σίγουρο πως δεν περιέχουν κανένα. Επιπλέον, δεδομένης μιας λύσης του συγκεκριμένου προβλήματος από τον εν λόγω αλγόριθμο υψηλού επιπέδου, χρησιμοποιούμε αρμονικούς μετασχηματισμούς και δυναμικά πεδία για να πραγματοποιήσουμε ασφαλείς μεταβάσεις μεταξύ γειτονικών κελιών, εξασφαλίζοντας έτσι σχεδόν καθολική σύγκλιση στην επιθυμητή διαμόρφωση. Παρατηρούμε ότι, καθώς το κατανομημένο σχήμα ελέγχου χαμηλού επιπέδου δεν χρειάζεται ρητή γνώση της τρέχουσας θέσης κάθε ρομπότ προκειμένου να υπολογίσει τις εισόδους ελέγχου, απαιτεί πολύ χαμηλούς πόρους στην επικοινωνία, αφού τα ρομπότ χρειάζεται να ανταλλάσσουν πληροφορία μόνο όταν ολοκληρώνουν επιτυχώς τη μετάβαση από το ένα κελί στο επόμενο. Τέλος, παρουσιάζουμε αποτελέσματα προσομοίωσης που επιδεικνύουν την αποτελεσματικότητα του προτεινόμενου σχήματος ελέγχου.

Σημειογραφία: Σε αυτό το κεφάλαιο, θα χρησιμοποιούμε το $\mathcal{I}_N \triangleq \{1, 2, \dots, N\}$ (αντίστοιχα, το $\mathcal{I}_N^* \triangleq \{0\} \cup \mathcal{I}_N$) για να δηλώσουμε το σύνολο που αποτελείται από όλους τους φυσικούς αριθμούς ξεκινώντας από το 1 (αντίστοιχα, το 0). Επιπλέον, δεδομένων συνόλων A και B , χρησιμοποιούμε τα ∂A , $\text{int}(A)$, $\text{cl}(A)$ για να δηλώσουμε το σύνορο, εσωτερικό και κλειστότητα A , αντίστοιχα, και το $A \setminus B$ για να δηλώσουμε το συμπλήρωμα του B ως προς το A .

Ορισμός του Προβλήματος

Θεωρούμε μία ομάδα από $N_{\mathcal{R}}$ ρομπότ που λειτουργούν εντός ενός συμπαγούς επίπεδου χώρου εργασίας $\mathcal{W} \subseteq \mathbb{R}^2$ που καταλαμβάνεται από ένα σύνολο N_0 σταθερών εσωτερικών εμποδίων $\mathcal{O}_i, i \in \mathcal{I}_{N_0}$. Υποθέτουμε ότι κάθε ρομπότ i έχει ένα δισκοειδές σώμα $\mathcal{R}_i \subset \mathbb{R}^2$ με ακτίνα $r_i > 0$. Έστω \mathcal{F}_w και $\mathcal{F}_i, i \in \mathcal{I}_{N_{\mathcal{R}}}$ συστήματα συντεταγμένων εντός των \mathcal{W} και $\mathcal{R}_i, i \in \mathcal{I}_{N_{\mathcal{R}}}$, αντίστοιχα. Θα αναφερόμαστε στην αρχή των αξόνων κάθε $\mathcal{F}_i, i \in \mathcal{I}_{N_{\mathcal{R}}}$ ως το σημείο αναφοράς του αντίστοιχου ρομπότ. Επιπλέον, χωρίς βλάβη της γενικότητας, υποθέτουμε ότι το σημείο αναφοράς του κάθε ρομπότ συμπίπτει με το κέντρο του σώματος του. Έστω $p_i \triangleq [x_i, y_i]^T \in \mathbb{R}^2$ η σχετική θέση του σημείου αναφοράς του i ρομπότ ως προς το \mathcal{F}_w , και έστω $\mathcal{R}_i(p)$ το αποτύπωμά του, δλδ. η περιοχή που καταλαμβάνει το \mathcal{R}_i όταν είναι τοποθετημένο στην θέση p . Σε αυτό το κεφάλαιο, θα χρησιμοποιούμε το $\mathcal{C} \subseteq \mathbb{R}^{2N_{\mathcal{R}}}$ για να δηλώσουμε τον χώρο διαμόρφωσης του ρομποτικού συστήματος και το $P \triangleq [p_1^T, p_2^T, \dots, p_{N_{\mathcal{R}}}^T]^T \in \mathcal{C}$ για το διάνυσμα θέσεων αυτών. Για συντομία, θα χρησιμοποιήσουμε επίσης το $P[i]$ για να δηλώσουμε την i συνιστώσα του P , δλδ. $P[i] = p_i$. Έστω \mathcal{W}^o το συμπλήρωμα του \mathcal{W} , δλδ. $\mathcal{W}^o \triangleq \mathbb{R}^2 \setminus \mathcal{W}$. Ορίζουμε επίσης μια διαμόρφωση P ως εφικτή εάν οι παρακάτω συνθήκες ισχύουν:

$$\begin{aligned} \mathcal{R}_i(P[i]) \cap \mathcal{R}_j(P[j]) &= \emptyset, \quad \forall i \neq j \in \mathcal{I}_{N_{\mathcal{R}}} \\ \mathcal{R}_i(P[i]) \cap \mathcal{W}^o &= \emptyset, \quad \forall i \in \mathcal{I}_{N_{\mathcal{R}}} \end{aligned} \quad (56)$$

και θα χρησιμοποιούμε το $\mathcal{C}^f \subset \mathcal{C}$ για να δηλώσουμε το σύνολο όλων των εφικτών διαμορφώσεων του ρομποτικού συστήματος, ενώ το συμπλήρωμα $\mathcal{C}^o \triangleq \mathcal{C} \setminus \mathcal{C}^f$ αυτού αντιστοιχεί στο σύνολο όλων των μη-εφικτών διαμορφώσεων. Επίσης, υποθέτουμε ότι η κίνηση κάθε ρομπότ i υπόκειται στο μοντέλο απλού ολοκληρωτή:

$$\dot{p}_i = u_i, \quad i \in \mathcal{I}_{N_{\mathcal{R}}} \quad (57)$$

όπου u_i είναι η είσοδος ελέγχου.

Έστω P_{init} και P_{des} δύο δεδομένες εφικτές διαμορφώσεις του πολυ-ρομποτικού συστήματος. Ο στόχος μας είναι ο σχεδιασμός σχήματος ελέγχου το οποίο θα οδηγήσει κάθε ρομπότ i , αρχικοποιημένο στην θέση $p_{\text{init},i} = P_{\text{init}}[i]$, στην επιθυμητική θέση $p_{\text{des},i} = P_{\text{des}}[i]$, ενώ παράλληλα αποφεύγει συγχρούσεις ρομπότ-ρομπότ και ρομπότ-χώρου εργασίας, δλδ, $P(t) \in \mathcal{C}^f$ για κάθε $t \geq 0$.

Σχεδιασμός Σχήματος Ελέγχου

Για την αντιμετώπιση του παραπάνω προβλήματος, πρώτα χρησιμοποιούμε ένα ιεραρχικό σχήμα αποσύνθεσης σε κελιά για την κατακερματισμό του χώρου διαμόρφωσης \mathcal{C} σε κελιά. Στην συνέχεια, σχεδιάζουμε ένα σχήμα υψηλού επιπέδου, το οποίο επεκτείνει την προαναφερθείσα δομή μέχρι να βρεθεί μια αλληλουχία συνδεδεμένων κελιών που οδηγούν από τη P_{init} στη P_{des} . Τέλος, χρησιμοποιούμε ένα σχήμα ελέγχου χαμηλού επιπέδου για να εξασφαλίσουμε την ασφαλή μετάβαση μεταξύ κελιών μέχρι.

Αποσύνθεση του Χώρου Διαμόρφωσης

Σε αυτή την υποενότητα, παρουσιάζουμε το ιεραρχικό σύστημα αποσύνθεσης του χώρου διαμόρφωσης που θα χρησιμοποιηθούν στην προσέγγισή μας. Ξεκινάμε παραβλέποντας προσωρινά συγκρούσεις μεταξύ ρομπότ και λαμβάνοντας υπόψη το χώρο διαμόρφωσης του κάθε μεμονωμένου ρομπότ. Πιο συγκεκριμένα, ο χώρος διαμόρφωσης του ρομπότ i , που συμβολίζεται με $\mathcal{A}_i(\mathcal{W})$, αντιστοιχεί στο μεγαλύτερο υποσύνολο του \mathcal{W} όπου το σημείο αναφοράς του ρομπότ i μπορεί να τοποθετηθεί ώστε $\mathcal{R}_i(p_i) \cap \mathcal{W}^o = \emptyset$, για κάθε $p_i \in \mathcal{A}_i(\mathcal{W})$. Επίσης, δεδομένου υποσυνόλου \mathcal{Z} του \mathcal{W} , θα χρησιμοποιούμε το $\mathcal{A}_i(\mathcal{Z})$ για να δηλώσουμε το σύνολο των εφικτών θέσεων του ρομπότ i που ανήκουν στο \mathcal{Z} , δηλ.:

$$\mathcal{A}_i(\mathcal{Z}) \triangleq \{ p \mid p \in \mathcal{Z} \text{ ανδ } \mathcal{R}_i(p) \cap \mathcal{W}^o = \emptyset \}, \quad \forall i \in \mathcal{I}_{N_{\mathcal{R}}}. \quad (58)$$

Επιπλέον, θεωρούμε δύο εκτιμήσεις της επιφάνειας που πιθανώς καταλαμβάνεται από το \mathcal{R}_i όταν η θέση p_i είναι περιορισμένη σε ένα υποσύνολο \mathcal{Z} του \mathcal{W} . Πιο συγκεκριμένα, δεδομένων ρομπότ $i \in \mathcal{I}_{N_{\mathcal{R}}}$ και συνόλου $\mathcal{Z} \subseteq \mathcal{A}_i(\mathcal{W})$, έστω $\overline{\mathcal{R}}_i(\mathcal{Z})$ και $\underline{\mathcal{R}}_i(\mathcal{Z})$ υπερ- και υπό-προσέγγιση, αντίστοιχα, του αποτυπώματος του \mathcal{R}_i όταν το ρομπότ i σαρώνει το σύνολο \mathcal{Z} τέτοια ώστε:

$$\begin{aligned} \overline{\mathcal{R}}_i(\mathcal{Z}) &\supseteq \bigcup_{p \in \mathcal{Z}} \mathcal{R}_i(p) \\ \underline{\mathcal{R}}_i(\mathcal{Z}) &\subseteq \bigcap_{p \in \mathcal{Z}} \mathcal{R}_i(p) \end{aligned} \quad (59)$$

και

$$\begin{aligned} \overline{\mathcal{R}}_i(\mathcal{Z}) &\subseteq \overline{\mathcal{R}}_i(\mathcal{Z}'), \quad \forall \mathcal{Z} \subseteq \mathcal{Z}' \\ \underline{\mathcal{R}}_i(\mathcal{Z}) &\supseteq \underline{\mathcal{R}}_i(\mathcal{Z}'), \quad \forall \mathcal{Z} \subseteq \mathcal{Z}'. \end{aligned} \quad (60)$$

Ένα τέτοιο παράδειγμα φαίνεται στην **Εικόνα 4.2**.

Θεωρούμε τώρα σύνολο $\mathcal{S} \subseteq \mathbb{R}^2$ μορφής $[x_1, x_2] \times [y_1, y_2]$. Θα αναφερόμαστε σε τέτοια σύνολα ως απλά τμήματα του \mathbb{R}^2 . Δεδομένου απλού τμήματος \mathcal{S} και ρομπότ $i \in \mathcal{I}_{N_{\mathcal{R}}}$, θα χρησιμοποιούμε το $\mathcal{W}_{\mathcal{S}}^i \triangleq \mathcal{A}_i(\mathcal{W}) \cap \mathcal{S}$ για να δηλώσουμε το σύνολο των εφικτών θέσεων του ρομπότ i (αμελώντας συγκρούσεις μεταξύ ρομπότ) που περιέχονται εντός του \mathcal{S} . Ένα σύνολο $\mathfrak{S} = \{ \mathcal{S}_i \mid i \in \mathcal{I}_{N_{\mathcal{S}}} \}$ από $N_{\mathcal{S}}$ απλά τμήματα θα λέγεται κάλυμμα του \mathcal{W} εάν

$$\mathcal{W} = \bigcup_{j \in \mathcal{I}_{N_{\mathcal{S}}}} \mathcal{S}_j \cap \mathcal{W}. \quad (61)$$

Παρατηρούμε ότι ένα κάλυμμα \mathfrak{S} χωρίζει το $\mathcal{A}_i(\mathcal{W})$ σε ένα σύνολο από περιοχές $\mathcal{W}_{\mathcal{S}_i}^i$, $\mathcal{S} \in \mathfrak{S}$ με κάθε μία από αυτές να αποτελείται από κανένα ή περισσότερα μη επικαλυπτόμενα υποσύνολα $\mathcal{C}_{\mathcal{S}_i, j}^i$, $j \in \mathcal{I}_{N_{\mathcal{S}}(\mathcal{W}_{\mathcal{S}_i}^i)}$, τα οποία θα αποκαλούμε απλά κελιά. Ένα κάλυμμα $\widehat{\mathfrak{S}} = \{ (i, \mathcal{S}_i) \mid i \in \mathcal{I}_{N_{\mathcal{R}}} \}$ του \mathcal{C} ορίζεται αντιστοιχίζοντας ένα απλό κάλυμμα σε κάθε ρομπότ. Ένα σύνθετο τμήμα $\widehat{\mathcal{S}}$ ορίζεται ως $\widehat{\mathcal{S}} = \{ (i, \mathcal{S}_i) \mid i \in \mathcal{I}_{N_{\mathcal{R}}} \}$, όπου \mathcal{S}_i , $i \in \mathcal{I}_{N_{\mathcal{R}}}$ είναι ένα σύνολο από απλά τμήματα. Παρομοίως, ένα κάλυμμα $\widehat{\mathfrak{S}} = \{ (i, \mathcal{S}_i) \mid i \in \mathcal{I}_{N_{\mathcal{R}}} \}$ του χώρου διαμόρφωσης επιφέρει μία διαμέριση του \mathcal{C} σε περιοχές $\widehat{\mathcal{W}}_{\widehat{\mathfrak{S}}} \triangleq \mathcal{W}_{\mathcal{S}_1}^1 \times \mathcal{W}_{\mathcal{S}_2}^2 \times \dots \times \mathcal{W}_{\mathcal{S}_{N_{\mathcal{R}}}}^{N_{\mathcal{R}}}$, όπου $\widehat{\mathcal{S}} = \{ (i, \mathcal{S}_i) \mid i \in \mathcal{I}_{N_{\mathcal{R}}} \}$ είναι στοιχείο του $\widehat{\mathfrak{S}}$.

Παρατηρούμε ότι κάθε μία από αυτές τις περιοχές μπορεί να αποτελείται από κανένα ή περισσότερα μη επικαλυπτόμενα υποσύνολα $\widehat{\mathcal{C}}_{\mathcal{S}_i}$, $i \in \mathcal{I}_{N_{\mathcal{E}}}$, στα οποία θα αναφερόμαστε εδώ ως σύνθετα κελιά. Δεδομένου σύνθετου κελιού $\widehat{\mathcal{C}}_{\mathcal{S}_i}$, θα χρησιμοποιούμε το $\widehat{\mathcal{C}}_{\mathcal{S}_i}^{[j]}$ για να δηλώσουμε την j - συνιστώσα του, δηλ. $\widehat{\mathcal{C}}_{\mathcal{S}_i}^{[j]} \triangleq \mathcal{C}_{\mathcal{S}_i}^j$, για κάθε $j \in \mathcal{I}_{N_{\mathcal{R}}}$. Σημειώνουμε ότι, σε αντίθεση με σχήματα ιεραρχικής αποσύνθεσης που συνήθως συναντώνται στη βιβλιογραφία, τα οποία χρησιμοποιούν κελιά από απλές γεωμετρίες (π.χ., υπερκύβοι ή υπερορθογώνια), τα κελιά που εξετάζονται στο παρόν κεφάλαιο έχουν γενικώς αυθαίρετη γεωμετρία. Αν και αυτή η επιλογή περιπλέκει την πλοήγηση μέσα σε αυτά, επιφέρει γενικά πιο χονδρή διαμέριση καθώς κάθε στοιχείο $\widehat{\mathcal{C}}^{[i]}$ του $\widehat{\mathcal{C}}$ ανήκει στο $\mathcal{A}_i(\mathcal{W})$ εκ κατασκευής.

Όσον αφορά τώρα τη μετάβαση από κελί σε κελί, εισάγουμε τώρα κάποιες απαραίτητες έννοιες συνδεσιμότητας. Ξεκινάμε θεωρώντας δύο ξεχωριστά απλά τμήματα \mathcal{S}_i και \mathcal{S}_j τα οποία θα αποκαλούμε γειτονικά εάν η τομή τους $\mathcal{S}_i \cap \mathcal{S}_j$ είναι μη-κενή. Επιπλέον, έστω $\mathcal{C}_{\mathcal{S}_m,i}$ και $\mathcal{C}_{\mathcal{S}_n,j}$ δύο ξεχωριστά απλά κελιά. Θα λέμε ότι αυτά τα κελιά είναι γειτονικά εάν $\mathcal{C}_{\mathcal{S}_m,i} \cap \mathcal{C}_{\mathcal{S}_n,j} \neq \emptyset$. Προφανώς, γειτνίαση των $\mathcal{C}_{\mathcal{S}_m,i}$ και $\mathcal{C}_{\mathcal{S}_n,j}$ συνεπάγεται ότι τα \mathcal{S}_m και \mathcal{S}_n είναι επίσης γειτονικά. Οι παραπάνω ορισμοί μπορεί να επεκταθεί εύκολα για σύνθετα τμήματα και κελιά. Ένα μονοπάτι Π από σύνθετα ορίζεται ως μια πεπερασμένη ακολουθία γειτονικών σύνθετων κελιών. Προφανώς, ένα μονοπάτι Π που αποτελείται από κελιά που βρίσκονται εξολοκλήρου εντός του \mathcal{E}^f και περιέχει τις διαμορφώσεις P_{init} και P_{des} είναι μία λύση του προβλήματος μας. Για να βρούμε ένα τέτοιο μονοπάτι, κτίζουμε μια ιεραρχική αποσύνθεση $\mathcal{H} = \{ (i, \mathcal{H}) \mid i \in \mathcal{I}_{N_{\mathcal{R}}} \}$ του \mathcal{E} αντιστοιχίζοντας σε κάθε ρομπότ i μία ιεραρχική διαμέριση του \mathcal{W} , που ορίζεται ως ένα δέντρο $\mathcal{H} \triangleq (\mathcal{N}_{\mathcal{H}}, \mathcal{E}_{\mathcal{H}})$ τέτοιο ώστε:

- Κάθε κόμβος $\mathcal{S} \in \mathcal{N}_{\mathcal{H}}$ είναι ένα απλό τμήμα.
- Κάθε παιδί \mathcal{S}_j ενός κόμβου \mathcal{S}_i (δηλ. $(\mathcal{S}_i, \mathcal{S}_j) \in \mathcal{E}_{\mathcal{H}}$) είναι ένα υποσύνολο του \mathcal{S}_i .
- Το σύνολο όλων των φύλλων πρέπει να ορίζει ένα κάλυμμα του \mathcal{W} .

Ελεγκτής Υψηλού Επιπέδου

Σε αυτή την υποενότητα, παρουσιάζουμε έναν ελεγκτή υψηλού επιπέδου για την εύρεση μονοπατιού Π που συνδέει την αρχική P_{init} και τελική P_{des} διαμόρφωση. Ένα από τα κύρια πλεονεκτήματα του προτεινόμενου αλγορίθμου είναι η χρήση κατάλληλου σχήματος σήμανσης, το οποίο επιτρέπει αναδρομική υποδιαίρεση, σε κάθε επανάληψη, των σύνθετων κελιών που βρίσκονται στο σύνορο μεταξύ \mathcal{E}^f και \mathcal{E}^o , ενώ αγνοεί κελιά που βρίσκονται εξ ολοκλήρου εντός του \mathcal{E}^f ή του \mathcal{E}^o . Για να το επιτύχει αυτό, χρησιμοποιεί τις υπερ- και υπο-προσεγγίσεις $\overline{\mathcal{R}}_i$ και $\underline{\mathcal{R}}_i$ του αποτυπώματος κάθε ρομπότ για να προσδιορίσει εάν ένα ρομπότ δύναται να συγκρουστεί με κάποιο άλλο ενώ πλοηγούνται ανεξάρτητα το καθένα εντός του απλού κελιού που τους αντιστοιχεί. Πιο συγκεκριμένα, δεδομένου σύνθετου κελιού $\widehat{\mathcal{C}}$:

- Εάν η τομή όλων των $\overline{\mathcal{R}}_i(\widehat{\mathcal{C}}^{[i]})$, $i \in \mathcal{I}_{N_{\mathcal{R}}}$ είναι κενή, τότε, λόγω της ($;$), κανένα ρομπότ δεν μπορεί να συγκρουστεί με κάποιο άλλο ενώ $P \in \widehat{\mathcal{C}}$, με αποτέλεσμα το $\widehat{\mathcal{C}}$ να ανήκει στο \mathcal{E}^f . Ένα τέτοιο σύνθετο κελί λέγεται αποδεκτό.

- Εάν η τομή όλων των $\underline{\mathcal{R}}_i(\widehat{\mathcal{C}}^{[i]})$, $i \in \mathcal{I}_{N_{\mathcal{R}}}$ είναι μη-κενή, τότε, λόγω της (;), για κάθε $P \in \widehat{\mathcal{C}}$ υπάρχει ένα τουλάχιστον ζεύγος ρομπότ σε σύγκρουση, με αποτέλεσμα το $\widehat{\mathcal{C}}$ να ανήκει στο \mathcal{C}^o . Ένα τέτοιο σύνθετο κελί λέγεται μη-αποδεκτό.
- Εάν το $\widehat{\mathcal{C}}$ δεν είναι ούτε αποδεκτό ούτε μη-αποδεκτό, τότε λέγεται ανάμεικτο.

Σε γενικές γραμμές, μιστά κελιά περιέχουν εφικτές και ανέφικτες διαμορφώσεις και υποδιαίρεσή τους (αναδρομικά) αναμένεται να δώσει αποδεκτά και μη-αποδεκτά υπο-κελιά. Από την άλλη πλευρά, υποδιαίρεση αποδεκτών (μη-αποδεκτών) κελιών επιφέρει μόνο αποδεκτά (μη-αποδεκτά) κελιά, χωρίς να συνεισφέρει περαιτέρω στην εξερεύνηση του χώρου διαμόρφωσης.

Ο αλγόριθμος εξερεύνησης που παρουσιάζεται στο **Αλγόριθμος 4**, αρχικά κατασκευάζει μία τραχιά διαμέριση που περιέχει ολόκληρο τον \mathcal{C}^f (συναρτήσεις INITIALIZEHIERARCHY και INITIALIZECCELLS). Στην συνέχεια, τα αρχικά υπολογισμένα κελιά υποδιαιρούνται μέχρι να βρεθούν αποδεκτά κελιά που περιέχουν τις P_{init} και P_{des} , (συνάρτηση FINDENCLOSINGACCELL), όπου αδυναμία εύρεσης τέτοιων κελιών συνεπάγεται ανυπαρξία λύσης. Έπειτα, κατασκευάζεται μονοπάτι Π που συνδέει τα $\widehat{\mathcal{C}}_{\text{init}}$ και $\widehat{\mathcal{C}}_{\text{goal}}$ και αποτελείται από σύνθετα κελιά που ανήκουν στο σύνορο της εξερεύνησης $S_{\widehat{\mathcal{C}},F}$ (συνάρτηση CONNECTSTRINGS). Σε κάθε επανάληψη, το πρώτο ανάμεικτο κελί του Π (συνάρτηση GETFIRSTMIXEDCCELL) αφαιρείται από το σύνορο και υποδιαιρείται (συνάρτηση EXPANDCCELL) κατάλληλα (συναρτήσεις GETCONFLICTINGSCELLS και SELECTCELLWITHWIDESTSSLICE) σε μικρότερα κελιά, όπως φαίνεται στο **Αλγόριθμος 5**. Τέλος, ένα νέο μονοπάτι κατασκευάζεται έως ότου το Π αποτελείται από μόνο αποδεκτά κελιά ή δεν μπορεί να βρεθεί πλέον μονοπάτι από ανάμεικτα και αποδεκτά κελιά που οδηγεί στην P_{des} .

Κινηματικός Νόμος Ελέγχου

Δεδομένου μονοπατιού Π που αποτελείται από N_{Π} αποδεκτά κελιά, παρουσιάζουμε τώρα έναν κατανομημένο νόμο ελέγχου για την ασφαλή πλοήγηση των ρομπότ. Αρχικά, θεωρούμε δύο διαδοχικά σύνθετα κελιά $\widehat{\mathcal{C}}_{\ell}$ και $\widehat{\mathcal{C}}_{\ell+1}$ του Π , για τα οποία υπολογίζουμε το σύνολο-στόχο $\mathcal{G}_{\ell,\ell} \triangleq \widehat{\mathcal{C}}_{\ell}^{[i]} \cap \widehat{\mathcal{C}}_{\ell+1}^{[i]}$ του κάθε ρομπότ i , το οποίο περιέχει εφικτές διαμορφώσεις που ανήκουν από κοινού στα $\widehat{\mathcal{C}}_{\ell}^{[i]}$ και $\widehat{\mathcal{C}}_{\ell+1}^{[i]}$ και είναι μη-κενό εκ κατασκευής. Το σύνολο-στόχος του τελευταίου κελιού του Π αποτελείται μόνο από την επιθυμητή διαμόρφωση P_{des} , δλδ. $\widehat{\mathcal{C}}_{N_{\Pi}} = \{P_{\text{des}}\}$ (**Εικόνα 4.4**). Έστω $\mathcal{F}_{\ell,i} \triangleq \widehat{\mathcal{C}}_{\ell}^{[i]} \setminus \text{int}(\mathcal{G}_{\ell,i})$ και $\mathcal{G}'_{\ell,i} \triangleq \mathcal{G}_{\ell,i} \cap \mathcal{F}_{\ell,i}$, για κάθε $k \in \mathcal{I}_{N_{\Pi}}$. Παρατηρούμε ότι $\mathcal{G}'_{\ell,i}$ αποτελείται από ένα ή περισσότερα ξεχωριστά υποσύνολα καθώς και ότι το ρομπότ i πρέπει να πλοηγηθεί σε κάποιο από αυτά χωρίς να διαφύγει από το $\widehat{\mathcal{C}}_{\ell}^{[i]}$ έτσι ώστε να μεταβεί στο επόμενο κελί επιτυχώς. Αντίστοιχα, η μετάβαση από το $\widehat{\mathcal{C}}_{\ell}$ στο $\widehat{\mathcal{C}}_{\ell+1}$ θεωρείται ολοκληρωμένη όταν κάθε ρομπότ i φτάσει το $\widehat{\mathcal{C}}_{\ell+1}^{[i]}$.

Για να ικανοποιήσουν την παραπάνω προδιαγραφή, εξοπλίζουμε κάθε ρομπότ i με έναν ελεγκτή u_i που βασίζεται σε κατάλληλους μετασχηματισμούς του χώρου εργασίας και προσαρμοστικά τεχνητά δυναμικά πεδία, τα οποία διαθέτουν εγγυημένες ιδιότητες

ασφάλειας και σύγκλισης. Πιο συγκεκριμένα, χτίζουμε έναν διφαιομορφισμό $q_i^\ell = T_i^\ell(p_i)$ που απεικονίζει το $\mathcal{F}_{\ell,i}$ στον μοναδιαίο δίσκο \mathcal{D} , το εξωτερικό σύνορο του $\mathcal{F}_{\ell,i}$ στο μοναδιαίο κύκλο $\partial\mathcal{D}$ και κάθε εσωτερικό εμπόδιο σε ξεχωριστό σημείο $q_{i,j}^\ell$, $j \in \mathcal{J}_{N_i^\ell}$, όπου N_i^ℓ είναι το γένος του $\mathcal{F}_{\ell,i}$. Διακρίνουμε τις παρακάτω περιπτώσεις: α) το $\mathcal{G}'_{\ell,i}$ είναι εσωτερικό σύνορο του $\mathcal{F}_{\ell,i}$, και β) το $\mathcal{G}'_{\ell,i}$ είναι μέρος του εξωτερικού συνόρου του $\mathcal{F}_{\ell,i}$. Αναλόγως, ο μετασχηματισμός T_i^ℓ προσαρμόζεται κατάλληλα για να απλοποιήσει τον σχεδιασμό του αντίστοιχου δυναμικού πεδίου. Συγκεκριμένα, στην περίπτωση (α) τροποποιείται ο T_i^ℓ έτσι ώστε το $\mathcal{G}_{\ell,i}$ απεικονίζεται σε εσωτερικό σημείο $q_{\text{des},i}^\ell$ του \mathcal{D} , ενώ στην περίπτωση (β) σχεδιάζεται T_i^ℓ ώστε το $\mathcal{G}'_{\ell,i}$ να απεικονίζεται σε σημείο $q_{\text{des},i}^\ell$ του $\partial\mathcal{D}$. Στην συνέχεια, ορίζουμε ένα αρμονικό πεδίο ϕ_i^ℓ , για το κάθε ρομπότ i κατά το βήμα ℓ , τοποθετώντας αρμονικές πηγές επί του $q_{\text{des},i}^\ell$ και των $q_{i,j}^\ell$, το οποίο δίνεται τελικώς από:

$$\phi_i^\ell = k_{i,d}^\ell \ln \left(\frac{\|q_i^\ell - q_{\text{des},i}^\ell\|}{2} \right) - \sum_{j \in \mathcal{J}_{N_i^\ell}} k_{i,j}^\ell \ln \left(\frac{\|q_i^\ell - q_{i,j}^\ell\|}{2} \right) \quad (62)$$

όπου $k_{i,d}^\ell > 0$ και $k_{i,j}^\ell \geq 0$ είναι μεταβλητές παράμετροι. Τέλος, ο νόμος ελέγχου u_i^ℓ του ρομπότ i κατά το βήμα ℓ δίνεται από

$$u_i^\ell = -Ks(q_i^\ell, k_{v,i}^\ell) \left(J_i^\ell(q_i^\ell) \right)^{-1} \nabla_{q_i^\ell} \psi_i^\ell(q_i^\ell, k_{v,i}^\ell) \quad (63)$$

όπου K είναι θετικό κέρδος, J_i^ℓ είναι η Ιακωβιανή του T_i^ℓ , s είναι συνάρτηση που εξασφαλίζει αποφυγή σύγκρουσης με το εξωτερικό όριο και $\psi_i^\ell = 1 + \tanh(w\phi_i^\ell)/2$, όπου w θετική σταθερά.

Αποτελέσματα Προσομοιώσεων

Σε αυτή την ενότητα, παρουσιάζουμε αποτελέσματα προσομοίωσης που επιδεικνύουν την αποτελεσματικότητα της προτεινόμενης μεθοδολογίας. Συγκεκριμένα, θεωρούμε 5 σενάρια όπου ένα σύστημα που αποτελείται από 2, 4, 6, 8 και 10 ρομπότ, αντίστοιχα, αρχικοποιημένα εντός του χώρου εργασίας που απεικονίζεται στην **Εικόνα 4.5**, το οποίο πρέπει να πλοηγηθεί σε συγκεκριμένους τελικούς προορισμούς. Ο χρόνος που απαιτείται από τον προτεινόμενο αλγόριθμο, καθώς και το συνολικό πλήθος σύνθετων κελιών που παρήχθησαν κατά τις προσομοιώσεις φαίνεται στην **Πίνακα 4.1**. Τα προφίλ κίνησης των ρομπότ φαίνονται αντίστοιχα στις **Εικόνα 4.5**, **Εικόνα 4.6**, **Εικόνα 4.7**, **Εικόνα 4.8**, **Εικόνα 4.9**. Επιπλέον, οι **Εικόνα 4.10** και **Εικόνα 4.11** δείχνουν την αρχική και επιθυμητή θέση των ρομπότ καθώς και τα αντίστοιχα κελιά για το σενάριο με τα 8 ρομπότ.

Συνεργατική Μεταφορά Αντικειμένου

Εισαγωγή

Σε αυτό το κεφάλαιο, παρουσιάζουμε μια μεθοδολογία για τον συντονισμό της μεταφοράς ενός αντικειμένου από ομάδα ρομπότ που λειτουργούν μέσα ένα συμπαγή χώρο εργασίας. Λόγω της περιστροφής του αντικειμένου και της διαμόρφωσης κάθε βραχίονα, η μεθοδολογία μας θα πρέπει να λάβει υπόψη το μεταβαλλόμενο σχήμα του ρομποτικού συστήματος για να κτίσει ένα πλάνο που θα οδηγήσει με ασφάλεια το ρομποτικό σύστημα στην επιθυμητή τελική διαμόρφωση. Ως εκ τούτου, σχεδιάζουμε έναν αλγόριθμο υψηλού επιπέδου ο οποίος πρέπει να κατασκευάσει μια αλληλουχία κελιών του χώρου διαμόρφωσης, καθένα από τα οποία ορίζει ένα επιτρεπόμενο σύνολο καταστάσεων για το αντικείμενο, καθώς και ρητά διαστήματα για τους βαθμούς ελευθερίας του κάθε βραχίονα. Επιπλέον, κατάλληλα σχεδιασμένες υπο- και υπερ-προσεγγίσεις του ελεύθερου χώρου χρησιμοποιούνται για την καθοδήγηση της εξερεύνησης του χώρου των καταστάσεων, επιλέγοντας τα κελιά που χρειάζονται περαιτέρω υποδιαίρεση. Επίσης, μεθοδολογίες με βάση κατευθυντών αναφοράς και ελέγχου προδιαγραφόμενης επίδοσης χρησιμοποιούνται μαζί με αρμονικούς χάρτες προκειμένου να σχεδιαστεί ένας κατανεμημένος νόμος ελέγχου για την πραγματοποίηση των μεταβάσεων μεταξύ διαδοχικών κελιών, ο οποίος έχει εγγυημένες ιδιότητες σύγκλισης. Τέλος, παρατηρούμε ότι ο προτεινόμενος νόμος ελέγχου χαμηλού επιπέδου δεν απαιτεί συνεχή ανταλλαγή πληροφορίας μεταξύ των ρομπότ, τα οποία βασίζονται αποκλειστικά σε μετρήσεις της τρέχουσας διαμόρφωσης του αντικειμένου και των δικών τους καταστάσεων, προκειμένου να υπολογίσουν τις αντίστοιχες εισόδους ελέγχου.

Ορισμοί

Σε αυτό το κεφάλαιο, χρησιμοποιούμε το \mathbb{R} για να δηλώσουμε το σύνολο των πραγματικών αριθμών και το \mathbb{N} για το σύνολο των πραγματικών αριθμών ξεκινώντας από το μηδέν. Επίσης, χρησιμοποιούμε το $\mathcal{I}_N \triangleq \{1, 2, \dots, N\}$ ($\mathcal{I}_N^* \triangleq \{0, 1, 2, \dots, N\}$) για να δηλώσουμε το σύνολο που αποτελείται από όλους τους φυσικούς αριθμούς μέχρι το N , ξεκινώντας από το 1 (0). Επιπλέον, δεδομένων συνόλων A και B , χρησιμοποιούμε τα ∂A , $\text{int}(A)$, $\text{cl}(A)$ για να δηλώσουμε το σύνορο, εσωτερικό, κλειστότητα αντίστοιχα, και το $A \setminus B$ για να δηλώσουμε το συμπλήρωμα του B ως προς το A .

Δεδομένου συστήματος συντεταγμένων \mathcal{F}_O εντός του \mathbb{R}^2 και δύο σημεία $P_A, P_B \in \mathbb{R}^2$, χρησιμοποιούμε το ${}^O P_B$ για να δηλώσουμε την σχετική θέση του σημείου P_B από το σημείο P_A , οι συντεταγμένες του οποίου εκφράζονται ως προς το \mathcal{F}_O . Δεδομένων

πλαισίων $\mathcal{F}_A, \mathcal{F}_B, \mathcal{F}_C$, χρησιμοποιούμε το ${}^{\{A\}}P_{\{C\}} \in \mathbb{R}^2$ για να δηλώσουμε την σχετική θέση της αρχής των αξόνων του \mathcal{F}_C από την αρχή των αξόνων του \mathcal{F}_B , εκφρασμένη ως προς το \mathcal{F}_A . Αναλόγως, δεδομένων πλαισίων $\mathcal{F}_A, \mathcal{F}_B$, χρησιμοποιήσουμε ${}^{\{A\}}R_{\{B\}} \in \mathbb{R}^{2 \times 2}$ για να δηλώσουμε τον πίνακα περιστροφής που αντιστοιχεί στον σχετικό προσανατολισμό του \mathcal{F}_B ως προς το \mathcal{F}_A .

Δεδομένης γωνίας στροφής θ , έστω $R(\theta)$ ο πίνακας στροφής που ορίζεται ως

$$R(\theta) \triangleq \begin{bmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{bmatrix}.$$

Για δεδομένα πλαίσια $\mathcal{F}_A, \mathcal{F}_B$, ορίζουμε ${}^{\{A\}}\mathcal{T}_{\{B\}}$ τον ομογενή μετασχηματισμό συντεταγμένων από το πλαίσιο \mathcal{F}_B στο \mathcal{F}_A , που ορίζεται ως

$${}^{\{A\}}\mathcal{T}_{\{B\}} \triangleq \begin{bmatrix} {}^{\{A\}}R_{\{B\}} & {}^{\{A\}}P_{\{B\}} \\ 0 & 1 \end{bmatrix}.$$

Σημειώνουμε ότι οι παρακάτω εξισώσεις ισχύουν για δεδομένο σημείο P :

$$\begin{bmatrix} {}^{\{A\}}P \\ {}^{\{A\}}P \\ 1 \end{bmatrix} = {}^{\{A\}}\mathcal{T}_{\{B\}} \cdot \begin{bmatrix} {}^{\{B\}}P \\ {}^{\{B\}}P \\ 1 \end{bmatrix}$$

όπου ${}^{\{A\}}P$ είναι η θέση του P ως προς το πλαίσιο \mathcal{F}_A και ${}^{\{B\}}P$ είναι η θέση του P ως προς το πλαίσιο \mathcal{F}_B . Για λόγους συντομίας, θα χρησιμοποιούμε το ${}^{\{A\}}P = {}^{\{A\}}\mathcal{T}_{\{B\}} \cdot {}^{\{B\}}P$ αντί του παραπάνω συμβολισμού όταν είναι βολικό.

Ορισμός του Προβλήματος

Θεωρούμε συμπαγή χώρο εργασίας $\mathcal{W} \subseteq \mathbb{R}^2$ που περικλείεται από ένα στατικό εξωτερικό σύνορο $\partial\mathcal{W}_0$ και N_0 εσωτερικά σύνορα $\partial\mathcal{W}_i$, $i \in \mathcal{I}_{N_0}$, με $N_0 \in \mathbb{N}$. Πιο συγκεκριμένα, υποθέτουμε ότι \mathcal{W} μπορεί να γραφτεί ως εξής:

$$\mathcal{W} \triangleq \overline{\mathcal{O}_0} \setminus \bigcup_{i \in \mathcal{I}_{N_0}} \mathcal{O}_i. \quad (64)$$

όπου \mathcal{O}_0 δηλώνει την επιφάνεια που βρίσκεται εκτός του $\partial\mathcal{W}_0$ με $\overline{\mathcal{O}_0} \triangleq \mathbb{R}^2 \setminus \mathcal{O}_0$, και \mathcal{O}_i δηλώνει την επιφάνεια που περικλείεται από το $\partial\mathcal{W}_i$, για κάθε $i \in \mathcal{I}_{N_0}$ (Εικόνα 5.2). Έστω $\overline{\mathcal{W}}$ το συμπλήρωμα του \mathcal{W} ως προς το \mathbb{R}^2 , δηλ. $\overline{\mathcal{W}} \triangleq \mathbb{R}^2 \setminus \mathcal{W}$, το οποίο θεωρείται κλειστό. Επιπλέον, το εξωτερικό όριο του χώρου εργασίας $\partial\mathcal{W}_0 \triangleq \partial\mathcal{O}_0$ και τα εσωτερικά του εμπόδια $\partial\mathcal{W}_i \triangleq \partial\mathcal{O}_i$, $i \in \mathcal{I}_{N_0}$ θεωρούνται μη-τεμνόμενες καμπύλες Jordan. Χωρίς βλάβη της γενικότητας, υποθέτουμε ότι ο \mathcal{W} έχει ένα αυθαίρετο αδρανειακό πλαίσιο $\mathcal{F}_{\mathcal{W}}$.

Θεωρούμε τώρα ένα αντικείμενο $\mathcal{L} \subset \mathbb{R}^2$ το σώμα του οποίου είναι μία συμπαγής, κλειστή, πολυγωνική 2-πολλαπλότητα, με την δυνατότητα να μετατοπίζεται και να περιστρέφεται εντός του \mathcal{W} υπό την προϋπόθεση ότι δεν βρίσκεται σε επαφή με τα σύνορα

του χώρου εργασίας. Έστω $\mathcal{F}_{\mathcal{L}}$ σταθερό πλαίσιο τοποθετημένο στο \mathcal{L} . Θα χρησιμοποιούμε το $p_{\mathcal{L}}$ και το $\theta_{\mathcal{L}}$ για να δηλώνουμε την τρέχουσα θέση και προσανατολισμό του \mathcal{L} ως προς το $\mathcal{F}_{\mathcal{W}}$, δλδ.:

$$p_{\mathcal{L}} \triangleq \begin{matrix} \{W\} \\ \{L\} \end{matrix} P_{\{L\}} \quad R(\theta_{\mathcal{L}}) \triangleq \begin{matrix} \{W\} \\ \{L\} \end{matrix} R.$$

Το αντικείμενο \mathcal{L} θεωρείται στερεό σώμα και έστω $M_{\mathcal{L}}$, $P_{\mathcal{L},\text{com}}$, $I_{\mathcal{L}}$ η μάζα, το κέντρο μάζας, και η ροπή αδράνειας γύρω από το $P_{\mathcal{L},\text{com}}$, αντίστοιχα, εκφρασμένα ως προς το πλαίσιο $\mathcal{F}_{\mathcal{L}}$. Υποθέτοντας ότι το $P_{\mathcal{L},\text{com}}$ συμπίπτει με την αρχή των αξόνων του $\mathcal{F}_{\mathcal{L}}$, η δυναμική του \mathcal{L} δίνεται από:

$$\begin{aligned} M_{\mathcal{L}} \cdot \ddot{p}_{\mathcal{L}} &= \tau_{\mathcal{L},p} \\ I_{\mathcal{L}} \cdot \ddot{\theta}_{\mathcal{L}} &= \tau_{\mathcal{L},\theta} \end{aligned}$$

όπου $\tau_{\mathcal{L},p} \in \mathbb{R}^2$ και $\tau_{\mathcal{L},\theta} \in \mathbb{R}$ είναι η δύναμη και η ροπή που ασκείται εξωτερικά στο αντικείμενο. Τέλος, ορίζουμε το $\mathcal{L}(p, \theta)$ ως το αποτύπωμα του \mathcal{L} , δλδ., τον χώρο του \mathcal{W} που το σώμα του \mathcal{L} καταλαμβάνει όταν $p_{\mathcal{L}} = p$ ανδ $\theta_{\mathcal{L}} = \theta$.

Για την μεταφορά του αντικειμένου \mathcal{L} από μία αρχική σε μία τελική διαμόρφωση, χρησιμοποιούμε μία ομάδα από $N_{\mathcal{R}} \geq 2$ συνεργαζόμενους κινούμενους βραχίονες. Συγκεκριμένα, κάθε ρομπότ \mathcal{R}_i , $i \in \mathcal{I}_{N_{\mathcal{R}}}$ αποτελείται από μια ολονομική βάση \mathcal{B}_i και έναν βραχίονα \mathcal{A}_i ο οποίος είναι συνδεδεμένος στην βάση και εφοδιασμένος με τελικό στοιχείο δράσης $\mathcal{A}_{i,E}$ το οποίο πιάνει σταθερά το αντικείμενο \mathcal{L} σε συγκεκριμένο σημείο. Η κινηματική και δυναμική κάθε ρομπότ \mathcal{R}_i , $i \in \mathcal{I}_{N_{\mathcal{R}}}$ περιγράφεται στις επόμενες υποενότητες. Υποτίθεται επίσης ότι τα σώματα των \mathcal{B}_i και \mathcal{A}_i περιγράφονται από συμπαγείς, κλειστές ανδ συνδεδεμένες 2-πολλαπλότητες, για κάθε $i \in \mathcal{I}_{N_{\mathcal{R}}}$.

Δεδομένης αρχικής $q_{\mathcal{L},\text{init}} = [p_{\mathcal{L},\text{init}}^T, \theta_{\mathcal{L},\text{init}}^T]^T$ και επιθυμητής διαμόρφωσης $q_{\mathcal{L},\text{goal}} = [p_{\mathcal{L},\text{goal}}^T, \theta_{\mathcal{L},\text{goal}}^T]^T$ για το αντικείμενο \mathcal{L} , ο στόχος μας είναι να σχεδιάσουμε σχήμα ελέγχου για τα ρομπότ \mathcal{R}_i , $i \in \mathcal{I}_{N_{\mathcal{R}}}$ το οποίο μπορεί να οδηγήσει το αντικείμενο στον επιθυμητό προσορισμό εάν μία τέτοια τροχιά υπάρχει, ενώ παράλληλα εξασφαλίζει ότι ούτε το αντικείμενο ούτε τα ρομπότ θα συγκρουστούν με το σύνορο $\partial\mathcal{W}$ του χώρου εργασίας.

Κινηματική Κινούμενων Βραχιόνων

Για κάθε $i \in \mathcal{I}_{N_{\mathcal{R}}}$, έστω $\mathcal{F}_{\mathcal{B}_i}$ σωματόδετο πλαίσιο εντός του \mathcal{B}_i . Χωρίς βλάβη της γενικότητας, υποθέτουμε ότι η αρχή των αξόνων του $\mathcal{F}_{\mathcal{B}_i}$ συμπίπτει με το κέντρο περιστροφής της βάσης \mathcal{B}_i . Για λόγους συντομίας, έστω p_i και θ_i η τρέχουσα θέση και προσανατολισμός του $\mathcal{F}_{\mathcal{B}_i}$ ως προς το $\mathcal{F}_{\mathcal{W}}$, δλδ.:

$$p_i \triangleq \begin{matrix} \{W\} \\ \{B_i\} \end{matrix} P_{\{B_i\}} \quad R(\theta_i) \triangleq \begin{matrix} \{W\} \\ \{B_i\} \end{matrix} R.$$

Επιπλέον, θα χρησιμοποιούμε το $\mathcal{B}_i(p, \theta)$ για να δηλώνουμε το αποτύπωμα της βάσης του ρομπότ \mathcal{R}_i όταν αυτή βρίσκεται στο p με προσανατολισμό θ .

Αναφορικά με τον βραχίονα \mathcal{A}_i , υποθέτουμε ότι αποτελείται από έναν ή περισσότερους συνδέσμους $\mathcal{A}_{i,j}$, $j \in N_{\mathcal{A}_i}$ που είναι συνδεδεμένοι έτσι ώστε να σχηματίζουν μία

ανοιχτή αλυσίδα. Επιπλέον, ο πρώτος σύνδεσμος $\mathcal{A}_{i,1}$ είναι σταθερά συνδεδεμένος στην βάση \mathcal{B}_i , ενώ το τελικό στοιχείο δράσης είναι σταθερά συνδεδεμένο στο τελευταίο σύνδεσμο $\mathcal{A}_{i,N_{\mathcal{A}_i}}$, για κάθε $i \in \mathcal{I}_{N_{\mathcal{R}}}$. Η αρίθμηση των υπόλοιπων συνδέσμων είναι τέτοια ώστε το σώμα του συνδέσμου $\mathcal{A}_{i,j+1}$ να μπορεί να περιστρέφεται/μετατοπίζεται ως προς την κοινή άρθρωση με τον σύνδεσμο $\mathcal{A}_{i,j}$. Για κάθε βραχίονα \mathcal{A}_i , θα χρησιμοποιούμε το $q_{i,j}$ και $\mathcal{D}_{q_{i,j}}$ για να δηλώσουμε την κατάσταση και εύρος, αντίστοιχα, του j βαθμού ελευθερίας, που αντιστοιχεί στην άρθρωση μεταξύ των συνδέσμων $\mathcal{A}_{i,j}$ και $\mathcal{A}_{i,j+1}$, για κάθε $j \in \mathcal{I}_{N_{\mathcal{A}_i}-1}$ και $i \in \mathcal{I}_{N_{\mathcal{R}}}$. Σημειώνουμε ότι κάθε εύρος $\mathcal{D}_{q_{i,j}}$ είναι ένα υποσύνολο είτε του \mathbb{R} ή του \mathbb{S}^1 . Το επαυξημένο διάνυσμα κατάστασης z_i του ρομπότ \mathcal{R}_i ορίζεται ως:

$$z_i \triangleq [p_i^T, \theta_i, q_i^T]^T$$

όπου q_i είναι το διάνυσμα των καταστάσεων των αρθρώσεων του \mathcal{A}_i , για κάθε $i \in \mathcal{I}_{N_{\mathcal{R}}}$. Παρομοίως, για κάθε $i \in \mathcal{I}_{N_{\mathcal{R}}}$ και $j \in \mathcal{I}_{N_{\mathcal{A}_i}}$, έστω $\mathcal{F}_{\mathcal{A}_{i,j}}$ ένα σωματόδετο πλαίσιο εντός του $\mathcal{A}_{i,j}$. Επιπλέον, τοποθετούμε αυθαίρετο πλαίσιο \mathcal{F}_{E_i} στα σημείο επαφής μεταξύ του τελικού στοιχείου δράσης του βραχίονα \mathcal{A}_i και του αντικειμένου \mathcal{L} . Υποθέτουμε ότι α) η αρχή των αξόνων του $\mathcal{F}_{\mathcal{A}_{i,j+1}}$ βρίσκεται στον άξονα περιστροφής της άρθρωσης j , και β) η αρχή των αξόνων του \mathcal{F}_{E_i} συμπίπτει με το αντίστοιχο σημείο επαφής (Φιγυρε 5.3).

Αναφορικά με την ευθεία κινηματική κάθε ρομπότ, θα χρησιμοποιούμε το $\mathcal{T}_{\mathcal{B}_i}(p, \theta)$ για να δηλώσουμε τον μετασχηματισμό από το $\mathcal{F}_{\mathcal{B}_i}$ στο $\mathcal{F}_{\mathcal{W}}$ όταν το ρομπότ είναι τοποθετημένο στο p με προσανατολισμό θ , δλδ. ${}^{\{\mathcal{W}\}}P_{\{\mathcal{B}_i\}} = p$ και ${}^{\{\mathcal{W}\}}R_{\{\mathcal{B}_i\}} = R(\theta)$. Έστω $\mathcal{T}_{\mathcal{A}_i}(q_i)$ η ευθεία κινηματική του βραχίονα \mathcal{A}_i , δλδ. $\mathcal{T}_{\mathcal{A}_i}(q_i) \triangleq {}^{\{\mathcal{A}_{i,1}\}}\mathcal{T}_{\{E_i\}}$. Καθώς ο βραχίονας είναι σταθερά τοποθετημένος στην βάση, υπάρχει ομογενής μετασχηματισμός $\mathcal{T}_{\mathcal{B}_i, \mathcal{A}_i}$ μεταξύ της βάσης \mathcal{B}_i και του πρώτου συνδέσμου $\mathcal{A}_{i,1}$ του βραχίονα, δλδ. $\mathcal{T}_{\mathcal{B}_i, \mathcal{A}_i} \triangleq {}^{\{\mathcal{B}_i\}}\mathcal{T}_{\{\mathcal{A}_{i,1}\}}$. Η ευθεία κινηματική $\mathcal{T}_{\mathcal{R}_i}(p, \theta, q)$ του ρομπότ \mathcal{R}_i δίνεται από:

$$\mathcal{T}_{\mathcal{R}_i}(p, \theta, q) \triangleq \mathcal{T}_{\mathcal{B}_i}(p, \theta) \cdot \mathcal{T}_{\mathcal{B}_i, \mathcal{A}_i} \cdot \mathcal{T}_{\mathcal{A}_i}(q).$$

Τέλος, για κάθε $i \in \mathcal{I}_{N_{\mathcal{R}}}$, θα χρησιμοποιούμε το $\mathcal{J}_{\mathcal{R}_i}$ και $\mathcal{J}_{\mathcal{A}_i}$ για να δηλώσουμε την Ιακωβιανή του ρομπότ \mathcal{R}_i και του βραχίονα \mathcal{A}_i , δλδ.:

$$\begin{bmatrix} {}^{\{\mathcal{W}\}}\dot{P}_{\{E_i\}} \\ {}^{\{\mathcal{W}\}}\Omega_{\{E_i\}} \end{bmatrix} = \mathcal{J}_{\mathcal{R}_i}(z_i) \cdot z_i$$

και

$$\begin{bmatrix} {}^{\{\mathcal{B}_i\}}\dot{P}_{\{E_i\}} \\ {}^{\{\mathcal{B}_i\}}\Omega_{\{E_i\}} \end{bmatrix} = \mathcal{J}_{\mathcal{A}_i}(q_i) \cdot \dot{q}_i$$

όπου $\Omega_{\{E_i\}}$ είναι η γωνιακή ταχύτητα του τελικού στοιχείου δράσης $\mathcal{A}_{i,E}$.

Δυναμική Κινούμενων Βραχιόνων

Η δυναμική κάθε ρομπότ \mathcal{R}_i , $i \in \mathcal{I}_{N_{\mathcal{R}}}$ υπόκειται στο καθιερωμένο μοντέλο Euler-Lagrange, δλδ.:

$$M_{\mathcal{R}_i}(z_i) \cdot \ddot{z}_i + C_{\mathcal{R}_i}(z_i, \dot{z}_i) \cdot \dot{z}_i + G_{\mathcal{R}_i}(z_i) = \tau_{m,i} - (\mathcal{J}_{\mathcal{R}_i}(z_i))^T \cdot \tau_{e,i} \quad (65)$$

όπου $M_{\mathcal{R}_i}$, $C_{\mathcal{R}_i}$, $G_{\mathcal{R}_i} \in \mathbb{R}^{(3+N_{\mathcal{A}_i}) \times (3+N_{\mathcal{A}_i})}$ είναι τα μητρώα μάζας, κοριόλης και βαρύτητας, $\tau_{m,i} \in \mathbb{R}^{(3+N_{\mathcal{A}_i})}$ είναι η ροπή που ασκούν οι επενεργητές του ρομπότ, και $\tau_{e,i} \in \mathbb{R}^3$ είναι η ροπή που ασκείται από το ρομπότ στο αντικείμενο \mathcal{L} μέσω του τελικού στοιχείου δράσης.

Σχήμα Ελέγχου

Για την αντιμετώπιση του παραπάνω προβλήματος, σχεδιάζουμε ένα υβριδικό σχήμα ελέγχου, το οποίο αποτελείται από:

- α') έναν αλγόριθμο υψηλού επιπέδου ο οποίος δεδομένης αρχικής διαμόρφωσης $q_{\mathcal{L},\text{init}}$ και τελικής διαμόρφωσης $q_{\mathcal{L},\text{goal}}$, μπορεί να υπολογίσει μια ακολουθία ενδιάμεσων στόχων για το ρομποτικό σύστημα, αν υπάρχει λύση στο παραπάνω πρόβλημα, ή να καθορίσει ανυπαρξία λύσης, και
- β') έναν ελεγκτή χαμηλού επιπέδου ο οποίος χρησιμοποιεί κατάλληλους μετασχηματισμούς του χώρου εργασίας προκειμένου να οδηγήσει το αντικείμενο και τους κινούμενους βραχίονες από κάθε κελί στην επόμενο αποφεύγοντας συγκρούσεις με τα φυσικά εμπόδια του χώρου εργασίας.

Πιο συγκεκριμένα, ο ελεγκτής υψηλού επιπέδου κατασκευάζει μία διαμέριση του χώρου διαμόρφωσης του συστήματος σε κελιά υποδιαιρώντας τα διαστήματα των βαθμών ελευθερίας του ρομποτικού συστήματος μέχρι να βρεθεί μια αλληλουχία συνδεδεμένων κελιών που περιέχει τις $q_{\mathcal{L},\text{init}}$ και $q_{\mathcal{L},\text{goal}}$ (εάν υπάρχει). Στην συνέχεια, για κάθε κελί, υπολογίζονται κατάλληλες προσωρινές επιθυμητές θέσεις $p_{\mathcal{L}}$ και προσανατολισμοί $\theta_{\mathcal{L}}$ και κατάλληλοι νόμοι ελέγχου χρησιμοποιούνται για την κίνηση του συστήματος.

Αποσύνθεση του Χώρου Διαμόρφωσης

Σε αυτή την υποενότητα, παρουσιάζουμε το ιεραρχικό σχήμα αποσύνθεσης σε κελιά που θα χρησιμοποιηθεί για το σχεδιασμό μιας σειράς από υψηλού επιπέδου, εφικτές εντολές που ορίζουν ένα 'μονοπάτι' που οδηγεί στην επιθυμητή διαμόρφωση. Πριν προχωρήσουμε παρακάτω, πρώτα θα ρίξουμε μια πιο προσεκτική ματιά στον χώρο διαμόρφωσης \mathcal{C} του ρομποτικού συστήματος. Σε αυτή την υποενότητα, θα μοντελοποιήσουμε το ρομποτικό σύστημα ως ένα ενιαίο εικονικό ρομπότ \mathfrak{R} που αποτελείται από $N_{\mathfrak{R}} = 1 + \sum_{i \in \mathcal{I}_{N_{\mathcal{R}}}} N_{\mathcal{A}_i}$ συνδέσμους, που αντιστοιχούν στο αντικείμενο \mathcal{L} , στην κινούμενη βάση \mathcal{B}_i και στους συνδέσμους $\mathcal{A}_{i,j}$ κάθε κινούμενου βραχίονα \mathcal{R}_i , για κάθε $j \in \mathcal{I}_{N_{\mathcal{A}_i}}$ και $i \in \mathcal{I}_{N_{\mathcal{R}}}$.

Ξεκινάμε παρατηρώντας ότι οι σύνδεσμοι του \mathfrak{A} σχηματίζουν ένα δένδρο $\mathfrak{T}(\mathbf{n}, \mathbf{e})$, όπου \mathbf{n} είναι το σύνολο των κόμβων και $\mathbf{e} \subset \mathbf{n} \times \mathbf{n}$ είναι το σύνολο των ακμών μεταξύ των κόμβων. Θα χρησιμοποιούμε το \mathfrak{A}_i για να δηλώσουμε τον i σύνδεσμο του \mathfrak{A} . Μία ακμή $(i, j) \in \mathbf{e}$ συνεπάγεται ότι ο j κόμβος μπορεί να κινηθεί σε σχέση με τον i κόμβο γύρω από το σημείο $\mathfrak{P}_{i,j}$. Επιπλέον, δεδομένου $i \in \mathcal{I}_{N_{\mathfrak{A}}}$, θα χρησιμοποιούμε το \mathbf{n}_i^c για να δηλώσουμε τα παιδιά του κόμβου \mathfrak{A}_i , δηλ. το σύνολο των κόμβων \mathfrak{A}_j που $(i, j) \in \mathbf{e}$, για κάθε $j \in \mathcal{I}_{N_{\mathfrak{A}}}$. Επίσης, το \mathbf{n}_p^i δηλώνει τον γονέα \mathfrak{A}_j του \mathfrak{A}_i , δηλ. το μοναδικό σύνδεσμο για τον οποίο ισχύει $(j, i) \in \mathbf{e}$, εάν υπάρχει. Αναλόγως, ορίζουμε τα \mathbf{n}_a^i και \mathbf{n}_a^i ως τα σύνολο από απογόνους και προγόνους, αντίστοιχα, του κόμβου \mathfrak{A}_i . Χωρίς βλάβη της γενικότητας, επιλέγουμε αρίθμηση τέτοια ώστε ο πρώτος κόμβος του \mathfrak{A} (δηλ. η ρίζα του \mathfrak{T}), να αντιστοιχεί στο αντικείμενο \mathcal{L} . Για χάρη απλότητας, θα χρησιμοποιούμε το $\mathcal{F}_{\mathfrak{A}_i}$, $i \in \mathcal{I}_{N_{\mathfrak{A}}}$ για να δηλώσουμε το σωματόδετο πλαίσιο κάθε συνδέσμου του \mathfrak{A} και θα αναφερόμαστε στις αρχές των αξόνων ως σημεία αναφοράς. Έστω $p_{\mathfrak{A}} \triangleq [x_{\mathfrak{A}}, y_{\mathfrak{A}}]^T \in \mathbb{R}^2$ και $\theta_{\mathfrak{A}} \in \mathcal{D}_{\theta} \subseteq \mathbb{S}^1$ η σχετική θέση του σημείου αναφοράς του ρομποτικού συστήματος και ο σχετικός προσανατολισμός του πλαισίου του $\mathcal{F}_{\mathfrak{A}_0}$ ως προς το πλαίσιο $\mathcal{F}_{\mathcal{W}}$, αντίστοιχα.

Ορίζουμε τα \mathbf{q}_i και $\mathcal{D}_{\mathbf{q}_i}$, $i \in \mathcal{I}_{N_{\mathfrak{q}}}$, ως τον βαθμό ελευθερίας και το εύρος τιμών αυτού που αντιστοιχεί στην άρθρωση μεταξύ του συνδέσμου i και του γονέα του, όπου $N_{\mathfrak{q}} \triangleq \sum_{j \in \mathcal{I}_{N_{\mathfrak{R}}}} (N_{A_j} - 1)$. Χωρίς απώλεια της γενικότητας, αφού κάθε σύνδεσμος πέρα από τη ρίζα έχει ακριβώς ένα γονέα, υποθέτουμε ότι το σημείο περιστροφής $\mathfrak{P}_{i,j}$ συμπίπτει με την αρχή των αξόνων του $\mathcal{F}_{\mathfrak{A}_j}$. Επιπλέον, θεωρώντας τον προσανατολισμό $\theta_{\mathcal{L}}$ του αντικειμένου ως εικονική άρθρωση, το διάνυσμα κατάστασης \mathbf{z} του εικονικού ρομποτικού συστήματος \mathfrak{A} ορίζεται ως:

$$\mathbf{z} \triangleq [p_{\mathfrak{A}}^T, \theta_{\mathfrak{A}}, \mathbf{q}^T]^T = [p_{\mathfrak{A}}^T, \mathbf{q}^T]^T$$

όπου $\mathbf{q} \triangleq [\mathbf{q}_i]_{i \in \mathcal{I}_{N_{\mathfrak{q}}}}$, είναι το στοιβαγμένο διάνυσμα κατάστασης των εικονικών αρθρώσεων με $\mathbf{q}_0 \triangleq \theta_{\mathfrak{A}}$ και $\mathbf{q}_i \triangleq \mathbf{q}_i$ για κάθε $i \in \mathcal{I}_{N_{\mathfrak{q}}}$.

Εξετάζουμε τώρα το αποτύπωμα του ρομποτικού συστήματος ενώ αυτό κινείται μέσα στον χώρο εργασίας. Παρατηρούμε ότι, για κάθε $i \in \mathcal{I}_{N_{\mathfrak{A}}}$, το αποτύπωμα του κάθε συνδέσμου \mathfrak{A}_i , δηλ. η επιφάνεια που καταλαμβάνει σε δεδομένη διαμόρφωση, καθορίζεται από την θέση του σημείου περιστροφής και τις καταστάσεις των (εικονικών) αρθρώσεων. Θα χρησιμοποιούμε το $\mathfrak{A}_i(p, \mathbf{q})$ για να δηλώσουμε το αποτύπωμα όταν το σημείο περιστροφής βρίσκεται στο p και οι καταστάσεις των αρθρώσεων είναι \mathbf{q} . Παρατηρούμε επίσης ότι, ενώ κάθε σύνδεσμος μπορεί να κινηθεί ως προς το σημείο περιστροφής, οποιαδήποτε κίνηση αυτού μεταφέρεται και στα παιδιά του, προκαλώντας μετατόπιση και/ή περιστροφή κάθε ενός από τους απογόνους \mathbf{n}_a^i αυτού. Ως εκ τούτου, το αποτύπωμα του συνδέσμου \mathfrak{A}_i μπορεί να οριστεί σε σχέση με την τρέχουσα θέση $p_{\mathfrak{A}}$ του ρομποτικού συστήματος και της κατάστασης της (εικονικής) άρθρωσης του κάθε \mathfrak{A}_j που ανήκει στους προγόνους \mathbf{n}_a^i . Παρατηρώντας ότι το αποτύπωμα $\mathfrak{A}(\mathbf{z})$ του ρομποτικού συστήματος σε δεδομένη

διαμόρφωση z είναι απλό η ένωση των αποτυπωμάτων των επιμέρους συνδέσμων, δηλ.:

$$\mathfrak{R}(p_{\mathfrak{R}}, \mathbf{q}) \triangleq \bigcup_{i \in \mathcal{I}_{N_{\mathfrak{R}}}} \mathfrak{R}_i(p_{\mathfrak{R}}, [\mathbf{q}_j]_{j \in \mathfrak{n}_i^* \cup i}^T) \quad (66)$$

μπορούμε να ορίσουμε το σύνολο των αποδεκτών διαμορφώσεων του προβλήματός μας. Για χάρη συντομίας, το $\mathfrak{R}(\mathbf{q})$ θα χρησιμοποιείται αντί του $\mathfrak{R}(0, \mathbf{q})$ όπου θεωρείται προτιμότερο. Παρατηρώντας ότι ο χώρος διαμόρφωσης \mathcal{C} του ρομποτικού συστήματος είναι μία πολλαπλότητα διφερομορφική στο $\mathbb{R}^2 \times \mathbb{S}^1 \times \mathfrak{D}_{\mathfrak{q}_1} \times \dots \times \mathfrak{D}_{\mathfrak{q}_{N_{\mathfrak{q}}}}$ και ανακαλώντας ότι ούτε το αντικείμενο \mathcal{L} ούτε τα ρομπότ \mathcal{R}_i , $i \in \mathcal{I}_{N_{\mathfrak{R}}}$ επιτρέπεται να τέμνουν το σύνορο $\partial \mathcal{W}$, το σύνολο \mathcal{C}_f διαμορφώσεων χωρίς συγκρούσεις δίδεται από:

$$\mathcal{C}_f = \{z \mid \mathcal{W} \cap \mathfrak{R}(z) = \emptyset \text{ και } z \in \mathcal{C}\}. \quad (67)$$

Τέλος, ορίζουμε $\mathcal{C}_o \triangleq \mathcal{C} \setminus \mathcal{C}_f$.

Τώρα, για να σχεδιάσουμε ένα συνεχές ‘μονοπάτι’ εντός του \mathcal{C}_f που συνδέει τις διαμορφώσεις $q_{\mathcal{L}, \text{init}}$ και $q_{\mathcal{L}, \text{goal}}$, επεκτείνουμε την μεθοδολογία που παρουσιάστηκε σε προηγούμενο κεφάλαιο για ένα πολυγωνικό ρομπότ. Πιο συγκεκριμένα, σχεδιάζοντας ένα κατάλληλο κάλυμμα του ελεύθερου χώρου διαμόρφωσης υποδιαίρωντας αναδρομικά τα εύρη τιμών των \mathbf{q} , ο στόχος μας είναι να κατασκευάσουμε μία ιεραρχική διαμέριση του \mathcal{C}_f . Για κάθε έμμεσα ορισμένο κελί, υπολογίζουμε μία κατάλληλη υπερ- και υπό-προσέγγιση αυτού, με απλούστερο σχήμα, τα οποία χρησιμοποιούνται για την καθοδήγηση της εξερεύνησης του χώρου διαμόρφωσης καθώς και για τον σχεδιασμό ενός πλάνου υψηλού επιπέδου που θα οδηγήσει το ρομποτικό σύστημα στον προορισμό του. Με αυτό τον στόχο, πρώτα θεωρούμε το πεδίο ορισμού $\mathfrak{D}_{\mathfrak{q}_i}$ της κατάστασης \mathfrak{q}_i , με $i \in \mathcal{I}_{N_{\mathfrak{q}}}^*$. Θα αναφερόμαστε σε σύνολα της μορφής $\mathcal{S}_{[q_{i1}, q_{i2}]}^{\mathfrak{q}_i}$ ως απλά τμήματα της παραμέτρου \mathfrak{q}_i , όπου $q_{i1}, q_{i2} \in \mathfrak{D}_{\mathfrak{q}_i}$. Επίσης, ένα σύνολο $\mathfrak{S}_i = \{\mathcal{S}_j^{\mathfrak{q}_i} \mid j \in \mathcal{I}_{N_{\mathfrak{S}_i}}\}$ που αποτελείται από $N_{\mathfrak{S}_i}$ απλά τμήματα της \mathfrak{q}_i θα αποκαλείται κάλυμμα του $\mathfrak{D}_{\mathfrak{q}_i}$ εάν

$$\mathfrak{D}_{\mathfrak{q}_i} = \bigcup_{j \in \mathcal{I}_{N_{\mathfrak{S}_i}}} \mathcal{S}_j^{\mathfrak{q}_i}$$

για κάθε $k, \ell \in \mathcal{I}_{N_{\mathfrak{S}_i}}$ με $k \neq \ell$. Ένα σύνθετο τμήμα $\widehat{\mathcal{S}}$ ορίζεται ως ένα σύνολο από απλά τμήματα, δηλ. $\widehat{\mathcal{S}} = \{\mathcal{S}^{\mathfrak{q}_i} \mid i \in \mathcal{I}_{N_{\mathfrak{q}}}^*\}$. Αντιστοίχως, ένα σύνολο $\widehat{\mathfrak{S}} = \{\mathfrak{S}_i \mid i \in \mathcal{I}_{N_{\mathfrak{q}}}^*\}$ θα λέγεται κάλυμμα του ελεύθερου χώρου διαμόρφωσης \mathcal{C}_f εάν κάθε \mathfrak{S}_i είναι κάλυμμα του $\mathfrak{D}_{\mathfrak{q}_i}$. Σημειώνουμε ότι ένα κάλυμμα $\widehat{\mathfrak{S}}$ αντιστοιχεί σε μία διαμέριση του \mathcal{C}_f σε περιοχές

$$\mathcal{C}_{\widehat{\mathfrak{S}}} = \{[p_{\mathfrak{R}}^T, \mathbf{q}^T]^T \mid p_{\mathfrak{R}} \in \mathcal{W} \text{ και } \mathbf{q} \in \widehat{\mathfrak{S}}\} \cap \mathcal{C}_f, \quad \widehat{\mathfrak{S}} \in \widehat{\mathfrak{S}} \quad (68)$$

κάθε μία από τις οποίες αποτελείται από $N_{\widehat{\mathfrak{S}}} \geq 0$ απομονωμένα υποσύνολα $\mathcal{C}_{\widehat{\mathfrak{S}}_i}$, $i \in \mathcal{I}_{N_{\widehat{\mathfrak{S}}}}$. Παρατηρούμε ότι δύο κελιά $\mathcal{C}_{\widehat{\mathfrak{S}}_i}$ ανδ $\mathcal{C}_{\widehat{\mathfrak{S}}_j}$ είναι συνδεδεμένα εάν $\widehat{\mathfrak{S}}_i, \widehat{\mathfrak{S}}_j$ είναι γειτονικά και οι προβολές των $\mathcal{C}_{\widehat{\mathfrak{S}}_i}, \mathcal{C}_{\widehat{\mathfrak{S}}_j}$ στο επίπεδο τέμνονται. Θυμόμαστε ότι δύο διαφορετικά απλά τμήματα $\mathcal{S}_i^{\mathfrak{q}_k}$ και $\mathcal{S}_j^{\mathfrak{q}_k}$ είναι γειτονικά εάν η τομή τους $\mathcal{S}_i^{\mathfrak{q}_k} \cap \mathcal{S}_j^{\mathfrak{q}_k}$ είναι μη-κενή, ενώ δύο

σύνθετα τμήματα $\widehat{\mathcal{S}}_i = \{\mathcal{S}_i^{qk} \mid k \in \mathcal{I}_{N_q}^*\}$, και $\widehat{\mathcal{S}}_j = \{\mathcal{S}_j^{qk} \mid k \in \mathcal{I}_{N_q}^*\}$, λέγονται γειτονικά εάν \mathcal{S}_i^{qk} , \mathcal{S}_j^{qk} είναι γειτονικά, για κάθε $k \in \mathcal{I}_{N_q}^*$.

Για να αποφύγουμε τον άμεσο υπολογισμό του σχήματος ενός κελιού, ορίζουμε κατάλληλες υπερ- και υπο-προσεγγίσεις αυτού. Για την κατασκευή αυτών των προσεγγίσεων του συνόλου των ελεύθερων διαμορφώσεων που αντιστοιχεί στο σύνθετο τμήμα $\widehat{\mathcal{S}} = \{\mathcal{S}^{q_i} \mid i \in \mathcal{I}_{N_q}^*\}$, πρώτα υπολογίζουμε μία υπερ-προσέγγιση $\overline{\mathfrak{R}}(\widehat{\mathcal{S}})$ και μία υπο-προσέγγιση $\underline{\mathfrak{R}}(\widehat{\mathcal{S}})$ του αποτυπώματος του ρομποτικού συστήματος ως εξής:

$$\begin{aligned}\overline{\mathfrak{R}}(\widehat{\mathcal{S}}) &= \bigcup_{q \in \mathcal{I}_{\widehat{\mathcal{S}}}} \mathfrak{R}(q) \\ \underline{\mathfrak{R}}(\widehat{\mathcal{S}}) &= \bigcap_{q \in \mathcal{I}_{\widehat{\mathcal{S}}}} \mathfrak{R}(q)\end{aligned}\tag{69}$$

όπου

$$\mathcal{I}_{\widehat{\mathcal{S}}} = \mathcal{S}^{q_0} \times \mathcal{S}^{q_1} \times \dots \times \mathcal{S}^q.\tag{70}$$

Ένα παράδειγμα τέτοιων προσεγγίσεων για ρομποτικό σύστημα δύο συνδέσμων φαίνεται στην [Εικόνα 5.4](#). Η υπερ-προσέγγιση $\overline{\mathcal{C}}_{\widehat{\mathcal{S}}}$ και υπο-προσέγγιση $\underline{\mathcal{C}}_{\widehat{\mathcal{S}}}$ ενός δεδομένου διαμερίσματος $\mathcal{C}_{\widehat{\mathcal{S}}}$, υπολογίζεται από:

$$\begin{aligned}\overline{\mathcal{C}}_{\widehat{\mathcal{S}}} &\triangleq \{[p^T, q^T]^T \mid p \in \overline{\mathcal{W}}_{\widehat{\mathcal{S}}} \text{ και } q \in \mathcal{I}_{\widehat{\mathcal{S}}}\} \\ \underline{\mathcal{C}}_{\widehat{\mathcal{S}}} &\triangleq \{[p^T, q^T]^T \mid p \in \underline{\mathcal{W}}_{\widehat{\mathcal{S}}} \text{ και } q \in \mathcal{I}_{\widehat{\mathcal{S}}}\}\end{aligned}\tag{71}$$

όπου

$$\begin{aligned}\overline{\mathcal{W}}_{\widehat{\mathcal{S}}} &\triangleq \mathcal{W} \ominus \overline{\mathfrak{R}}(\widehat{\mathcal{S}}) \\ \underline{\mathcal{W}}_{\widehat{\mathcal{S}}} &\triangleq \mathcal{W} \ominus \underline{\mathfrak{R}}(\widehat{\mathcal{S}})\end{aligned}\tag{72}$$

με $A \ominus B$ να δηλώνει την Minkowski διαφορά των συνόλων A και B , και $\mathcal{I}_{\widehat{\mathcal{S}}} = \mathcal{S}^{q_0} \times \mathcal{S}^{q_1} \times \dots \times \mathcal{S}^q$. Προφανώς, κάθε ένα από τα $\overline{\mathcal{C}}_{\widehat{\mathcal{S}}}$ και $\underline{\mathcal{C}}_{\widehat{\mathcal{S}}}$ αποτελούνται από απομονωμένα κελιά $\overline{\mathcal{C}}_{\widehat{\mathcal{S}},i}$, $i \in \mathcal{I}_{N_{\overline{\mathcal{C}}_{\widehat{\mathcal{S}}}}}$ και $\underline{\mathcal{C}}_{\widehat{\mathcal{S}},i}$, $i \in \mathcal{I}_{N_{\underline{\mathcal{C}}_{\widehat{\mathcal{S}}}}}$, αντίστοιχα, που περιέχουν ή περιέχονται από τα κελιά του $\mathcal{C}_{\widehat{\mathcal{S}}}$.

Σε αυτό το σημείο, σημειώνουμε ότι η προσέγγιση του \mathcal{C}_f βελτιώνεται καθώς υποδιαιρείται ο χώρος διαμόρφωσης σε όλο και περισσότερα κελιά. Ως εκ τούτου, επιλέγοντας επαρκώς λεπτή διαμέριση του \mathcal{C} , μια αλληλουχία από γειτονικές υπο-προσεγγίσεις που συνδέουν τις $q_{\mathcal{L},\text{init}}$ και $q_{\mathcal{L},\text{goal}}$ θα εμφανιστεί, υπό την προϋπόθεση να υπάρχει λύση εξ αρχής. Αντί τις επιλογής αυθαίρετης, επαρκώς λεπτής διαμέρισης, σχεδιάζουμε ένα προσαρμοστικό σχήμα υποδιαίρεσης που χρησιμοποιεί υπερ-προσεγγίσεις για την επιλογή του τμήματος προς υποδιαίρεση σε κάθε βήμα. Πιο συγκεκριμένα, σχεδιάζουμε έναν αλγόριθμο που δεδομένου σύνθετου καλύμματος $\widehat{\mathcal{S}}$, προσπαθεί να βρει αλληλουχία $\underline{\Pi}$ από γειτονικές υπερ-προσεγγίσεις κελιών που συνδέουν τις δεδομένες διαμορφώσεις. Αν αυτό δεν είναι εφικτό, τότε ο αλγόριθμος προσπαθεί να ενώσει τις δύο διαμορφώσεις με μια αλληλουχία $\overline{\Pi}$ από γειτονικές υπερ-προσεγγίσεις κελιών. Αν μια τέτοια διαδρομή υπάρχει,

τότε το τμήμα που αντιστοιχεί σε κάποιο κελί του $\bar{\Pi}$ επιλέγεται βάσει κατάλληλου ευριστικού κανόνα και υποδιαιρείται, επιστρέφοντας μια νέα διαμέριση του \mathcal{C}_f . Διαφορετικά, δεν υπάρχει λύση στο δεδομένο πρόβλημα και ο αλγόριθμος επιστρέφει. Εν συντομία:

1. Εάν υπάρχει διαδρομή από γειτονικές υπο-προσεγγίσεις κελιών που περιέχουν τις $q_{\mathcal{L},\text{init}}$ και $q_{\mathcal{L},\text{goal}}$, τότε το πρόβλημα μας επιδέχεται λύση.
2. Εάν υπάρχει διαδρομή από γειτονικές υπερ-προσεγγίσεις κελιών που περιέχουν τις $q_{\mathcal{L},\text{init}}$ και $q_{\mathcal{L},\text{goal}}$, τότε δεν είναι ξεκάθαρο ακόμα αν το πρόβλημα έχει λύση και χρειάζεται περαιτέρω επέκταση του $\widehat{\mathcal{S}}$.
3. Εάν δεν υπάρχει διαδρομή από γειτονικές υπερ-προσεγγίσεις κελιών που περιέχουν τις $q_{\mathcal{L},\text{init}}$ και $q_{\mathcal{L},\text{goal}}$, τότε το πρόβλημα μας δεν επιδέχεται λύση.

Ο προτεινόμενος αλγόριθμος φαίνεται στο [Αλγόριθμος 6](#).

Κατανεμημένος Νόμος Ελέγχου

Δεδομένου μονοπατιού $\underline{\Pi}$ από κελιά σχεδιάζουμε κατανεμημένο σχήμα ελέγχου για τα ρομπότ που εγγυάται την ασφαλή μετάβαση από κελί σε κελί μέχρι να φτάσουν τον τελικό προορισμό $q_{\mathcal{L},\text{goal}}$. Έστω $\underline{\mathcal{C}}_{\widehat{\mathcal{S}}}$ κελί του $\underline{\Pi}$ και $\underline{\mathcal{C}}_{\widehat{\mathcal{S}}}$ η προβολή του στο επίπεδο. Θυμόμαστε ότι $\underline{\mathcal{C}}_{\widehat{\mathcal{S}}}$ είναι μία υπο-προσέγγιση του ελεύθερου χώρου διαμόρφωσης, που κατασκευάζεται βάσει του $\underline{\mathcal{W}}_{\widehat{\mathcal{S}}}$, το οποίο συνεπάγεται ότι, αν $q \in \widehat{\mathcal{S}}$, τότε η $p_{\mathcal{L}}$ μπορεί να βρísκεται ασφαλώς σε οποιοδήποτε σημείο του $\underline{\mathcal{C}}_{\widehat{\mathcal{S}}}$. Επίσης, το $\underline{\mathcal{C}}_{\widehat{\mathcal{S}}}$ είναι μη-κενή, συμπαγής περιοχή του \mathbb{R}^2 με αυθαίρετη συνδεσιμότητα και σχήμα. Ως εκ τούτου, μπορούμε να απεμπλέξουμε τον νόμο ελέγχου για: α) την θέση του αντικειμένου $p_{\mathcal{L}}$, β) τον προσανατολισμό του αντικειμένου $\theta_{\mathcal{L}}$, και γ) τις αρθρώσεις q_i κάθε ρομπότ \mathcal{R}_i , $i \in \mathcal{I}_{N_{\mathcal{R}}}$.

Για κάθε ενδιαμέσο κελί του $\underline{\Pi}$, υπολογίζουμε σύνολα-στόχους για τα $p_{\mathcal{L}}$, $\theta_{\mathcal{L}}$ και q ξεχωριστά, από της τομή του με το επόμενο κελί. Θεωρούμε ζεύγος διαδοχικών κελιών $\underline{\mathcal{C}}_{\widehat{\mathcal{S}}_i}$ και $\underline{\mathcal{C}}_{\widehat{\mathcal{S}}_j}$ in $\underline{\Pi}$. Αναφορικά με την θέση του αντικειμένου, αρκεί το $p_{\mathcal{L}}$ να φτάσει στο σύνολο

$$\mathcal{G}_{p_{\mathcal{L}}}(\underline{\mathcal{C}}_{\widehat{\mathcal{S}}_i}, \underline{\mathcal{C}}_{\widehat{\mathcal{S}}_j}) \triangleq \underline{\mathcal{C}}_{\widehat{\mathcal{S}}_i} \cap \underline{\mathcal{C}}_{\widehat{\mathcal{S}}_j}. \quad (73)$$

Αντίστοιχα, ο στόχος για τον προσανατολισμό του αντικειμένου προκύπτει από την τομή των τμημάτων των $\widehat{\mathcal{S}}_i$ και $\widehat{\mathcal{S}}_j$, δηλ.:

$$\mathcal{G}_{\theta_{\mathcal{L}}}(\underline{\mathcal{C}}_{\widehat{\mathcal{S}}_i}, \underline{\mathcal{C}}_{\widehat{\mathcal{S}}_j}) \triangleq \mathcal{S}_i^{\text{qo}} \cap \mathcal{S}_j^{\text{qo}}. \quad (74)$$

Τα σύνολα-στόχοι για του βαθμούς ελευθερίας κάθε ρομπότ προκύπτουν με παρόμοιο τρόπο. Συγκεκριμένα, έστω $\mathcal{P}_{\mathcal{A}_k}(\underline{\mathcal{C}}_{\widehat{\mathcal{S}}})$ η προβολή $\underline{\mathcal{C}}_{\widehat{\mathcal{S}}}$ κατά τον άξονα του βαθμού ελευθερίας του \mathcal{A}_k . Προφανώς, $\mathcal{P}_{\mathcal{A}_k}(\underline{\mathcal{C}}_{\widehat{\mathcal{S}}})$ ισούται με το γινόμενο των απλών τμημάτων $\widehat{\mathcal{S}}$ που αντιστοιχούν στο q_k . Έτσι, το αντίστοιχο σύνολο-στόχος του q_k δίνεται από

$$\mathcal{G}_{\mathcal{A}_k}(\underline{\mathcal{C}}_{\widehat{\mathcal{S}}_i}, \underline{\mathcal{C}}_{\widehat{\mathcal{S}}_j}) \triangleq \mathcal{P}_{\mathcal{A}_k}(\underline{\mathcal{C}}_{\widehat{\mathcal{S}}_i}) \cap \mathcal{P}_{\mathcal{A}_k}(\underline{\mathcal{C}}_{\widehat{\mathcal{S}}_j}), \quad \forall k \in \mathcal{I}_{N_{\mathcal{R}}}. \quad (75)$$

Ως εκ τούτου, για την ασφαλή πλοήγηση του ρομποτικού συστήματος από το $\underline{\mathcal{C}}_{\hat{\mathcal{S}}_i}$ στο $\underline{\mathcal{C}}_{\hat{\mathcal{S}}_j}$, πρέπει να σχεδιαστούν ελεγχτές για του κινούμενους βραχίονες που:

- διασφαλίζουν αναλλοίωτο του τρέχοντος κελιού, δλδ. $p_{\mathcal{L}} \in \mathcal{P}_{p_{\mathcal{L}}}(\underline{\mathcal{C}}_{\hat{\mathcal{S}}_i})$, $\theta_{\mathcal{L}} \in \mathcal{S}_i^{\text{qo}}$ και $q_k \in \mathcal{P}_{\mathcal{A}_k}(\underline{\mathcal{C}}_{\hat{\mathcal{S}}_i})$, $\forall k \in \mathcal{I}_{N_{\mathcal{R}}}$, έως ότου η μετάβαση ολοκληρωθεί, και
- διασφαλίζουν σύγκλιση των καταστάσεων στα σύνολα-στόχους $\mathcal{G}_{p_{\mathcal{L}}}(\underline{\mathcal{C}}_{\hat{\mathcal{S}}_i}, \underline{\mathcal{C}}_{\hat{\mathcal{S}}_j})$, $\mathcal{G}_{\theta_{\mathcal{L}}}(\underline{\mathcal{C}}_{\hat{\mathcal{S}}_i}, \underline{\mathcal{C}}_{\hat{\mathcal{S}}_j})$ και $\mathcal{G}_{\mathcal{A}_k}(\underline{\mathcal{C}}_{\hat{\mathcal{S}}_i}, \underline{\mathcal{C}}_{\hat{\mathcal{S}}_j})$, $k \in \mathcal{I}_{N_{\mathcal{R}}}$.

Πριν συνεχίσουμε με τον ορισμό των παραπάνω νόμων ελέγχου, κάνουμε τις παρακάτω υποθέσεις για το σύστημά μας.

Υπόθεση 1. Κάθε ρομπότ \mathcal{R}_k , $k \in \mathcal{I}_{N_{\mathcal{R}}}$ γνωρίζει ακριβώς δυναμικό μοντέλο αυτού και του αντικειμένου, δλδ. τα $M_{\mathcal{L}}$, $I_{\mathcal{L}}$, $P_{\mathcal{L}, \text{com}}$ και $M_{\mathcal{R}_k}$, $C_{\mathcal{R}_k}$, $D_{\mathcal{R}_k}$, $G_{\mathcal{R}_k}$ θεωρούνται γνωστά.

Υπόθεση 2. Κάθε ρομπότ \mathcal{R}_k , $k \in \mathcal{I}_{N_{\mathcal{R}}}$ γνωρίζει ακριβώς την κατάσταση του z_k και την τρέχουσα διαμόρφωση $q_{\mathcal{L}}$ του αντικειμένου \mathcal{L} .

Υπόθεση 3. Το πλάνο του αλγορίθμου υψηλού επιπέδου είναι διαθέσιμο σε όλα τα ρομπότ. Επίσης, κάθε ρομπότ \mathcal{R}_k , $k \in \mathcal{I}_{N_{\mathcal{R}}}$ μπορεί να επικοινωνεί με τα υπόλοιπα μόνο για να δηλώσει πως είναι έτοιμο να μεταβεί στο επόμενο κελί, δλδ. ότι $p_{\mathcal{L}} \in \mathcal{G}_{p_{\mathcal{L}}}(\underline{\mathcal{C}}_{\hat{\mathcal{S}}_i}, \underline{\mathcal{C}}_{\hat{\mathcal{S}}_j})$, $\theta_{\mathcal{L}} \in \mathcal{G}_{\theta_{\mathcal{L}}}(\underline{\mathcal{C}}_{\hat{\mathcal{S}}_i}, \underline{\mathcal{C}}_{\hat{\mathcal{S}}_j})$ και $q_k \in \mathcal{G}_{\mathcal{A}_k}(\underline{\mathcal{C}}_{\hat{\mathcal{S}}_i}, \underline{\mathcal{C}}_{\hat{\mathcal{S}}_j})$.

Υπόθεση 4. Κάθε ρομπότ \mathcal{R}_k , $k \in \mathcal{I}_{N_{\mathcal{R}}}$ μπορεί να ασκήσει ανεξάρτητα μια επιθυμητή δύναμη και ροπή μέσω του τελικού στοιχείου δράσης ενώ διατηρεί το q_k εντός του $\mathcal{P}_{\mathcal{A}_k}(\underline{\mathcal{C}}_{\hat{\mathcal{S}}_i})$. Επίσης, ο κάτω διαγώνιος υπο-πίνακας $N_{\mathcal{A}_k} - 1 \times N_{\mathcal{A}_k} - 1$ του $(M_{\mathcal{R}_k})^{-1} (I - (\mathcal{J}_{\mathcal{R}_k}(z_k))^{\dagger} \cdot \mathcal{J}_{\mathcal{R}_k})$ είναι μη-ιδιόμορφος.

Θέση του Αντικειμένου

Πρώτα, σχεδιάζουμε κατάλληλο διανυσματικό πεδίο για την σύγκλιση της θέσης του αντικειμένου $p_{\mathcal{L}}$ στο $\mathcal{G}_{p_{\mathcal{L}}}(\underline{\mathcal{C}}_{\hat{\mathcal{S}}_i}, \underline{\mathcal{C}}_{\hat{\mathcal{S}}_j})$. Για αυτόν τον λόγο, κατασκευάζουμε έναν μετασχηματισμό T_i του $\mathcal{P}_{p_{\mathcal{L}}}(\underline{\mathcal{C}}_{\hat{\mathcal{S}}_i}, \underline{\mathcal{C}}_{\hat{\mathcal{S}}_j}) \triangleq \mathcal{P}_{p_{\mathcal{L}}}(\underline{\mathcal{C}}_{\hat{\mathcal{S}}_i}) \setminus \mathcal{G}_{p_{\mathcal{L}}}(\underline{\mathcal{C}}_{\hat{\mathcal{S}}_i}, \underline{\mathcal{C}}_{\hat{\mathcal{S}}_j})$ στον μοναδιαίο δίσκο και απεικονίζουμε την περιοχή του $\mathcal{G}_{p_{\mathcal{L}}}(\underline{\mathcal{C}}_{\hat{\mathcal{S}}_i}, \underline{\mathcal{C}}_{\hat{\mathcal{S}}_j})$ σε ένα σημείο, χρησιμοποιώντας την διαδικασία που περιγράφεται στην **Υποενότητα 3.3.3**. Ο παρακάτω νόμος ελέγχου για την γραμμική ταχύτητα θα οδηγούσε ασφαλώς το αντικείμενο στο σύνολο-στόχο από σχεδόν κάθε αρχική διαμόρφωση:

$$v_{p_{\mathcal{L}}}^{[i]}(p_{\mathcal{L}}) \triangleq (J_{T_i}(p_{\mathcal{L}}))^{-1} \cdot (q_{\mathcal{L}, d}^{[i]} - q_{\mathcal{L}}^{[i]}) \quad (76)$$

όπου J_{T_i} είναι η Ιακωβιανή του T_i . Για να σχεδιάσουμε την επιθυμητή δύναμη που πρέπει να ασκηθεί στο αντικείμενο από τα ρομπότ, χρησιμοποιούμε την μεθοδολογία που παρουσιάστηκε στην εργασία [2] η οποία μας επιτρέπει να επεκτείνουμε ασφαλώς το διανυσματικό πεδίο **Εξίσωση 76** στο δυναμικό σύστημα δεύτερης τάξης. Πιο συγκεκριμένα, η

επιθυμητή δύναμη προκύπτει από:

$$\begin{aligned} \tau_{\mathcal{L},p}^{des} &= -M_{\mathcal{L}} \cdot \left(K_{p_{\mathcal{L}}}^{[i]} \cdot (p_{\mathcal{L}} - p_{\mathcal{L},G}^{[i]}) + \zeta_{p_{\mathcal{L}}}^{[i]} \cdot \dot{p}_{\mathcal{L}} \right) \\ \dot{p}_{\mathcal{L},G}^{[i]} &= K_{p_{\mathcal{L},G}}^{[i]} \cdot \frac{v_{p_{\mathcal{L}}}^{[i]}(p_{\mathcal{L},G}^{[i]})}{\|v_{p_{\mathcal{L}}}^{[i]}(p_{\mathcal{L},G}^{[i]})\|} \cdot \min \left(\|v_{p_{\mathcal{L}}}^{[i]}(p_{\mathcal{L},G}^{[i]})\|, \sqrt{\Delta E_{p_{\mathcal{L}}}^{[i]}(p_{\mathcal{L}}, p_{\mathcal{L},G}^{[i]}) / K_{p_{\mathcal{L}}}^{[i]}} \right) \end{aligned} \quad (77)$$

όπου

$$\Delta E_{p_{\mathcal{L}}}^{[i]}(p_{\mathcal{L}}, p_{\mathcal{L},G}^{[i]}) \triangleq K_{p_{\mathcal{L}}}^{[i]} \cdot d \left(p_{\mathcal{L},G}^{[i]}, \partial \mathcal{F}_{p_{\mathcal{L}}} \left(\underline{\mathcal{C}}_{\hat{\mathcal{S}}_i}, \underline{\mathcal{C}}_{\hat{\mathcal{S}}_j} \right) \right) - E_{p_{\mathcal{L}}}^{[i]}(p_{\mathcal{L}}, p_{\mathcal{L},G}^{[i]}) \quad (78)$$

$$E_{p_{\mathcal{L}}}^{[i]}(p_{\mathcal{L}}, p_{\mathcal{L},G}^{[i]}) \triangleq \frac{1}{2} \cdot \left(\|\dot{p}_{\mathcal{L}}\|^2 + K_{p_{\mathcal{L}}}^{[i]} \cdot \|p_{\mathcal{L}} - p_{\mathcal{L},G}^{[i]}\|^2 \right) \quad (79)$$

το $p_{\mathcal{L},G}^{[i]}$ είναι η (εικονική) κατάσταση του κυβερνήτη αναφοράς, το $d(x, \mathcal{X})$ είναι η απόσταση του x από το σύνολο \mathcal{X} , τα $K_{p_{\mathcal{L}}}^{[i]}$, $K_{p_{\mathcal{L},G}}^{[i]}$ είναι θετικές σταθερές και το $\zeta_{p_{\mathcal{L}}}^{[i]}$ είναι μια σταθερά εικονικής απόσβεσης.

Προσανατολισμός του Αντικειμένου

Για να οδηγήσουμε τον προσανατολισμό $\theta_{\mathcal{L}}$ στο αντίστοιχο σύνολο-στόχο $\mathcal{G}_{\theta_{\mathcal{L}}}(\underline{\mathcal{C}}_{\hat{\mathcal{S}}_i}, \underline{\mathcal{C}}_{\hat{\mathcal{S}}_j})$ ενώ εξασφαλίζουμε ότι παραμένει εντός του $\mathcal{S}_i^{q_0} = [\theta_l^{[i]}, \theta_l^{[i]}]$, σχεδιάζουμε επιθυμητή ροπή $\tau_{\mathcal{L},\theta}$ βάσει της μεθοδολογίας έλεγχου προδιαγεγραμμένης απόκρισης. Υποθέτουμε ότι το $\mathcal{G}_{\theta_{\mathcal{L}}}(\underline{\mathcal{C}}_{\hat{\mathcal{S}}_i}, \underline{\mathcal{C}}_{\hat{\mathcal{S}}_j})$ είναι της μορφής $[\theta_{l,G}^{[i]}, \theta_{u,G}^{[i]}]$, το οποίο μπορεί να εξασφαλιστεί σχεδιάζοντας το σχήμα διαμέρισης του χώρου διαμόρφωσης έτσι ώστε τα σύνθετα τμήματα να αλληλεπικαλύπτονται. Ορίζουμε τώρα τις παρακάτω συναρτήσεις απόκρισης

$$\begin{aligned} \underline{\rho}_{\theta_{\mathcal{L}}}^{[i]}(t) &\triangleq \theta_{l,G}^{[i]} + (\theta_l^{[i]} - \theta_{l,G}^{[i]}) \cdot e^{-\lambda_{\theta_{\mathcal{L}}} \cdot t} \\ \bar{\rho}_{\theta_{\mathcal{L}}}^{[i]}(t) &\triangleq \theta_{u,G}^{[i]} + (\theta_l^{[i]} - \theta_{u,G}^{[i]}) \cdot e^{-\lambda_{\theta_{\mathcal{L}}} \cdot t} \end{aligned} \quad (80)$$

όπου t είναι ο χρόνος και $\lambda_{\theta_{\mathcal{L}}}$ είναι θετική σταθερά. Ο αντίστοιχος νόμος ελέγχου δίνεται από:

$$\tau_{\mathcal{L},\theta}^{des} = I_{\mathcal{L}} \cdot \left(-K_{\theta_{\mathcal{L},2}}^{[i]} \cdot (\dot{\theta}_{\mathcal{L}} - v_{\theta_{\mathcal{L}}}^{[i]}) + \dot{v}_{\theta_{\mathcal{L}}}^{[i]} - a_{\theta_{\mathcal{L}}}^{[i]} \cdot \ln \left(\frac{\theta_{\mathcal{L}} - \underline{\rho}_{\theta_{\mathcal{L}}}^{[i]}}{\bar{\rho}_{\theta_{\mathcal{L}}}^{[i]} - \theta_{\mathcal{L}}} \right) \right) \quad (81)$$

$$v_{\theta_{\mathcal{L}}}^{[i]} \triangleq \frac{b_{\theta_{\mathcal{L}}}^{[i]} - K_{\theta_{\mathcal{L},1}}^{[i]} \cdot \ln \left(\frac{\theta_{\mathcal{L}} - \underline{\rho}_{\theta_{\mathcal{L}}}^{[i]}}{\bar{\rho}_{\theta_{\mathcal{L}}}^{[i]} - \theta_{\mathcal{L}}} \right)}{a_{\theta_{\mathcal{L}}}^{[i]}} \quad (82)$$

$$\begin{aligned} a_{\theta_{\mathcal{L}}}^{[i]} &\triangleq \frac{1}{\theta_{\mathcal{L}} - \underline{\rho}_{\theta_{\mathcal{L}}}^{[i]}} + \frac{1}{\bar{\rho}_{\theta_{\mathcal{L}}}^{[i]} - \theta_{\mathcal{L}}} \\ b_{\theta_{\mathcal{L}}}^{[i]} &\triangleq \frac{\dot{\underline{\rho}}_{\theta_{\mathcal{L}}}^{[i]}}{\theta_{\mathcal{L}} - \underline{\rho}_{\theta_{\mathcal{L}}}^{[i]}} + \frac{\dot{\bar{\rho}}_{\theta_{\mathcal{L}}}^{[i]}}{\bar{\rho}_{\theta_{\mathcal{L}}}^{[i]} - \theta_{\mathcal{L}}} \end{aligned} \quad (83)$$

με $K_{\theta_{\mathcal{L},1}}^{[i]}$ και $K_{\theta_{\mathcal{L},2}}^{[i]}$ θετικές σταθερές.

Κινούμενοι Ρομποτικοί Βραχίονες

Θεωρώντας τώρα το σχήμα ελέγχου των κινούμενων βραχιόνων, τονίζουμε ότι, λόγω των **Ασυμπτίων 1** και **Ασυμπτίων 2** και υποθέτοντας μια κοινή πολιτική αρχικοποίησης των εικονικών καταστάσεων των κυβερνητών αναφοράς, κάθε ρομπότ μπορεί να υπολογίσει την επιθυμητή συνολική δύναμη $\tau_{\mathcal{L},p}$ και ροπή $\tau_{\mathcal{L},\theta}$ που πρέπει να ασκηθεί στο αντικείμενο. Έτσι, κάθε ρομπότ \mathcal{R}_k , $k \in \mathcal{I}_{N_{\mathcal{R}}}$ αρκεί να ασκήσει στο αντικείμενο:

$$\tau_{e,k} = \frac{1}{N_{\mathcal{R}}} \cdot \begin{bmatrix} \tau_{\mathcal{L},p} \\ \tau_{\mathcal{L},\theta} - (\tau_{\mathcal{L},p})^T \cdot {}_{\{E_k\}}P_{\mathcal{L},com}^\perp \end{bmatrix} \quad (84)$$

όπου ${}_{\{E_k\}}P_{\mathcal{L},com}^\perp = R\left(\frac{\pi}{2}\right) \cdot R(\theta_{\mathcal{L}}) \cdot {}_{\{E_k\}}P_{\mathcal{L},com}$ με ${}_{\{E_k\}}P_{\mathcal{L},com}$ η θέση του κέντρου μάζας του αντικειμένου ως προς το σημείο επαφής με τον βραχίονα \mathcal{A}_k . Επιπλέον, κάθε ρομπότ πρέπει να διασφαλίσει ότι $q_k \in \mathcal{P}_{\mathcal{A}_k}(\underline{\mathcal{C}}_{\hat{\mathcal{S}}_i})$ ενώ οδηγεί τις καταστάσεις q_k στο $\mathcal{G}_{\mathcal{A}_k}(\underline{\mathcal{C}}_{\hat{\mathcal{S}}_i}, \underline{\mathcal{C}}_{\hat{\mathcal{S}}_j})$. Για αυτόν τον λόγο, σχεδιάζουμε μία δύναμη στο μηδενοχώρο της $\mathcal{J}_{\mathcal{R}_k}$ που εξασφαλίζει τις παραπάνω προδιαγραφές χωρίς να επηρεάζει την δύναμη που ασκείται στο αντικείμενο. Θυμόμαστε ότι η δυναμική του κινούμενου βραχίονα \mathcal{R}_k είναι:

$$M_{\mathcal{R}_k}(z_k) \cdot \ddot{z}_k + C_{\mathcal{R}_k}(z_k, \dot{z}_k) \cdot \dot{z}_k + G_{\mathcal{R}_k}(z_k) = \tau_{m,k} - (\mathcal{J}_{\mathcal{R}_k}(z_k))^T \cdot \tau_{e,k} \quad (85)$$

Υποθέτοντας γνωστές δυναμικές παραμέτρους και κατάσταση, σχεδιάζουμε

$$\tau_{m,k} = C_{\mathcal{R}_k}(z_k, \dot{z}_k) \cdot \dot{z}_k + G_{\mathcal{R}_k}(z_k) + \tau_{m,k,1} + \left(I - (\mathcal{J}_{\mathcal{R}_k}(z_k))^\dagger \cdot \mathcal{J}_{\mathcal{R}_k}(z_k)\right) \cdot \tau_{m,k,2} \quad (86)$$

όπου $\tau_{m,k,1}$ και $\tau_{m,k,2}$ είναι νέες εικονικές εισοδοί που θα οριστούν παρακάτω και $(\mathcal{J}_{\mathcal{R}_k})^\dagger$ δηλώνει τον ψευδό-αντίστροφο του $\mathcal{J}_{\mathcal{R}_k}$. Αντικαθιστώντας την παραπάνω σχέση στην **Εξίσωση 85** δίνει:

$$M_{\mathcal{R}_k}(z_k) \cdot \ddot{z}_k = \tau_{m,k,1} + \left(I - (\mathcal{J}_{\mathcal{R}_k}(z_k))^\dagger \cdot \mathcal{J}_{\mathcal{R}_k}(z_k)\right) \cdot \tau_{m,k,2} - (\mathcal{J}_{\mathcal{R}_k}(z_k))^T \cdot \tau_{e,k} \quad (87)$$

Θεωρούμε τώρα το παραπάνω δυναμικό μοντέλο στον χώρο εργασίας:

$$M'_{\mathcal{R}_k} \cdot \begin{bmatrix} \ddot{p}_{E_k} \\ \ddot{\theta}_{E_k} \end{bmatrix} + C'_{\mathcal{R}_k} \cdot \begin{bmatrix} \dot{p}_{E_k} \\ \dot{\theta}_{E_k} \end{bmatrix} = (\mathcal{J}_{\mathcal{R}_k}^T)^\dagger \cdot \tau_{m,k,1} - \tau_{e,k} \quad (88)$$

όπου \ddot{p}_{E_k} και $\ddot{\theta}_{E_k}$ είναι η θέση και ο προσανατολισμός του σημείου επαφής του τελικού στοιχείου δράσης και

$$\begin{aligned} M'_{\mathcal{R}_k} &= (\mathcal{J}_{\mathcal{R}_k}^T)^\dagger \cdot M_{\mathcal{R}_k} \cdot (\mathcal{J}_{\mathcal{R}_k})^\dagger \\ C'_{\mathcal{R}_k} &= -(\mathcal{J}_{\mathcal{R}_k}^T)^\dagger \cdot M_{\mathcal{R}_k} \cdot (\mathcal{J}_{\mathcal{R}_k})^\dagger \cdot \dot{\mathcal{J}}_{\mathcal{R}_k} \cdot (\mathcal{J}_{\mathcal{R}_k})^\dagger. \end{aligned} \quad (89)$$

Έστω $[p_{E_k}^T \ \theta_{E_k}]^T = \mathcal{T}_{\mathcal{L},E_k}(p_{\mathcal{L}}, \theta_{\mathcal{L}})$ ο μετασχηματισμός μεταξύ των θέσεων και προσανατολισμών των αντίστοιχων σημείων. Ισχύει ότι

$$\begin{bmatrix} \dot{p}_{E_k} \\ \dot{\theta}_{E_k} \end{bmatrix} = J_{\mathcal{L},E_k}(p_{\mathcal{L}}, \theta_{\mathcal{L}}) \cdot \begin{bmatrix} \dot{p}_{\mathcal{L}} \\ \dot{\theta}_{\mathcal{L}} \end{bmatrix} \quad (90)$$

$$\begin{bmatrix} \ddot{p}_{E_k} \\ \ddot{\theta}_{E_k} \end{bmatrix} = J_{\mathcal{L},E_k}(p_{\mathcal{L}}, \theta_{\mathcal{L}}) \cdot \begin{bmatrix} \ddot{p}_{\mathcal{L}} \\ \ddot{\theta}_{\mathcal{L}} \end{bmatrix} + \dot{J}_{\mathcal{L},E_k}(p_{\mathcal{L}}, \theta_{\mathcal{L}}) \cdot \begin{bmatrix} \dot{p}_{\mathcal{L}} \\ \dot{\theta}_{\mathcal{L}} \end{bmatrix} \quad (91)$$

με το $J_{\mathcal{L},E_k}$ να δηλώνει την Ιακωβιανή του μετασχηματισμού. Επομένως, η Εξίσωση 88 μπορεί να γραφτεί ως προς την κατάσταση του αντικειμένου ως εξής:

$$M''_{\mathcal{R}_k} \cdot \begin{bmatrix} \ddot{p}_{\mathcal{L}} \\ \ddot{\theta}_{\mathcal{L}} \end{bmatrix} + C''_{\mathcal{R}_k} \cdot \begin{bmatrix} \dot{p}_{\mathcal{L}} \\ \dot{\theta}_{\mathcal{L}} \end{bmatrix} = (\mathcal{J}_{\mathcal{R}_k}^T)^\dagger \cdot \tau_{m,k,1} - \tau_{e,k} \quad (92)$$

με

$$\begin{aligned} M''_{\mathcal{R}_k} &= M'_{\mathcal{R}_k} \cdot J_{\mathcal{L},E_k} \\ C''_{\mathcal{R}_k} &= C'_{\mathcal{R}_k} \cdot J_{\mathcal{L},E_k} + M'_{\mathcal{R}_k} \cdot \dot{J}_{\mathcal{L},E_k}. \end{aligned} \quad (93)$$

Παρατηρούμε ότι επιτυγχάνοντας την επιθυμητή συμπεριφορά, δλδ. το αντικείμενο να υπακούει την δυναμική που προδιαγράφουν οι Εξίσωση 77 και Εξίσωση 81 ενώ το φορτίο κατανέμεται εξίσου στα ρομπότ, ισοδυναμεί με

$$\tau_{e,k} = \frac{1}{N_{\mathcal{R}}} \left[I_{\mathcal{L}} \cdot \ddot{\theta}_{\mathcal{L}} - M_{\mathcal{L}} \cdot (\ddot{p}_{\mathcal{L}})^T \cdot {}_{\{E_k\}}P_{\mathcal{L},com}^\perp \right] = \frac{1}{N_{\mathcal{R}}} M_{\mathcal{L},E_k} \cdot \begin{bmatrix} \ddot{p}_{\mathcal{L}} \\ \ddot{\theta}_{\mathcal{L}} \end{bmatrix} \quad (94)$$

όπου $M_{\mathcal{L},E_k}$ είναι το μέρος της αδράνειας του αντικειμένου που αντιλαμβάνεται το ρομπότ \mathcal{R}_k . Αντικαθιστώντας την παραπάνω σχέση στην Εξίσωση 88 δίνει

$$\left(M''_{\mathcal{R}_k} + \frac{1}{N_{\mathcal{R}}} \cdot M_{\mathcal{L},E_k} \right) \cdot \begin{bmatrix} \ddot{p}_{\mathcal{L}} \\ \ddot{\theta}_{\mathcal{L}} \end{bmatrix} + C''_{\mathcal{R}_k} \cdot \begin{bmatrix} \dot{p}_{\mathcal{L}} \\ \dot{\theta}_{\mathcal{L}} \end{bmatrix} = (\mathcal{J}_{\mathcal{R}_k}^T)^\dagger \cdot \tau_{m,k,1}. \quad (95)$$

Επομένως, αρκεί να επιλέξουμε

$$\tau_{m,k,1} \triangleq \mathcal{J}_{\mathcal{R}_k} \cdot \left(\left(M''_{\mathcal{R}_k} + \frac{1}{N_{\mathcal{R}}} \cdot M_{\mathcal{L},E_k} \right) \cdot \begin{bmatrix} \tau_{\mathcal{L},p}^{des}/M_{\mathcal{L}} \\ \tau_{\mathcal{L},\theta}^{des}/I_{\mathcal{L}} \end{bmatrix} + C''_{\mathcal{R}_k} \cdot \begin{bmatrix} \dot{p}_{\mathcal{L}} \\ \dot{\theta}_{\mathcal{L}} \end{bmatrix} \right) \quad (96)$$

υποθέτοντας ότι εφοδιάζουμε τα $N_{\mathcal{R}}$ ρομπότ με τον ίδιο νόμο ελέγχου.

Θεωρώντας ξανά την Εξίσωση 87, θα σχεδιάσουμε τώρα το $\tau_{m,k,2}$ κατάλληλα έτσι ώστε να ικανοποιούνται οι προδιαγραφές των γωνιών των αρθρώσεων του βραχίονα. Θυμόμαστε ότι η προβολή του $\tau_{m,k,2}$ ως προς το $I - (\mathcal{J}_{\mathcal{R}_k}(z_k))^\dagger \cdot \mathcal{J}_{\mathcal{R}_k}(z_k)$ δεν επιδρά στην δύναμη και ροπή που ασκείται στο αντικείμενο. Έστω, λοιπόν, $L_{A,k}$, $L_{B,k}$, $L_{C,k}$, $L_{D,k}$ πίνακες τέτοιοι ώστε $L_{A,k} \in \mathbb{R}^{3 \times 3}$ και

$$L_k \triangleq M_{\mathcal{R}_k}^{-1} \cdot \left(I - (\mathcal{J}_{\mathcal{R}_k}(z_k))^\dagger \cdot \mathcal{J}_{\mathcal{R}_k}(z_k) \right) = \begin{bmatrix} L_{A,k} & L_{B,k} \\ L_{C,k} & L_{D,k} \end{bmatrix}. \quad (97)$$

Καθώς το $L_{D,k}$ θεωρείται αντιστρέψιμο βάσει της **Υπόθεση 4**, χρησιμοποιούμε την μεθοδολογία ελέγχου προδιαγεγραμμένης απόδοσης μαζί με **backstepping** για να σχεδιάσουμε το $\tau_{m,k,2}$ ως εξής:

$$\tau_{m,k,2} \triangleq \begin{bmatrix} 0 \\ 0 \\ 0 \\ (L_{D,k})^{-1} \cdot v_{k,B} \end{bmatrix} \quad (98)$$

όπου

$$v_{k,B} \triangleq \begin{bmatrix} -(\dot{q}_{k,1} - v_{k,b,1}) + \dot{v}_{k,b,1} - a_{q_{k,1}} \cdot \ln\left(\frac{q_{k,1} - \rho_{q_{k,1}}}{\bar{\rho}_{q_{k,1}} - q_{k,1}}\right) \\ -(\dot{q}_{k,1} - v_{k,b,1}) + \dot{v}_{k,b,2} - a_{q_{k,2}} \cdot \ln\left(\frac{q_{k,2} - \rho_{q_{k,2}}}{\bar{\rho}_{q_{k,2}} - q_{k,2}}\right) \\ \vdots \\ -(\dot{q}_{k,N_{A_k}-1} - v_{k,b,N_{A_k}-1}) + \dot{v}_{k,b,N_{A_k}-1} - a_{q_{k,N_{A_k}-1}} \cdot \ln\left(\frac{q_{k,N_{A_k}-1} - \rho_{q_{k,N_{A_k}-1}}}{\bar{\rho}_{q_{k,N_{A_k}-1}} - q_{k,N_{A_k}-1}}\right) \end{bmatrix} \quad (99)$$

$$v_{k,b,\ell} \triangleq \frac{b_{q_{k,\ell}} - \ln\left(\frac{q_{k,\ell} - \rho_{q_{k,\ell}}}{\bar{\rho}_{q_{k,\ell}} - q_{k,\ell}}\right)}{a_{q_{k,\ell}}} \quad (100)$$

$$a_{q_{k,l}} \triangleq \frac{1}{q_{k,l} - \rho_{q_{k,l}}} + \frac{1}{\bar{\rho}_{q_{k,l}} - q_{k,l}} \quad (101)$$

$$b_{q_{k,l}} \triangleq \frac{\dot{\rho}_{q_{k,l}}}{q_{k,l} - \rho_{q_{k,l}}} + \frac{\dot{\bar{\rho}}_{q_{k,l}}}{\bar{\rho}_{q_{k,l}} - q_{k,l}}$$

είναι ο νόμος ελέγχου αναφοράς της ταχύτητας, $\rho_{q_{k,\ell}}$ και $\bar{\rho}_{q_{k,\ell}}$ είναι συναρτήσεις απόδοσης οι οποίες αλλάζουν ομαλά από το $\mathcal{P}_{A_k}(\underline{C}_{\hat{S}_i})$ στο $\mathcal{G}_{A_k}(\underline{C}_{\hat{S}_i}, \underline{C}_{\hat{S}_j})$, και ορίζονται ως

$$\begin{aligned} \rho_{q_{k,\ell}}(t) &\triangleq \bar{q}_{G,k,\ell} + (\bar{q}_{k,\ell} - \bar{q}_{G,k,\ell}) \cdot e^{-\lambda_q \cdot t} \\ \bar{\rho}_{q_{k,\ell}}(t) &\triangleq \underline{q}_{G,k,\ell} + (\underline{q}_{k,\ell} - \underline{q}_{G,k,\ell}) \cdot e^{-\lambda_q \cdot t} \end{aligned} \quad (102)$$

με $\bar{q}_{k,\ell}$, $\underline{q}_{k,\ell}$ και $\bar{q}_{G,k,\ell}$, $\underline{q}_{G,k,\ell}$ τα κάτω και άνω φράγματα των γωνιών των αρθρώσεων του A_k που αναλογούν στο $\mathcal{P}_{A_k}(\underline{C}_{\hat{S}_i})$ και $\mathcal{G}_{A_k}(\underline{C}_{\hat{S}_i}, \underline{C}_{\hat{S}_j})$ αντίστοιχα.

Αποτελέσματα Προσομοιώσεων

Για να δείξουμε την αποτελεσματικότητα του προτεινόμενου σχήματος ελέγχου, θεωρούμε ρομποτικό σύστημα από δύο κινούμενους βραχίονες που κρατάνε ένα ορθογωνικό αντικείμενο, όπως φαίνεται στη **Εικόνα 5.3**, που λειτουργεί εντός του χώρου εργασίας που απεικονίζεται στην **Εικόνα 5.5**. Το ρομποτικό σύστημα αρχικοποιήθηκε με διαμόρφωση $q_{\mathcal{L},\text{init}} = [0.9, 2, 1.57]^T$ και $q_{1,1} = q_{2,1} = 0$ ενώ η τελική επιθυμητή διαμόρφωση του αντικειμένου ορίστηκε ως $q_{\mathcal{L},\text{goal}} = [5, 8, 4.663]^T$. Τα τμήματα των κελιών που

υπολογίστηκαν από τον αλγόριθμο φαίνονται στον Πίνακας 5.1, ενώ οι παράμετροι ελέγχου που επιλέχθηκαν δίνονται στον Πίνακας 5.2. Η Εικόνα 5.6 δείχνει την τροχιά του ρομποτικού συστήματος υπό τον προτεινόμενο νόμο ελέγχου, ενώ διαγράμματα της θέσης και προσανατολισμού του αντικειμένου και των ταχυτήτων αυτών, φαίνονται στις Εικόνα 5.7, Εικόνα 5.8, Εικόνα 5.9, Εικόνα 5.10, αντίστοιχα. Επιπλέον, η Εικόνα 5.13 και η Εικόνα 5.13 δείχνουν την εξέλιξη των αρθρώσεων των δύο βραχιόνων, καθώς και τα υπολογισμένα κάτω και άνω όρια που αντιστοιχούν σε κάθε κελί. Η συνολική δύναμη και ροπή που ασκήθηκε στο αντικείμενο φαίνεται στις Εικόνα 5.11 και Εικόνα 5.12, αντίστοιχα. Από τα παραπάνω διαγράμματα, μπορεί κανείς να επιβεβαιώσει ότι το ρομποτικό σύστημα συνέκλινε επιτυχώς στην επιθυμητή διαμόρφωση διατηρώντας τις προδιαγραφές των θ_L , $q_{1,1}$, $q_{2,1}$.