UNIVERSIDADE FEDERAL DE MINAS GERAIS
Instituto de Ciências Exatas
Programa de Pós-Graduação em Ciência da Computação

Caio Conti Guidote Ribeiro

Socially-aware Navigation and Object Transportation for Mobile Manipulators in Static Environments

Belo Horizonte
2025

Caio Conti Guidote Ribeiro

Socially-aware Navigation and Object Transportation for Mobile Manipulators in Static Environments

Final Version

Thesis presented to the Graduate Program in Computer Science of the Federal University of Minas Gerais in partial fulfillment of the requirements for the degree of Master in Computer Science.

Advisor: Douglas Guimarães Macharet

Belo Horizonte
2025

Ribeiro, Caio Conti Guidote.

R484s Socially-aware navigation and object transportation for mobile manipulators in static environments [recurso eletrônico] / Caio Conti Guidote Ribeiro. – 2025.

1 recurso online (109 f. il.) : pdf.

Orientador: Douglas Guimarães Macharet.

Dissertação (mestrado) - Universidade Federal de Minas Gerais, Instituto de Ciências Exatas, Departamento de Ciência da Computação.

Referências: f. 99-109.

1. Computação - Teses. 2. Robótica - Teses. 3. Inteligência Artificial - Teses. I. Macharet, Douglas Guimarães Macharet. II. Universidade Federal de Minas Gerais, Instituto de Ciências Exatas, Departamento de Ciência da Computação. III. Título.

CDU 519.6*82.9 (043)

Ficha catalográfica elaborada por Célio Resende Diniz, bibliotecário CRB 6/2403 - Universidade Federal de Minas Gerais – ICEx.

UNIVERSIDADE FEDERAL DE MINAS GERAIS
INSTITUTO DE CIÊNCIAS EXATAS
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

FOLHA DE APROVAÇÃO

Socially-aware Navigation and Object Transportation for Mobile Manipulators in Static Environments

CAIO CONTI GUIDOTE RIBEIRO

Dissertação defendida e aprovada pela banca examinadora constituída pelos Senhores:

Douglas Guimarães Macharet

PROF. DOUGLAS GUIMARÃES MACHARET - Orientador
Departamento de Ciência da Computação - UFMG

PROF. ARMANDO ALVES NETO
Departamento de Engenharia Eletrônica - UFMG

PROF. LUIZ CHAIMOWICZ
Departamento de Ciência da Computação - UFMG

Belo Horizonte, 31 de outubro de 2025.

Acknowledgments

Agradeço aos meus pais, Cássia e Edivan, pelo constante suporte que me oferecem, independentemente do objetivo final desde que o mesmo me deixe feliz. À minha tia Escura, que cuida de mim desde bebê e sempre se preocupa com meu bem-estar. À minha namorada Terra, que mora no meu coração, pelo companheirismo, carinho e apoio constantes. A toda a minha família, pelas boas conversas, bons tempos, caminhadas e pela ternura com que me tratam.

Também agradeço ao meu orientador Douglas Macharet, pelas inúmeras conversas, orientações e planejamentos que fizemos juntos ao passar dos anos. À minha banca, composta pelos professores Armando Alves, que tive o prazer de conhecer durante a graduação, e o professor Luiz Chaimowicz, pelas discussões e propostas de melhoria da minha dissertação.

Agradeço a todos os meus amigos que fiz com o passar dos anos durante minha estadia em Minas Gerais, na UFMG. Amizades importantes, desde quando estava na graduação de Engenharia de Controle e Automação até o meu mestrado no VeRLab. Estou carregado de boas memórias dos projetos e das diversões que tivemos em conjunto, aguardo ansiosamente pelo dia em que nos encontraremos novamente.

Por fim, devo agradecer ao meu irmão Ícaro, que na realidade, esta é a primeira frase que escrevo nos agradecimentos, mas a última colocada, por propósitos de continuidade. Ao meu irmão, querido do coração, melhor do que concordar, é contrariar, como será demonstrado pela frase na página abaixo.

A todos que cruzaram meu caminho, sem vocês, a trajetória seria outra, e a trajetória que percorri me deixa imensamente feliz, muito obrigado!

"*Ain't that just the way.*"
(Greg, OTGW)

Resumo

Este trabalho aborda o desafio de navegar com um manipulador móvel robótico, transportando uma carga, em um ambiente estático com presença humana, respeitando normas sociais. Em outras palavras, trata do problema de Navegação Social para Manipuladores Móveis, com o objetivo de desenvolver um manipulador móvel socialmente consciente. Aqui, propomos uma técnica inovadora que não apenas garante navegação sem colisões, mas também respeita as preferências sociais humanas e considera o objeto sendo transportado, promovendo uma coexistência harmoniosa entre robôs e pessoas. Enquanto pesquisas anteriores em robótica social focaram em robôs móveis para navegação social, este trabalho busca integrar um braço robótico a uma base móvel e acionar tanto a plataforma móvel quanto o braço robótico de forma coordenada. Essa configuração permite ao sistema minimizar o desconforto social durante o transporte de objetos. Propomos duas técnicas distintas: a primeira computa movimentos coordenados base–braço para minimizar o desconforto, adequada para ambientes bidimensionais mais simples. A segunda é uma estratégia desacoplada, planejando primeiro a base e depois o braço, tornando mais rápida a busca por soluções em cenários tridimensionais. Além disso, introduzimos um novo modelo tridimensional de desconforto que, até onde sabemos, não havia sido definido anteriormente. Nossos resultados demonstram que os métodos propostos superam abordagens tradicionais que consideram apenas a base móvel.

Palavras-chave: Robótica social, navegação social, transporte social de objetos, manipuladores móveis, manipulação móvel social.

Abstract

This work addresses the challenge of navigating a robotic mobile manipulator, carrying a load, in a static, human-populated environment while respecting social norms. In other terms, it tackles the problem of *Social Navigation for Mobile Manipulators*, aiming to develop a socially-aware mobile manipulator. Here, we propose a novel technique that not only ensures collision-free navigation but also respects human social preferences and considers the object being transported, building harmonious coexistence between robots and people. While previous research in social robotics focused on mobile robots for social navigation, this work seeks to integrate a robotic arm to a mobile base and actuate both mobile platform and robotic arm concurrently. This setup allows the system to minimize social discomfort while transporting objects. We propose two distinct techniques: the first computes coordinated base–arm motions to minimize discomfort, suitable for simpler two-dimensional environments. The second is a decoupled strategy, planning first the base and then the arm, making it faster to find solutions for three-dimensional scenarios. In addition, we introduce a novel three-dimensional discomfort model, which, to the best of our knowledge, has not been previously defined. Our results demonstrate that the proposed methods outperform traditional approaches that consider only the mobile base.

Keywords: Social robotics, social navigation, social object transportation, mobile manipulators, social mobile manipulation.

List of Figures

List of Tables

List of Algorithms

List of Abbreviations and Acronyms

UFMG	Universidade Federal de Minas Gerais
HRI	*Human-Robot Interaction*
RRT	*Rapidly-exploring Random Tree*
RRT*	*Rapidly-exploring Random Tree Star*
IR-RRT*	*Informed-Risk-RRT**
MPP	*Maximum Permissible Pressure*
RH	*Relative Height*
MF	*Membership Functions*
PRM	*Probabilistic Roadmap*
RL	*Reinforcement Learning*

Contents

Chapter 1

Introduction

The presence of robots in environments shared with humans, whether through direct interaction or indirect cohabitation, has been increasing over the past few years and is expected to continue growing. Robots can cooperate with people on joint tasks or simply coexist in the same space with different activities. In these scenarios, it is important to consider how the interaction between robots and humans occurs.

In this context, it is essential that robots and humans coexist harmoniously, as they could otherwise interfere with each other's tasks in undesirable ways, leading to inefficiency or even accidents. Furthermore, for humans to accept autonomous vehicles performing tasks around them and to feel safe and secure, these autonomous systems must consider human nature, social restrictions, and expectations.

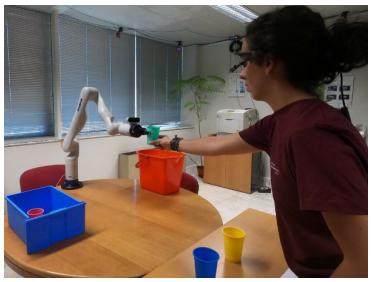
Therefore, developing techniques to navigate autonomous robots through environments with humans nearby is a valuable contribution to robotics. These advancements promote better coexistence between people and robots, leading to improved workplace environments, increased productivity, and enhanced quality of life.

This work proposes the development of a technique for social navigation and manipulation for mobile manipulators robots, with a task to carry an object in an environment having one or more humans nearby.

1.1 Contextualization

Robots come in various types, which we categorize as mobile robots, manipulators, and mobile manipulators. Mobile robots are capable of moving through the environment and are not fixed in one location. Manipulators, on the other hand, are robotic arms with a fixed base, allowing only the arm to move within a specific range. Mobile manipulators combine both types, featuring a robotic arm mounted on a mobile platform, allowing for both movement through the environment and manipulation capabilities. Figure 1.1 presents examples of the three categories of robots mentioned.

(a) Mobile Robot [20].



(b) Manipulator Robot [51].

(c) Mobile Manipulator [72].

Figure 1.1: One example for three different categories of robots: (a) A four-wheeled Mobile Robot, (b) A robotic arm Manipulator Robot, (c) A four-wheeled Mobile Manipulator with two robotic arms.

Social robotics is a rapidly growing area within robotics, driven by the increasing demand for robots that can interact and coexist with humans [54]. In fact, some robots are specifically designed for social environments, such as the Pepper robot shown in Figure 1.2. Social robotics is a field of research aimed at developing methods to enable robots to interact with people in a socially-aware manner. The general objective is to design robots that can recognize and respond appropriately to the presence of people nearby, ensuring their actions do not cause discomfort, one of the most important issues for human-robot symbiosis [64].

Social robotics can be applied to Human–Robot Interaction (HRI) tasks [10], and the terms are often used interchangeably in the literature. In this work, however, we make a distinction: while HRI refers broadly to any task in which humans and robots interact, social robotics specifically addresses scenarios in which the robot accounts for social behaviors. Social behavior refers to the actions individuals take to improve social acceptance and/or reduce social discomfort. An everyday example is the tendency to avoid walking through a group of friends engaged in conversation, thereby respecting social boundaries and personal space.

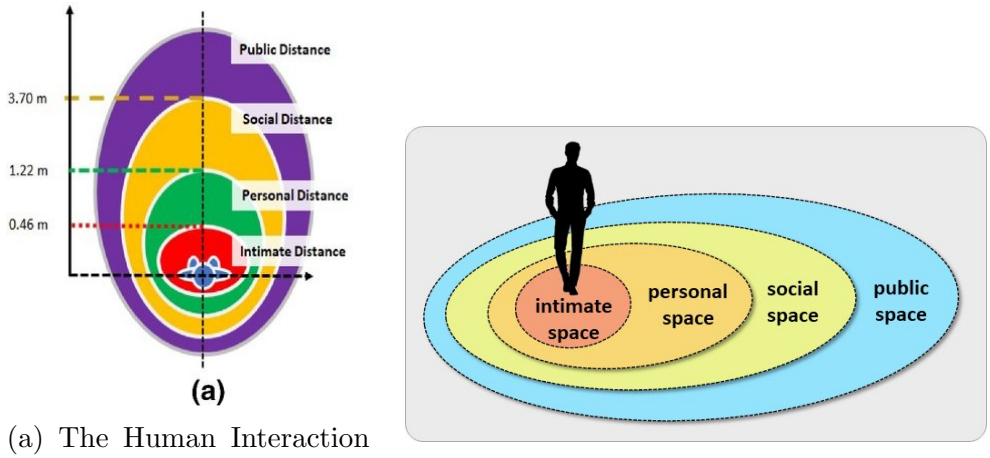
Figure 1.2: Pepper bot is an example of social bot. In the figure, it is helping enforce social distance and greeting during COVID-19 pandemic [79].

In social robotics, having the robot to move in a socially acceptable manner is commonly referred as social navigation. Social navigation is a desired outcome as robots become increasingly prevalent in daily activities with people around it, therefore the need for such capabilities grows. In this work, we include the social transportation problem, where the robot must not only navigate socially but also do so while transporting an object.

Problems related to social navigation include human prediction [60], crowded spaces [84], navigating around pedestrians [39], among others. Each work is unique given the problem proposed, the works may consider dynamic [63] or static environments [12], solo person or multiple groups [83], indoor [44] or outdoor spaces [106]. Because of the variety of scenarios that can be proposed, the methods used and the analysis of results can vary significantly. The common factor between all works is having a robot to navigate with people around it.

One way to evaluate the success of a social navigation solution is by considering how comfortable humans feel with the path taken by the robot. In order to take comfort into account, in robotics, one of the most common ways is by evaluating the robot's position in relation to the person. This approach is based on the concept of *proxemics/personal spaces* [64, 83], which delineates the areas around an individual where they feel most comfortable having others, including robots, nearby. Although nomenclature may change from work to work, the concept described remains similar.

The general idea in using personal spaces is to make the robot respect a defined region while navigating, resulting in a socially-aware robot. Research in this area focuses on navigating robots in a manner that avoids entering an individual's social space. Two examples of such defined personal space regions are illustrated in Figure 1.3a.



(a) The Human Interaction Area by [65].

(b) Personal space employed by [18].

Figure 1.3: Examples of defined personal spaces/proxemics from the literature.

Despite the growth of social robotics and the increasing use of mobile manipulators, few studies have jointly addressed social navigation for mobile manipulators, and even

fewer have considered social object transportation. Furthermore, there is a clear lack of personal space models specifically tailored to this type of robots. However, it is crucial that mobile manipulators take into account the human presence in their path, not just by avoiding individuals who are in the way, but by considering social behaviors. For general obstacle avoidance objectives in robotics, it usually does not matter whether the vehicle passes near or far from the obstacle, as long as there is no collision. When dealing with human avoidance, a robot carrying a pointed object must avoid approaching a person's eyes dangerously. In other words, when dealing with human interactions, avoiding collisions is only the starting point.

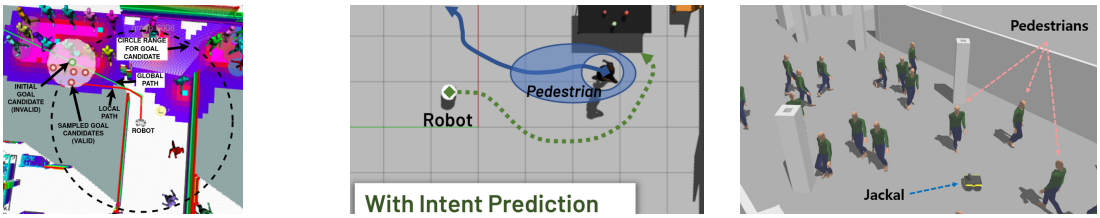
This work aims to develop a social autonomous navigation technique for a mobile manipulator robot operating in environments populated by people. The robot's task involves carrying a load from one location to another, our objective is to complete this task achieving a socially acceptable navigation given a certain metric, ensuring the safety and comfort of all humans involved.

1.2 Motivation

Addressing the problem of autonomous robot navigation in environments shared with humans is essential for activities that require cooperation, comfort, and safety. Robots are already being integrated into such activities, including operations in industrial sites [17], search and rescue missions [31], and domestic applications [24], all of which directly depend on the ability of robots to interact with people. For example, in disaster response scenarios [24], robots assist in searching for survivors under hazardous conditions, while in healthcare settings [1], they support medical staff by monitoring patients or delivering supplies.

In this context, where robots and humans coexist, the challenge of social navigation emerges. Social navigation is crucial in social robotics, yet it is difficult because robots must account for subjective human perceptions to be perceived as socially acceptable. Techniques developed in this area need to consider the presence of nearby humans so that tasks can be performed effectively without startling individuals, causing discomfort, or leading to accidents. To date, a variety of techniques and metrics have been proposed to promote socially aware behavior, however, most research on social robotics navigation has focused solely on mobile robots without manipulators. The Figure 1.4 shows three different works on social navigation with mobile robots.

These techniques could only be directly applied to mobile manipulators if the arm were assumed to remain static. Such an assumption not only reduces the robot's

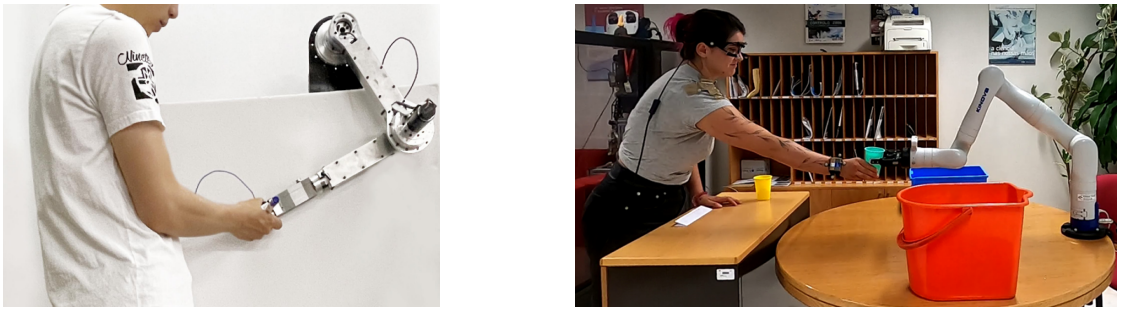


(a) Pepper mobile robot. [84] (b) Generic mobile robot. [60] (c) Jackal mobile robot. [63]

Figure 1.4: Social navigation works in mobile robots with different robots.

maneuverability but also forces the choice of a fixed arm position, which may not ensure comfort for nearby people throughout the entire path.

It is also worth noting that, within the field of manipulators, social aspects of manipulation have been overlooked, even though numerous works exist in HRI particularly in the area of cooperative human-robot manipulation such as the ones shown in Figure 1.5. Nevertheless, because manipulators are fixed, the concept of social navigation does not apply directly to manipulator-only robots.



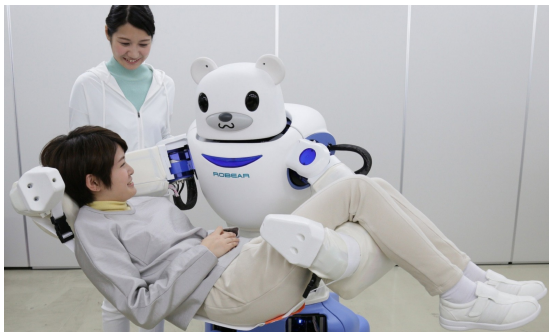
(a) Writing-aid task by a manipulator. [3] (b) Handover task by a manipulator. [51]

Figure 1.5: Manipulators performing HRI tasks.

For tasks of carrying and transporting, mobile manipulators represents a versatile tool by having a mobile base and one or more robotic arm mounted to the base. Due to this configuration, the robot has the ability to perform dexterous tasks with the arm (manipulation), combined with mobility to move to different regions, giving these robots versatility. Therefore, mobile manipulators are especially used due to their ability to manipulate and transport objects, simultaneously or not. Given these capabilities, they have been increasingly studied and deployed in populated environments, such as to perform domestic chores or carry debilitated humans, as illustrated in Figure 1.6.

In this context, research on social navigation techniques for mobile manipulators becomes relevant. The importance of a specific approach can be justified by a everyday example: it is desirable for a mobile manipulator, when passing by a person, to not only avoid immediate collision but also adjust the position of its manipulated object to maintain additional distance from sensitive areas, such as the eyes. Achieving this requires consideration of the combined configuration of the robot's base, its manipulator arm, and

(a) Spot-arm [23]. A quadruped robot with a manipulator arm collecting clothing and placing it in a basket.



(b) Bearbot [62]. A mobile robot with two arms carrying a person for healthcare assistance.

Figure 1.6: Two examples of mobile manipulators designed for different applications: (a) Spot-arm for domestic tasks; (b) Bearbot for healthcare support.

the object being transported, highlighting the complexity of socially aware navigation for mobile manipulators.

Research focused on autonomous navigation methods for mobile manipulators in populated contexts is crucial for preventing accidents and minimizing potential harm. Furthermore, it increases comfort for the humans present, thus increasing the acceptability of robots. The study of the area of social navigation and transportation for mobile manipulators aims to achieve the successful and safe integration of these robots in the workplace.

1.3 Problem

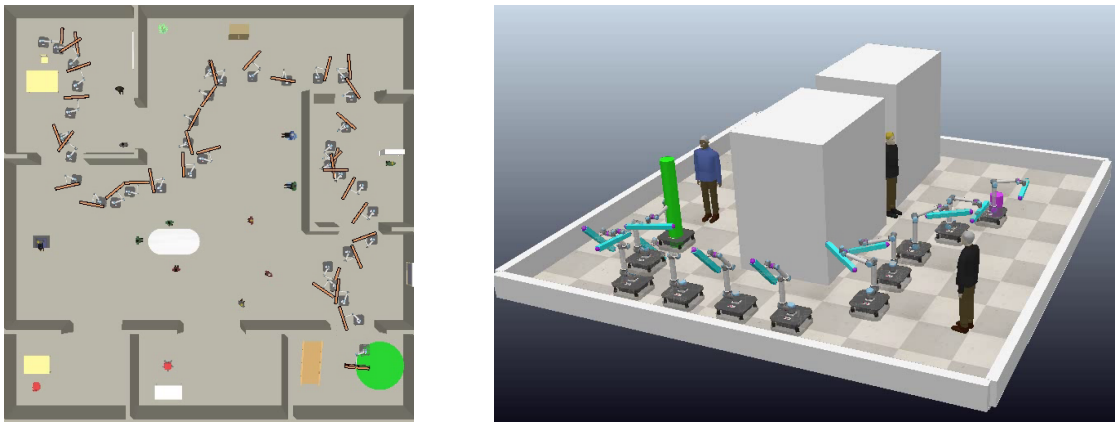
Despite advances in social navigation for mobile robots and works on HRI for manipulators, integrating the aspects into a single mobile manipulator system remains an open problem. Furthermore, research explicitly addressing object transportation that accounts for the object's social impact has, to the best of our knowledge, not been conducted, representing an important advance in our work.

In this work, we consider a static environment in which humans do not move, which represents our most significant limitation. This simplification allows us to begin exploring the field of social navigation for mobile manipulators. Although not entirely realistic, this scenario can be justified in environments such as offices, where humans are expected to be at certain points. The simplification is necessary because mobile manipulators have a high number of degrees of freedom, making full-body motion planning computationally intensive and slow [75].

Given the context described above, a high-level definition of the challenge we aim to tackle in this work is described in Problem 1.

Problem 1. *Given a known static and human-populated environment, a mobile manipulator needs to navigate from one place to another while transporting a non-negligible object in a socially-acceptable manner.*

We can split this problem into two distinct scenarios: a two-dimensional (2D) planar case and a three-dimensional (3D) one, with social navigation examples of both in Figure 1.7. The 2D scenario arises when the robot arm is a planar manipulator, which simplifies the problem formulation and allows for direct comparison with existing approaches, as most of them are constrained to two dimensions.



(a) 2D Social Navigation Result. (b) 3D Social Navigation Result.

Figure 1.7: 2D and 3D social navigation results. The figure shows an incremental plot illustrating the robot's movement, with its objective to reach the green target.

However, mobile manipulators are likely to operate in three-dimensional space, as the arms of most existing manipulators move in 3D. Therefore, even though the navigation occurs on a 2D plane, the manipulator can move and interact in three dimensions, making it necessary to explicitly account for the Z-axis in the planning process. Three-dimensional social navigation has been explored for aerial robots and drones; but since ground-based robots cannot fly, approaches developed for drones are not directly applicable. This creates a new challenge in social navigation: developing methods for mobile manipulators that explicitly consider three-dimensional environments.

Furthermore, given a mobile manipulator with an arm that manipulates in 3D, a subproblem arises. As most existing studies focus on planar environments, the associated social cost or discomfort functions are also limited to two dimensions (x-axis and y-axis). Consequently, the subproblem of formulation a three-dimensional social discomfort model suitable for mobile manipulators is also addressed in this work.

1.4 Objectives

Our main objective in this work is to propose a framework for social navigation and transportation for mobile manipulators in static scenarios. We understand social navigation as the action of navigating while considering a social metric, in our case, discomfort, that will be objectively defined further in this work. To achieve this objective, we need to accomplish the following specific objectives:

1. Perform socially-aware navigation actuating both mobile base and manipulator concurrently;
2. Consider different types of objects being transported;
3. Select preference towards robots parts to be closer than others to people;
4. Create a three-dimensional discomfort model to compute discomfort in all 3D.

1.5 Contributions

Overall, this dissertation introduces two complementary approaches for socially aware navigation for mobile manipulators. The first is a full-body motion planner that jointly coordinates the base and the manipulator. While slower, exploits the coupled motion of the entire robot, being able to find paths otherwise decoupled motions would struggle. This method is best suited 2D scenarios, given they are simpler. The second is a decoupled planner, where the base is planned first and the manipulator subsequently. This strategy yields faster results, making it more appropriate for 3D environments, although it does not have the same level of coordination as the coupled approach. Both methods are guided by a novel discomfort model proposed in this work, the multi-point approach, that allows for consideration for the mobile base, the arm and the object being transported. The last one also uses a novel discomfort three-dimensional model we propose for 3D problems.

The contributions can be summarized as follows:

1. **Full-body Socially-aware 2D Motion Planner.** We propose a 2D coupled motion planner that minimizes social discomfort in static scenarios by jointly actuating both the mobile base and the manipulator in a coordinated manner. The planner operates by minimizing a discomfort function specifically defined in this work.

a) **Social multi-point cost function.** We introduce a novel discomfort function based on a multi-point formulation. Unlike previous works that only considered the robot base, this approach allows to account for the entire robot and the object being transported. Furthermore, it allows for weighting different points on the robot, making it possible to prioritize certain parts over others to better reflect human comfort preferences.

2. **A three-dimensional discomfort model.** To the best of our knowledge, no existing work has defined a 3D social discomfort model targeted for ground-based robots. We address this gap by proposing a discomfort formulation that combines a novel Z-axis discomfort equation with established XY-plane models.

   a) **Proposal of a Z-axis discomfort equation.** Building on the relationship between discomfort, human body sensitivity, and safety, we design a new Z-axis discomfort equation. This model enables consideration of vertical sensitivity regions around humans, not possible with other models.

3. **Decoupled Socially-aware 3D Motion Planner.** Extending beyond planar navigation, we propose a framework for full 3D social navigation. Unlike the full-body coupled planner in the first contribution, this method employs a decoupled approach, which, while not guaranteeing optimality, ensures reliable computation times. As a result, it enables the generation of socially aware paths suitable for three-dimensional environments which the first method struggles with.

   Part of the work presented in this dissertation has been published at the *2025 IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN 2025)*. The first publication corresponds to the bidimensional component of this study and is entitled *Socially-Aware Object Transportation by a Mobile Manipulator in Static Planar Environments with Obstacles* by Caio C. G. Ribeiro, Leonardo R. D. Paes, and Douglas G. Macharet. The second publication focuses on the development of the proposed three-dimensional discomfort function, presented in the paper *Beyond the Plane: A 3D Representation of Human Personal Space for Socially-Aware Robotics* by Caio C. G. Ribeiro and Douglas G. Macharet.

Chapter 2

Background

In this chapter, we first introduce mobile manipulators, outlining their main characteristics, typical applications in research, and a brief overview of traditional motion planning methods that do not account for social aspects. We then review relevant work in social robotics, with a particular focus on social navigation. Finally, we discuss social metrics, summarizing how researchers define and evaluate social acceptability in robot behavior.

Regarding social navigation and manipulation, in the literature, works related can be divided into three major robots types, each employing different approaches: mobile robots, manipulators, and mobile manipulators. Within each of these categories, social robotics can be applied with varying strategies and objectives. It is crucial to examine social navigation independently for each type to address their unique requirements, hence, we distinguish each in different sections inside social robotics.

We also highlight recent surveys and reviews in the fields of social robotics and human–robot interaction (HRI). These works provide comprehensive overviews of the current state of the art, summarizing key methodologies, experimental findings, and identified research gaps. By consolidating and analyzing the contributions across multiple studies, these surveys serve as valuable resources for researchers aiming to gain understanding of social robotics and to identify directions for future investigations. The following studies provided information that guided the development of this dissertation.

In the field of Human-Robot Interaction (HRI), safety has been a central concern. Lasota *et al.* [50] present an extensive survey addressing motion planning, control strategies, prediction, and psychological factors that influence human-robot interactions. Similarly, Rubagotti *et al.* [73] focus specifically on perceived safety, defining the concept and evaluating six classes of robots with respect to human comfort. Still on HRI, Wang *et al.* [97] reviews machine learning methods applied to HRI and discuss factors that affect human comfort and strategies to improve it.

Focusing more specifically on social robotics, Henschel *et al.* [33] analyze the characteristics that define a social robot, providing insight into what qualifies a system as socially interactive. Mahdi *et al.* [54] present an exhaustive survey of 344 robots explicitly designed for social interaction, identifying trends, gaps, and offering guidelines for future designs. Sheridan *et al.* [78] provide a psychological perspective on social robotics,

categorizing current research into three primary areas.

Finally, Mavrogiannis *et al.* [56] offer a comprehensive review of social navigation, detailing the remaining challenges in enabling robots to navigate while respecting human social expectations. Collectively, these surveys provide a foundation for understanding the state of HRI and social robotics.

2.1 Mobile Manipulators

Mobile manipulators are a class of robots characterized by being able to move, like mobile robots, and manipulate objects, like robotic manipulators [91]. These robots typically consist of a mobile base, which allows them to navigate through various environments, and an robotic arm (manipulator), which enables them to perform manipulation tasks [53].

Figure 2.1: Mobile Manipulator performing a pick-and-place action using suction cups for supermarket applications, at TU-Delft during 2024 Robotics: Science and Systems conference. [90]

To better understand mobile manipulators we first describe mobile robots and manipulators separately.

Mobile robots are capable of moving freely throughout their environment, as they are not anchored to a fixed point on the ground. Drones and wheeled robots are examples of mobile robots [43]. Their mobility mechanisms, which may include wheels, tracks, or legs, allow them to adapt to various types of surroundings [80]. These specific type of robots are useful for scenarios in which it is needed constant reallocation, which there are several applications in a wide range of fields. They can function as inspection vehicles in warehouses [70], planetary rovers [80], service robotics, such as vacuum cleaning and

transporting medication [84]. The ability of mobile robots to autonomously navigate and perform tasks in dynamic and unstructured environments underscores their significance in both research and industry.

Robotic manipulators are robots which mimics the articulated human arm, consisting of multiple links and joints [43], providing a high degree of dexterity and flexibility and are designed to interact (manipulate) with objects. Robotic manipulators are widely used in manufacturing sites [52], where they perform repetitive and precise tasks [42] such as assembly, welding, and painting. They are also increasingly employed in medical applications, like robotic surgery [107]. The ability of robotic manipulators to perform intricate and repetitive tasks with high precision makes them indispensable in various domains, driving advancements in automation and productivity.

Mobile Manipulators combine both types of robots described, gaining the base mobility and dexterity of an articulated manipulator [91]. The combination of these two functionalities allows mobile manipulators to interact with and manipulate objects within their surroundings while also being able to move to different locations. This versatility makes them suitable for a wide range of applications, including industrial automation [108] , service robots [90], healthcare [25], and more [91]. It is evident that these applications will likely involve people in the surroundings, making social navigation for mobile manipulators highly relevant.

2.1.1 Non-Social Motion and Control Approaches

To lay the groundwork for developing a socially aware motion planner for mobile manipulators, we first review traditional, not social, motion planning approaches. For an in-depth overview of these approaches, we rely on recent systematic reviews, particularly the works of Youakin and Ridao [103], Sandakalum and Ang [75] and Thakar *et al.* [91], which provide comprehensive and relatively recent discussions on the subject.

Since mobile manipulators typically have a high number of degrees of freedom (DoF), their motion planning is inherently challenging. Broadly, two main approaches are commonly employed: decoupled and coupled planning. Coupled approaches plan the motion of both the mobile base and the manipulator arm simultaneously. While this enables greater coordination between the base and arm, it significantly increases the computational complexity of the problem due to the higher dimensionality [75]. In contrast, decoupled approaches compute the motion of the base and the arm separately. This strategy reduces computational effort but may sacrifice coordination, potentially resulting in less efficient or suboptimal motions [75].

For coupled motion planning, several new methods have been proposed in recent years [75]. Each presents distinct advantages and limitations, yet two approaches remain the most established and widely used: sampling-based and optimization-based methods.

Sampling-based methods still are popular [5, 75]. A study from 2018 [103] compared several planning approaches and found that sampling-based methods outperform others in terms of solution time, although often at the cost of solution quality. Algorithms such as Probabilistic Roadmap (PRM) and the Rapidly-exploring Random Tree (RRT) are well-suited for high-DoF planning, particularly as offline planners. They can generate whole-body motions relatively quickly, but may require significant time to optimize the complete trajectory from start to goal, especially when the mobile manipulator involves many joints. Sampling-based methods are also integrated into MoveIt! [45], further demonstrating their practicality and widespread adoption. However, while often faster than other planners, sampling-based methods can slow down for complex manipulators due to the higher DoF and cost of collision checking.

Optimization-based techniques are also being explored for whole-body motion planning in mobile manipulators, often with Model Predictive Control (MPC) [58, 32, 91]. Control methods can be robust and allow for responsive behaviors while handling constraints. However, they are often complex to implement and require detailed system-specific models[45], which limits their generalization. Adapting them to new robots demands significant recalculations to consider the dynamics and constraints of each robot.

Those two tends approaches to be the most classical methods, but in recent years alternative methods have emerged, although they still require further adoption to become well established. Machine learning techniques have been explored with reinforcement learning [105, 45] and imitation learning [100]. Hybrid methods combining machine learning with MPC or sampling-based planners have also been proposed [45, 75]. More recently, Riemannian Motion Policies (RMPs) and Geometric Fabrics was used for reactive navigation and collision-avoidance [5, 57].

However, the most common approach remains the decoupled strategy, rather than full whole-body planning [75], as it is generally easier to implement and faster to compute. In this approach, the mobile base can employ a variety of existing techniques developed specifically for mobile robots to compute its path, while the manipulator is planned independently as a separate problem.

2.2 Social Robotics

Social robotics is a relatively new field within robotics [54], aimed to give to robots social capabilities. The term social robotics is sometimes used interchangeably with Human-Robot Interaction (HRI), as discussed before. According to [7], HRI is any kind of action that involves another human being or robot, which includes social robotics. In this work, we will use social robotics as the primary term, although no universally agreed-upon definition exists [33], we will apply it when the robot account for any social behavior, rather than any interaction with humans. However, we will cite works within the general field of HRI, as they can also include socially-aware robots. In this section, we describe the current related works for each type of robot.

2.2.1 Social Robotic Manipulators

Manipulators are robotic arms, and they have been extensively studied in environments with humans [73], however, specially in HRI tasks and not necessarily in social robotics, where the robot would need to consider human preferences. Special attention to manipulators in social contexts represents a significant gap in the literature, where we could not find several works. One indicator of this gap is the comprehensive survey by Mahdi *et al.* [54], which, despite its broad definition of social robots, excludes robotic manipulators from its review precisely because no manipulators were found to be explicitly designed for interaction with people. This observation is further reinforced by [73] concluding that most works involving physiological assessment focus on industrial manipulators rather than socially oriented ones.

Still, some works labeled as human-robot [51], human-aware [52] on manipulators could be considered social robotic manipulators. For instance, Lastrico *et al.* [51] developed a system where the manipulator picks objects from a person in a handover task, simulating human carefulness and generating expressive movements. Their results showed that this approach improved the time required for humans to complete their part of the task.

In another collaborative task, a dual-armed robotic manipulator was used to build a bridge on a table in collaboration with a human partner [76], as shown in Figure 2.2. The authors defined five distinct strategies for the robot to cooperate, and participants consistently showed a preference for one particular strategy. Although the study does not explicitly use the term social robotic manipulator, the definition of multiple strategies

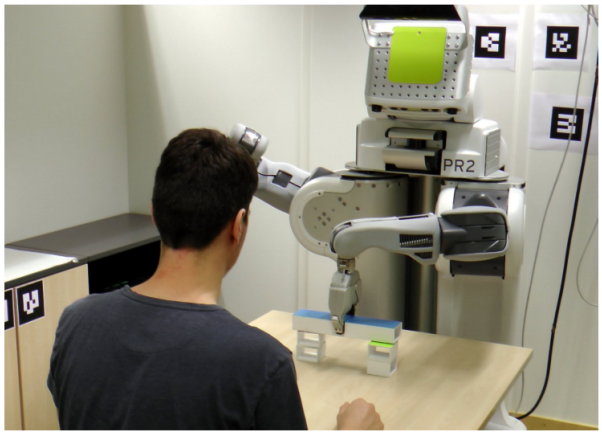


Figure 2.2: Example of a socially-aware task performed by the PR2, a two-armed robot, collaborating with a human to build a bridge on a table [76].

and the evaluation of human preferences can be defined as social robotics.

The use of emotions to guide a robot can also be considered social behavior. The robot presented by [42] is a social robotic manipulator capable of detecting facial expression, and when a person shows happiness, it handles a flower to the person. Therefore, the robotic manipulator can be considered socially-aware.

The works by Sisbot *et al.* [86] and Mainprice *et al.* [55] both proposes human-aware motion planners to solve manipulator socially deliver an object to a person. They use safety and visibility constraints, and the former solves with a A* algorithm over a constructed grid whereas the latter employs a sampling-based (T-RRT) tactic.

In contrast to collaborative tasks, our work focuses on coexistence, given that in social navigation robots and humans are not directly interacting with each other. Within this perspective, other studies have explored coexistence aspects of manipulators operating in human-populated environments. These works typically address collision avoidance, occasionally incorporating predictions of human motion, but do not consider explicit social metrics. Nonetheless, they represent the closest approaches to the direction pursued in this dissertation.

With the objective to prevent accidents [59] employs proximity sensors mounted on a manipulator and proposes a real-time path planning technique to avoid people around it. Similarly, [61] use a technique based on repulsive forces to control the robot, avoiding dynamic obstacles which include humans or objects. Continuing in the idea of collision avoidance, the work by [21] proposes a methodology based on potential fields for a manipulator to avoid a potential collision with a human entering its workspace, using Kinect-type sensors. The work by [52] proposes a human-aware motion planning framework, predicting human next movements using graph neural networks to compute collision-free motions for manipulators.

Although these collision-avoidance works are closer to the focus of this dissertation in the sense of coexistence and fall within the field of HRI, they arguably do not qualify

as social robotics, since social behavior was not explicitly considered.

The extensive review on perceived safety in physical HRI by Rubagotti [73] provides further insights into the role of manipulators in human-centered scenarios, where several works could be considered part of social robotics. The survey highlights that, within the field of perceived safety, manipulators account for the largest number of published studies. Perceived safety is primarily associated motion smoothness, distance to the person, and velocity. However, studies on manipulators have not addressed proxemics or personal space zones.

Overall, the field of social robotic manipulators seems restricted to collision-avoidance of humans and perceived safety in the context of coexistence, while collaboration appears to have more research towards socially-aware robots, but still with sparse works.

2.2.2 Social Mobile Robots

Mobile robots are the class of robots most frequently studied in social robotics. Unlike manipulators or other fixed systems, mobile robots possess a platform that allows them to move freely through the environment, making them especially suitable for research in human-populated spaces.

Socially-aware mobile robots are designed to account for human social expectations, and while several aspects of socially interactive mobile robots have been explored, this dissertation specifically focuses on robots capable of navigating environments while respecting human social preferences. Accordingly, our work centers on social robot navigation, or simply social navigation [84, 63, 12]. Social navigation differs from conventional navigation in that the robot adapts its trajectory based on human preferences, rather than solely on geometric or efficiency criteria. This distinction is illustrated in Figure 2.3.

Social navigation has advanced significantly since its initial applications in robotics. According to a survey by [56], the earliest works in this field can be traced back to tour guide robots and nowadays, the topic intersects with multiple adjacent areas and receives substantial funding from various organizations.

For mobile robots be able to be socially-aware, there are multiple tactics ones could employ, differing depending on the task proposed. Perhaps the simplest form of social mobile robot navigation is human collision avoidance. In this scenario, the mobile robot navigates the environment while avoiding collisions with humans. The main difference between avoiding a human and any other obstacle is the dynamic behavior of people. Early works in social navigation set this standard, however, humans were non-reactive and merely considered as dynamic objects [56], the social aspect barely holds truth.

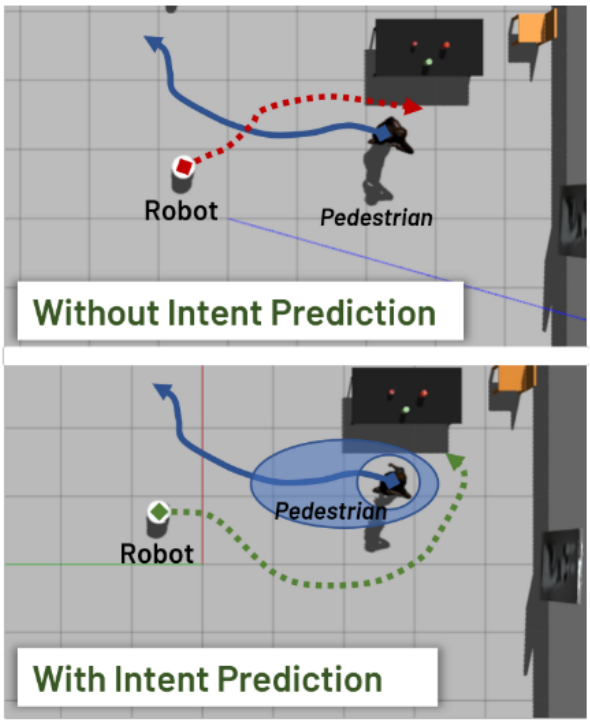


Figure 2.3: Comparison of paths taken by a mobile robot: The bottom path shows the robot predicting human intent for smoother navigation, while the top path lacks this prediction, resulting in the robot going right in front of the person [60].

Mobile robots begin to exhibit more social behavior when human-avoidance is combined with prediction techniques. Researchers aim to predict potential human movements and use these predictions to navigate the robot in a way that avoids collisions, even when humans are in motion [34, 60, 63]. Successfully incorporating human prediction into collision-avoidance navigation has been demonstrated using reinforcement learning [41], as well as neural networks [2, 74]. Still, predicting human behavior for navigating without collision not considering human preferences may not result in successful social navigation or a truly socially-aware mobile robot.

Closer to the approach in our work, recent studies have advanced by incorporating not only the prediction but also social metrics into the navigation of social mobile robots. The work by [63] predicts future states of humans and generates a path that considers these predictions, as well as efficiency and social acceptance. They use Monte Carlo tree search to generate the path and evaluate its quality with a novel socially-aware navigation score metric. Similarly, [60] predicts human intention information used for a novel navigation algorithm that uses social and personal space constraints, based on reinforcement learning and the quality of the path is calculated using Average Personal Space Deviation.

Other works also delve in this approach, with different metrics or tasks. The paper by [109] uses deep reinforcement learning for a social mobile robot navigation in a crowd. Using SLAM and neural networks, [85] uses a mobile robot to generate a map in an human-populated environment caring to social norms, ensuring comfort and safety. The

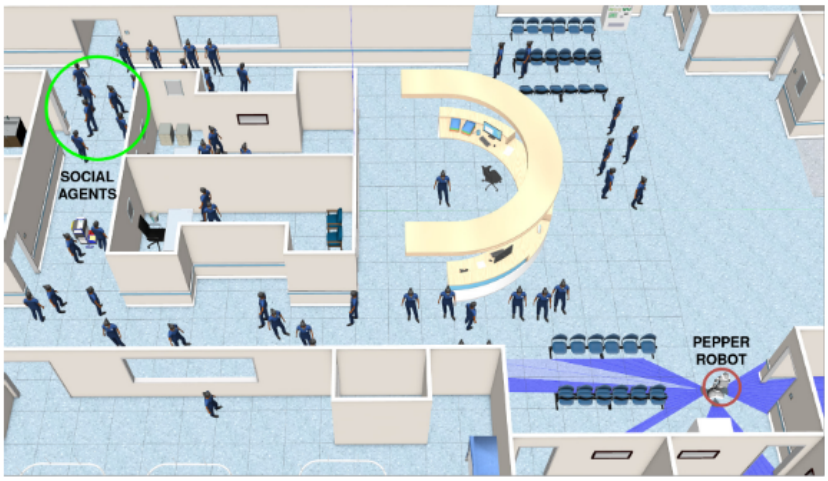


Figure 2.4: Example of a human-populated environment where a mobile robot needs to go from one place to another in a crowded scenario [84].

work by [12] extends multi-agent path finding to consider people personal intentions in constrained environments for a social navigation.

However, human intention prediction is just one tool for social mobile navigation. Some works do not directly rely heavily on predicting human behavior. For example, [84] presents a framework capable of socially navigating a mobile robot using a heatmap, multi-layer path-planning, and control. This approach incorporates a discomfort function to measure the social acceptance of the path. Additionally, the work by [83] computes paths for approaching static groups of people in a socially-acceptable manner, using proxemics to determine appropriate locations to reach the group. Furthermore, a common approach to the social navigation problem involves using Potential Fields with a Social Force Model [60]. This method calculates repulsive forces from obstacles and people, and attractive forces towards the goal destination, additional forces can be computed based on collision predictions for a better socially-aware navigation [47].

Sampling-based techniques, and specifically, RRT-based, have been employed for social navigation in mobile robots in several works. Barnaud *et al.* [6] investigated corridor crossing navigation with proxemics and employed Risk-RRT as the planner, also concluding that the method should be extendable for other costs types. The afromentioned work by Silva *et al.* [84] utilize RRT* and Informed RRT* algorithms in conjunction with a social heatmap to generate socially-aware navigation paths. Chi *et al.* [14] introduced the Risk-RRT algorithm in social settings, integrating a comfort and collision risk map into the RRT framework to facilitate socially-aware navigation. In subsequent years, they extended this work by developing Risk-RRT*[15] and Risk-Informed-RRT* [16], further enhancing the method's ability to account for both social comfort and safety during navigation.

In summary, the field of social mobile robots has seen a significant surge in research in recent years, incorporating a diverse array of techniques and tools. While many papers

claim superior performance, the evaluation largely depends on the metrics used, which vary across studies. It is also important to note the considerable variation in the techniques employed to address these challenges. The methods reviewed here are solely for mobile robots, and may not translate directly for mobile manipulators.

2.2.3 Social Mobile Manipulators

In mainstream media, social mobile manipulators robots such as the Asimo Robot [94], NAO [93], and Pepper [79] frequently appear, shown in Figure 2.5. These robots, classified as humanoid robots, possess arms, which are manipulators, and mobility, allowing them to be categorized as mobile manipulators. This common occurrence may happen due to humanoid bots being built to specially deal with humans, in social scenarios. In earlier developments, it was common for robots intended for human interaction to be equipped with two manipulators to emulate the human body [54]. Consequently, humanoid or human-inspired robots remain the most common type encountered in both the literature on social robotics and in commercially available platforms [25].

(a) Asimo Robot. [94]

(b) NAO Robot. [93]

(c) Pepper Robot. [79]

Figure 2.5: Three examples of social humanoid robots: Asimo, NAO, and Pepper. These robots possess two arms, which function as manipulators, and have the capability to move through their environment, thus classifying them as mobile manipulators.

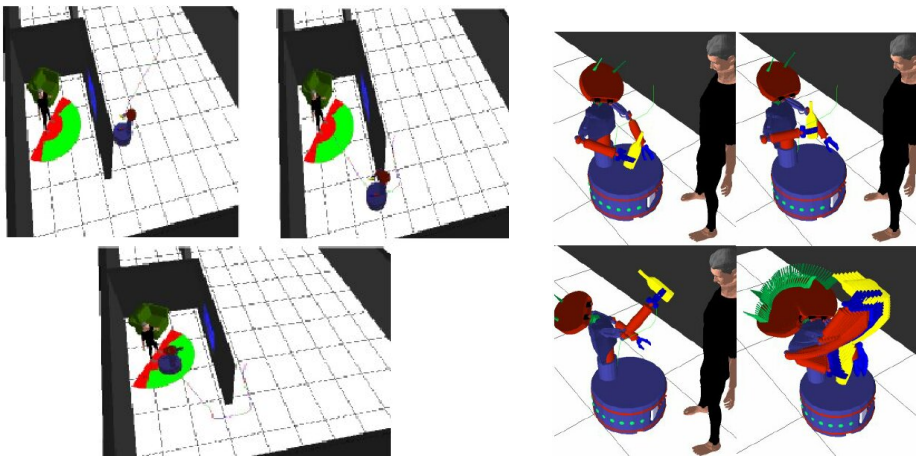
However, there is a noticeable lack of studies in social navigation that consider the motion of both the mobile base and the manipulator while carrying an object. This gap may be explained by the findings of Mahdi *et al.* [54], who report that most social robots either do not include a manipulator or use it primarily for gesturing rather than for manipulation tasks. Consequently, the scarcity of social robots equipped with functional manipulators likely contributes to the absence of studies in this area.

Furthermore, even when a mobile manipulator is capable of performing such operations, literature often describes scenarios where the robot remains stationary while using

the manipulator, thereby not utilizing the mobile base [76]. In other instances, the robot retracts its arms to simplify navigation.

The works by [84] and [40] theoretically consider mobile manipulators as their tested robots, but they never utilize the manipulator's capabilities, effectively reducing the problem to that of a mobile robot. Similarly, [76] considers the PR2 robot, a mobile manipulator, but only utilizes the robot's arms without employing the mobile base for operations, thus reducing the problem to that of a manipulator-only robot.

Research in social navigation for mobile manipulators that consider human preferences is sparse. The work by Sisbot *et al.*[88] proposes a human-aware motion planner and employed a mobile manipulator, however, focused on the mobile navigation aspect. Employing the same mobile manipulator, they continued to explored this area performing a fetch-and-carry task in [87] considering safety, visibility, and arm comfort. In this work, the proposed approach involved planning a socially aware motion for the mobile base, illustrated in Figure 2.6a, and once the base reached its goal, a second socially aware motion was planned for the manipulator to deliver the object, illustrated in Figure 2.6b. This study was the closest found to ours, as it integrates both mobile navigation and manipulation within the same context, although not in a concurrently.



(a) Social Navigation (b) Social Manipulation.

Figure 2.6: Results of a social fetch-and-carry task performed by a mobile manipulator [88].

Although the two studies are similar to the task proposed in our work, they differ in the focus, where their object is to interact with humans whereas our approach emphasizes coexistence. In addition, our work explicitly accounts for the object being transported.

More recently, there are works that also consider human-aware navigation of mobile manipulators. In [49] a mobile manipulator adapts its navigation when there are workers nearby, increasing safety and comfort, however, the manipulator remains static. Pupa *et al.* [68] propose a dynamic planner for safe navigation of mobile manipulators. Their approach employs a traditional non-social planner (RRT-Connect) to compute an initial

Figure 2.7: Mobile manipulator Tidybot++ cleaning a countertop [100].

path and then reactively adapts to dynamic human presence using an online optimizer that enforces human safety. However, the work relies on safety standards, it does not explicitly model or address social proxemic comfort zones.

In counterpart to human-aware navigation studies, other recent works involving mobile manipulators in human-populated environments often do not explicitly address social navigation. For instance, [36] presents a collision-free navigation method for mobile manipulators in cluttered, office-like environments using search-based planning. In other work, a new robot is designed to assist older adults by carrying heavy objects [13]. The work by [90] employs a mobile manipulator to pick supermarket groceries from shelves in a human-shared retail environment using behavior trees. In [89] a excavator is autonomously operated on a construction site. Wu *et al.* [100] introduce a platform for mobile manipulators to learn several household tasks, environments where humans would naturally be present but are not explicitly considered. One example of the tasks the Tidybot++ can perform is cleaning a countertop, showed in Figure 2.7. Despite these advances, none of these studies specifically address social navigation aspects.

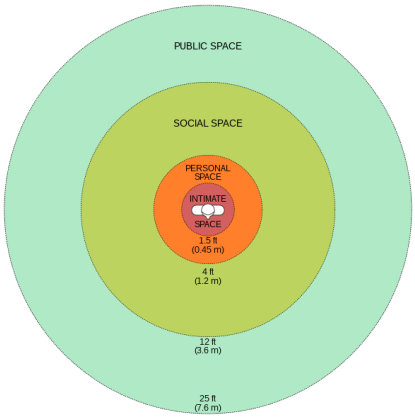
Overall, the field of mobile manipulation is not well explored in the sense of full-body motion for a socially-aware navigation, and even less when considering social object transportation.

2.3 Personal Space Models

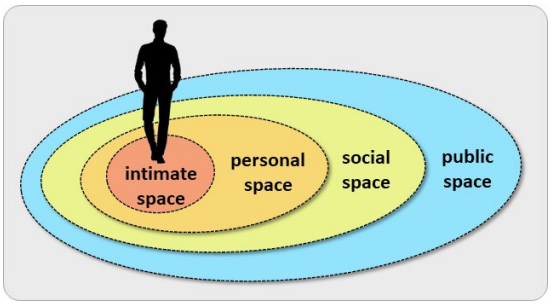
In social robotics, it is essential to define appropriate social metrics in order to quantify the degree to which a robot behaves in a socially acceptable manner. However, such metrics are highly subjective and can depend on numerous contextual and human factors, and to date no universally accepted approach exists.

One common approach in social navigation is to account for social norms, particularly the preservation of human personal space, which is essential for fostering positive human-robot interactions [56]. Building on this idea, prior research has incorporated social cues into planning frameworks, allowing robots to navigate in a socially-aware manner and adapt their motion to human comfort [84, 82].

Personal spaces, also called *Proxemics* is a concept introduced in psychology by Hall [29]. In their work, Proxemics describe the distances at which humans feel most comfortable interacting with others. Subsequent research has extended this concept to social robotics, demonstrating that proxemic considerations also apply when humans interact with robots [60]. There have been numerous attempts to mathematically define the proxemics. Initially, they were presented by Hall as concentric circular zones [29], illustrated in Figure 2.8a. Personal spaces were later refined using Gaussian functions to form egg-shaped regions that more accurately reflect human spatial boundaries [46], illustrated in Figure 2.8b. This shape has been observed experimentally in simulations [6], providing further evidence to support this model. This approach remain in use, with some variations, in recent works [64, 60, 82, 35, 18].



(a) Illustration of Hall personal spaces [29] by [98].



(b) Elongated personal spaces by [18].

Figure 2.8: Illustration of personal space models where (a) is based on Hall's proxemics and (b) based on more recent front-elongated models.

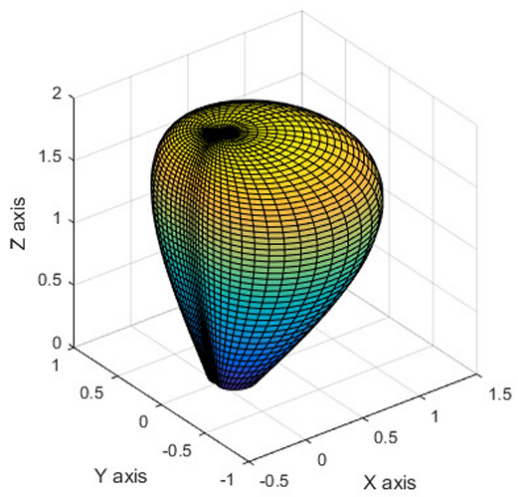
One aspect in common with these approaches to model personal space, is that they are restricted in two dimensions, on the XY plane, overlooking the vertical component. This bidimensional approach works for ground-based mobile robots, which operate predominantly on planar surfaces. However, when dealing with aerial robots (e.g., drones), robotic arms, or mobile manipulators, a three-dimensional representation of personal space becomes essential to ensure socially-aware interaction in the vertical dimension as well.

In recent years, three-dimensional representations of personal space have started to gain attention, particularly due to the growing popularity of drones. In human-drone interaction, several studies have explored proxemics with varying results. Duncan and

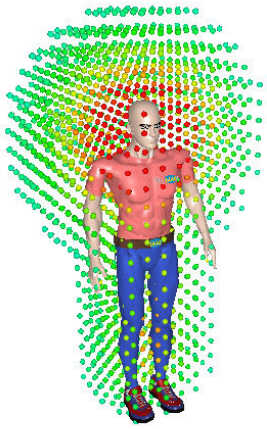
Murphy [22] investigated how drone height affects human comfort but found no significant preference. Similarly, studies by Han *et al.* [30] and Kunde *et al.* [48] also reported no clear preference for specific heights. In contrast, Yeh *et al.* [102] observed that people tended to approach drones more closely at a height of 1.2 meters compared to 1.8 meters. Likewise, Bevins *et al.* [9] found that the average preferred height for drone operations ranged from chest to eye level. On the other hand, Ariyasena *et al.* [4] reported that users preferred drones at higher altitudes, typically between 1.9 and 3 meters, in a follow-me task. More recently, Bretin *et al.* [11] also registered that participants preferred drones to operate above eye level rather than below, in co-existence scenarios.

Although findings vary, recent trends in drone proxemics indicate that humans tend to have altitude preferences, which appear to be task-dependent and context-sensitive. However, none of these studies have proposed a mathematical formulation to quantify discomfort along the vertical axis, let alone in a full three-dimensional personal space, as seen in existing 2D models.

The work by Garrell and Elizondo *et al.* [27], targeted for drone accompaniment tasks, is the only study found in the literature to propose a three-dimensional personal space formulation. The work extends an existing bidimensional model to a full three-dimensional representation showed in figure 2.9a, for the task of drone accompanying a person [27]. However, it assumes the 3D model as a direct extension of the 2D version, while this may be true, further research is desired.



(a) 3D Social Force Model [27].



(b) Distance costmap by [55].

Figure 2.9: Three-dimensional models used for social tasks. In (a), the Human's anisotropic factor in 3D view based on the Social Force Model, by [27]. In (b) The distance costmap that considers the region around the head more critical, by [55].

Beyond aerial robots, few studies have focused on comfort modeling in three-dimensional personal space. The closest approach found is the work by Sisbot *et al.* [86], which, in the context of socially-aware mobile manipulation, introduces a safety and vis-

ibility Criterion in 3D space. This model, among other factors, assumes that greater distances from the head and chest, as they are vulnerable regions of the human body, result in increased safety. Similarly [55] also adopts the region around the head is sensible, and creates a three-dimensional costmap, showed in figure 2.9b, merged with a visibility costmap to plan a comfortable motion for a manipulator to deliver an object to a person.

In our work, for a full socially-aware mobile manipulator, there is the need for a three-dimensional discomfort function. As reviewed, no previous works define a 3D model directed for coexistance targeted to manipulators. Hence, we propose a novel method for computing three-dimensional personal space, with a focus on robot navigation that prioritizes coexistence by minimizing discomfort to humans, rather than being tailored to specific interactions. While some works recognize that certain areas, such as the head and face [73, 19, 86, 55] are more sensitive and evoke greater stress in individuals, they have not fully explored the three-dimensional nature of personal space.

2.4 Contextualization of This Work

Our review indicates that the majority of studies in social robotics navigation focus on mobile platforms, typically incorporating Social Regions/Social Models, also commonly called Proxemics. There is a limited body of work on social robotics for manipulation applications, which primarily addresses human-robot cooperative tasks rather than coexistence in shared spaces, focus of our work. Research on social mobile manipulators is even more scarce; most approaches either retract the manipulator, reducing the problem to mobile robot navigation alone, or rely on metrics aimed solely at collision avoidance, neglecting considerations of human comfort and social models. In our work, we focus on social navigation of mobile manipulators, considering not only collision avoidance but also how comfortable people feel around the robot.

Table 2.1 summarizes the reality of the field, highlighting main works found in Social Robotics for different types of robots with different techniques and metrics. In this table, we prioritized the most recent works in the literature, considering different techniques or metrics and for Mobile Manipulators, only the ones in which the robot also moves its arms. The *Technique* column specifies the strategies used for robot navigation, while the *Metric* column describes the criteria for completing successfully the task proposed.

Table 2.1: Summary of key research works in social robotics, prioritizing recent advancements. The table categorizes different techniques and metrics employed.

Author	**Robot Type**	**Technique**	**Metric**
Silva S. (2023) [84]	Mobile	RRT* + Informed RRT*	Discomfort Function
Oh J. (2023) [63]	Mobile	Monte Carlo Tree Search	SANS (Score Metric)
Hirose N. (2024) [34]	Mobile	Vision Learning-based (SACSoN policy)	Personal Space
Zhou Z. (2021) [109]	Mobile	DRL + Online Planning	Collision Avoidance + Safe Distance
Truong X. (2017) [92]	Mobile	Proactive Social Motion Model	Social Force Model + Reciprocal Velocity
Singh K. (2022) [85]	Mobile	Adaptive Squashing Function based ANN	Proxemics and Social Norms
Kim M. (2024) [44]	Mobile	DRL + Image Planning	Individual Space Model
Nascimento H. (2020) [61]	Manipulator	Safety Contour Estimation + Repulsive Forces	Collision Avoidance
Lastrico L. (2023) [51]	Manipulator	Kinematics	Online Carefulness Classifier
Liu W. (2024) [52]	Manipulator	Graph + RNN human prediction	Collision Avoidance
Sisbot E. (2007) [87]	Mobile Manipulator	Human Aware Motion Planner	Safety + Visibility + Hidden Zones
Feidakis M. (2023) [25]	Mobile Manipulator	High-level Navigation	Element Intelligence Quotients (EIQs)
Bakker S. (2024) [5]	Mobile Manipulator	Symbolic Plans	Collision Avoidance
Ours	Mobile Manipulator	Risk-RRT*/ A* + Sampling-Based	Proxemics

In this dissertation, we employ a mobile manipulator to socially navigate a environment, were we employ two techniques: a coupled approach based on the Risk-RRT* algorithm, and a decoupled hybrid approach combining A* and a sampling-based planner. Both methods incorporate a social cost metric based in proxemic principles. The main contributions of this work, in comparison to existing studies, are the explicit consideration of the object being transported, the development of a whole-body motion planner based on proxemic reasoning using a novel multi-point formulation, and the proposal of a new metric for three-dimensional situations.

Chapter 3

Formalization

In this chapter, we present the formalization of our problem and the framework of our proposed approach. Establishing the characteristics of the environment is paramount, as these vary significantly depending on the specific scenario and task. The chapter is organized as follows. We begin by describing the overall problem, formulating it mathematically and formally. Following this, we introduce the framework that enables complete socially-aware navigation. Within the framework section, we provide a detailed description of its components.

3.1 Problem Formulation

In this section, we formally define the problem addressed in this work. At a high level, the problem can be described in more details as follows: given an initial and a goal configuration, the objective is to navigate a mobile manipulator from the initial to the goal configuration while transporting a load, in a static environment. The solution must jointly actuate both the mobile base and the manipulator arm in a coordinated manner and concurrently, ensuring that the resulting motion is not only collision-free but also socially aware, by avoiding human discomfort computed by a social cost function. The social cost function needs consider the base, the manipulator and the object as sources of discomfort.

Now, lets define it formally. Let $\mathcal{E} \in \mathbb{R}^3$ be a known static three-dimensional cluttered and human-populated environment, represented by a map $\mathcal{M}$. The space occupied by obstacles is denoted as $\mathcal{M}_{obs}$, while the free space is defined as $\mathcal{M}_{free} = \mathcal{M} \backslash \mathcal{M}_{obs}$. Obstacles and humans are distributed throughout the environment with different x and y positions, but are always set at ground level ($z = 0$). Although their z position is fixed, objects and people may have different heights, so the robot may need to move along the z-axis using its manipulator to reduce human discomfort and avoid collisions with obstacles.

Consider a set of spatially distributed individuals denoted as $\mathcal{N} = \{\mathbf{n}_1, \ldots, \mathbf{n}_m\}$. An individual $\mathbf{n}_i$ is characterized by its configuration $\mathbf{q}_i = (\mathbf{p}_i, \theta_i)$, where $\mathbf{p}_i = (x_i, y_i)$ denotes the position and θ_i the orientation. Each individual also has a height h_i assigned, meaning their stature while standing.

The robot chosen for this dissertation is a mobile manipulator with a holonomic base platform. Let $\mathcal{R}$ be the holonomic mobile base represented by its position $\mathbf{p}^{\mathrm{B}} = (x_{\mathrm{base}}, y_{\mathrm{base}})$, with kinematic model $\dot{\mathbf{p}}^{\mathrm{B}} = u$. The arm has k joints, with joint values of ψ.

Hence, the complete mobile manipulator configuration is:

$$\mathbf{q}^{\mathrm{MM}} = \begin{bmatrix} x_{\mathrm{base}} & y_{\mathrm{base}} & \psi_1 & \psi_{\ldots} & \psi_k \end{bmatrix}^{\mathrm{T}},$$

Given $\mathbf{q}^{\mathrm{MM}}_{start}$ and $\mathbf{q}^{\mathrm{MM}}_{end}$, start and end configurations, with $\mathbf{q}^{\mathrm{MM}}_{start}, \mathbf{q}^{\mathrm{MM}}_{end} \in \mathcal{M}_{free}$. The path connecting the start and end configurations is represented by $\sigma(\mathbf{q}^{\mathrm{MM}}_{start}, \mathbf{q}^{\mathrm{MM}}_{end}) \in \mathcal{M}_{free}$.

An optimal motion planner aims to determine an optimal sequence of configurations forming a path, $\sigma^*(\mathbf{q}^{\mathrm{MM}}_{start}, \mathbf{q}^{\mathrm{MM}}_{end})$, that minimizes a given cost function $\mathcal{F}(\cdot)$, where the cost is socially driven and depends on the three-dimensional spatial relationship between the robot and nearby individuals.

Formally, the objective is to find the optimal solution to the following problem:

$$\sigma^*(\mathbf{q}^{\mathrm{MM}}_{start}, \mathbf{q}^{\mathrm{MM}}_{end}) = \underset{\sigma(\mathbf{q}^{\mathrm{MM}}_{start}, \mathbf{q}^{\mathrm{MM}}_{end})}{\arg\min} \; \mathcal{F}(\sigma) \; . \tag{3.1}$$

3.2 Social Navigation Framework

To achieve complete social navigation in static scenarios, we propose the workflow illustrated in Figure 3.1. Our contributions are primarily embedded within the *Offline Planner* phase, where we introduce two distinct planning strategies, and within the *Social Metric*, where we propose novel formulations to evaluate social discomfort, targeted for mobile manipulators.

Our scheme is structured as follows. The *Offline Planner* is responsible for generating a motion plan for the robot, which consists of a sequence of movements specifying both the position of the mobile base and the joint values of the manipulator arm. We introduce two distinct planning methods tailored for the different scenarios defined.

The Offline Planner receives as input the global map information, which includes the locations of obstacles and people, as well as the goal configuration that the robot must achieve. In addition, it incorporates a social discomfort function that evaluates the level of discomfort associated with each robot configuration given people nearby. The main objective of the planner is to compute a sequence of motions that minimizes the overall

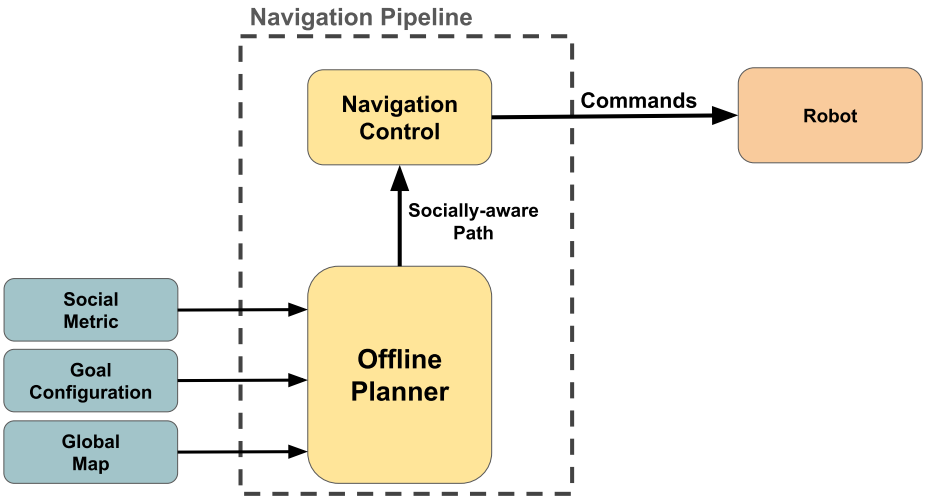


Figure 3.1: General motion planning and navigation workflow.

social metric informed along the navigation path, as defined on the previous section. The social metric defined is also important to notice, which we implement two main novelties for working with mobile manipulators, our multi-point approach and the three dimensional model.

The resulting motion sequence is then passed to the *Navigation Control Module*, which executes the plan by controlling the robot in real time and sending the appropriate commands to ensure that the desired motion is followed.

3.2.1 Offline Planner

The objective of the offline planner is to compute a socially-aware motion plan that guides the mobile manipulator from an initial configuration to a target configuration. This component represents the core contribution of this dissertation, as it explicitly integrates social considerations into the planning process, ensuring that the robot's trajectory not only achieves the task but also take proxemic constraints into account. In this section, we provide an overview of the offline planner, the two specific techniques developed will be discussed in detail in their own chapters.

As previously described, the mobile manipulator is a conjunction between a mobile base and at least one robotic arm (manipulator). Because of this configuration, for the planning approach, two primary methods can be considered: coupled and decoupled approaches.

In a coupled approach, both the mobile base and the robotic arm are treated as an

integrated system. This method involves coordinating the movements of the arm and the base simultaneously, accounting for their interactions throughout the process. Therefore, the configuration of the whole arm and mobile base are defined in conjunction.

In contrast, the decoupled approach treats the mobile base and the robotic arm independently. The central idea is to plan the movements separately: first, a final configuration is defined, where the mobile base navigates to the target position, and then the robotic arm adjusts to achieve the final desired pose. However, considerations needs to be made in this approach as there is the risk of the manipulator colliding with an obstacle or a human while the mobile base is in motion. To mitigate this, one possible solution is to first plan the path for the base and then, at each step, plan the manipulator's movements accordingly, ensuring that the entire trajectory remains collision-free and socially-aware. Other possibility is to let the manipulator retracted and only make online adjustments when needed, while the robot is in locomotion, and in the end adjust for the desired end position. We adopt the former method.

Our literature review did not reveal a definitive pattern regarding which method performs better in practice for socially-aware robots, but as discussed in the related work section, the decoupled approach is more common [75] given its simplicity for traditional motion planning. In this work, we develop both coupled and decoupled approach, to tackle two variations of the problem.

The coupled approach, allows us to address static social navigation in a single planning phase. Although there is the advantage of the planner be coordinated, given the high degrees of freedom (DoF) typical of mobile manipulators, full-body offline planning methods can be time-consuming. Finding an effective solution that simultaneously coordinates the base and the arm can take time. We employ this approach in a bidimensional simplification of the problem, modeling the manipulator as a 2R planar manipulator. This reduction significantly decreases the planning complexity, enabling the computation of feasible paths within a reasonable time frame. In principle, the approach works for any robot, with any arm or base, with time constraints being the only limitation.

For the three-dimensional problem, involving a manipulator with more links operating within a simulated environment, we adopt a decoupled approach. This strategy enables the generation of feasible, socially-aware paths with reduced computational demand, but the base never coordinates with the arm. The planning process is divided into two stages: first, the trajectory of the mobile base is computed while keeping the manipulator static; subsequently, for intersections of the base path, the motion of the manipulator is planned. Each stage independently aims to minimize social discomfort, allowing a practical balance between computational efficiency and socially compliant behavior.

3.2.2 Navigation Control

The navigation control module will actuate the robot to follow the motion planned, online. It is important to note that online navigation in this context does not refer to navigation in dynamic environments. Instead, is understood as the process of executing the precomputed motion plan in real time, guiding the robot along the planned motion. The robot does not adapt its path in response to unforeseen dynamic changes in the environment.

For the motion execution, the robot follows the path generated by the planner using a straightforward waypoint-following strategy. Each waypoint specifies a configuration $\mathbf{q}^{\mathrm{MM}}$ with the variables $x_{\mathrm{base}}, y_{\mathrm{base}}, \psi_1, ..., \psi_n$. These values represent the desired state of the robot at each waypoint, defining both the position of the mobile base and the joint angles of the manipulator.

We control the mobile base and manipulator independently but simultaneously. The mobile base chosen is equipped with omnidirectional four-wheel drive, showed in Figure 3.2.

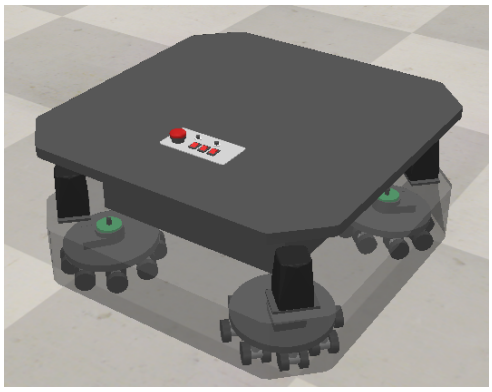


Figure 3.2: Omnidirectional base.

The control is managed using a proportional controller:

$$\begin{bmatrix} v_x \\ v_y \end{bmatrix} = K_p \begin{bmatrix} x_{\mathrm{current}} - x_{\mathrm{next}} \\ y_{\mathrm{current}} - y_{\mathrm{next}} \end{bmatrix} , \tag{3.2}$$

and this controller regulates the velocities of the four wheels:

$$\begin{bmatrix} v_1 & v_2 & v_3 & v_4 \end{bmatrix}^{\mathrm{T}} = \frac{1}{r} \begin{bmatrix} 1 & 1 & -1 & -1 \\ -1 & 1 & 1 & -1 \end{bmatrix}^{\mathrm{T}} \begin{bmatrix} v_x \\ v_y \end{bmatrix} , \tag{3.3}$$

where r is the radius of the wheels, v_x and v_y represent the linear velocities in the x and y directions, x_{current} and y_{current} are the current coordinates of the mobile base, while x_{next} and y_{next} are the coordinates of the next waypoint. The robot's velocity is limited by its maximum speed.

The manipulator, on the other hand, relies on an implementation of the Ruckig online trajectory generator [8][1], where we directly input the desired joint angles, tool built-in in the simulator.

The robot only proceeds to the next waypoint once both the base and manipulator have reached the current waypoint configuration within a defined acceptance distance. For the base, the acceptance distance is defined as a Euclidean distance of $0.2, \mathrm{m}$ from the goal, while for the manipulator, each joint angle must be within an absolute difference of $0.05, \mathrm{rad}$ from its target value. It is important to note that some joints are capable of full rotation, requiring a wrap-around computation to correctly evaluate the angular difference, defined as:

$$\Delta\psi = \mathrm{mod}\big((\psi_{goal} - \psi_{current}) + \pi,\, 2\pi\big) - \pi$$

This ensures alignment and coordinated movement before transitioning to the next segment of the path. The motion is complete when the robot reaches the final desired configuration.

We also highlight the navigation aspects that are beyond the scope of this work. The perception stage, for instance, is a whole area of robotics that demands dedicated research. In Social Robotics, one solution for human detection that can specially get the basic human structure employs Kinect sensors [52, 69], offering a more straightforward solution for human perception compared to traditional cameras and computer vision techniques. Moreover, accurately determining the robot's current position and configuration presents its own set of challenges, often necessitating techniques like Simultaneous Localization and Mapping (SLAM) for precise pose estimation. These challenges will not be the focus of our work, we conduct our research in a simulation environment where pose information for both the robot and humans is provided at all times.

[1] https://github.com/pantor/ruckig

Chapter 4

Full-body Socially-aware 2D Motion Planner

We propose a framework built around a socially-aware whole-body motion planner that accounts for the mobile base, attached manipulator, and transported object. We apply our method in a static human-populated environment. While assuming static humans is a simplification, office-like settings often have predictable human presence around desks and chairs. Despite this assumption, our work marks a significant step toward social navigation strategies for mobile manipulators, especially when transporting large objects that may cause discomfort.

Our approach integrates all system components into the planning process using a risk-aware, sampling-based method, where the traditional risk cost is replaced by a social-related cost function. In this chapter, we first define our planar environment, which serves as a simplified representation of the overall problem. Next, we describe the robot and the objects used. Afterwards, we present the developed method, beginning with the social function and our multi-point approach, which ensures that the discomfort experienced by people is induced by the entire robot rather than just the base. We then detail the motion planner, which employs a Risk-RRT* [67] algorithm.

4.1 Simplification: Planar Environment

We introduce a simplification of the three-dimensional environment into a two-dimensional (planar) setting. This simplification allows us to investigate social navigation for mobile manipulators in a more tractable manner. Moreover, since most existing research on social navigation in robotics has been carried out in two-dimensional spaces, adopting this setting facilitates a more direct comparison with related work.

The distinction between the general three-dimensional environment described in the problem formulation and the two-dimensional planar environment considered here is

straightforward: the motion of the manipulator is constrained to a plane, and the social metric is defined accordingly. Therefore, the configuration of the robot can be described as:

$$\mathbf{q}^{\mathrm{MM}} = \begin{bmatrix} x_{\mathrm{base}} & y_{\mathrm{base}} & \psi_1 & \psi_2 \end{bmatrix}^{\mathrm{T}},$$

4.2 Robot and Objects

The mobile manipulator considered in this section is a planar robot. Its manipulator is a 2-link RR planar manipulator, consisting of two revolute joints forming a 2-DoF arm, which is attached to a holonomic mobile base. The base is implemented as an omnidirectional platform in the CoppeliaSim simulator, equipped with four omnidirectional wheels, as shown in Figure 3.2.

For the manipulator, we employ the UR10 model[1]. We rotate it 90° about the global y-axis and activate only the second and third joints, as shown in Figure 4.1. This results in a simplified planar manipulator with two links, where the first link measures 0.75 m and the second 0.9 m.

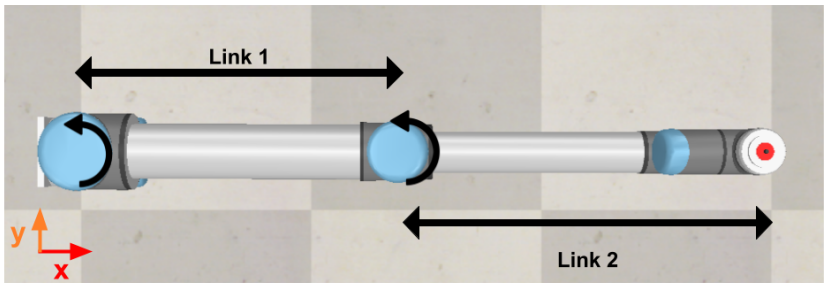


Figure 4.1: Planar 2R manipulator.

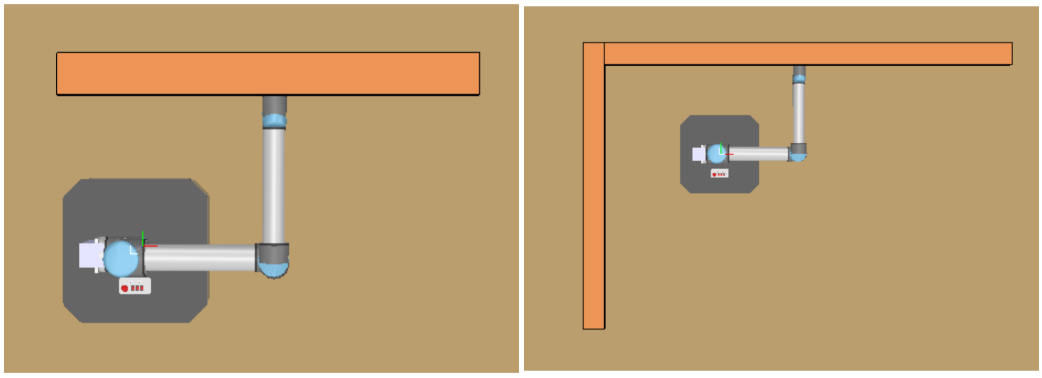
For the objects carried by the mobile manipulator, we consider and test two objects throughout this dissertation. However, our framework is general and can accommodate any object with minor adjustments to the social function, as will be described in the corresponding section.

The first object is a simple bar-like plank, 1.5 meters in length. Due to its great length, it must be explicitly accounted for to achieve successful social navigation. The

[1]Universal Robots. *User Manual – UR10 CB-Series, Version SW3.15 (English)*. Available at: https://www.universal-robots.com/download/manuals-cb-series/user/ur10/315/user-manual-ur10-cb-series-sw315-english-international-en/

second object is an L-shaped structure composed of two planks: the first measures 3 meters and the second 2 meters. Both objects have a width of 0.15 meters and a depth of 0.5 meters. The L-shaped object is larger and more geometrically complex, making it more challenging to find good paths. Therefore, it serves as a stress test for the planner. These dimensions are used in our planner. However, in robotics, it is common to use bigger objects dimensions (or environment) in the planning stage to accommodate real-world uncertainties. In our experiments, we used slightly shorter objects with the same geometry when executing the motions generated by the planner.

The robotic arm is mounted at the center of the mobile base, forming the complete planar mobile manipulator. The objects are placed at the tip of the manipulator, with the robot grasping each object at the midpoint of its longest plank. The full configuration of the mobile manipulator with the objects is shown in Figure 4.2.



(a) Mobile manipulator with Bar-shaped object.

(b) Mobile manipulator with L-shaped object.

Figure 4.2: Planar mobile manipulator used in this study, showcasing the two objects defined.

4.3 Social Cost Function

In order to achieve a socially-aware transportation, it is essential to quantitatively assess a person's level of comfort based on the motion occurring in their surroundings. In this work, we build upon the concept of *proxemics*, where we adopt the traditional Asymmetric Gaussian Function (AGF) [46] as the personal space model. This model represents personal space as an elongated region, a characterization widely used [6, 60, 35]. Although other models could be employed, Kirby's AGF continues to be applied [82] and provides a direct mathematical measure of discomfort as a function of distance, which aligns with the requirements of this work. The model will be employed to quantify a

person's comfort level based on the complete system's position (base, arm, object) relative to them.

Algorithm 1 can be used to determine the social cost of a specific coordinate (x, y) on the plane relative to an individual $\mathbf{n}_i$, with higher values indicating increased discomfort. The closer the coordinate is to the person, the higher the associated social cost. Following the model proposed by Kirby [46], the AGF is centered at $\mathbf{p}_i = (x_i, y_i)$, with orientation θ_i, and specified variances σ_h, σ_s, and σ_r along the person's direction, sides, and rear.

Algorithm 1 $\texttt{AGF}(\mathbf{n}_i, x, y)$ [46]

$\alpha \leftarrow \text{atan2}(y - y_i, x - x_i) - \theta_i + \pi/2$
Normalize α
$\sigma \leftarrow (\alpha \leq 0 \ ? \ \sigma_r : \sigma_h)$
$a \leftarrow (\cos\theta_i)^2/(2\sigma^2) + (\sin\theta_i)^2/(2\sigma_s^2)$
$b \leftarrow \sin(2\theta_i)/(4\sigma^2) - \sin(2\theta_i)/(4\sigma_s^2)$
$c \leftarrow (\sin\theta_i)^2/(2\sigma^2) + (\cos\theta_i)^2/(2\sigma_s^2)$
return $\exp(-(a(x - x_i)^2 + 2b(x - x_i)(y - y_i) + c(y - y_i)^2))$

Similar to other studies [71, 95, 81], which suggest that people are more protective of their frontal space, we model the AGF as elongated along the body orientation. Specifically, we set the following parameters:

$$\sigma_h = 2 \quad \text{and} \quad \sigma_s = \frac{2}{3}\sigma_h \quad \text{and} \quad \sigma_r = \frac{1}{2}\sigma_h \ .$$

This results in a bidimensional equation that quantifies the personal space, illustrated in Figure 4.3 using a contour plot. Notice the elongated shape, with the frontal region more sensitive where robots or people approaching from the front generate higher discomfort than from the rear, and discomfort increases as proximity to the person decreases.

Additionally, we apply a threshold of $\tau = 0.2$ to evaluate the AGF limited to the proxemic zone known as *public space* [64], which extends up to approximately 3.60m in a straight line from the person's front at its farthest point. If the AGF result is less than or equal to 0.2, it is considered 0, indicating no discomfort.

While the use of social functions has been successfully applied to guide mobile robots in socially acceptable navigation [15, 84, 64], additional considerations are required for mobile manipulators. Traditional methods typically focus on the robot's mobile base as the key reference point. However, when applied directly to mobile manipulators, these methods may prevent the base from entering socially restricted areas, yet fail to account for the manipulator arm or the carried object, which can still intrude into personal space.

In this context, we propose a multi-interest-point approach to quantify the social acceptability of the robot based on its complete current configuration. Additionally, it allows for assigning different weights to various points, giving priority to those that are

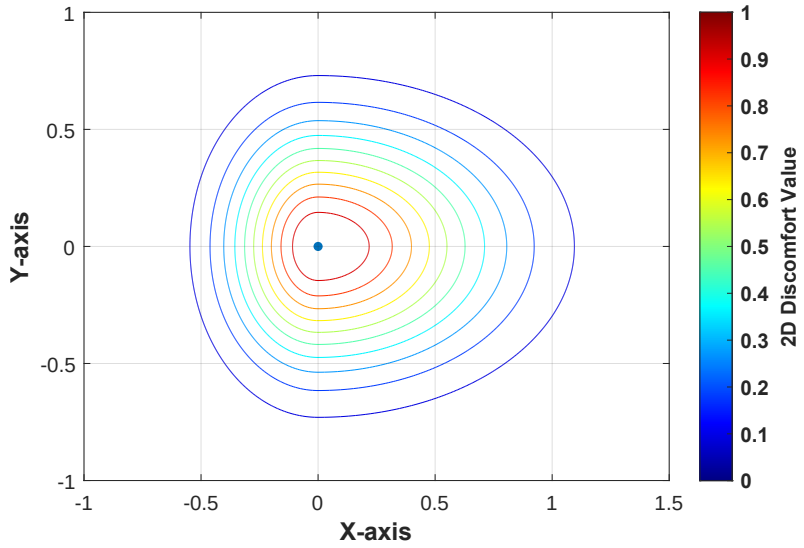


Figure 4.3: 2D personal space for an individual at $\mathbf{q} = ((0,0),0)$ ooriented along the positive x-axis, based on the model by Kirby [46], with $\sigma_h = 1/2$.

most critical. For instance, if the transported object is hazardous, higher weights can be assigned to the object's key points, emphasizing the importance of keeping it farther from people. In such cases, a robot's base being closer to a person is more socially acceptable than having the dangerous object near them.

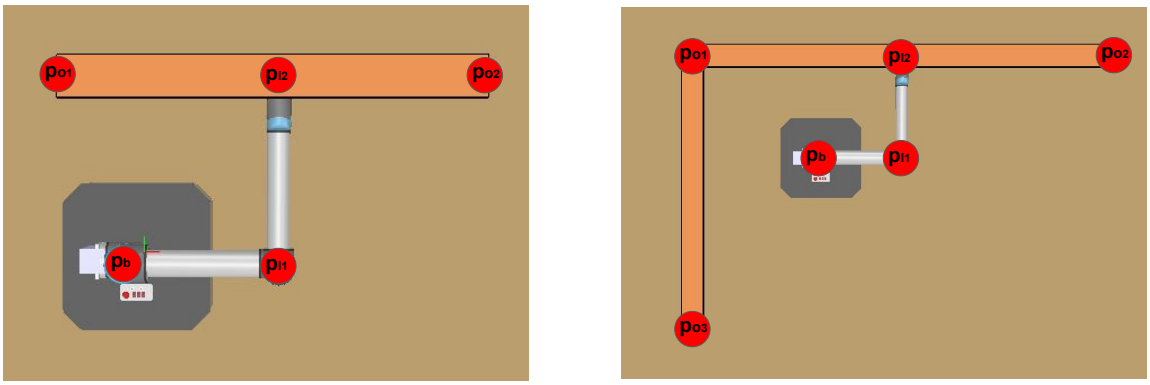
To apply this approach, key points on the robot must be defined to calculate the social cost, with these points being selected based on the robot's geometry. Intuitively, corners and tips are suitable choices, as they represent critical areas. For instance, on a planar arm with revolute joints, the tip of a link can serve as a key social point – if the tip is far from a person, it is likely that the entire arm remains distant as well. Moreover, corners and tips often pose higher risks due to their sharper features, making them important for social cost evaluation. The same principle applies to the object being carried, where extremities are logical points for defining social importance. These points are manually selected, future works could explore automatic selection methods.

In this work we consider a 2-link planar mobile manipulator with revolute joints carrying an object; examples of defined points for different objects are shown in Figure 4.4.

Given s selected points, the total social cost of a mobile manipulator at configuration $\mathbf{q}^{\mathrm{MM}}$ relative to all people n is calculated as a weighted sum of the social costs of each individual point/person, *i.e.*:

$$\mathcal{S}(\cdot) = \sum_{i=1}^{m} \sum_{j=1}^{s} w_j \texttt{AGF}_i(x_j, y_j), \tag{4.1}$$

where w_j is the point's weight, and `AGF` is the social cost related to an individual's personal space, defined by Alg. 1.



(a) Bar-shaped object. (b) L-shaped object.

Figure 4.4: The points $p_b, p_{l1}, p_{l2}, p_{o1}, p_{o2}$, and p_{o3} (only in (b)) represent the base, link tips, and object tips, including the final tip of the L-shaped object.

4.4 Motion Planner

Several techniques could be used to tackle the problem of social whole-body motion planner, in this work, we use a sampling-base technique based on the Rapidly Random Tree (RRT) method. The choice of an RRT-based planner for this task is supported by the literature, as RRT methods are well-suited for high-dimensional configuration spaces and can be extended to account for manipulator collisions, enabling whole-body motion planning for mobile manipulators [77]. Also, RRT variants have been successfully employed in various social navigation contexts for mobile robots as described in the Related Work section. Furthermore, although other techniques for social navigation in mobile robots has been explored using Reinforcement Learning [101, 60, 44, 108] and control-based techniques [66], applying these methods to mobile manipulators presents challenges: control-based approaches require new architectures, and Reinforcement Learning must account for the manipulator's influence, making them not directly transferable. In contrast, RRT-based methods can be naturally extended to consider the manipulator's configuration, making them a suitable choice for whole-body motion planning. Given their prior use in social navigation for mobile robots and their adaptability to mobile manipulators, we adopt this technique for developing our socially-aware whole-body motion planner. However, a notable limitation of this technique is its computational intensity, which increases as the degrees of freedom grow.

In this work, we adapt the Risk-based Optimal Rapidly-exploring Random Tree (Risk-RRT*) by [67], incorporating a social cost function as defined by Equation 4.1, instead of the traditional risk function. The pseudocode for our Social Risk-RRT* is reported on Algorithm 2, where the algorithm mainly differs from the original one in Line 14, where the social costs are introduced.

Algorithm 2 Social Risk-RRT*

1: $\mathcal{T} \leftarrow \{\mathbf{q}^{\mathrm{MM}}_{start}\}$
2: **for** $k = 1$ **to** K **do**
3: $\quad \mathbf{q}^{\mathrm{MM}}_{rand} \leftarrow$ `SampleRandomState()`
4: $\quad \mathbf{q}^{\mathrm{MM}}_{nearest} \leftarrow$ `NearestNode`$(\mathbf{q}^{\mathrm{MM}}_{rand}, \mathcal{T})$
5: $\quad$ **if** $||\mathbf{q}^{\mathrm{MM}}_{nearest} - \mathbf{q}^{\mathrm{MM}}_{rand}|| > \delta$ **then**
6: $\quad\quad \mathbf{q}^{\mathrm{MM}}_{new} \leftarrow$ `Steer`$(\mathbf{q}^{\mathrm{MM}}_{rand}, \mathbf{q}^{\mathrm{MM}}_{nearest}, \delta)$
7: $\quad$ **else**
8: $\quad\quad \mathbf{q}^{\mathrm{MM}}_{new} \leftarrow \mathbf{q}^{\mathrm{MM}}_{rand}$
9: $\quad$ **end if**
10: $\quad$ **if** `isValid`$\left(\mathbf{q}^{\mathrm{MM}}_{new}\right)$ **then**
11: $\quad\quad c_{min} \leftarrow \infty$
12: $\quad\quad \mathcal{Q}_{near} \leftarrow$ `AllNear`$(\mathcal{T}, \mathbf{q}^{\mathrm{MM}}_{new}, r_{near})$
13: $\quad\quad$ **for each** $\mathbf{q}^{\mathrm{MM}}_{near} \in \mathcal{Q}_{near}$ **do**
14: $\quad\quad\quad \Phi \leftarrow$ `Msc`$(\mathbf{q}^{\mathrm{MM}}_{near}, \mathbf{q}^{\mathrm{MM}}_{new}, \mathcal{M})$
15: $\quad\quad\quad$ **if** $\mathcal{F}(\mathbf{q}^{\mathrm{MM}}_{near}) < c_{min}$ **then**
16: $\quad\quad\quad\quad$ **if** `CollisionFree`$(\mathbf{q}^{\mathrm{MM}}_{near}, \mathbf{q}^{\mathrm{MM}}_{new}, \mathcal{M})$ **then**
17: $\quad\quad\quad\quad\quad \mathbf{q}^{\mathrm{MM}}_{new}$.parent $\leftarrow \mathbf{q}^{\mathrm{MM}}_{near}$
18: $\quad\quad\quad\quad\quad c_{min} \leftarrow \mathcal{F}(\mathbf{q}^{\mathrm{MM}}_{near})$
19: $\quad\quad\quad\quad$ **end if**
20: $\quad\quad\quad$ **end if**
21: $\quad\quad$ **end for**
22: $\quad\quad$ **if** $c_{min} \neq \infty$ **then**
23: $\quad\quad\quad \mathcal{T} \leftarrow \mathcal{T} \cup \mathbf{q}^{\mathrm{MM}}_{new}$
24: $\quad\quad\quad \mathcal{T} \leftarrow$ `Rewire`$(\mathcal{T}, \mathbf{q}^{\mathrm{MM}}_{new}, \mathcal{Q}_{near}, \mathcal{M})$
25: $\quad\quad$ **end if**
26: $\quad$ **end if**
27: **end for**
28: **return** `GetPath`$(\mathcal{T}, \mathbf{p}^{\mathrm{B}}_{end})$

The algorithm initializes by inserting the start configuration, $\mathbf{q}^{\mathrm{MM}}_{start}$, as the first node into an empty tree $\mathcal{T}$. It then proceeds iteratively for K iterations. In each iteration, a random node $\mathbf{q}^{\mathrm{MM}}_{rand}$ is uniformly sampled from the configuration space (C-space), and the nearest node $\mathbf{q}^{\mathrm{MM}}_{nearest}$ in the tree is determined based on the chosen distance metric. In this work, distances are computed as the norm of the difference between two configurations. Given a planning range δ, the algorithm first checks whether $\mathbf{q}^{\mathrm{MM}}_{nearest}$ lies within a distance δ from $\mathbf{q}^{\mathrm{MM}}_{rand}$. If this condition is met, the new node $\mathbf{q}^{\mathrm{MM}}_{new}$ is set to $\mathbf{q}^{\mathrm{MM}}_{rand}$; otherwise, a new node is created by moving from $\mathbf{q}^{\mathrm{MM}}_{nearest}$ toward $\mathbf{q}^{\mathrm{MM}}_{rand}$, at a distance of δ, using the following function:

$$\texttt{Steer}\left(\mathbf{q}^{\mathrm{MM}}_1, \mathbf{q}^{\mathrm{MM}}_2, \delta\right) = \mathbf{q}^{\mathrm{MM}}_1 + \left(\frac{\mathbf{q}^{\mathrm{MM}}_2 - \mathbf{q}^{\mathrm{MM}}_1}{||\mathbf{q}^{\mathrm{MM}}_1, \mathbf{q}^{\mathrm{MM}}_2||}\right)\delta \ .$$

At Line 10, the algorithm checks whether $\mathbf{q}^{\mathrm{MM}}_{new}$ is valid, ensuring it does not collide with any obstacles or people. In Line 12, all nodes within a distance r_{near} of $\mathbf{q}^{\mathrm{MM}}_{new}$ are put into set $\mathcal{Q}_{near}$. These nearby nodes are then used to determine the most suitable parent

for $\mathbf{q}_{new}^{\mathrm{MM}}$.

In Lines 13 to 21, the parent node for $\mathbf{q}_{new}^{\mathrm{MM}}$ is chosen by evaluating each node in $\mathcal{Q}_{near}$. The selection criterion minimizes the cost from the start node to $\mathbf{q}_{new}^{\mathrm{MM}}$, using:

$$\mathcal{F}(\mathbf{q}_{a}^{\mathrm{MM}}) = \mathcal{F}(\mathbf{q}_{ap}^{\mathrm{MM}}) + \texttt{Msc}(\mathbf{q}_{ap}^{\mathrm{MM}}, \mathbf{q}_{a}^{\mathrm{MM}}) \ , \tag{4.2}$$

where $\mathbf{q}_{ap}^{\mathrm{MM}}$ is the parent of $\mathbf{q}_{a}^{\mathrm{MM}}$ and $\texttt{Msc}(\cdot)$ computes the motion social cost, which quantifies the total social cost of moving between configurations. The function $\mathcal{F}(\cdot)$, as previously noted, represents the objective to be minimized, namely the social cost of the whole path up to a configuration $\mathbf{q}_{a}^{\mathrm{MM}}$. In Line 14 it is calculated the social cost of moving from a potential parent node to $\mathbf{q}_{new}^{\mathrm{MM}}$ by integrating the cost over the moved distance s, as done by Primatesta *et al.* [67], *i.e.*:

$$\texttt{Msc}(\mathbf{q}_{a}^{\mathrm{MM}}, \mathbf{q}_{b}^{\mathrm{MM}}) = \int_{\mathbf{q}_{a}^{\mathrm{MM}}}^{\mathbf{q}_{b}^{\mathrm{MM}}} \mathcal{S}(\mathbf{q}^{\mathrm{MM}})\mathrm{d}s \ . \tag{4.3}$$

In practice, the integral is computed numerically by linearly interpolating between the start and end configurations, then applying the trapezoidal method[2], as:

$$\int_{\mathbf{q}_{a}^{\mathrm{MM}}}^{\mathbf{q}_{b}^{\mathrm{MM}}} \mathcal{S}(\mathbf{q}^{\mathrm{MM}})\mathrm{d}s \approx \frac{1}{2}\sum_{j=2}^{N} ||\mathbf{q}_{j}^{\mathrm{MM}} - \mathbf{q}_{j-1}^{\mathrm{MM}}|| \left[\mathcal{S}(\mathbf{q}_{j}^{\mathrm{MM}}) + \mathcal{S}(\mathbf{q}_{j-1}^{\mathrm{MM}})\right] \ , \tag{4.4}$$

where $\mathbf{q}_{a}^{\mathrm{MM}} = \mathbf{q}_{1}^{\mathrm{MM}}$, $\mathbf{q}_{b}^{\mathrm{MM}} = \mathbf{q}_{N}^{\mathrm{MM}}$ and intermediate configurations are the linear interpolation of N steps between $\mathbf{q}_{a}^{\mathrm{MM}}$ and $\mathbf{q}_{b}^{\mathrm{MM}}$.

Figure 4.5 illustrates the linear interpolation step.

To ensure the feasibility of the path from $\mathbf{q}_{near}^{\mathrm{MM}}$ to $\mathbf{q}_{new}^{\mathrm{MM}}$, the algorithm utilizes the $\texttt{CollisionFree}(\cdot)$ function (Line 16). This function once again utilizes the linear interpolation between $\mathbf{q}_{near}^{\mathrm{MM}}$ and $\mathbf{q}_{new}^{\mathrm{MM}}$, checking each interpolated configuration to ensure it does not collide with any obstacles or people. In Lines 17 and 18, the best parent node and its corresponding cost are recorded.

In Lines 22 to 25, if a valid $\mathbf{q}_{new}^{\mathrm{MM}}$ with a parent is found, the tree is *rewired*, and $\mathbf{q}_{new}^{\mathrm{MM}}$ is added to the tree. During the rewiring process, each node in $\mathcal{Q}_{near}$ is evaluated to see if $\mathbf{q}_{new}^{\mathrm{MM}}$ can serve as a better parent. If using $\mathbf{q}_{new}^{\mathrm{MM}}$ as a parent results in a collision-free motion and reduces the current $\mathcal{F}(\mathbf{q}_{near}^{\mathrm{MM}})$, the parent of $\mathbf{q}_{near}^{\mathrm{MM}}$ is updated to $\mathbf{q}_{new}^{\mathrm{MM}}$. This step ensures the path is continuously optimized by incorporating lower-cost connections where possible.

Finally, after completing all K iterations, the algorithm retrieves the path using the $\texttt{GetPath}(\cdot)$ function (Line 28). This function first identifies the end node $\mathbf{q}_{end}^{\mathrm{MM}}$, which is the node within r_{near} distance of the goal with the lowest total cost $\mathcal{F}(\cdot)$, and then reconstructs the path by recursively tracing back through the parent nodes.

[2] https://www.mathworks.com/help/matlab/ref/trapz.html

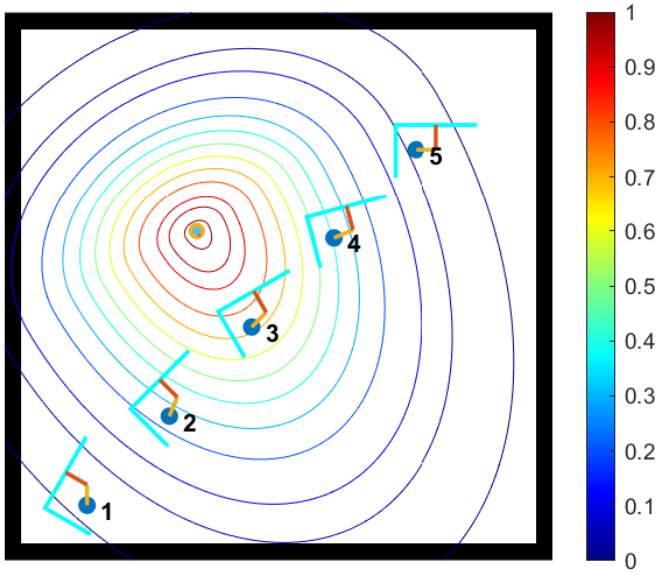


Figure 4.5: Linear interpolation example. The dark blue circle is the robot base, the yellow and orange rods are the first and second links, respectively, and the light-blue L-structure is the object. Starting from configuration 1, the robot reaches configuration 5 through intermediate steps 2–4. The colormap represents the social cost.

Chapter 5

Three Dimensional Discomfort Model

As discussed, most socially-aware methods define personal space in two dimensions, typically on the ground plane. This simplification aligns well with socially aware navigation, as many mobile robots can be approximated as points or simple shapes moving in a planar environment. However, existing techniques fall short for robots operating in 3D space, such as drones and mobile manipulators, as traditional 2D approaches neglect the Z-axis, potentially causing undesirable behaviors like moving near a person's head.

Given the lack of dedicated research of proxemics in three-dimensions, it is needed to develop our own to account for mobile manipulators. We propose a novel three-dimensional personal space representation (3D Proxemics). Our approach incorporates the concept of Maximum Permissible Pressure (MPP) from established robotic norms [37] and employs Fuzzy Logic to model a smooth comfort gradient along the Z-axis. By integrating this with a traditional formulation for personal space in the XY plane, we achieve a comprehensive 3D model. The resulting model quantifies a person's comfort level at any point in 3D space, ranging from 0 (completely comfortable) to 1 (completely uncomfortable).

This section outlines the derivation of the 3D Proxemics equation in three key steps. First, we define discrete discomfort values for specific body regions using a proposed segmentation, mapping each value to its corresponding height relative to the individual's total height. Second, we apply a Fuzzy Inference System (FIS) to transform these discrete values into a continuous Z-axis comfort function, capturing the relationship between height and discomfort. Finally, we integrate this function with a XY-plane personal space model, forming a complete 3D discomfort representation.

5.1 Problem Formulation

Let us formalize the problem using Section 3.1 as a starting point. An individual $\mathbf{n}$ has an associated personal space $\mathcal{P} \subset \mathcal{E}$. Comparing to the bidimensional model, $\mathcal{P}$ would be substituting the `AGF`. The personal space $\mathcal{P}$ is a function that computes the discomfort value, in the range $[0, 1]$, that an individual with configuration $\mathbf{q}$ and height h experiences at a given point $\mathbf{r} \in \mathbb{R}^3$. In other words, it answers the question: *How much discomfort would I feel with the robot at that location?* Therefore:

$$\mathcal{P}(\mathbf{n}, \mathbf{r}) \in [0, 1] \ . \tag{5.1}$$

The problem consists of defining a comprehensive three-dimensional personal space representation that quantifies the discomfort caused by proximity to the individual $\mathbf{n}$ at any location $\mathbf{r}$ in the environment. More formally, the objective is to compute the personal space $\mathcal{P}(\mathbf{n}, \mathbf{r}) \in [0, 1]$, *i.e.*, a discomfort value, for any given point $\mathbf{r} \in \mathcal{E}$. The discomfort value should reflect the individual's comfort level relative to their proximity to the point $\mathbf{r}$, with 0 indicating no discomfort (maximum comfort) and 1 indicating maximum discomfort (minimum comfort).

5.2 Body Discomfort-related Segmentation

To the best of our knowledge, no comprehensive studies have explicitly quantified human discomfort based on an object's or robot's position along the Z-axis (height). However, existing findings suggest that proximity to the face or head induces greater discomfort [73, 27, 86], implying a preference for external elements to be at lower regions.

Given the limited research on this topic, our approach builds on the established link between safety and comfort [96], where safety is a prerequisite for comfort. Prior studies in robotics identify varying levels of vulnerability across body regions, with discomfort or pain arising when pressure exceeds a certain threshold. Therefore, to promote safe and comfortable interactions, robots should maintain greater distances from these more sensitive areas.

The international standard ISO/TS 15066:2016 [37], revised in 2022, establishes safety guidelines for collaborative robots, defining precise pressure and force limits across different body regions to prevent minor injuries. In this work, we leverage the Maximum Permissible Pressure (MPP) as a proxy for sensitivity – lower MPP values indicate more

sensitive areas, implying that robots should maintain greater distances from these regions to enhance human comfort.

Since MPP values can vary significantly across closely spaced points, we first segment the human body into broader regions, selecting the lowest (most sensitive) MPP within each region. Following Haddadin *et al.* [28], who categorize safety-measured regions into parts such as the head, neck, torso, and lower extremities, we adopt a similar approach, dividing the body into regions based on spatial proximity.

We define four body regions: head, torso, hips, and legs, as illustrated in Figure 5.1. The figure also highlights the most sensitive area b within each region, along with its corresponding MPP value in $\mathrm{N/cm^2}$.

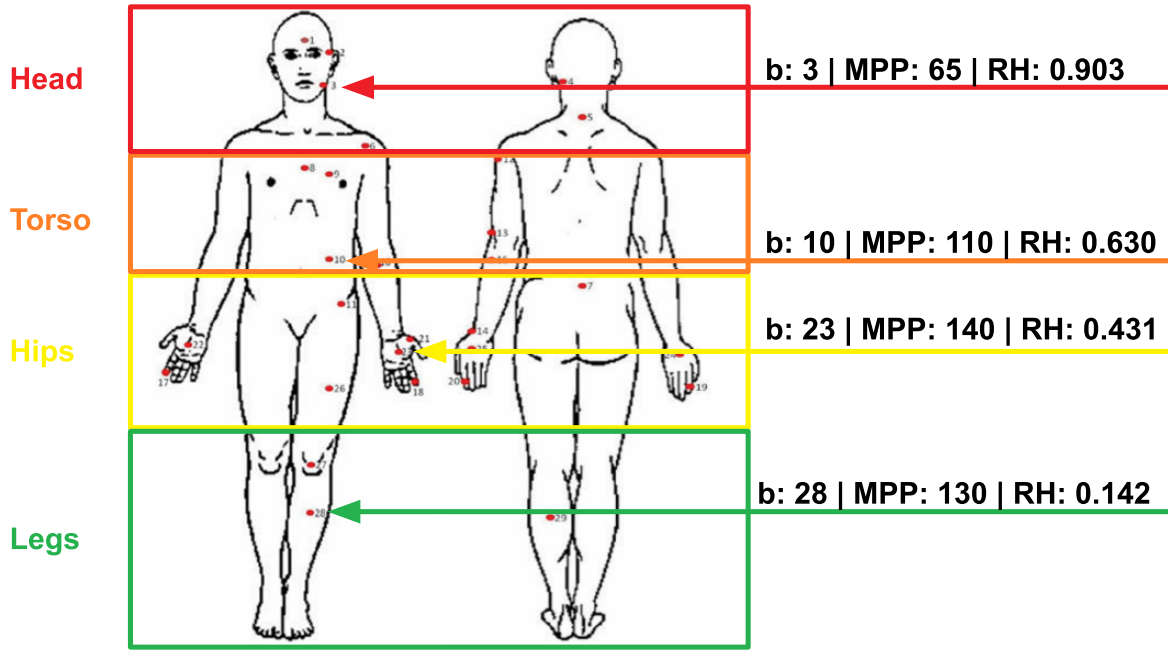


Figure 5.1: Adapted diagram from [37], with our segmentation, highlighting the most sensitive area b within each body region, along with its corresponding Maximum Permissible Pressure (MPP) and relative height (RH).

For generalization, it is necessary to determine the location of each selected point relative to an individual's height. Using body segment proportions from Winter *et al.* [99], expressed as fractions of the height, we estimate the relative height (RH) of each chosen point. These proportions offer a reliable approximation when only the individual's height is available [99]. Figure 5.1 shows the results, where each RH value represents a fraction of the height. For example, a body area with $RH = 0.903$ for a person with a height of $1.75\,m$ corresponds to a point located at $0.903 * 1.75 \approx 1.58\,m$.

Finally, we define the relationship between MPP and the degree of discomfort. As noted earlier, discomfort is inversely proportional to MPP, meaning that lower MPP values correspond to higher discomfort levels. To map the MPP values of each selected body area b to discomfort, we divide the lowest MPP value (65, for the head) by the MPP

of each region, ensuring the most sensitive point corresponds to a maximum discomfort value of 1.0 and the higher the value, the lower it gets.

We also define two additional regions outside the human height boundaries: the *ground* and the *top*. The ground is set at height 0.0. The top is defined as $h + 0.75\,m$, this height was fine tuned to ensure smooth transition between head and top, but higher values could be used in a conservative approach. For the ground, we assign a discomfort value of 1, as we aim to avoid robots making contact with the feet or the floor. For the top, we assign a discomfort value of 0, since there are no body parts above this height, and the robot's proximity in the Z-axis becomes negligible, thus no discomfort is caused.

The information for each selected body part b and the two additional regions is summarized in Table 5.1.

Table 5.1: Parameters by regions, including Maximum Permissible Pressure (MPP), discomfort, and relative height (RH) as a proportion of the person's height h.

b	**Region**	**MPP**	**Discomfort**	**RH**
3	Head	65	1.0	$0.903h$
10	Torso	110	0.591	$0.630h$
23	Hips	140	0.464	$0.431h$
28	Legs	130	0.500	$0.142h$
-	Ground	-	1.0	0.0
-	Top	-	0.0	$h + 0.75\,m$

5.3 Fuzzy Inference System

We use a Fuzzy Inference System (FIS) to derive a continuous equation that generalizes the relationship between discomfort and height, based on the six discrete data points described earlier. Fuzzy Inference Systems are effective at modeling human expertise (in this case, discomfort values) and are widely used as universal approximators [38, 104]. Other approaches could also be used to transform discrete data into a continuous function, such as spline interpolation. The FIS has the advantage of offering an interpretable and flexible framework for representing human-related phenomena, and it has been widely used to model human subjective responses. Our FIS ensures smooth transitions between the discrete regions, generating a continuous function that accurately captures comfort variations along the Z-axis.

For this task, we employ a Zero-order Sugeno FIS [38], utilizing primarily Gaussian-type Membership Functions (MFs) for the antecedents and constant values for the consequents. The system has a single input variable z, *i.e.*, the height, and produces a single

output: discomfort ($[0, 1]$). We define six fuzzy sets to represent the regions: Legs, Hips, Torso, Head, Ground, and Top. Defuzzification is performed using a weighted average of all rule outputs, ensuring a smooth transition across the defined regions.

Rules are logical IF-THEN statements that define the relationship between the antecedent and the consequent. An example of a rule diagram is shown in Figure 5.2.

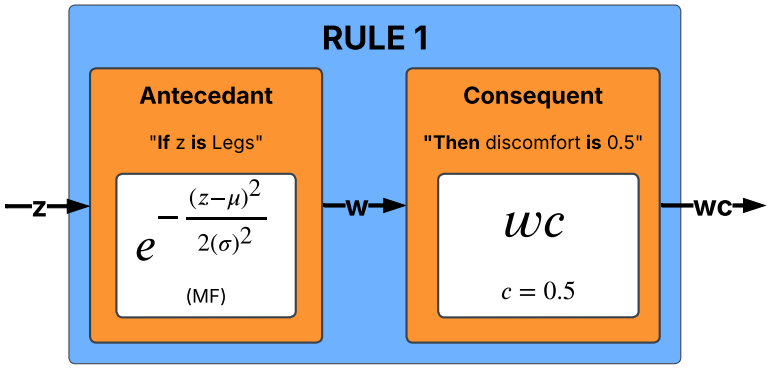


Figure 5.2: Diagram demonstrating how fuzzy rules operate.

The antecedent corresponds to the IF part, and the consequent corresponds to the THEN part. We have formulated one rule for each fuzzy set, with the consequent of each rule representing the discomfort values, as the following:

1. If z is **legs**, then discomfort is 0.500.
2. If z is **hips**, then discomfort is 0.464.
3. If z is **torso**, then discomfort is 0.591.
4. If z is **head**, then discomfort is 1.000.
5. If z is **ground**, then discomfort is 1.000.
6. If z is **top**, then discomfort is 0.000.

Given the rules, the first step is to determine the degree to which the input z belongs to a specific region, known as the *is region* part of the rule. In a FIS, the input has a membership degree for each fuzzy set, ranging from 0 (no membership) to 1 (full membership). In other words, the farther the input is from a region, the lower the degree of membership. This process is achieved using Membership Functions (MFs), which allow the system to handle uncertainty and imprecision, enabling smooth transitions between fuzzy sets and producing a more flexible and adaptive model for representing discomfort levels.

To model the degree of membership of an input z to each of the six regions, we define six different MFs, one for each region, showed in Figure 5.3. We employ Gaussian functions to model these MFs, given by the following equation:

$$G_k(z) = e^{-\frac{(z-\mu_k)^2}{2(\sigma_k)^2}} , \tag{5.2}$$

where μ_k and σ_k are the center and standard deviation of the k-th Gaussian, respectively, and $G_k(z)$ indicates the degree of membership of input z to region k. Centers μ_k correspond to the relative heights listed in Table 5.1, while σ_k values were empirically tuned for smooth transitions: 0.3 for most regions, 0.25 for the head, 0.1 for the ground, and 0.3 for the top. Though more precise values may exist, these rounded ones offer a good balance between accuracy and practicality.

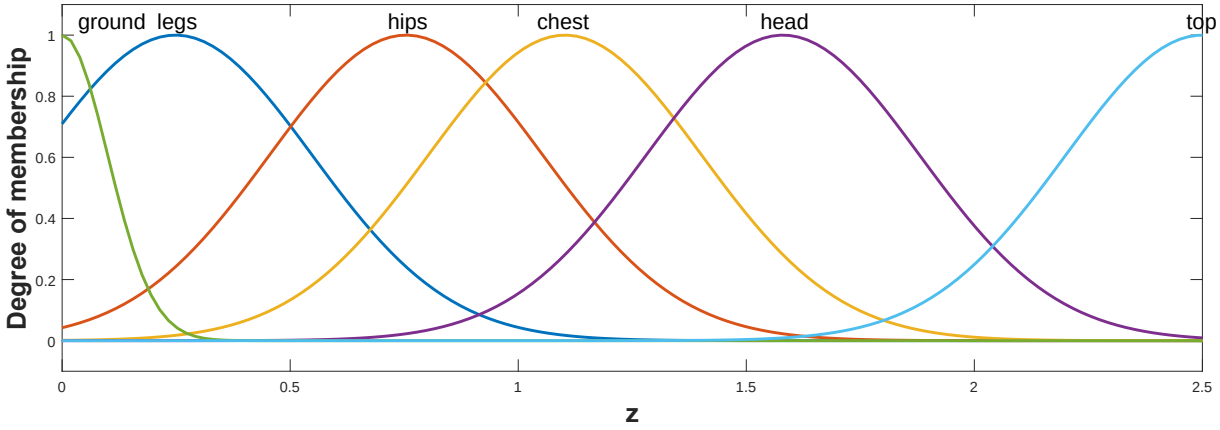


Figure 5.3: Visualization of the input Membership Functions (MFs) for an individual with a height of $h = 1.75\,m$.

With the MFs established and no use of AND/OR operators in our rules, the antecedent consists of a single condition: If z is in a given region. Consequently, the output of the MF directly determines the antecedent (w), which is then used to weight the consequent of each rule without additional computations. This process is illustrated in Figure 5.2.

Since our consequents are constant discomfort values c (zero-order system), the rule output is computed as a simple product of the antecedent activation level and the consequent. These rule outputs are then aggregated and processed by the defuzzifier. The defuzzification step employs a weighted average, producing the final discomfort value as follows:

$$f(z) = \frac{\sum_{i=1}^{n} w_i c_i}{\sum_{i=1}^{n} w_i} , \tag{5.3}$$

where n denotes the total number of rules, i represents the i-th rule, w_i is the antecedent activation (which corresponds to the MF output in our case), and c_i is the consequent, representing the associated discomfort value. Equation 5.3 encapsulates the

complete formulation of discomfort as a function of height along the Z-axis. Properly defined MFs ensure a smooth and continuous transition across all height levels, maintaining consistency in discomfort estimation.

Since we use Gaussian membership functions, with a constant consequent and an antecedent output that directly corresponds to the membership function values, we can substitute w with the Gaussian equation (Equation 5.2). This results in an equivalent but more explicit formulation of the complete Z-axis discomfort function:

$$f(z) = \frac{\sum_{k=1}^{n} e^{-\frac{(z-\mu_k)^2}{2\sigma_k^2}} c_k}{\sum_{k=1}^{n} e^{-\frac{(z-\mu_k)^2}{2\sigma_k^2}}}, \tag{5.4}$$

where k iterates over the six defined regions, with c_k representing the discomfort value associated with each region.

Figure 5.4 presents the discomfort values over the height range $[0, 2.5\,m]$ for a $1.75\,m$ tall person, with peak discomfort near the head and minimal discomfort around the legs.

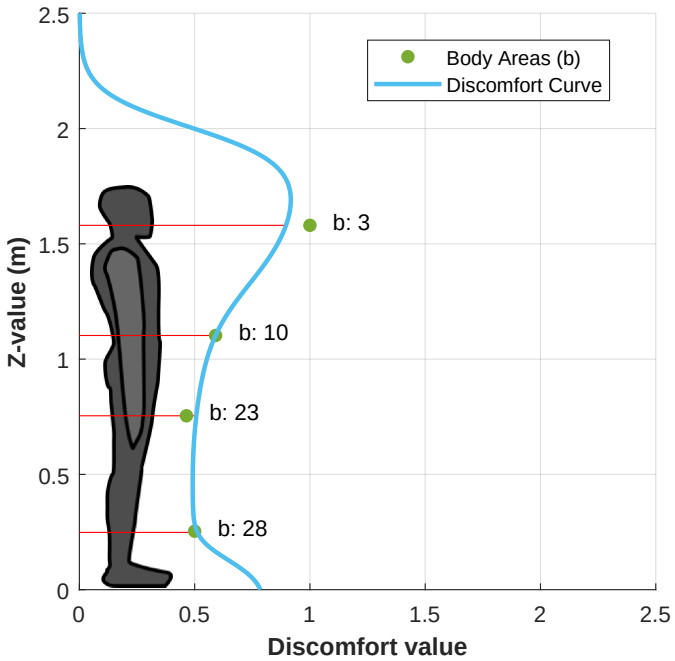


Figure 5.4: Z-axis discomfort function (blue curve), with green dots representing the most sensitive areas within each region, plotted by their corresponding z and discomfort value.

When fine-tuning the Gaussian functions in our model, users should be aware that, in some configurations, discomfort values may increase again beyond a certain height due to the nature of fuzzy sets. To prevent unintended behavior, it may be necessary to manually set discomfort to zero above a specific threshold. If the goal is to extend the comfort zone above the head, replacing the top Gaussian membership function with an S-shaped function is recommended for a smoother decay. Additionally, while normalization is typically required when working exclusively with the Z-axis discomfort function, it is

unnecessary at this stage since the function will later be unified with the XY discomfort function and normalized as a whole.

5.4 Combining Discomfort in Z with XY

After deriving the Z-axis discomfort function in Equation 5.4, integrating it with an XY-plane discomfort model yields a comprehensive XYZ discomfort representation, defining the full 3D Personal Space. We employ the the planar (XY) personal space using the traditional Asymmetric Gaussian Function (AGF) [46], described in the previous Section 4.3 and showed in Figure 4.3.

Integrating the fuzzy Z-axis function with the AGF-based XY-plane model is straightforward, as both output discomfort values are within $[0, 1]$. This combination follows an intuitive rationale: if an object is positioned at a height where the fuzzy Z-axis function returns a near-zero value, the overall discomfort remains negligible, regardless of its proximity in the XY-plane. Similarly, if an object is far in the XY-plane, leading to a low AGF output, the overall discomfort remains low regardless of height. In intermediate cases, the combined discomfort follows: when one component (Z or XY) is high and the other is intermediate, the resulting discomfort exceeds the intermediate level; if one is intermediate and the other low, the overall discomfort remains low.

To achieve this behavior, we use a geometric mean to combine both components. This approach naturally meets the described requirements, ensuring a smooth and consistent integration into a unified 3D proxemics discomfort function:

$$\mathcal{P}(\mathbf{n}, \mathbf{r}) = \sqrt{\mathrm{AGF}_{\mathbf{n}}(\mathbf{r}_x, \mathbf{r}_y) * f(\mathbf{r}_z)} \ . \tag{5.5}$$

Finally, a normalization step is applied to scale the maximum values of the AGF and $f(z)$ integration to 1. This guarantees consistency and preserves the relative contributions of both components in the final discomfort function. The overall equation to compute the three-dimensional discomfort is straightforward to implement. We have provided a Python implementation to facilitate its use[1].

To make the model applicable to our motion planners, we integrate it with the multi-point formulation and extend it to multiple individuals, yielding a 3D $\mathcal{S}$. To achieve this, we simply replace the original `AGF` in Equation 4.1 with the proposed $\mathcal{P}$, as shown below by Equation 5.6:

[1]Code: https://github.com/verlab/roman2025-3d-personal-space

$$\mathcal{S}(\cdot) = \sum_{i=1}^{m}\sum_{j=1}^{s} w_j \mathcal{P}_i(\mathbf{n}, \mathbf{r}) = \sum_{i=1}^{m}\sum_{j=1}^{s} w_j \sqrt{\mathrm{AGF}_i(\mathbf{r}_{jx}, \mathbf{r}_{jy}) * f_i(\mathbf{r}_{jz})}, \tag{5.6}$$

Chapter 6

Decoupled Socially-aware 3D Motion Planner

In this chapter, we address the scenario of socially-aware navigation in a three-dimensional environment. In previous chapters, we employed the Risk-RRT* technique for planar (2D) environments. Extending this method to 3D is conceptually straightforward; however, its practical implementation introduces additional complexities.

First, simulating and executing the planner in 3D requires a dedicated three-dimensional simulator, particularly for accurate collision checking. This significantly increases computational cost, as the environment must be continuously queried during sampling and path validation. Furthermore, when increasing the number of links in the manipulator to better reflect realistic mobile manipulators, the dimensionality of the configuration space rises, making sampling-based approaches take longer to find good solution. Another major bottleneck is the communication latency between the planner and the simulator, which was observed that can dominate the computation time compared to the relatively lightweight 2D implementation that does not rely on simulation feedback. There are alternative solutions for faster collision checking that do not rely on a dedicated simulator, such as physics-based engines or specialized collision-checking libraries. However, they come with trade-offs in terms of flexibility and implementation complexity. Modifying the environment or robot model within can be considerably more difficult compared to using a simulator with a graphical interface, hence, those alternatives were not tested.

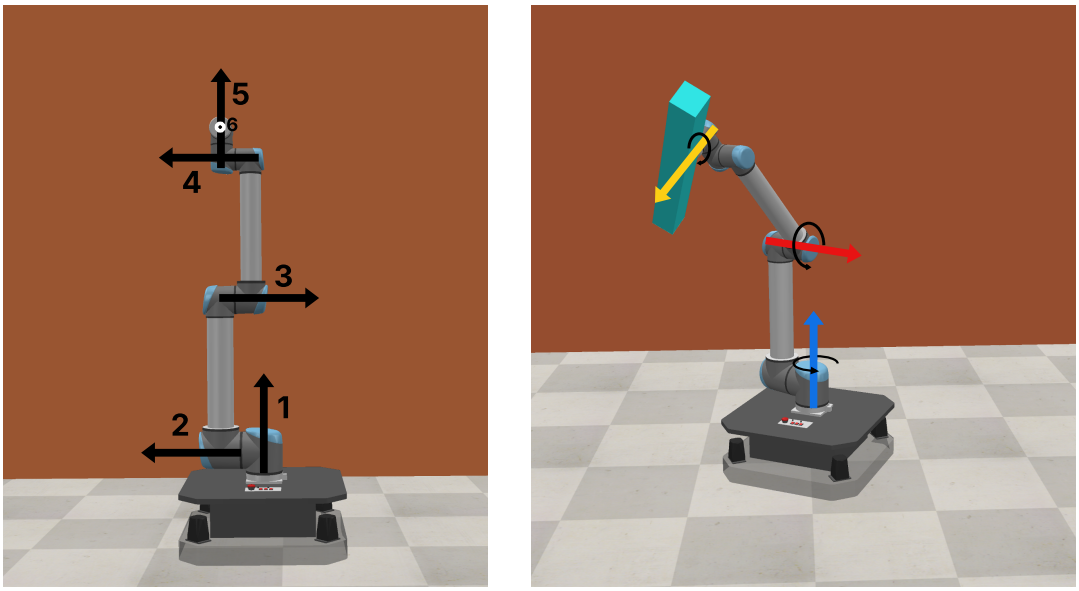
Given the limitations described, an alternative approach to the Risk-RRT* is proposed. Here, we introduce a decoupled motion planning strategy, where the robot's motion is divided into two sequential stages: first, planning the path for the mobile base, and subsequently, planning the manipulator's motion along that path. This decomposition allows the system to maintain social awareness while significantly reducing computational demand.

Several elements from the planar case remain unchanged or are directly translated to the three-dimensional setup, and thus are not discussed further in detail. The social cost function used here corresponds to the three-dimensional discomfort model presented in the

previous chapter, which quantifies the social cost at each point in space considering vertical (height-dependent) sensitivity. The multi-point mehtod remains the same, but notice that now the points have z-coordinate to represent elevation, also the points are distributed following the same principles described in the 2D framework. In this environment, we perform experiments with the bar-object only.

6.1 Robot

The mobile manipulator considered in this section employs the same holonomic mobile base and manipulator used in the 2D environment and is always carrying the bar-like object. However, the manipulator UR10 model is mounted in a different configuration and with other joints activated.



(a) All joints numbered. (b) Selected active joints.

Figure 6.1: Mobile manipulator used in the three-dimensional environment. The manipulator is mounted at the center of the omnidirectional base, with arrows indicating the rotational axes of the joints.

The manipulator is centrally positioned on top of the omnidirectional base, and the object is attached to the end-effector. The Figure 6.1a shows the mobile manipulator and all the joints numbered, from 1 to 6, with arrows indicating the rotational direction of the joints. At joint six, the arrow is pointed outside towards outside the paper. We begin our experimentation with three independent joints activated, one more than in the bidimensional case, resulting in a 3R manipulator arm. The robot initial configuration

and joint activation scheme are defined such that each joint performs rotation about one of the three principal axes, x, y, and z. This initial setup, including the object and the rotational directions of the joints, is illustrated in Figure 6.1b.

The multi-interest points are defined at the three manipulator joints and at the extremities of the object, resulting in a total of five points. Although additional points could be assigned to the mobile base, the base center coincides with the location of the first joint of the manipulator; therefore, an extra point was deemed unnecessary.

6.2 Motion Planner

As we introduced, the motion planner is a decoupled approach. We first plan the path that the base will do with the manipulator static in a position. The manipulator is fixed with the arm lower to the ground, as it was theorized that lower induces less discomfort as it is farther from sensible head parts. The Figure 6.2 shows the mobile manipulator start configuration with a person on the side.

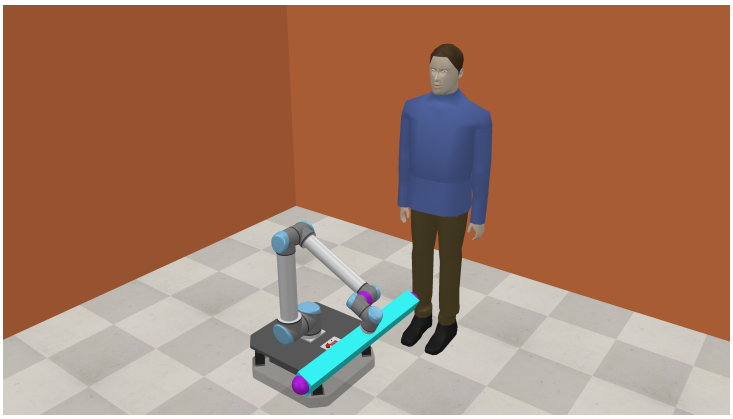
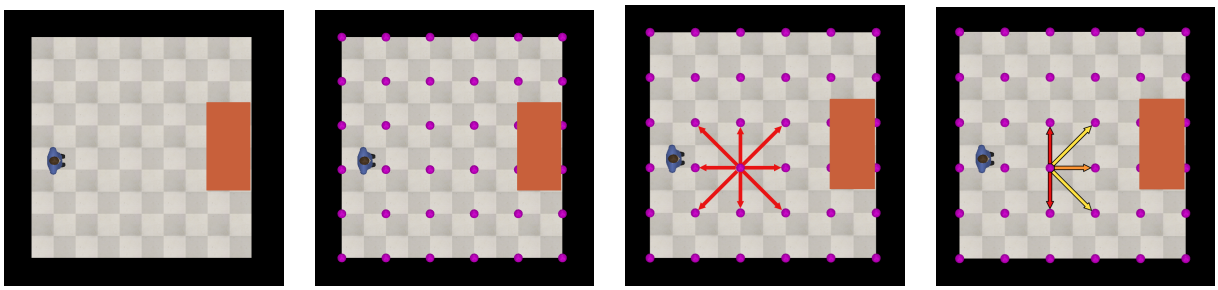


Figure 6.2: Start configuration of the mobile manipulator, where the manipulator is static.

For fast computation of the path for the mobile base, we employ the A* algorithm. Prior to applying A*, a graph representation of the environment must be constructed. We adopt a grid-based graph, where nodes are uniformly spaced at one-meter intervals. Each node is connected to up to eight neighboring nodes: left, right, down, up and the four diagonals. To achieve socially-aware navigation, edge weights are assigned according to the social cost associated with moving from one node to another. This process is illustrated in figure 6.3 for one node.

This cost is computed using the same integration process described in the bidimensional motion planner; however, instead of the 2D social cost function, we employ the proposed three-dimensional formulation. During graph construction, edges are added



(a) Environment. (b) Adding nodes. (c) Adding edges. (d) Updating edges.

Figure 6.3: Graph construction illustration. (a) The raw environment, containing one obstacle and one person. (b) Nodes added to the environment, shown as purple spheres with a 1-meter resolution. (c) Edge generation from a selected node. (d) Edge weighting based on social cost, color-coded from best to worst: yellow, orange, and red. Edges that would result the robot colliding with the human are excluded.

only if the interpolated path between nodes remains valid, meaning it is collision-free. Once the graph is built, the A* algorithm is executed. We highlight here that as the environment becomes larger, the graph construction time increases primarily due to the higher number of edge weight calculations required. This effect can slow down the process since the computation is performed through the simulator's API for collision checking.

The result of the A* yields a list of nodes with the robot configurations that navigates the robot from the start point to the goal, minimizing the discomfort (edge weights) considering a fixed manipulator. After obtaining the motion from the A* algorithm, we apply a simplification step to remove redundant intermediate nodes that lie on straight segments. For instance, if the computed path contains consecutive nodes such as (1, 0), (2, 0), and (3, 0), the intermediate node (2, 0) is removed, resulting in a simplified path defined by the endpoints (1, 0) and (3, 0). This reduces unnecessary waypoints.

Next, we plan the manipulator motion using a sampling-based technique, shown in Algorithm 3. Given the sequence of base positions obtained by A* with the arm fixed, we plan the manipulator's motion between consecutive nodes in a socially-aware manner. The approach is as follows: given the previous and following position, we randomly sample manipulator configurations for the following position, and, similarly to RRT, use the `Steer` function to obtain the actual sampled configuration. The sampler, indicated in the algorithm (line 8) as `SampleRandomAngles()`, generates the first two samples in a specific manner. The first sample returns a configuration where the manipulator angles remain the same as in the previous configuration, but with the position of the next configuration, in other words, it does not move the arm only the base. The second sample corresponds to the manipulator first fixed-position configuration, which is known to yield low discomfort. After these two predefined samples, the remaining $N_{samples} - 2$ are generated uniformly randomly within the joint limits.

We compute the motion social cost of going from the previous configuration to the

next sampled. If the cost is lower than the current lowest, interpolate from the previous to the following and test for collisions (considering the base motion). Multiple samples ($N_{samples}$) are evaluated to go from the previous to the next configuration, and the motion with the lowest social cost is selected. It is worth noting that the algorithm's runtime depends on the size of the path $\mathcal{C}$, if the A* algorithm produces a path with many nodes, the manipulator sampling process will take longer.

Algorithm 3 Manipulator Sampling

$\mathcal{C} \leftarrow [\mathbf{q}_1^{\mathrm{MM}}, \mathbf{q}_2^{\mathrm{MM}}, ..., \mathbf{q}_n^{\mathrm{MM}}]$
for $i = 1$ **to** $n - 1$ **do**
  $\mathbf{q}_{prev}^{\mathrm{MM}} \leftarrow \mathcal{C}[i]$
  $\mathbf{q}_{next}^{\mathrm{MM}} \leftarrow \mathcal{C}[i+1]$
  $\mathbf{q}_{best}^{\mathrm{MM}} \leftarrow \mathbf{q}_{next}^{\mathrm{MM}}$
  `bestCost` $\leftarrow$ inf
  while $j < N_{samples}$ **do**
    $\mathbf{q}_{sample}^{\mathrm{MM}} \leftarrow$ `SampleRandomAngles(`$\mathbf{q}_{next}^{\mathrm{MM}}$`,`$\mathbf{q}_{prev}^{\mathrm{MM}}$`,`$j$`)`
    $\mathbf{q}_{sample}^{\mathrm{MM}} \leftarrow$ `Steer(`$\mathbf{q}_{prev}^{\mathrm{MM}}$`,`$\mathbf{q}_{sample}^{\mathrm{MM}}$`,`$\delta$`)`
    $\Phi \leftarrow$ `Msc`$(\mathbf{q}_{prev}^{\mathrm{MM}}, \mathbf{q}_{sample}^{\mathrm{MM}}, \mathcal{M})$
    if $\Phi <$ `bestCost` **then**
      if `CollisionFree(`$\mathbf{q}_{prev}^{\mathrm{MM}}, \mathbf{q}_{sample}^{\mathrm{MM}}, \mathcal{M}$`)` **then**
        `bestCost` $\leftarrow \Phi$
        $\mathbf{q}_{best}^{\mathrm{MM}} \leftarrow \mathbf{q}_{sample}^{\mathrm{MM}}$
      end if
    end if
  end while
  $\mathcal{C}[i+1] \leftarrow \mathbf{q}_{best}^{\mathrm{MM}}$
end for
return $\mathcal{C}$

We made one refinement to the base algorithm provided, that improved the results. Instead of integrating the full mobile manipulator configuration for computing the social cost, we integrate only over the base distance. This change avoids excessive penalization being observed when the manipulator moves slightly. Those modifications suggests a possible direction for future improvement even for the 2D case.

Finally, this manipulator planning stage produces a corresponding manipulator configuration for each base position determined by the A* path. As a result, the complete motion of the mobile manipulator is generated in a socially-aware manner, ensuring that both the base and the arm move coherently from the initial configuration to the final goal while minimizing social discomfort.

Chapter 7

Experimentation and Results

In this chapter, we present the experiments conducted and discuss their results. We begin with the bidimensional scenario, using the 2D whole-body motion planner implemented in MATLAB. Then, we generate and analyze visualizations of the three-dimensional discomfort model, also using MATLAB. Finally, we perform experiments in a three-dimensional environment employing the 3D decoupled planner implemented in Python. All simulations were carried out in CoppeliaSim. In particular, for the 3D decoupled approach, we used CoppeliaSim during the planning process through its Python API.

All experiments were executed on a computer running Windows 11, equipped with an Intel(R) Core(TM) i7-10510U CPU @ 1.80,GHz (up to 2.30,GHz) and 8,GB of RAM.

7.1 Environments

We defined two environments for our experiments: a generic scenario and an office-like scenario. As previously described, two objects were defined: a bar-like object and an L-shaped object. The multi-interest points for the planar arm with the bar object are denoted as M_{pb}, for the planar arm with the L-shaped object as M_{pl}, and for the full three-dimensional manipulator with the bar object as M_{fb}. All coordinates are expressed with respect to the mobile base frame as imported into CoppeliaSim; note that the z coordinate does not represent the vertical axis in this context. The robots are positioned according to Figures 4.4a and 4.4b, and as 6.1a.

$$M_{pb}(y,z) = \{(0,0),\ (0.6,0.0),\ (0.6,0.7),\ (1.2,0.7),\ (0.0,0.7)\}$$
$$M_{pl}(y,z) = \{(0,0),\ (0.6,0.0),\ (0.6,0.7),\ (1.9,0.7),\ (-0.7,0.7),\ (-0.7,-1.0)\}$$
$$M_{fb}(x,y,z) = \{(0.25,0.0,0.25),\ (0.88,-0.11,0.026),\ (1.45,-0.10,0.02),$$
$$(1.58,-0.93,-0.11),\ (1.58,-0.57,-0.11)\}$$

The generic environment measures 20 × 20m and contains 10 people distributed

throughout the space with varying orientations, showed in Figure 7.1a. The office environment measures 25×25m and contains 12 people, showed in Figure 7.1b. The configurations of people in each environment are denoted by $\mathcal{N}_{generic}$ and $\mathcal{N}_{office}$ showed below, specified in meters and radians, following the pattern $q_i = (x, y, \theta)$.

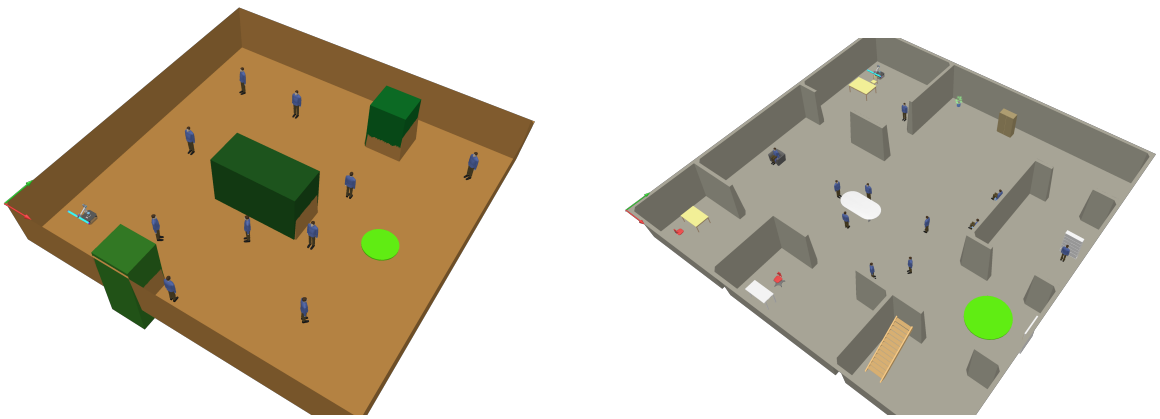
$$\begin{aligned}\mathcal{N}_{generic} = \{&(6, 4, 0.63), (15, 4, 0.78), (3.5, 10, -1.57), (13, 12, 1.57),\\ &(10, 6.5, 3.77), (18, 17, 4.19), (3, 16, 4.19), (7, 16, -1.57),\\ &(9, 1.5, 1.57), (13, 8, -1.57)\}\end{aligned}$$

$$\begin{aligned}\mathcal{N}_{office} = \{&(6.975, 20, -1.48), (7.225, 16.625, 1.66), (10.025, 11.85, -1.48), (8.125, 10.775, 0.0),\\ &(10.425, 9.025, 1.66), (2.05, 10.3, -1.57), (17.275, 14.1, 3.14), (17.125, 17.25, 3.05),\\ &(14.425, 6.925, 0.78), (16.075, 8.725, -2.27), (23.525, 15.475, 1.57), (15, 12, -3.05)\}\end{aligned}$$

The positions and orientations of individuals were chosen not to reflect realistic scenarios, but to test the social navigation method. The configuration is designed to encourage the planner to identify longer yet socially preferable paths. In particular, individuals are oriented to face the probable path of the robot, thereby creating challenging situations that stress the planner's ability to respect personal spaces while navigating.

The start and goal positions were fixed for each environment as follows:

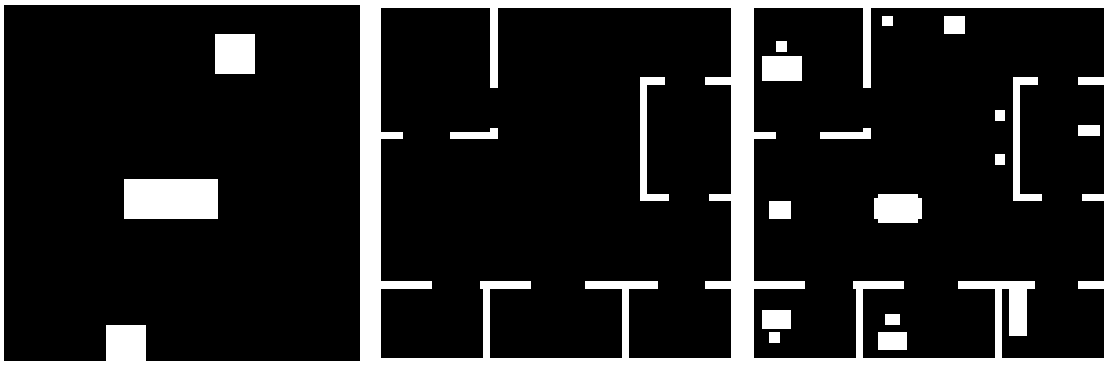
$$\begin{aligned}\mathbf{q}^{\mathrm{MM}}_{startOffice} &= (2, 23, \dots), \quad \mathbf{q}^{\mathrm{MM}}_{endOffice} = (23, 3, \dots),\\ \mathbf{q}^{\mathrm{MM}}_{startGeneric} &= (2, 3, \dots), \quad \mathbf{q}^{\mathrm{MM}}_{endGeneric} = (16, 10, \dots).\end{aligned}$$



(a) Generic (20x20m). (b) Office (25x25m).

Figure 7.1: Simulated environments used in the experiments. The reference frame is located at $(x, y) = (0, 0)$, with the red and green axes representing the x- and y-directions, respectively. Robot is in the initial position and the green circle is the goal.

The environments are positioned such that their bottom-left corner corresponds to $(0, 0)$. We also provide the corresponding black-and-white overhead image of the environments used as heightfields in Figure 7.2, where black represents free space and white indicates obstacles.



(a) Generic. (b) Office Walls. (c) Office with furniture.

Figure 7.2: Overhead black-and-white heightfield representation of the environments. Black areas indicate free space, while white areas represent obstacles.

7.2 Bidimensional Navigation Experimentation

To evaluate our method, we first compare social navigation algorithms with a non-social baseline (RRT*) in two distinct environments (Sections 7.2.1 and 7.2.2). Section 7.2.3 provides a detailed comparison between our approach and other social navigation methods, while Section 7.2.4 examines the effect of varying social cost weights on the resulting trajectories. Each algorithm was executed 10 times per environment with $K = 2000$. Unless stated otherwise, all multi-point approaches use uniform weights ($\{w\}_{j=1}^{5} = 1$).

For these experiments, we evaluate four social navigation algorithms. First, our multi-point social Risk-RRT* approach, which considers the full mobile manipulator, including the base, arm, and carried object. Next, we extend this approach with the Informed-RRT* [26] forming a Social Informed-Risk-RRT* (IR-RRT*) inspired by its recent applications in social navigation [84, 16], where sampling is restricted to an ellipsoid in the XY-plane after the first solution is found. We note that IR-RRT* is not the main choice, as it is biased toward the first path it finds and is not an optimal solver. However, its results may be desirable for applications where a balance between distance and comfort is more desired. Finally, since most existing methods focus solely on the base, we compare both algorithms with their corresponding base-only variants, implemented by assigning zero weight to the manipulator and object while keeping a weight of one for the base in the social cost function. The purpose of the base-only approach is to evaluate whether existing traditional base-only methods are able produce results comparable to our multi-point planner. Since our discomfort metric is multi-point, the traditional methods are expected to perform worse by default. The comparison aims to confirm this expectation.

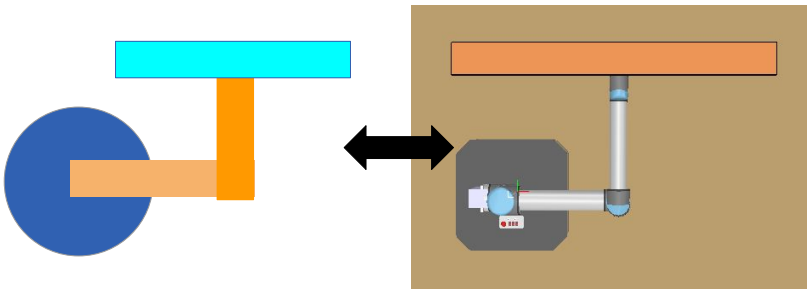


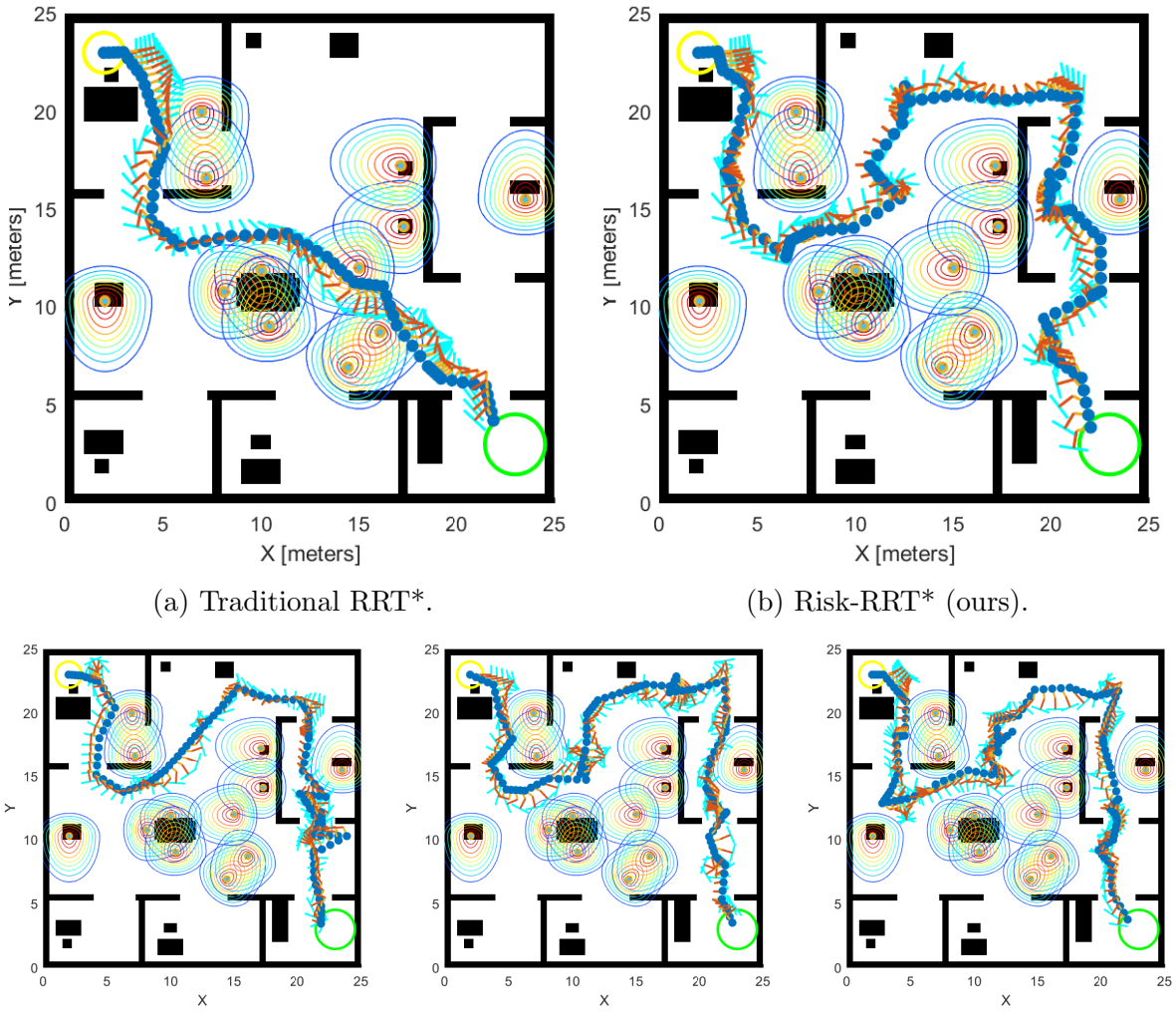
Figure 7.3: Comparison between plotted robot to real robot. Base is the blue circle, link one the light orange, link two the dark orange and the object in cyan blue.

We also provide video demonstrations of the best result for each scenario, for both the Social Risk-RRT* and the traditional RRT*, showing the robot executing the offline-planned motions in the simulated environment. To simplify the plots, the colorbar representing social cost around humans has been omitted (see Figure **??**). The results are plotted in MATLAB, and the correspondence between the plotted robot and the real robot is illustrated in Figure 7.3.

7.2.1 Complex environment

The first environment is a complex office-like setting, where the robot transports a 1.5-meter-long bar (Figure 4.4a). We show the best results from every algorithm in Figure 7.4, providing a meaningful comparison of social versus non-social versus base-only planning. A detailed analysis of social cost versus distance for all algorithms is presented in Figure 7.5.

Figure 7.4 shows that our approach effectively directs both the robot's base and manipulator away from high-discomfort regions, reducing social disturbances. This trend is further supported by Figure 7.5, which shows that our method, along with other social approaches, consistently achieves lower social costs compared to RRT*. However, as expected, our paths tend to be longer, since the RRT* minimize distance, while our approach optimizes for social acceptability.



(a) Traditional RRT*.

(b) Risk-RRT* (ours).

(c) Risk-RRT* base-only.

(d) IR-RRT* (ours).

(e) IR-RRT* base-only.

Figure 7.4: Comparison of navigation results in an office-like environment, robot transporting a bar object, and all weights uniformly set to 1. Video: https://youtu.be/lLFQ12lf_go.

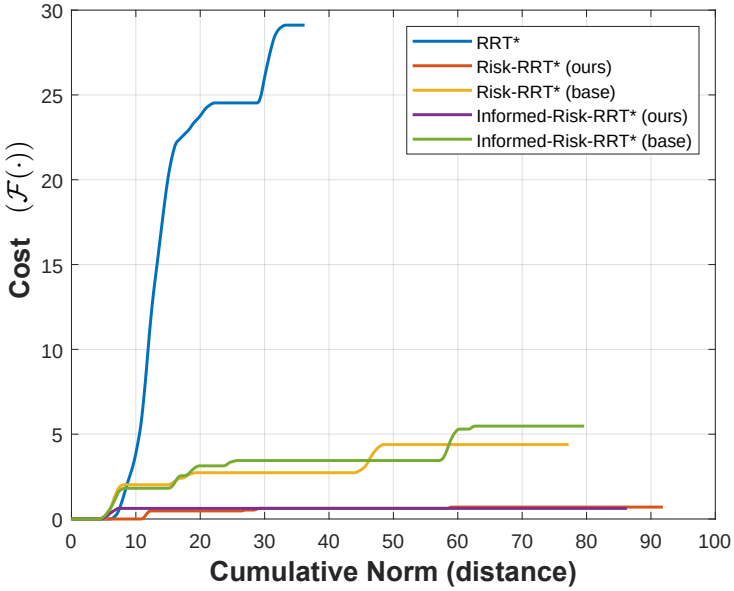
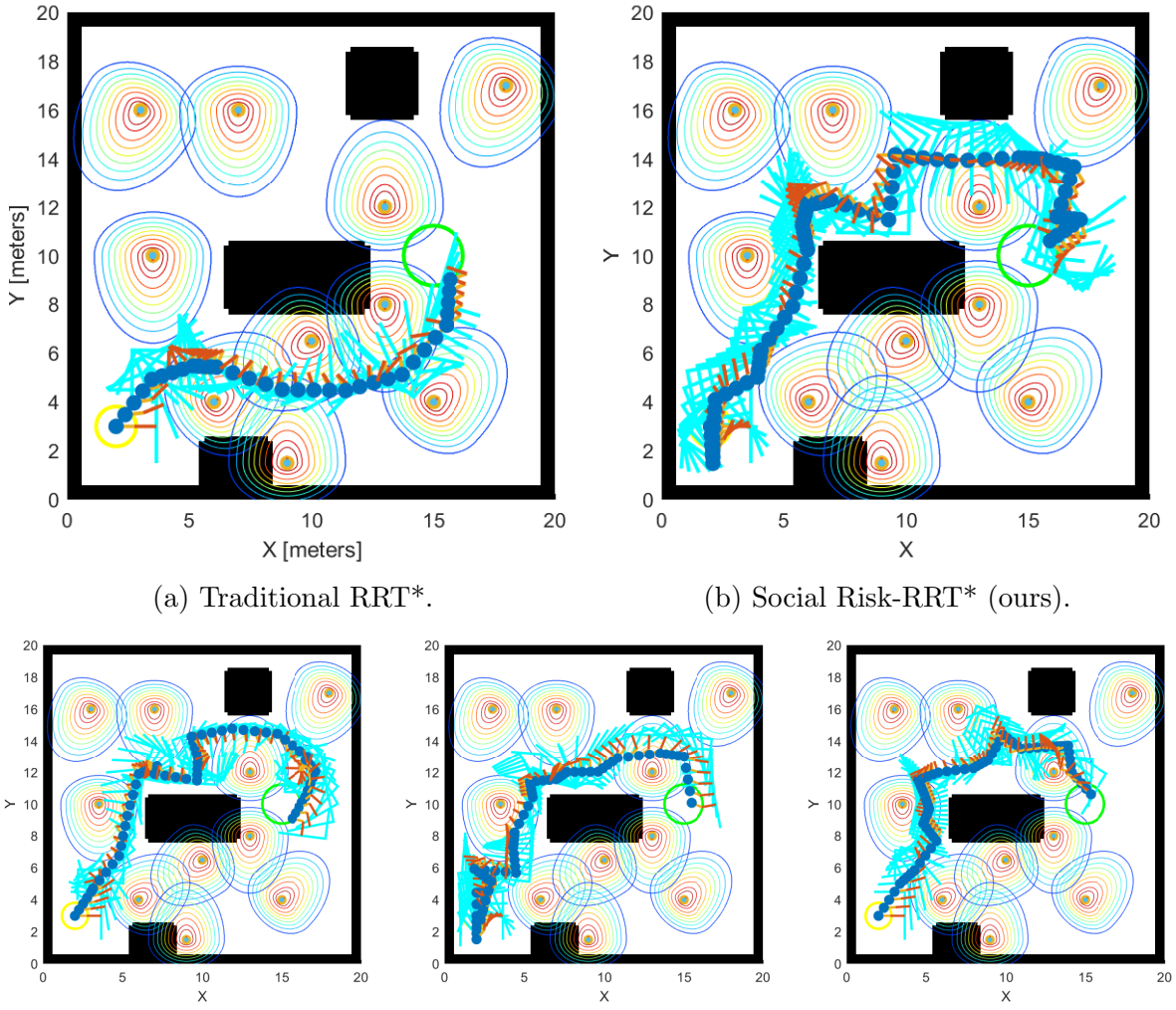


Figure 7.5: Graph comparing the results of social methods with RRT* in an office-like environment with a bar-shaped object. The Y-axis represents the cost, while the X-axis shows the cumulative norm.

7.2.2 Complex Object

In this section, we evaluate our method in a generic map with a complex L-shaped object composed of two bars measuring 3.0 and 2.0 meters in length (Fig. 4.4b). The tighter environment and intricate object geometry make finding an optimal social path more challenging. Figure 7.6 presents the best results for all algorithms tested, while Fig. 7.7 displays the cost *vs.* distance graph.



(a) Traditional RRT*.

(b) Social Risk-RRT* (ours).

(c) Risk-RRT* base-only.

(d) IR-RRT* (ours).

(e) IR-RRT* base-only.

Figure 7.6: Comparison of navigation results in a tighter environment, robot carrying an L-shaped object, and all weights uniformly set to 1. Video: https://youtu.be/c8PtqI2iLMM.

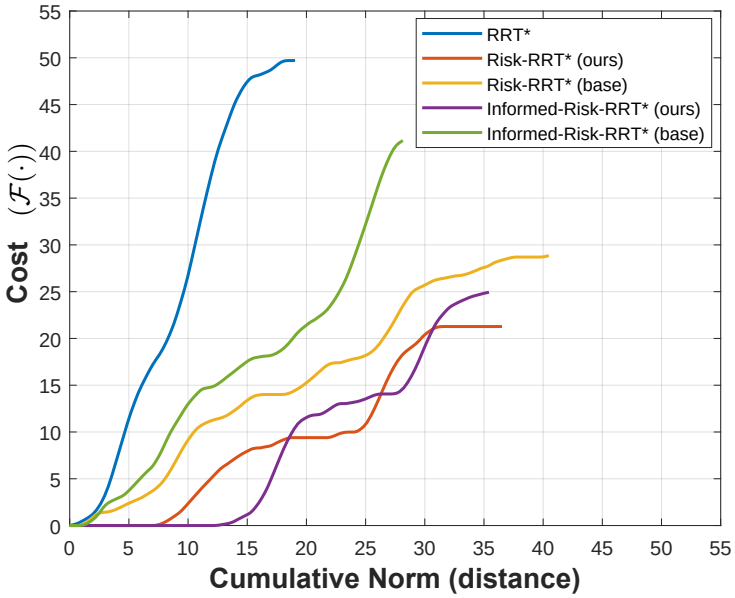


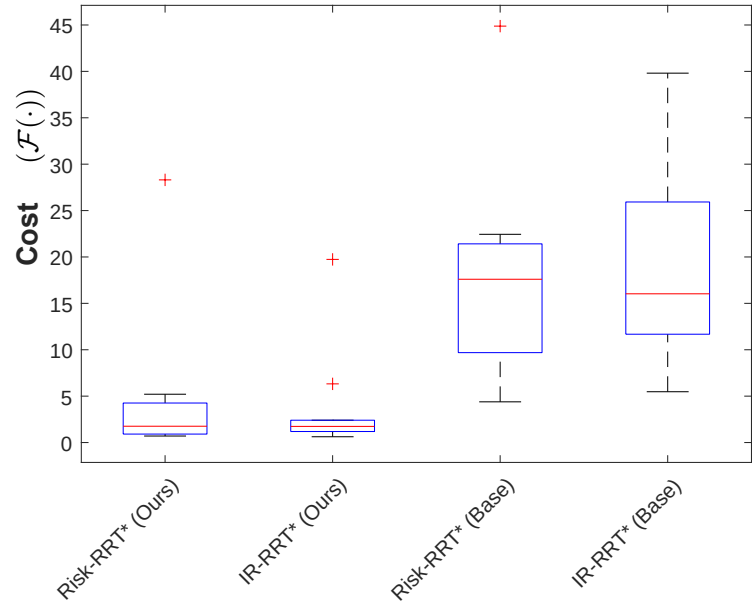
Figure 7.7: Graph comparing the results of social methods with RRT* in a tighter generic environment with a L-shaped object. The Y-axis represents social cost, while the X-axis shows the cumulative norm.

The results demonstrate that our method effectively navigates the robot while minimizing proximity to people whenever possible. However, in narrow corridors, limited path options force the robot or the object to get close to individuals. Compared to the standard RRT*, which often cuts through densely populated areas, our approach actively avoids these regions, generating more socially-aware paths. This is further supported by Figure 7.7, which shows that, despite producing longer paths, all social methods consistently achieve lower social costs than the baseline RRT*. This result indicates that the primary limiting factor in the scenario was collision avoidance. The constrained maneuvering space, particularly caused by the presence of the large object, restricted the planner's ability to optimize social cost effectively. As a consequence, the robot's trajectory was largely dictated by the need to avoid obstacles and maintain feasible configurations, leaving limited flexibility to improve comfort around humans. This highlights the inherent trade-off between collision-free motion and social optimization in environments with tight spatial constraints.

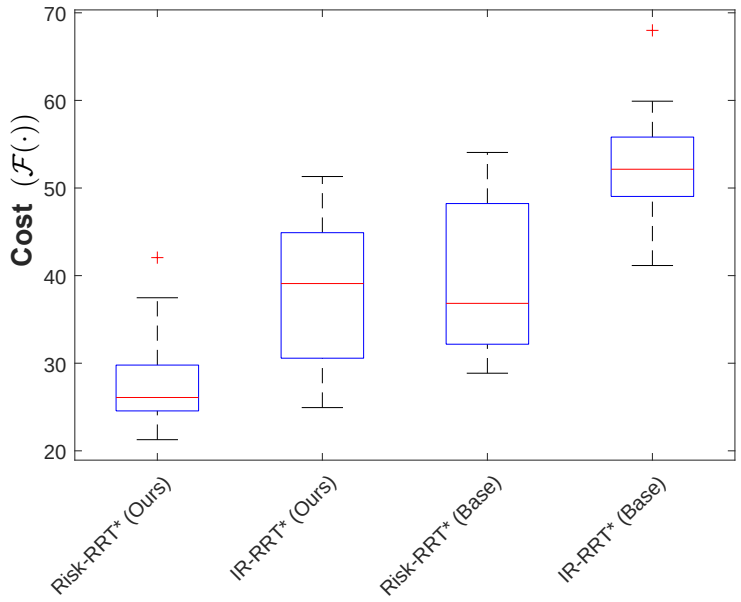
7.2.3 Quantitative Results

We quantitatively evaluate the four previously described social algorithms, comparing their performance against each other and their base-only counterparts. To this end, we analyze boxplots of the 10 resulting path costs, i.e., $\mathcal{F}(\mathbf{q}_{goal}^{\mathrm{MM}})$. The results for

the office-like environment (Section 7.2.1) are shown in Figure 7.8a, while those for the generic environment (Section 7.2.2) appear in Figure 7.8b.



(a) Office-like environment with Bar object.



(b) Generic environment with L-shaped object.

Figure 7.8: Boxplots of the final cost $\mathcal{F}(\mathbf{q}_{end}^{\mathrm{MM}})$ for each run of the social algorithms. The Y-axis represents the cost, while the X-axis indicates the algorithm applied.

In the office scenario, the multi-point Informed-Risk-RRT* achieved the best performance, closely followed by the multi-point Risk-RRT*. However, in the generic environment, the multi-point Risk-RRT* significantly outperformed the informed variant.

This behavior is expected: Informed approaches become highly biased toward the first path found since sampling is restricted after the initial solution. In contrast, the

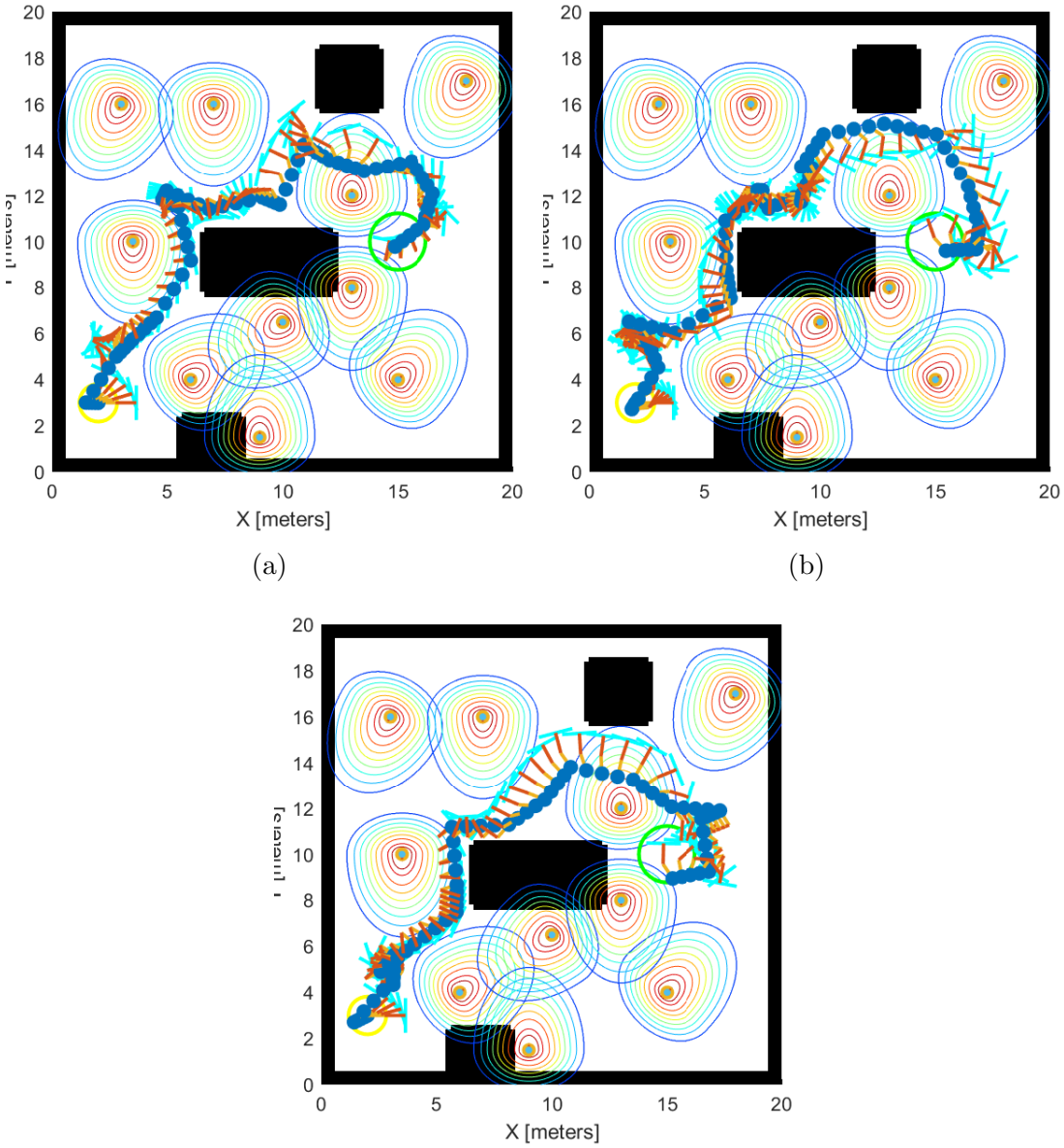
original Risk-RRT* continues exploring the entire space, even after identifying a feasible path. As a result, it has a higher chance of discovering longer but more socially acceptable paths. Once an informed variant is biased by a suboptimal initial path, it is less likely to sample and identify better alternatives.

Overall, it is clear that in both scenarios, the multi-point approaches consistently outperformed their base-only counterparts, demonstrating the superiority of our proposed method and the need for specific mobile manipulator techniques for socially-aware mobile manipulators.

7.2.4 Influence of Social Cost Weight

In this test, we used the generic map with the bar object. We evaluated three distinct weighting strategies. In the first case, we prioritized the points associated with the object, assigning them a weight of 0.95, while the remaining points were weighted at 0.05. In the second case, the focus shifted to the base, with the base point receiving a weight of 0.95 and the other points weighted at 0.05. Lastly, in the third scenario, we applied equal weights to all points, as in the previous sections. The best results are shown in Figure 7.9.

We observe that assigning greater weight to the base leads it to maintain a larger distance from people, while the object moves closer, and vice versa when the object is prioritized. These behaviors align with expectations, showcasing the method's flexibility. However, assigning equal weights yields similar results to prioritizing the object, as the object has more interest points than other parts, giving it an inherent advantage. This effect could be mitigated by rebalacing the weigths. In summary, our approach allows for prioritizing which parts of the robot to be more cautious around people.



(a) (b)

(c)

Figure 7.9: Social navigation results for different weights. (a) Weight of 0.95 for the object and 0.05 for the rest. (b) Weight of 0.95 for the base and 0.05 for the rest. (c) Equal weights. Video: https://youtu.be/tNo6pqw-BAA.

7.3 Three-dimensional Discomfort Model Results

7.3.1 Illustrative Example

To illustrate the spatial distribution of discomfort[1], we present the 3D isosurface corresponding to a discomfort threshold value of 0.5 in Figure 7.10. Additionally, Figure 7.11 includes 2D surface plots of the ZX and ZY planes, where the color coding represents the corresponding discomfort values.

Discomfort values are highest near the head, reflecting greater sensitivity, and lowest around the legs, where sensitivity is typically lower. This reinforces the idea that robots should position components like arms or tools closer to the legs, rather than near the head, aligning with human comfort expectations. These findings emphasize the importance of considering both horizontal and vertical dimensions when designing robots to respect human personal space.

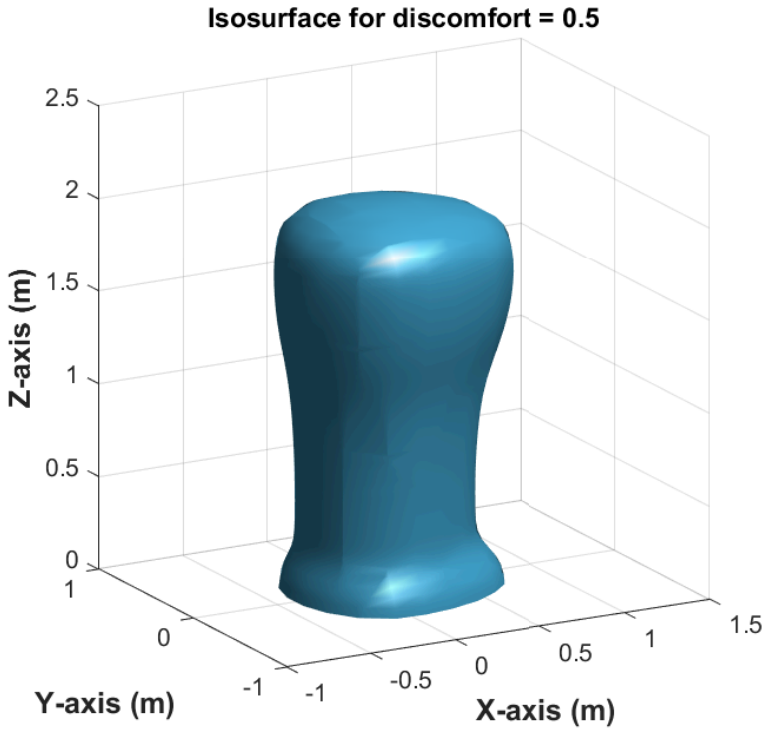


Figure 7.10: 3D isosurface representing a discomfort threshold value of 0.5 for a person of height 1.75 m, centered at (0,0) and facing along the X-axis.

We also note that, although the Z-axis has an effect, the X- and Y-axes generally dominate, except when very close to the person. Analyzing Figure 7.11, consider the

[1]Code: https://github.com/verlab/roman2025-3d-personal-space

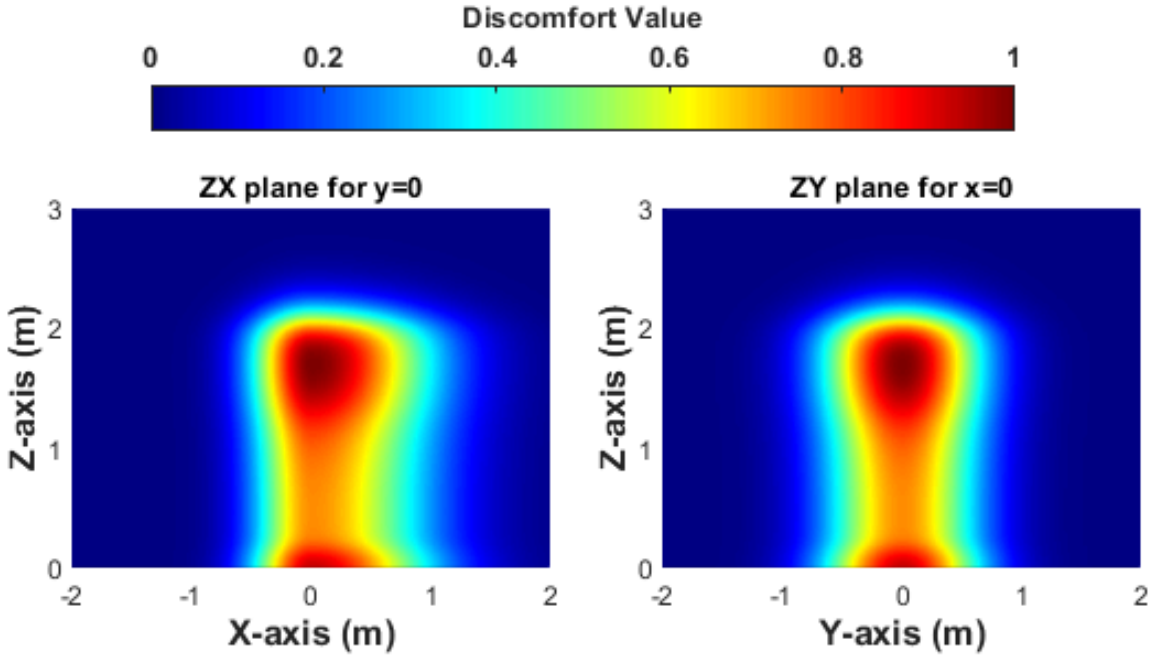
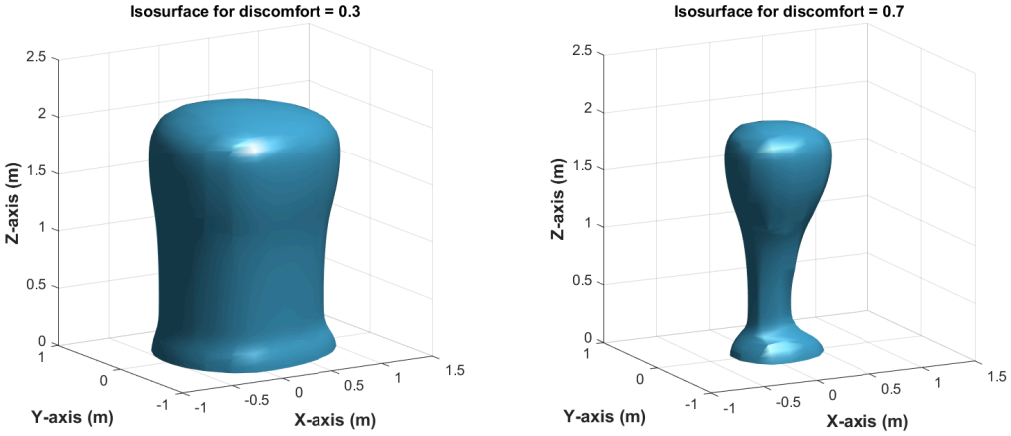


Figure 7.11: Surface plots of the XZ and YZ planes for a person at $\mathbf{q} = ((0, 0), 0)$ and with a height of $1.75\,m$.

1-meter mark on the x-axis along the ZX plane: moving straight upward causes the value to change slightly; a similar behavior is observed in the ZY plane. However, at a horizontal distance of around 0.3 meters, moving straight up on the z-axis results in a more substantial variation in the discomfort value.

7.3.2 Different Discomfort Level Thresholds

Figure 7.12 shows the 3D isosurfaces for two discomfort thresholds. At a low discomfort level (0.3), the surface is relatively uniform, indicating minimal variation across body regions. In contrast, at a higher discomfort level (0.7), greater sensitivity near the head becomes evident, requiring a robot to be much closer to the legs to induce the same discomfort as it would farther from the head.

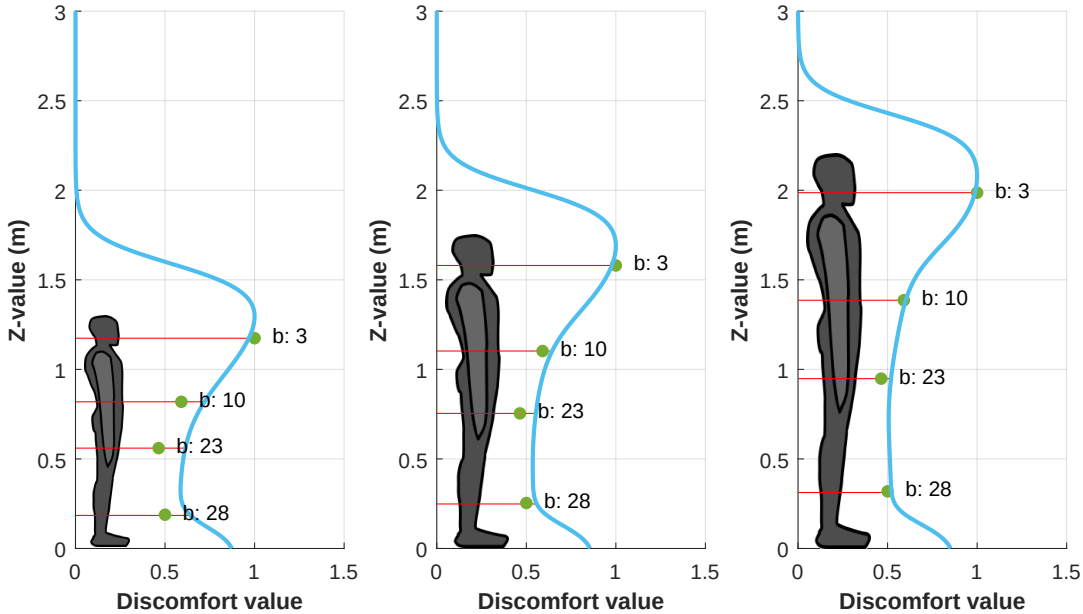


(a) (b)

Figure 7.12: Comparison of 3D isosurfaces for discomfort levels at 0.3 and 0.7. Subfigures (a) and (b) illustrate the spatial variations in discomfort sensitivity at each threshold.

7.3.3 Height Variability

Our model also adapts to varying heights, as illustrated in Figure 7.13, which presents normalized Z-axis discomfort profiles for individuals of $1.30\,m$, $1.75\,m$, and $2.20\,m$. The vertical distribution scales proportionally, ensuring proper alignment of key regions such as the head, torso, and lower body. This adaptability preserves consistency across diverse profiles without necessitating structural modifications.



(a) $h = 1.30\,m$ (b) $h = 1.75\,m$ (c) $h = 2.20\,m$

Figure 7.13: Adaptability to different height profiles.

Notably, discomfort drops to near zero approximately $0.75\,\mathrm{m}$ above a person's head, and is already low at around $0.25\,\mathrm{m}$ above the head. In this work, we assume that the robot cannot move above the person height, so this behavior would only be exploited by aerial robots. While the literature lacks full consensus, recent studies suggest that flying above humans is preferred for safe coexistence [11]; thus, our model remains applicable in such scenarios.

7.4 Three-dimensional Navigation Experimentation

For this section, we employ the generic environment setup and utilize the bar-shaped object. All weighting parameters in the cost function are set to one, unless stated otherwise. This environment was selected because it is more populated and smaller compared to the office scenario, which allows faster planning while amplifying the differences observed between methods, as the resulting paths are expected to pass closer to individuals. The bar-shaped object was chosen over the L-shaped alternative to simplify the analysis and reduce the complexity of manipulator motion. Furthermore, we use the the 3D discomfort model and the UR10 arm mounted on the 3D scenario, except for the first test to compare with the decoupled planner.

Since we employ the 3D discomfort model, it is necessary to define the height of each individual. To avoid introducing an additional variable into the analysis, we set everyone's height to the same value of $2.0\,\mathrm{m}$. We chose this taller height because the robot's large arm could otherwise lead to path planning beyond head level, although this assumption would require further experimentation to validate. A subsequent section explores the impact of varying people's height to assess the adequacy of this choice.

Similar to the bidimensional experiments, each scenario is executed ten times to account for the stochastic nature of the method. In the manipulator motion planning step, up to 100 samples are typically generated between each node of the A* simplified result. The sampling process is terminated early if the computed discomfort value falls below 0.1, indicating that the solution is already sufficiently comfortable.

All experiments illustrated in this section are summarized in the following video: https://youtu.be/wqhSNGuErAg.

7.4.1 Decoupled vs Coupled

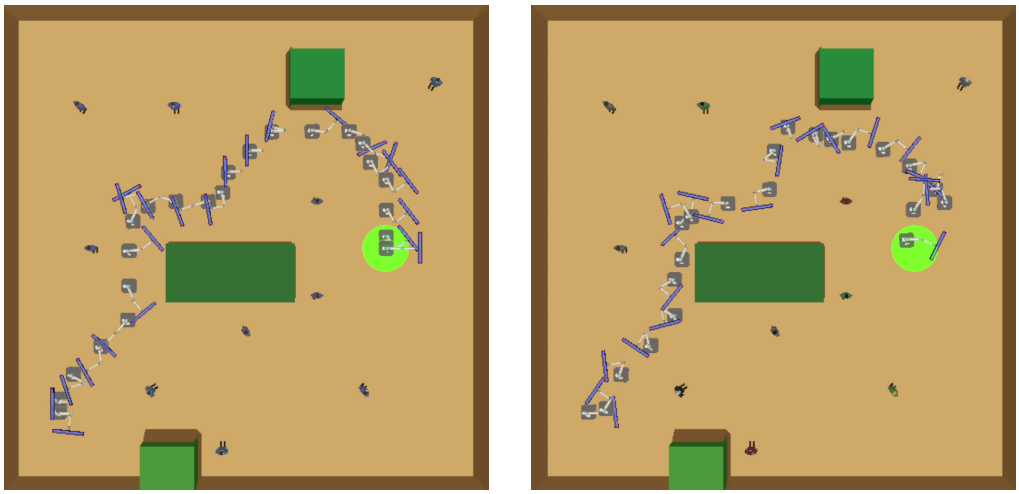
In this section, we compare the performance of the decoupled and coupled planning strategies. Theoretically, the coupled approach is expected to produce superior results, as it allows coordinated motion between the mobile base and the manipulator, optimizing the overall trajectory. In contrast, the decoupled approach plans the motion of each component separately, which may reduce planner time at the expense of coordination efficiency.

For this comparison, we employ the planar robot from the bidimensional environment using the decoupled approach. This setup allows direct comparison with the methods analyzed in the previous bidimensional experiments. However, our decoupled approach tries to optmize the 3D discomfort function, while the Risk-RRT* the 2D discomfort function. Given the planar robot, no z-axis variability is analyzed in this subsection, therefore, the 3D optmization should not have very different results than if we use the 2D discomfort function.

We re-executed the Risk-RRT* experiments, modifying the integration process used to compute the motion's social cost to be same of the decoupled approach. Specifically, the norm was applied only over the displacement of the mobile base, rather than the full configuration space. This adjustment was motivated by the improvements observed during the decoupled approach, to ensure a fairer comparison between. In the end, this change improved the overall best result.

Given the best solutions of each, we re-computed the social cost of the solution using the simulated environment and employing the 2D discomfort function, to compare the motions in the same situation. In theory, the coupled approach should yield superior results, as it simultaneously optimizes the motion of both the base and the manipulator in the full configuration space. However, among ten executions, the best Risk-RRT* solution achieved a total discomfort value of 6.992, whereas the decoupled approach achieved a lower value of 5.666. The resulting trajectories for both planners are shown in the cumulative plot Figure 7.14.

On the contrary to the expected, even in a scenario where the coupled Risk-RRT* is expected to outperform, the decoupled planner produced a better result. This deviation from the theoretical expectation can be attributed to the limited number of iterations allowed for the Risk-RRT* (2000 in this case). Since RRT* is asymptotically optimal, its solution quality improves only as the number of samples approaches infinity. However, within the sampling limits set in this dissertation, the decoupled approach produced results that were comparable to, or even better than, those of the coupled method, while also requiring significantly less time. Analyzing the figure, it is noticeable that the overall motions are similar and mainly differ in the manipulator's trajectory. This observation



(a) Decoupled approach. (b) Coupled approach.

Figure 7.14: Incremental plots showcasing the social navigation of the mobile manipulator in the bidimensional scenario. The robot's goal position is the green circle.

suggests that both techniques effectively aim to minimize social discomfort. Furthermore, the decoupled approach appears to produce smoother motions. The sampling-based approach appears less smooth because it interpolates directly in the combined arm plus base configuration space, causing the base and arm to move simultaneously without independent smoothing. In contrast, the decoupled approach optimizes each subsystem separately, allowing smoother transitions for both the base and manipulator.

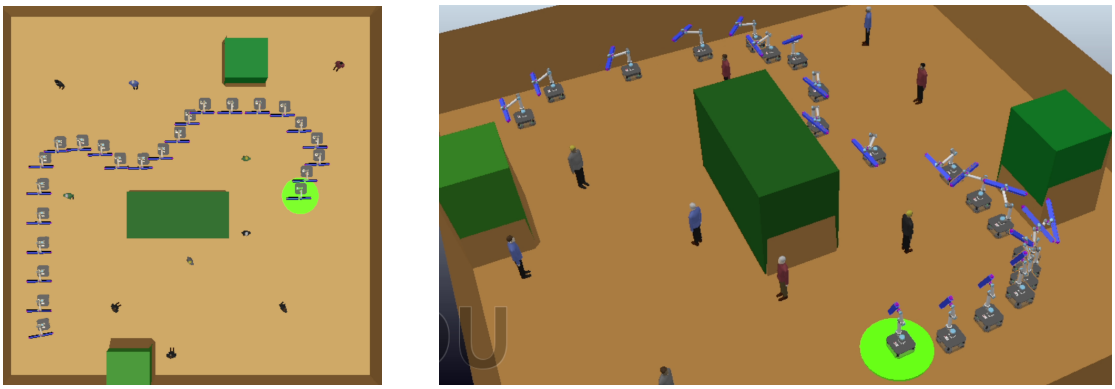
Nevertheless, it is important to highlight a limitation of the decoupled approach in three-dimensional scenarios. Although not explicitly tested here, the proposed method fail in situations where the base-only planner (A*) cannot find a path to the goal while considering a static manipulator. This could happen in the office scenario with the L-shaped object, because it has to go through tight entrances, the robot may not even find its way over the first room. Another impossibility is if the manipulator must move vertically to avoid an obstacle, the A* planner would be unable to account for this, whereas a coupled planner could still find a feasible path. Alternative decoupled strategies could mitigate this issue, such as planning the base trajectory independently and subsequently adjusting the manipulator motion when collisions are detected. However, this was not the strategy adopted in the present work.

During the navigation phase in the simulator, we also observed that the real-time navigator followed the coupled approach more smoothly, which may seem counterintuitive compared to the results shown in the figure. In practice, the arm and base appeared to move in a more coordinated manner when using the sampling-based method. In contrast, with the decoupled approach, the motion occurred in noticeable "bursts," as if the arm needed to move more slowly to keep pace with the base. For a decoupled planner, the real-time path follower can be improved to generate more coordination.

7.4.2 Base Only Comparison

We compare the results obtained when the manipulator is active to those when it remains fixed. The base-only result corresponds to the trajectory generated by the A* algorithm with a static manipulator configuration. Since A* is a deterministic algorithm, the resulting path remains consistent across executions unless different grid resolutions are used or a new graph is employed, details that are beyond the scope of this work. We evaluate how much, if at all, allowing manipulator motion improves the overall solution.

The base-only solution total discomfort motion yield a value of 15.358, this is our baseline. In ten executions of the decoupled approach the minimum discomfort was 8.542 with a average of 10.236 with a standard deviation of 0.944. Both solutions are shown in Figure 7.15, the A* solution with the manipulator fixed in a bird-eye view and the decoupled in a angle manner.



(a) A* solution. (b) Full decoupled with joints 1,3 and 6 active.

Figure 7.15: Incremental social navigation plots: (a) the base-only A* solution with the arm fixed and (b) the result after the manipulator sampling.

Therefore, the discomfort decreased in 44.38%, confirming that, as expected, arm movement enhances overall social navigation. The improvement occurs because manipulating the arm repositions the points used to compute discomfort, increasing their distance from nearby humans in the plane while also adjusting the height. Hence, the decoupled approach successfully improves upon the baseline, indicating that base-only strategies may be suboptimal for mobile manipulators, given the assumptions of the discomfort metric employed in this work.

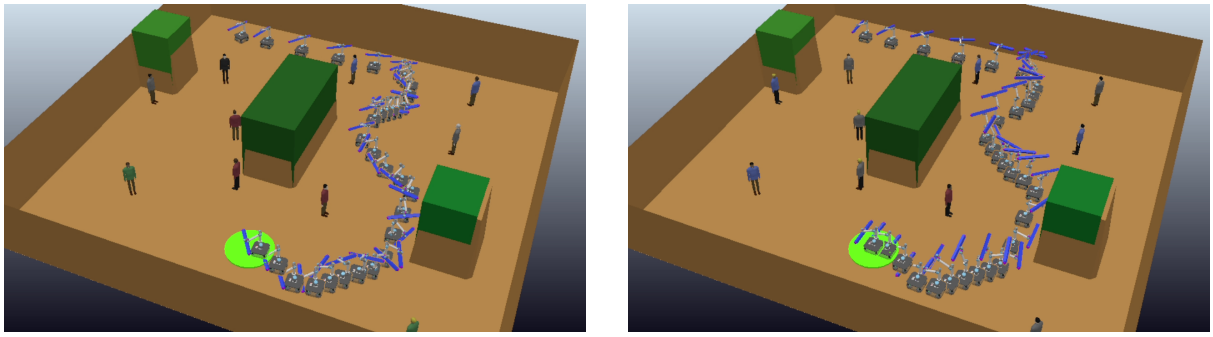
Further improvements are expected by selecting alternative joints to move, which will be explored in following sections, or by increasing the number of samples. Additionally, assigning higher weights to the object could potentially yield even greater reductions in discomfort compared to the baseline.

7.4.3 Z-axis Effect

We evaluate the impact of variable height in our solution, analyzing what happens when the manipulator is allowed to move along the Z-axis. First, we activate only joint 1 to allow motion only in the XY plane, keeping the Z position fixed. Then, we introduce an additional joint 3 to enable vertical displacement and compare the results. Figure 7.16 show the best solutions for each situation.

The first configuration with only Joint 1 is active, allows the robot to rotate the arm around the Z-axis, thus varying only the X and Y position. This configuration resulted in a minimum discomfort value of 11.733, with an average of 11.989 and a standard deviation of 0.332.

To introduce vertical motion, we keep Joint 1 active and additionally activate Joint 3, which enables changes in the height of the transported object while maintaining the other points of interest fixed. This configuration achieved a minimum discomfort value of 9.685, an average of 10.998, and a standard deviation of 0.855.



(a) Joint 1 active. (b) Joints 1 and 3 active.

Figure 7.16: Incremental social navigation plots comparing z-axis effect on solution. (a) Only joint 1 active, restricted to the plane and (b) Joints 1 and 3 active, can vary height.

Analyzing the results, we observe a slight improvement when height variation is allowed. Hence, it appears small variations happens when allowing for a z-axis movement. However, note that the experiments start with the object positioned relatively low to the ground—an already comfortable height configuration. Therefore, even without vertical motion, the robot begins in a favorable position, resulting in limited potential improvement from height adjustments. If the initial object height were in a less comfortable region, a greater improvement from vertical movement would likely be observed. It is also important to notice that joint 2 also introduce variation on the plane, therefore, the difference in the discomfort level can also be explained by the difference on the x-axis and y-axis.

To further investigate this effect, we compare the result of the base case where the

manipulator remains fixed in a lower position with a case in which the manipulator is fixed in an upright configuration, both performing the A* path. The base case yielded a total discomfort of 15.358 as seen in the previous section, whereas the upright case resulted in 18.919. These results indicate that height variation indeed influences overall discomfort, although the XY-plane effects dominate.

We also tested whether varying people's height significantly affects the total motion cost. In our experiments, we first considered all individuals with a height of 2 meters, and generated the corresponding motion. To test the impact, we set everyone with a height of 1.5 meters and apply the same motions found for individuals of 2.0 meters. After testing every solution, the average difference in discomfort between short and tall individuals was 0.696, meaning that shorter people would feel slightly more discomfort when following a path optimized for taller individuals. However, an average variation of less than 0.7, with the highest difference being 0.984, is not highly expressive in the context of the overall path.

To further analyze this effect, we re-do the sampling process, now setting everyone with 1.5 meters. The solutions average discomfort was of 11.081, however, when the robot used the motion found for tall individuals in short individuals, the mean discomfort was 10.932, both results are close enough to be deemed the same. Therefore, we acknowledge that, given our metric and planner, differences in human height have a shallow impact on the overall results. Consequently, we argue that a generalized motion planned assuming a uniform height should perform adequately for individuals within the range tested and with the metrics employed.

7.4.4 Joints Comparison

In this section, we observe the impact of number of joints being activated and which joints yield best results. Experiment 0 corresponds to the case used in the previous base comparison with joints 1,3 and 6 activated. In Experiments 1 through 6, one additional joint is activated sequentially in each trial from 1 joint to all 6 joints. In Experiment 7, only the joints that produced the most significant improvements are activated. The results are summarized in Table 7.1.

Table 7.1: Active joints experiments results. SD = standard deviation, CI = confidence interval (t-Student).

Experiment	Active Joints	Lowest Cost	Average Cost	SD	95% CI
0	1,3,6	8.542	10.236	0.944	0.675
1	1	11.733	11.989	0.332	0.237
2	1,3	9.685	10.998	0.855	0.612
3	1,2,3	6.994	8.080	0.598	0.428
4	1,...,4	6.722	8.330	0.683	0.489
5	1,...,5	7.116	8.040	0.780	0.558
6	1,...,6	5.720	7.331	1.028	0.735
7	1,2,3,6	6.275	7.527	0.873	0.624

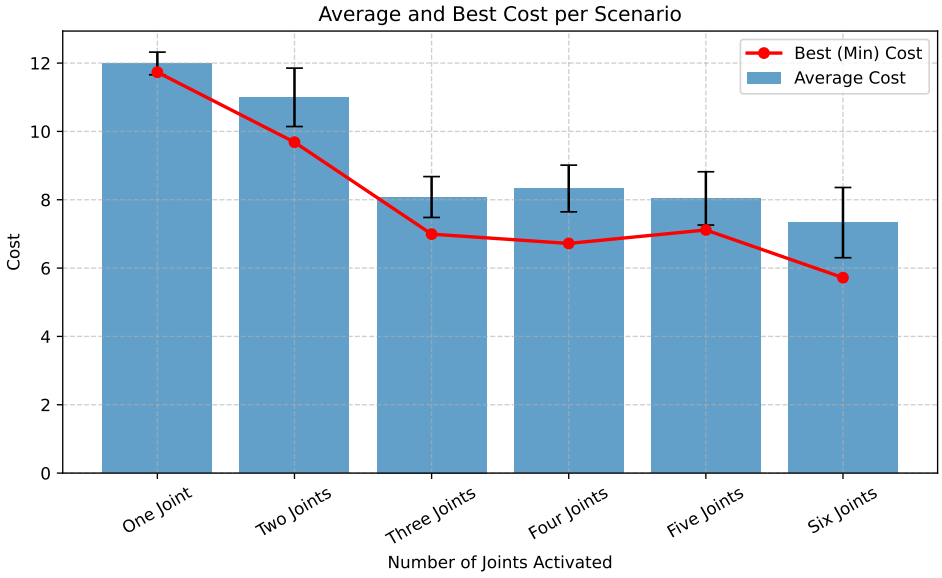


Figure 7.17: Graph comparing the results of the sequential experiments (1-6) varying number of joints activated, using the decoupled approach,

Let us discuss the results in parts, beginning with the sequential experiments (1-6). In the first test, Joint 1 controls the rotation of the entire arm around the z-axis. In the second test, Joints 1 and 3 are activated, introducing height and distance variation as discussed in the previous sections, improving the result. A even more significant improvement occurs when Joint 2 is also activated in the third test. Although Joints 2 and 3 rotate around the same axis, having both active allows the robot to position the object farther away from nearby people.

When additional joints are activated beyond these three, the results 4 and 5 exhibit slight changes. This may seem counterintuitive, since adding more joints should, in principle, provide the robot with greater maneuverability. In theory, this would never worsen performance, as the robot could always replicate previous motions or discover improved configurations. However, in practice, this advantage depends on the number of

samples available for the motion planner. Given the limit of 100 samples per node and 10 executions per test, the planner was unable to find much better solutions than those achieved with three joints. Reducing the number of active joints simplifies the planning problem, enabling the planner to find comparable solutions with less samples. Another important observation is that the joints 4 and 5 have an effect over short links, meaning that likely they will not provide meaningful increase in the distance between the arm and people.

Activating the last joint (joint 6), thus employing the full manipulator, had a greater improvement compared to the previous two experiments. This joint directly rotates the object being transported, which may explain the observed performance increase. The results of the sequential activation experiments are presented in Figure 7.17.

When comparing these outcomes with Experiment 0 where joints 1, 3, and 6 were active we observe that this configuration produced a worse result than activating joints 1, 2, and 3, although it still outperformed the case with only joints 1 and 3. This reinforces the previous observation that joints controlling longer links tend to have a greater influence on the overall performance, as expected.

Finally, we conducted an additional experiment activating joints 1, 2, 3, and 6, while excluding joints 4 and 5, since the former set of joints demonstrated higher individual impact. This configuration achieved the best performance among all setups except when using the entire manipulator, and its results remained comparable to the full configuration considering the standard deviation and average cost observed. The graph with all results is shown in Figure 7.18, in order of higher average to lowest.

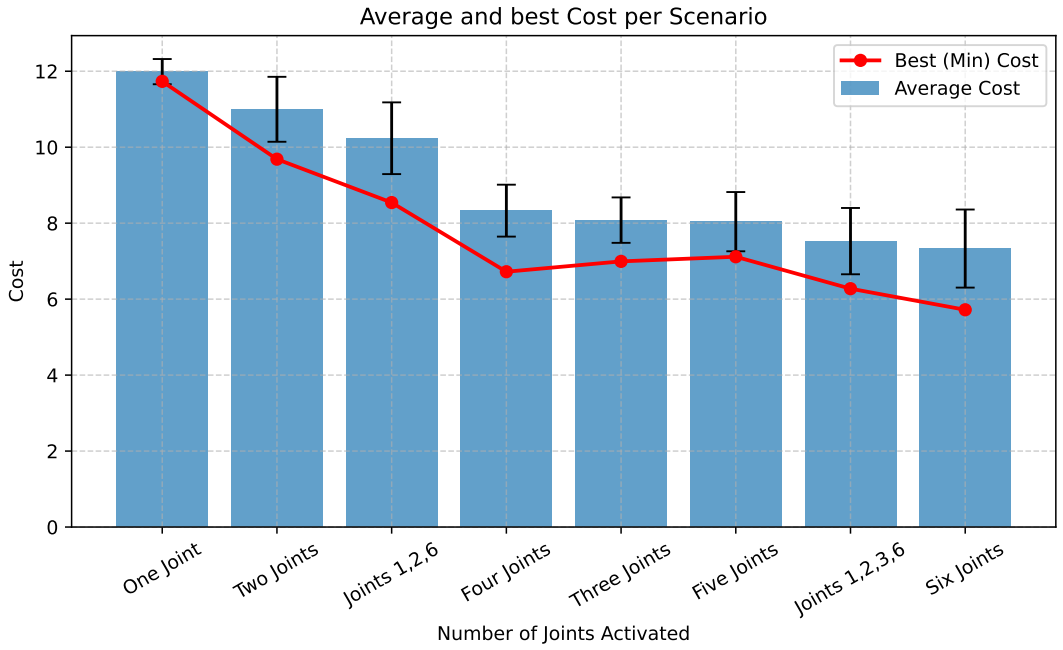
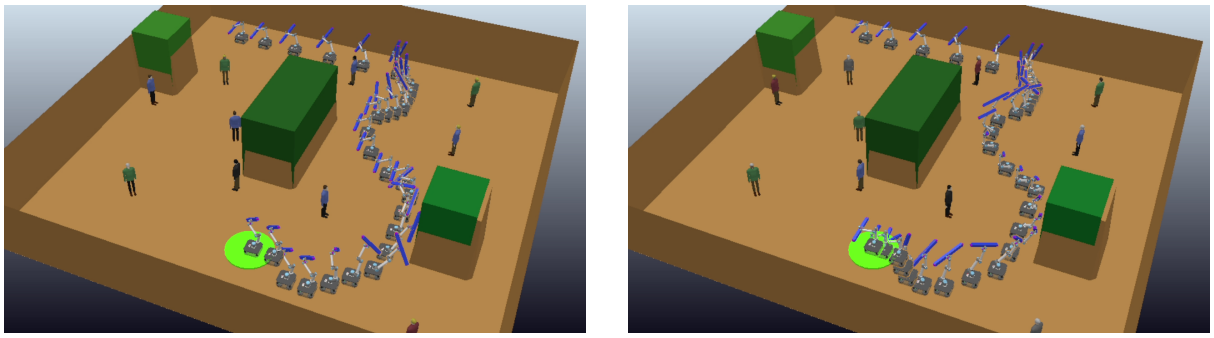


Figure 7.18: Graph with results of the decoupled approach varying joints of all 7 experiments, ordering by average cost.

Overall, we observed that selecting the appropriate three joints provides a balance between minimizing the social cost and limiting the number of active joints. Beyond four joints, further improvements can occur but tend to be less impactful. This outcome can be attributed to the fact that, using the UR10 robotic arm, after the third joint, the subsequent joints primarily control shorter links, which have a smaller influence on the robot's overall configuration and consequently on the discomfort cost. Furthermore, using the best configuration obtained (with a total discomfort cost of 5.720), we achieved a 62.75% reduction in discomfort compared to the base-only scenario, which presented a cost of 15.358.



(a) Joints 1,2,3 and 6 active. (b) All joints active (1-6).

Figure 7.19: Best overall results for the full 3D scenario.

7.5 Dangerous Object

We analyze the effects of considering that the manipulator is carrying a dangerous object. To model this condition, the weights along the object are adjusted as follows: the edge of the object is assigned a weight of 5, the midpoint a weight of 1, and the remaining points on the manipulator and base a weight of 0.1. All manipulator joints are active in this experiment. The result is shown in Figure 7.20.

Analyzing the results, we observe that the object indeed tends to move farther away from the nearest person when higher weights are assigned to it. However, this behavior is not substantially different from the best results obtained previously with uniform weights. Still, it is noticeable that the object is more consistently projected away from people, whereas this effect was less evident when all weights were set to one.

We also highlight an observed pattern towards the object, it has a tendency to be up-straight, aligned with the vertical/z-axis. We hypothesize that this behavior emerges because such an orientation allows the robot to manipulate and rotate the object more

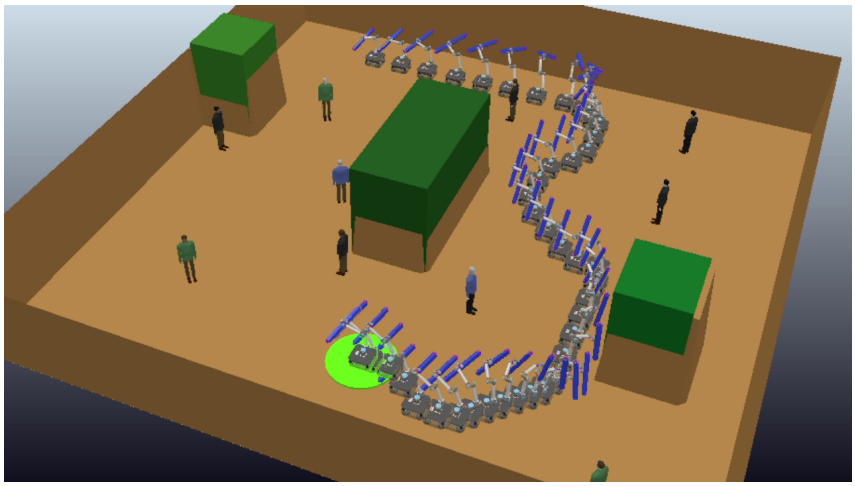


Figure 7.20: Incremental social navigation plot showing the best result for the scenario in which the object is considered dangerous, with higher weights assigned to the object.

easily without colliding. Consequently, the sampler tends to find these configurations. Additionally, keeping the object vertical restricts its edges within the XY plane, which makes the most difference in the discomfort function. With the object vertical, both edges are optimized equally. From a qualitative perspective, this behavior is also intuitively human-like: when carrying a bar-shaped object, is reasonable to assume that people would hold it vertically to navigate narrow spaces to avoid collisions. Moreover, when the bar is upright, the edges will not with a person, reducing potential harm.

When comparing to the coupled approach, a key difference emerges. In the coupled method, the robot's base moves closer to the person, allowing the object to be positioned farther away through coordinated motion. In contrast, the decoupled approach constrains the base to a fixed path, limiting how much the object can be adjusted. Consequently, the overall trajectory in the decoupled case shows least variation compared to the coupled approach, that exhibits more pronounced path differences due to the joint coordination between the base and the manipulator.

7.6 Time Analysis

When testing the coupled Risk-RRT* approach in the 2D generic scenario with the bar object, we executed 2000 iterations and recorded six trials, yielding an average runtime of approximately 52.80 minutes. Notably, the algorithm was able to find an initial feasible solution relatively quickly, in an average of 2.28 minutes. However, this initial solution was not decent, as the overall discomfort decreased by 68.21% from the first feasible solution to the best one obtained after 2000 iterations.

Across 10 executions, the decoupled approach was evaluated using a maximum of 100 samples between each pair of nodes, with the A* path consisting of 13 nodes and all manipulator joints active. The average computation time obtained was 7.96 minutes. When analyzing the planning time for different numbers of joints, no correlation between the number of joints and the total computation time was observed.

The generic map used in the decoupled approach measures 20×20 meters. With a resolution of 1 meter, this results in 400 nodes. Constructing the graph and computing the edges, including collision checking, required approximately 13.3 minutes. Since the environment contains both obstacles and people, not all possible edges were tested as edge generation was interrupted whenever a collision was detected. Once the graph is built, however, the A* algorithm can be executed from any node to any other node to compute a path. For this graph size, the computation time is on the order of milliseconds and therefore negligible.

Chapter 8

Conclusion and Future Work

8.1 Limitations

The primary limitation of the coupled implementation lies in the scalability of the Risk-RRT* planner. As the number of degrees of freedom increases, the configuration space grows, making it computationally expensive to find feasible and socially acceptable paths. Consequently, the method may become impractical for manipulators with a large number of links.

The decoupled approach, while plans faster, presents a different set of challenges. Since the base and manipulator are planned independently, the method cannot fully coordinate their motions. The arm trajectory is influenced by the base path, yet the base path remains fixed once computed. Moreover, although faster than a full-body planner, the decoupled strategy still relies on a sampling-based approach for the manipulator, which can still struggle with time constraints.

Also, we propose an offline planning framework without real-time replanning capabilities. This design choice simplifies computation but comes with a clear trade-off: the robot is unable to adapt to changes in the environment, even minor ones that may occur during execution. Consequently, the planned motion remains valid only as long as the environment remains static. Additionally, the method assumes prior knowledge of the environment map, which may not be feasible in real-world applications where the robot operates in partially known environments.

It is also important to note the limitations of the three dimensional discomfort model proposed. While our model accounts for the height of the person, it assumes a generalized discomfort function. In reality, perceptions of comfort vary across individuals due to personal and cultural differences, our model, is a generalization. Furthermore, the current formulation considers only spatial proximity when estimating discomfort. In real human-robot interactions, other factors such as the robot's velocity, motion smoothness, and even sound, influence perceived comfort.

8.2 Conclusion

As robots are increasingly integrated into human-populated environments, the field of social robotics has gained significant importance. The development of socially acceptable behaviors in robots enhances their acceptance, reduces potential risks, and enables more natural and effective collaboration between humans and robots in shared spaces.

In this dissertation, we tackle the problem of socially-aware object transportation using a mobile manipulator. By incorporating a social cost function with a multi-interest-point evaluation into the planning process, we ensure that the robot not only respects personal spaces but also accounts for the positions of the mobile base, the manipulator arm, and the transported object. This approach allows the mobile manipulator to carry out tasks efficiently and safely, while adhering to social norms, ensuring a higher degree of acceptance in human-populated environments.

We developed two approaches to address two different scenarios. The first scenario involves a bidimensional environment with a mobile manipulator composed of a holonomic base and a 2R planar arm. For this setup, we adapted a Risk-RRT* algorithm as a social whole-body motion planner. The results show that the robot's path indeed achieves more socially appropriate behavior compared to base-only approaches. However, the sampling method is computationally intensive. Therefore, it is better suited for simplified scenarios or systems with high computational power.

The second scenario considers a three-dimensional environment with a mobile manipulator featuring a holonomic base and the UR10 robotic arm, with over 6 revolute joints. Given the increased complexity, we adopted a decoupled approach: first, we plan the base trajectory using an A* algorithm that minimizes social discomfort; then, we apply a social sampling method between the A* waypoints to control the arm. This approach provides a good social response while significantly reducing computation time, making it more suitable for complex scenarios where whole-body sampling-based struggle with time constraints.

In our results, we observed that the decoupled method outperforms the coupled method, producing motions that are more comfortable according to our metric. Hence, the decoupled method overall is preferred. However, there are situations in which the A* planner may fail to find a path to the goal, particularly when coordinated motion between the base and the manipulator is required to navigate through narrow passages. In such cases, a coupled approach can provide a feasible solution, demonstrating its advantage in tightly constrained environments. Another advantage observed of the coupled method is related to varying the social weights, such as when transporting dangerous objects. In these cases, it effectively adjusted its motion by keeping the object farther from people while moving the base closer when necessary. In contrast, the decoupled method, because

its first step maintains the arm fixed, showed limited adaptability.

Furthermore, this work introduced a novel 3D discomfort model that extends traditional formulations by incorporating vertical discomfort, enabling a more comprehensive representation of human personal space. We applied this model in the generic scenario and observed that the robot adjusts the height of its manipulator to minimize social discomfort, demonstrating the necessity of a three-dimensional discomfort representation. Our experiments also indicate that while variations in height do affect discomfort, the impact is relatively modest compared to when the robot's manipulator already operates at favorable, lower heights.

We also investigated whether activating additional joints, or selecting different combinations of joints, influences the overall outcome. We found that, in general, activating a joint tends to improve performance, although this is not always the case. The effect is more pronounced when the joint controls a longer link, as it has a greater potential impact on the robot's configuration; conversely, activating shorter or already well-positioned joints may have little to no effect. Our results also indicate that it is possible to achieve satisfactory performance by selectively activating only a subset of joints.

Overall, we conclude that having the manipulator in motion is preferable to keeping it static, which justifies the development of techniques specifically designed for mobile manipulators in social robotics. Our method outperforms socially-aware base-only approaches and offers the flexibility to prioritize certain parts of the robot to be more cautious. Although we do not deal with dynamic environments, and humans being static is an unrealistic assumption, we believe that our work is a start to research for social navigation targeted to mobile manipulators.

8.3 Future Work

In future work, explore dynamic scenarios involving moving people is the next step, enhancing the robot's ability to adapt to changing environments. This can be done using reactive or hybrid approaches that integrate our proposed multi-point sampling strategy. Preliminary experiments have explored reinforcement learning techniques for reactive, socially-aware navigation of mobile robots, yielding promising initial results. Additionally, real-time adaptations of RRT-based methods have been investigated, with mixed results. These approaches showed potential primarily in simpler scenarios, nevertheless, further validation is necessary to determine whether such methods could achieve real-time avoidance in practice, if the computer that computes the motion is able to do it in time. The main time bottleneck is the collision checking, more efficient methods can be explored

to highly improve the time taken for planning.

Regarding the three-dimensional discomfort function, our current approach is based solely on theoretical assumptions, highlighting the need for real-world validation. We have initiated preliminary user studies to assess the alignment between predicted and actual discomfort levels and to refine the model parameters. Early results from these experiments show consistency with the theoretical predictions of our proposed model.

8.3.1 Real Deployment

For practitioners, the proposed system can be deployed in real-world scenarios by integrating perception, mapping, and control components within the robot's software architecture.

To begin with, the map of the environment is required. This can be obtained by mapping the robot's surroundings using 3D sensors such as LiDAR or depth cameras and representing the resulting data as an octomap. This representation enables collision checking and path planning in three dimensions. Additionally, a simulated digital twin of the robot with the object being carried must be available to replicate its kinematic and dynamic behavior, ensuring that the computed paths are consistent with the physical system's constraints. Human presence and height estimation are also essential to compute the social discomfort model. Human positions and heights can be defined manually for controlled experiments or estimated automatically using cameras or motion capture systems such as the Kinect. Once this information is acquired, it can be provided to one of the two proposed motion planners to compute the socially aware motion. The resulting motion can then be executed by the real robot using the same control framework employed in simulation, possibly with fine-tuning of the controller gains.

Finally, for real-world operation, it is advisable to include a real-time obstacle avoidance module to handle unforeseen dynamic obstacles and ensure safe human–robot coexistence during navigation.

References

[1] Hojjat Abdollahi, Mohammad H. Mahoor, Rohola Zandie, Jarid Siewierski, and Sara H. Qualls. Artificial emotional intelligence in socially assistive robots for older adults: A pilot study. *IEEE Transactions on Affective Computing*, 14(3):2020–2032, 2023.

[2] Alexandre Alahi, Kratarth Goel, Vignesh Ramanathan, Alexandre Robicquet, Li Fei-Fei, and Silvio Savarese. Social lstm: Human trajectory prediction in crowded spaces. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 961–971, 2016.

[3] Tianjiao An, Yuexi Wang, Guangjun Liu, Yuanchun Li, and Bo Dong. Cooperative game-based approximate optimal control of modular robot manipulators for human–robot collaboration. *IEEE Transactions on Cybernetics*, 53(7):4691–4703, 2023.

[4] Erangi P Ariyasena, Udaka A Manawadu, Kasun R Abeyratne, and P Ravindra S De Silva. Exploring a ground-air personal space in human-drone interactions. In *2020 International Conference on Image Processing and Robotics (ICIP)*, pages 1–6, 2020.

[5] Saray Bakker, Andreu Matoses Gimenez, Clarence Chen, Javier Alonso-Mora, and Wendelin Bohmer. INTERACT: Intuitive Interaction for Robots among Humans. https://autonomousrobots.nl/projects/interact, 2024. [Online; accessed 01-August-2024].

[6] Marie-Lou Barnaud, Nicolas Morgado, Richard Palluel-Germain, Julien Diard, and Anne Spalanzani. Proxemics models for human-aware navigation in robotics: Grounding interaction and personal space models in experimental data from psychology. In *Proceedings of the 3rd IROS'2014 workshop "Assistance and Service Robotics in a Human Environment"*, Chicago, United States, September 2014.

[7] Andrea Bauer, Dirk Wollherr, and Martin Buss. Human-robot collaboration: a survey. *I. J. Humanoid Robotics*, 5:47–66, 03 2008.

[8] Lars Berscheid and Torsten Kröger. Jerk-limited real-time trajectory generation with arbitrary target states. *Robotics: Science and Systems XVII*, 2021.

[9] Alisha Bevins, Siya Kunde, and Brittany A. Duncan. User-designed human-uav interaction in a social indoor environment. In *19th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, 2024.

[10] Cynthia Breazeal, Kerstin Dautenhahn, and Takayuki Kanda. *Social Robotics*, pages 1935–1972. 06 2016.

[11] Robin Bretin, Emily Cross, and Mohamed Khamis. Co-existing with drones: A virtual exploration of proxemic behaviours and users' insights on social drones. *International Journal of Social Robotics*, 16(3):547–567, Mar 2024.

[12] Rohan Chandra, Rahul Maligi, Arya Anantula, and Joydeep Biswas. Socialmapf: Optimal and efficient multi-agent path finding with strategic agents for social navigation. *IEEE Robotics and Automation Letters*, 8(6):3214–3221, 2023.

[13] Balasubramaniyan Chandrasekaran and James M. Conrad. Human-robot collaboration: A survey. In *SoutheastCon 2015*, pages 1–8, 2015.

[14] Wenzheng Chi, Hitoshi Kono, Yusuke Tamura, Atsushi Yamashita, Hajime Asama, and Max Q.-H. Meng. A human-friendly robot navigation algorithm using the risk-rrt approach. In *2016 IEEE International Conference on Real-time Computing and Robotics (RCAR)*, pages 227–232, 2016.

[15] Wenzheng Chi and Max Q.-H. Meng. Risk-rrt*: A robot motion planning algorithm for the human robot coexisting environment. In *2017 18th International Conference on Advanced Robotics (ICAR)*, pages 583–588, 2017.

[16] Wenzheng Chi, Jiankun Wang, and Max Qing-Hu Meng. Risk-informed-rrt*: A sampling-based human-friendly motion planning algorithm for mobile service robots in indoor environments. In *2018 IEEE International Conference on Information and Automation (ICIA)*, pages 1101–1106, 2018.

[17] Sara Cooper, Raquel Ros, and Séverin Lemaignan. Challenges of deploying assistive robots in real-life scenarios: an industrial perspective. pages 435–442, 08 2023.

[18] Marco Cristani, Alessio Del Bue, Vittorio Murino, Francesco Setti, and Alessandro Vinciarelli. The visual social distancing problem. *IEEE Access*, PP:1–1, 07 2020.

[19] Frédéric Dehais, Emrah Akin Sisbot, Rachid Alami, and Mickaël Causse. Physiological and subjective evaluation of a human–robot object hand-over task. *Applied Ergonomics*, 42(6):785–791, 2011.

[20] TU Delft. Designing behaviour for mobile robots. https://www.tudelft.nl/en/ide/research/mobility/designing-behaviour-for-mobile-robots, 2020. [Online; accessed 16-June-2024].

[21] Guanglong Du, Shuaiying Long, Fang Li, and Xin Huang. Active collision avoidance for human-robot interaction with ukf, expert system, and artificial potential field method. *Frontiers in Robotics and AI*, 5, 2018.

[22] Brittany A. Duncan and Robin R. Murphy. Comfortable approach distance with small unmanned aerial vehicles. In *IEEE International Conference on Robot and Human Interactive Communication*, pages 786–792, 2013.

[23] Boston Dynamics. Spot's Got an Arm! https://www.youtube.com/watch?v=6Zbhvaac68Y, 2021. [Online; accessed 31-July-2024].

[24] Kevin Fan, Melanie Jouaiti, Ali Noormohammadi-As, Chrystopher L. Nehaniv, and Kerstin Dautenhahn. A social referencing disambiguation framework for domestic service robots. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 11979–11985, 2023.

[25] Michalis Feidakis, Iro Gkolompia, Athina Marnelaki, Konstantina Marathaki, Sofia Emmanouilidou, and Eleni Agrianiti. Nao robot, an educational assistant in training, educational and therapeutic sessions. In *2023 IEEE Global Engineering Education Conference (EDUCON)*, pages 1–6, 2023.

[26] Jonathan D. Gammell, Siddhartha S. Srinivasa, and Timothy D. Barfoot. Informed rrt*: Optimal sampling-based path planning focused via direct sampling of an admissible ellipsoidal heuristic. In *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2997–3004, 2014.

[27] A. Garrell, Luis Garza-Elizondo, M. Villamizar, F. Herrero, and A. Sanfeliu. Aerial social force model: A new framework to accompany people using autonomous flying robots. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 7011–7017, 2017.

[28] Sami Haddadin. *Towards Safe Robots - Approaching Asimov's 1st Law*, volume 1, chapter 04 - Biomechanics and Forensics, page 69. Springer Berlin, Heidelberg, September 2013.

[29] Edward Twitchell Hall. *The Hidden Dimension: Man's Use of Space in Public and Private.* The Bodley Head Ltd, 01 1966.

[30] JeongHye Han, Dylan Moore, and Ilhan Bae. Exploring the social proxemics of human-drone interaction. *The International Journal of Advanced Smart Convergence*, 8(2):1–7, 2019.

[31] Sangchul Han, Sanguk Chon, JungYeong Kim, Jeahong Seo, Dong Gwan Shin, Sangshin Park, Jin Tak Kim, Jinhyeon Kim, Maolin Jin, and Jungsan Cho. Snake

robot gripper module for search and rescue in narrow spaces. *IEEE Robotics and Automation Letters*, 7(2):1667–1673, 2022.

[32] Adam Heins and Angela P. Schoellig. Keep it upright: Model predictive control for nonprehensile object transportation with obstacle avoidance on a mobile manipulator. *IEEE Robotics and Automation Letters*, 8(12):7986–7993, 2023.

[33] Anna Henschel, Guy Laban, and Emily S Cross. What makes a robot social? a review of social robots from science fiction to a home or hospital near you. *Curr. Robot. Rep.*, 2(1):9–19, February 2021.

[34] Noriaki Hirose, Dhruv Shah, Ajay Sridhar, and Sergey Levine. Sacson: Scalable autonomous control for social navigation. *IEEE Robotics and Automation Letters*, 9(1):49–56, 2024.

[35] Van Bay Hoang, Van Hung Nguyen, Trung Dung Ngo, and Xuan-Tung Truong. Socially aware robot navigation framework: Where and how to approach people in dynamic social environments. *IEEE Transactions on Automation Science and Engineering*, 20(2):1322–1336, 2023.

[36] Armin Hornung, Mike Phillips, E. Gil Jones, Maren Bennewitz, Maxim Likhachev, and Sachin Chitta. Navigation in three-dimensional cluttered environments for mobile manipulation. In *2012 IEEE International Conference on Robotics and Automation*, pages 423–429, 2012.

[37] ISO/TS 15066:2016. Robots and robotic devices — Collaborative robots. Technical specification, International Organization for Standardization, Geneva, Switzerland, February 2016.

[38] Jyh-Shing Roger Jang, Chuen-Tsai Sun, and Eiji Mizutani. *Neuro-Fuzzy and Soft Computing - A Computational Approach to Learning and Machine Intelligence*, chapter 01 - Introduction to Neuro-Fuzzy and Soft Computing, page 6. Prentice-Hall, Inc., 1997.

[39] Maria Kabtoul, Manon Prédhumeau, Anne Spalanzani, Julie Dugdale, and Philippe Martinet. How to evaluate the navigation of autonomous vehicles around pedestrians? *IEEE Transactions on Intelligent Transportation Systems*, 25(3):2311–2321, 2024.

[40] Maryam Kameli, Thomas Zurales, and Arshia Khan. Improved navigation for social robots through process offloading. In *2022 International Conference on Computational Science and Computational Intelligence (CSCI)*, pages 1743–1748, 2022.

[41] Kapil D. Katyal, Gregory D. Hager, and Chien-Ming Huang. Intent-aware pedestrian prediction for adaptive crowd navigation. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3277–3283, 2020.

[42] Pouria Khajehpour and Esmaeil Najafi. Design and manual control of a 3 degrees of freedom social robotic manipulator. Institute of Electrical and Electronics Engineers Inc., 12 2020.

[43] Wisama Khalil. Modeling and Control of Manipulators - Part I: Geometric and Kinematic Models. Lecture, January 2019.

[44] Mincheul Kim, Youngsun Kwon, Sebin Lee, and Sung eui Yoon. Cctv-informed human-aware robot navigation in crowded indoor environments. *IEEE Robotics and Automation Letters*, 6 2024.

[45] Julien Kindle, Fadri Furrer, Tonci Novkovic, Jen Jen Chung, Roland Siegwart, and Juan Nieto. Whole-body control of a mobile manipulator using end-to-end reinforcement learning, 2020.

[46] Rachel Kirby. *Social Robot Navigation*. Phd dissertation, Carnegie Mellon University, Pittsburgh, PA, May 2010. Available at https://www.ri.cmu.edu/pub_files/2010/5/rk_thesis.pdf.

[47] Hasan Kivrak, Furkan Cakmak, Hatice Kose, and Sirma Yavuz. Social navigation framework for assistive robots in human inhabited unknown environments. *Engineering Science and Technology, an International Journal*, 24(2):284–298, 2021.

[48] Siya Kunde, Nathan Simms, Gerson Uriarte, and Brittany Duncan. Let's run an online proxemics study! but, how do results compare to in-person? In *Social Robotics: 14th International Conference, ICSR 2022, Florence, Italy, December 13–16, 2022, Proceedings, Part I*, page 24–37, Berlin, Heidelberg, 2022. Springer-Verlag.

[49] Lorenzo Landolfi, Dario Pasquali, Alice Nardelli, Jasmin Bernotat, and Francesco Rea. Working memory-based architecture for human-aware navigation in industrial settings. In *2023 32nd IEEE International Conference on Robot and Human Interactive Communication (RO-MAN)*, pages 1878–1885, 2023.

[50] Przemyslaw A. Lasota, Terrence Song, and Julie A. Shah. *A Survey of Methods for Safe Human-Robot Interaction*. 2017.

[51] Linda Lastrico, Nuno Ferreira Duarte, Alessandro Carfí, Francesco Rea, Alessandra Sciutti, Fulvio Mastrogiovanni, and José Santos-Victor. Expressing and inferring

action carefulness in human-to-robot handovers. In *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, October 2023.

[52] Wansong Liu, Kareem Eltouny, Sibo Tian, Xiao Liang, and Minghui Zheng. Integrating uncertainty-aware human motion prediction into graph-based manipulator motion planning, 2024.

[53] Kevin M. Lynch and Frank C. Park. In *Modern Robotics: Mechanics, Planning, and Control*, chapter 1, page 10. Cambridge University Press, USA, 1st edition, 2017.

[54] Hamza Mahdi, Sami Alperen Akgun, Shahed Saleh, and Kerstin Dautenhahn. A survey on the design and evolution of social robots — past, present and future. *Robotics and Autonomous Systems*, 156, 10 2022.

[55] Jim Mainprice, E. Akin Sisbot, Léonard Jaillet, Juan Cortés, Rachid Alami, and Thierry Siméon. Planning human-aware motions using a sampling-based costmap planner. In *2011 IEEE International Conference on Robotics and Automation*, pages 5012–5017, 2011.

[56] Christoforos Mavrogiannis, Francesca Baldini, Allan Wang, Dapeng Zhao, Pete Trautman, Aaron Steinfeld, and Jean Oh. Core challenges of social robot navigation: A survey. 12(3), apr 2023.

[57] Tomas Merva, Saray Bakker, Max Spahn, Ivan Virgala, and Javier Alonso-Mora. Reactive grasp and motion planning for adaptive mobile manipulation among obstacles. In *Robotics: Science and Systems (RSS), Workshop on Frontiers of Optimization for Robotics*, 2024. W14.

[58] Maria Vittoria Minniti, Farbod Farshidian, Ruben Grandia, and Marco Hutter. Whole-body MPC for a dynamically stable mobile manipulator. *CoRR*, abs/1902.10415, 2019.

[59] Seung Jae Moon, Jinsol Kim, Hongsik Yim, Yeeun Kim, and Hyouk Ryeol Choi. Real-time obstacle avoidance using dual-type proximity sensor for safe human-robot interaction. *IEEE Robotics and Automation Letters*, 6(4):8021–8028, 2021.

[60] Venkatraman Narayanan, Bala Murali Manoghar, Rama Prashanth RV, and Aniket Bera. Ewarenet: Emotion-aware pedestrian intent prediction and adaptive spatial profile fusion for social robot navigation. pages 7569–7575. IEEE, 5 2023.

[61] Hugo Nascimento, Martin Mujica, and Mourad Benoussaad. Collision avoidance in human-robot interaction using kinect vision system combined with robot's model and data. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 10293–10298, 2020.

[62] Megan Nichols. Robots are Transforming the Healthcare Industry. https://community.robotshop.com/blog/show/robots-are-transforming-the-healthcare-industry?fbclid=IwAR0oxON1UTYcBZm366KtMNcR4P6HFb5o9Acnrrmyt6P-tBk_Lu2iPuh6DF4, 2019. [Online; accessed 31-July-2024].

[63] Jeongwoo Oh, Jaeseok Heo, Junseo Lee, Gunmin Lee, Minjae Kang, Jeongho Park, and Songhwai Oh. Scan: Socially-aware navigation using monte carlo tree search. pages 7576–7582. IEEE, 5 2023.

[64] Pakpoom Patompak, Sungmoon Jeong, Nak Young Chong, and Itthisek Nilkhamhang. Mobile robot navigation for human-robot social interaction. *International Conference on Control, Automation and Systems*, 0:1298–1303, 1 2016.

[65] Pakpoom Patompak, Sungmoon Jeong, Itthisek Nilkhamhang, and Nak Chong. Learning proxemics for personalized human–robot social interaction. *International Journal of Social Robotics*, 12, 01 2020.

[66] Sriyash Poddar, Christoforos Mavrogiannis, and Siddhartha S. Srinivasa. From crowd motion prediction to robot navigation in crowds, 2023.

[67] Stefano Primatesta, Matteo Scanavino, Giorgio Guglieri, and Alessandro Rizzo. A risk-based path planning strategy to compute optimum risk path for unmanned aircraft systems over populated areas. In *2020 International Conference on Unmanned Aircraft Systems (ICUAS)*, pages 641–650, 2020.

[68] Andrea Pupa, Marco Minelli, and Cristian Secchi. A dynamic planner for safe and predictable human-robot collaboration. *IEEE Robotics and Automation Letters*, 9(1):507–514, 2024.

[69] Photchara Ratsamee, Yasushi Mae, Kenichi Ohara, Tomohito Takubo, and Takero Arai. Human-robot collision avoidance using a modified social force model with body pose and face orientation. *International Journal of Humanoid Robotics*, 10, 04 2013.

[70] Caio C. G. Ribeiro, Leonardo H. M. C. dos Santos, and Douglas G. Macharet. Collaborative ugv/uav path planning for inventory management in warehouses. In *2022 Latin American Robotics Symposium (LARS), 2022 Brazilian Symposium on Robotics (SBR), and 2022 Workshop on Robotics in Education (WRE)*, pages 121–126, 2022.

[71] J. Rios-Martinez, A. Spalanzani, and C. Laugier. Understanding human interaction for probabilistic autonomous navigation using risk-rrt approach. In *IEEE/RSJ In-*

ternational Conference on Intelligent Robots and Systems (IROS), pages 2014–2019, Sept 2011.

[72] Clearpath Robotics. Gallery, 2024.

[73] Matteo Rubagotti, Inara Tusseyeva, Sara Baltabayeva, Danna Summers, and Anara Sandygulova. Perceived safety in physical human–robot interaction—a survey. *Robotics and Autonomous Systems*, 151:104047, 2022.

[74] Andrey Rudenko, Luigi Palmieri, Michael Herman, Kris M. Kitani, Dariu M. Gavrila, and Kai Oliver Arras. Human motion trajectory prediction: A survey. *CoRR*, abs/1905.06113, 2019.

[75] Thushara Sandakalum and Marcelo H. Ang. Motion planning for mobile manipulators—a systematic review. *Machines*, 10(2), 2022.

[76] Ruth Schulz, Philipp Kratzer, and Marc Toussaint. Building a bridge with a robot: A system for collaborative on-table task execution. *HAI 2017 - Proceedings of the 5th International Conference on Human Agent Interaction*, pages 399–403, 10 2017.

[77] Jun Shao, Hao Xiong, Jianfeng Liao, Wei Song, Zheng Chen, Jason Gu, and Shiqiang Zhu. Rrt-goalbias and path smoothing based motion planning of mobile manipulators with obstacle avoidance. In *IEEE International Conference on Real-time Computing and Robotics*, pages 217–222, 2021.

[78] Thomas B Sheridan. A review of recent research in social robotics. *Current Opinion in Psychology*, 36:7–12, 2020. Cyberpsychology.

[79] Asahi Shimbun. Softbank's famous robot Pepper is helping helping enforce social distancing and greeting COVID-19 patients around the world. https://www.businessinsider.com/softbank-pepper-robot-coronavirus-japan-and-germany-2020-5, 2020. [Online; accessed 16-June-2024].

[80] Roland Siegwart, Illah R. Nourbakhsh, and Davide Scaramuzza. In *Introduction to Autonomous Mobile Robots*, chapter 1, pages 1–12. The MIT Press, 2ed edition, 2011.

[81] Alan D. G. Silva and Douglas G. Macharet. Are You With Me? Determining the Association of Individuals and the Collective Social Space. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 313–318, 2019.

[82] Aline F. F. Silva, Luciano E. Almeida, and Douglas G. Macharet. Efficiently approaching groups of people in a socially acceptable manner in environments with

obstacles. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 12038–12044, 2023.

[83] Aline F.F. Silva, Luciano E. Almeida, and Douglas G. MacHaret. Efficiently approaching groups of people in a socially acceptable manner in environments with obstacles. *Proceedings - IEEE International Conference on Robotics and Automation*, 2023-May:12038–12044, 2023.

[84] Steven Silva, Nervo Verdezoto, Dennys Paillacho, Samuel Millan-Norman, and Juan David Hernández. Online social robot navigation in indoor, large and crowded environments. pages 9749–9756. IEEE, 5 2023.

[85] Kiran Jot Singh, Divneet S. Kapoor, Khushal Thakur, Anshul Sharma, Anand Nayyar, Shubham Mahajan, and Mohd. Asif Shah. Map making in social indoor environment through robot navigation using active slam. *IEEE Access*, 10:134455–134465, 2022.

[86] Emrah Akin Sisbot and Rachid Alami. A human-aware manipulation planner. *IEEE Transactions on Robotics*, 28(5):1045–1057, 2012.

[87] Emrah Akin Sisbot, Luis F. Marin, and Rachid Alami. Spatial reasoning for human robot interaction. In *2007 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2281–2287, 2007.

[88] Emrah Akin Sisbot, Luis F. Marin-Urias, Rachid Alami, and Thierry Simeon. A human aware mobile robot motion planner. *IEEE Transactions on Robotics*, 23(5):874–883, 2007.

[89] Ruitao Song, Samuel Ong, Liuwang Kang, Shiyu Jin, Yuan-Chih Peng, Zhenpeng He, Lingfeng Qian, and Liangjun Zhang. Autonomous excavator system for construction automation. In *Robotics: Science and Systems 2023*, 2023.

[90] Max Spahn, Corrado Pezzato, Chadi Salmi, Rick Dekker, Cong Wang, Christian Pek, Jens Kober, Javier Alonso-Mora, Carlos Hernandez Corbato, and Martijn Wisse. Demonstrating adaptive mobile manipulation in retail environments. In *Robotics: Science and Systems 2024*, 2024.

[91] Shantanu Thakar, Srivatsan Srinivasan, Sarah Al-Hussaini, Prahar M. Bhatt, Pradeep Rajendran, Yeo Jung Yoon, Neel Dhanaraj, Rishi K. Malhan, Matthias Schmid, Venkat N. Krovi, and Satyandra K. Gupta. A Survey of Wheeled Mobile Manipulation: A Decision-Making Perspective. *Journal of Mechanisms and Robotics*, 15(2):020801, 07 2022.

[92] Xuan-Tung Truong and Trung Dung Ngo. Toward socially aware robot navigation in dynamic and crowded environments: A proactive social motion model. *IEEE Transactions on Automation Science and Engineering*, 14(4):1743–1760, 2017.

[93] Ubahnverleih. Nao Robot (Robocup 2016). https://commons.wikimedia.org/wiki/File:Nao_Robot_(Robocup_2016).jpg, 2016. [Online; accessed 18-June-2024].

[94] Vanillase. Asimo at a Honda factory. https://commons.wikimedia.org/wiki/File:ASIMO_4.28.11.jpg, 2011. [Online; accessed 18-June-2024].

[95] Dizan Vasquez, Procópio Stein, Jorge Rios-Martinez, Arturo Escobedo, Anne Spalanzani, and Christian Laugier. Human aware navigation for assistive robotics. In *Experimental Robotics: The 13th International Symposium on Experimental Robotics*, pages 449–462. Springer, 2013.

[96] Jacqueline C. Vischer. The effects of the physical environment on job performance: towards a theoretical model of workspace stress. *Stress and Health*, 23(3):175–184, 2007.

[97] Weitian Wang, Yi Chen, Rui Li, and Yunyi Jia. Learning and comfort in human–robot interaction: A review. *Applied Sciences*, 9(23), 2019.

[98] WebHamster. Diagram representation of personal space limits, according to Edward T. Hall's interpersonal distances of man, showing radius in feet and meters. Inspired by Reaction-bubble.png by Libb Thims. https://commons.wikimedia.org/wiki/File:Personal_Space.svg#filehistory, 2009. [Online; accessed 20-October-2025].

[99] David A. Winter. *Biomechanics and Motor Control of Human Movement*, volume 4, chapter 04 - Anthropometry, page 83. John Wiley & Sons, Inc., Hoboken, New Jersey, 4 edition, September 2009.

[100] Jimmy Wu, William Chong, Robert Holmberg, Aaditya Prasad, Yihuai Gao, Oussama Khatib, Shuran Song, Szymon Rusinkiewicz, and Jeannette Bohg. Tidybot++: An open-source holonomic mobile manipulator for robot learning, 2024.

[101] Yifan Xu, Theodor Chakhachiro, Tribhi Kathuria, and Maani Ghaffari. Solo t-dirl: Socially-aware dynamic local planner based on trajectory-ranked deep inverse reinforcement learning. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 12045–12051, 2023.

[102] Alexander Yeh, Photchara Ratsamee, Kiyoshi Kiyokawa, Yuki Uranishi, Tomohiro Mashita, Haruo Takemura, Morten Fjeld, and Mohammad Obaid. Exploring proxemics for human-drone interaction. In *Proceedings of the 5th International Conference on Human Agent Interaction*, HAI '17', page 81–88, New York, NY, USA, 2017. Association for Computing Machinery.

[103] Dina Youakim and Pere Ridao. Motion planning survey for autonomous mobile manipulators underwater manipulator case study. *Robotics and Autonomous Systems*, 107:20–44, 2018.

[104] Tao Zhao, Chengsen Chen, Hongyi Cao, Songyi Dian, and Xiangpeng Xie. Multiobjective optimization design of interpretable evolutionary fuzzy systems with type self-organizing learning of fuzzy sets. *IEEE Transactions on Fuzzy Systems*, 31(5):1638–1652, 2023.

[105] Zunchao Zheng. Learning to avoid collisions in mobile manipulator reaching task. In *2024 7th International Conference on Intelligent Robotics and Control Engineering (IRCE)*, pages 111–116, 2024.

[106] Jinghui Zhong, Dongrui Li, Wentong Cai, Wei-Neng Chen, and Yuhui Shi. Automatic crowd navigation path planning in public scenes through multiobjective differential evolution. *IEEE Transactions on Computational Social Systems*, 11(1):905–918, 2024.

[107] Wei Zhou, Shuxiang Guo, Jin Guo, Fanxu Meng, Zhengyang Chen, and Chuqiao Lyu. A surgeon's habits-based novel master manipulator for the vascular interventional surgical master-slave robotic system. *IEEE Sensors Journal*, 22(10):9922–9931, 2022.

[108] Zhengxue Zhou, Leihui Li, Riwei Wang, and Xuping Zhang. Experimental eye-in-hand calibration for industrial mobile manipulators. In *2020 IEEE International Conference on Mechatronics and Automation (ICMA)*, pages 582–587, 2020.

[109] Zhiqian Zhou, Pengming Zhu, Zhiwen Zeng, Junhao Xiao, Huimin Lu, and Zongtan Zhou. Robot navigation in a crowd by integrating deep reinforcement learning and online planning. *Appl. Intell.*, 52(13):15600–15616, October 2022.