A Framework for Cooperative Human-Aware Navigation and Coordination of Multi-Robot Systems in Social Environments

THÈSE N^{O} 9175 (2018)

PRÉSENTÉE LE 14 DÉCEMBRE 2018

À LA FACULTÉ DE L'ENVIRONNEMENT NATUREL, ARCHITECTURAL ET CONSTRUIT
LABORATOIRE DE SYSTÈMES ET ALGORITHMES INTELLIGENTS DISTRIBUÉS
PROGRAMME DOCTORAL EN ROBOTIQUE, CONTRÔLE ET SYSTÈMES INTELLIGENTS

ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE

POUR L'OBTENTION DU GRADE DE DOCTEUR ÈS SCIENCES

PAR

Zeynab TALEBPOUR

acceptée sur proposition du jury:

Prof. A. Ijspeert, président du jury
Prof. A. Martinoli, directeur de thèse
Prof. R. Alami, rapporteur
Prof. P. Lima, rapporteur
Prof. P. Dillenbourg, rapporteur

ÉCOLE POLYTECHNIQUE
FÉDÉRALE DE LAUSANNE
Suisse
2018

This thesis is dedicated to my parents, Mina and Alireza.
For their endless love, kindness, patience, support and encouragement …

Acknowledgements

THE accomplishment of this work would not have been possible without the help and support of many people. I would like to start by thanking my advisor, Alcherio Martinoli for giving me the opportunity to pursue this research and for his unconditional support over the years.

I would like to thank all members of the MOnarCH project for the truly great experience of working with them. In particular, I am grateful to Prof. Rodrigo Ventura and Deepak Viswanathan with whom I had the pleasure to closely collaborate. I would also like to thank José Carlos Castillo for helping me to get started with the Mbot interactive features and Paulo Alvito from IDMind for his invaluable technical support.

I would like to express special thanks to Francesco Mondada, Daniel Burnier, Norbert Crot, Florian Vaussard and Philippe Rétornaz from the Laboratoire de Systèmes Robotiques at EPFL, to Séverin Lemaignan, and to David Mansolino for their great help with the Ranger robot. I would also like to thank Gianni di Caro, Armando Pesenti and Alessandro Giusti for their valuable support on using and integrating their people tracker software.

I have been lucky to have supervised talented and hard-working students during my doctoral studies. I would like to thank Alessio Canepa, Stefano Savarè, Jose Manuel Palacios-Gasos and Paul Prevel for their genuine enthusiasm and for their dedication in transforming the project ideas to real robot functionalities.

I would like to thank all my former and current lab colleagues for the moments that we have shared together. Special thanks go to José Nuno Pereira, Alicja Wasik, Lorenzo Sarti and Iñaki Navarro whom I had the pleasure to closely work with, for all their help and indispensable support, for the discussions that we have had over the years and for everything that I have learned from them.

I would like to especially thank Faezeh Rahbar, Anwar Quraishi, Duarte Dias, Ali Marjovi and Chiara Ercolani who have been very kind and supportive and have been directly involved in my —sometimes long— human-robot experiments. Furthermore, I would like to thank Corinne Farquharson for her kind support as well as the amazing job that she does with making sure that all lab operations run smoothly. I would also like to acknowledge Denis Rochat and Emmanuel Droz for their invaluable help on the IT and robotic infrastructure.

Acknowledgements

I am especially grateful to my closest friends for always being there for me; to those who are here in Lausanne and to the ones who may be far in distance but certainly close to my heart. In particular, I am very thankful to Zeinab Sadatian, for being with me every step of the way and for the friendship that we have shared for more than twenty years.

My deepest gratitude and heartfelt thanks goes to my parents, Mina and Alireza for their incredible kindness, love and support throughout my life. I feel extremely blessed to have such wonderful parents, without them, none of this would have been possible. I would like to thank my brother Saied for his unfailing support, unbounded kindness and intelligent wit, my sister Leila for her unconditional encouragement and sincere love, Amin, and Yasmin, who has filled our lives with joy ever since she has opened her eyes.

Lausanne, 14 December 2018 Zeynab Talebpour

Abstract

CONTINUAL developments in robotic technology have enabled the use of robots in everyday applications in domestic, office and public spaces. Although single robot problems have been the main focus of social robotics research, applications of robots in social environments will not be limited to a single robot due to the increasing demand for robotic assistants and multi-robot operations. Multi-robot systems can achieve performances exceeding the sum of the individual robot contributions by exploiting the full potential of the team through information sharing, coordination, and joint decision-making.

Robots operating in human-populated environments either directly interact with people or have to share the space with the humans. It is of utmost importance that people co-existing with robots feel safe and comfortable around them. This makes human-awareness essential for long-term sustainable deployment of robots in such environments. Furthermore, for co-operative robots, the presence of humans and their actions can directly affect the robot and team plans, making human-awareness more essential for ensuring high performance as well as social acceptability. Research in the area of socially-aware navigation has received substantial attention in recent years. However, despite their great potential, human-aware teams of robots considering social factors at both individual navigation and collective coordination and planning levels, are currently largely unexplored.

In this thesis, we address the problem of human-aware cooperative navigation and coordination for multi-robot systems in realistic social environments. We focus on a class of multi-robot coordination problems known as multi-robot task allocation using a market-based approach. We explicitly consider the challenges of noisy, dynamic and stochastic human-populated environments by means of accounting for perception and prediction limitations and uncertainties in social cost modeling, bid estimation, coordination and replanning. We construct an end-to-end framework comprising three main components of (i) human-aware navigation, (ii) human-aware coordination and planning for multi-robot systems, and (iii) human-robot interaction in the presence of multiple cooperative robots.

We opt for an incremental approach to this problem starting from single robot human-aware navigation with expectation-based social costmaps. Subsequently, we move to multi-robot cooperative navigation in highly stochastic social environments. We propose human-aware

coordination strategies based on social costs and social risks. The concept of risk introduced in this thesis incorporates perception and prediction uncertainties as well as social costs for estimating the stochastic costs of tasks that the robots should bid on in the market. Additionally, we introduce an adaptive risk-based replanning method for dealing with the limitations of local perception and unpredicted human behavior in the social environment. Finally, we demonstrate the interactive potential of the team of robots for social multi-robot task allocation by integrating an interaction that actively requests human collaboration and assistance in socially costly and blocking situations, into our adaptive replanning strategy. Extensive experiments with up to four robots and 12 humans in simulation, and up to two robots and two humans in reality have been carried out for evaluating the performance of the proposed methods in this thesis.

Keywords: human-aware multi-robot coordination, human-aware cooperative navigation, risk-based multi-robot coordination, human-robot-interaction for multi-robot systems, adaptive replanning, social robotics, multi-robot systems

Résumé

LES développements continus de la technologie robotique favorisent l'utilisation des robots pour des applications quotidiennes, dans les milieux domestiques, les bureaux et les espaces publics. Bien que la recherche en robotique sociale se concentre principalement sur des problèmes à un seul robot, les applications des robots dans des environnements sociaux ne se limiteront pas à un seul robot, en raison de la demande croissante d'assistants robotiques et d'autres systèmes multi-robots. En comparaison avec leurs homologues individuels, les systèmes multi-robots peuvent présenter des propriétés supérieures qui ne peuvent être obtenues qu'en exploitant le potentiel de l'équipe, grâce au partage d'informations, à la coordination et à la prise de décision commune.

Lorsque des robots opèrent dans un environnement commun avec des humains, soit ils interagissent directement avec eux, soit ils doivent correctement partager l'espace. C'est de la plus haute importance que les personnes coexistant avec des robots se sentent en sécurité et à l'aise avec eux. Ce qui signifie que la prise en compte des humains est essentielle pour un déploiement durable à long terme dans de tels environnements. De plus, pour les robots coopératifs, la présence des humains et leurs actions peuvent affecter directement les plans des robots et ceux de l'équipe, ce qui rend la prise en compte de l'homme plus essentielle pour garantir des performances élevées, ainsi que l'acceptabilité sociale.

La recherche dans le domaine de la navigation socialement consciente a fait l'objet d'une attention considérable ces dernières années. Cependant, malgré leur grand potentiel, des systèmes multi-robot conscients de l'homme, tenant compte des facteurs sociaux, tant pour la navigation individuelle que pour la coordination et la planification collectives, sont actuellement en grande partie inexplorées.

Dans cette thèse, nous abordons le problème de la navigation et de la coordination coopératives conscientes à l'homme, pour des systèmes multi-robots dans des environnements sociaux réels et non contrôlés. Nous nous concentrons sur une classe de problèmes de coordination multi-robots appelée allocation de tâches multi-robots, en utilisant une approche basée sur le mécanisme du marché. Nous considérons explicitement les défis posés par les environnements bruités, dynamiques et stochastiques, en prenant en compte les limites et les incertitudes de la perception et de la prédiction dans la modélisation des coûts sociaux, l'estimation des offres, la coordination et la replanification. Nous construisons un cadre intégral

comprenant trois composantes principales : (i) la navigation consciente de l'homme, (ii) la coordination et la planification conscientes de l'homme pour des systèmes multi-robots, et (iii) interaction homme-robot en présence de plusieurs robots coopératifs.

Nous optons pour une approche incrémentale face à ce problème, commençant par la navigation d'un seul robot conscient des humains avec des cartes de coûts sociaux basées sur les attentes. Par la suite, nous avançons vers la navigation coopérative multi-robot dans des environnements sociaux hautement stochastiques. Nous proposons des stratégies de coordination conscientes à l'homme, tenant compte des coûts et des risques sociaux. Le concept de risque introduit dans cette thèse intègre des incertitudes de perception et de prédiction, ainsi que des coûts sociaux pour estimer les coûts stochastiques des tâches pour lesquelles les robots devraient faire des offres sur le marché. De plus, nous introduisons une méthode de replanification adaptative relative au risque, pour traiter les limitations de la perception locale et du comportement imprévu des humains dans l'environnement social. Enfin, nous démontrons le potentiel interactif de l'équipe de robots pour l'allocation sociale des tâches entre plusieurs robots, en intégrant une interaction qui sollicite activement la collaboration et l'assistance humaines dans des situations socialement coûteuses et bloquantes, dans notre stratégie de replanification adaptative. Des expériences approfondies, avec jusqu'à quatre robots et 12 hommes en simulation, et jusqu'à deux robots et deux hommes en réalité, ont été réalisées pour évaluer la performance des méthodes proposées dans cette thèse.

Mots clefs : coordination multi-robot consciente de l'homme, navigation coopérative consciente de l'homme, coordination multi-robot basée sur les risques, interaction homme-robot pour des systèmes multi-robots, replanification adaptative, robotique sociale, systèmes multi-robots

Contents

Part I Introduction and Background

1 Coordinated Multi-Robot Human-Aware Navigation

THIS chapter highlights the motivation of this research and describes the importance of this work. Furthermore, it provides a definition of our research problem and details the main relevant challenges.

1.1 Motivation

The use of robots in workspaces and home environments is increasingly becoming a common reality. There exist numerous applications for robots as personal assistants at homes, tutors at schools, and helpers at hospitals and nursing homes. Armar[1] Pepper[2], and Spencer[3] are examples of such robots designed to be integrated in social environments. One common key factor in all social environments is the presence of people and all the technical and social challenges it brings. Robots in such environments either directly interact with people or have to operate in spaces shared with humans. Thus, for long-term sustainable deployment in social environments, it is essential that robots gain the acceptance of people by providing

(a) (b) (c)

Figure 1.1 – Examples of robots operating in social environments. a) Armar at the kitchen, b) Pepper at the office and c) Spencer at Schiphol airport.

[1] By Karlsruhe University (KIT)
[2] By SoftBanks Robotics
[3] By SPENCER project

satisfactory levels of safety, comfort, and conforming to basic social rules [1] while maintaining efficiency in performing their tasks.

One of the most fundamental aspects of any robotic application is the ability to properly navigate in the environment. In environments shared with humans, especially in situations where contact between robot and human is not sporadic but rather repetitive, it is important that the robot is perceived as being intelligent and efficient and that it does not become a detrimental constraint to the motion and actions of the humans. Extensive research has been done on indoor navigation, a subset of which involves human-aware navigation. Human-aware navigation, defined as "The intersection between research on Human Robot Interaction (HRI) and robot motion planning"[1], focuses on finding solutions for socially acceptable robot navigation [2], [3] by constraining the navigation of the robots (both in terms of motion planning and of reactive response to dynamic environments) to respect personal spaces and social norms associated with human presence and motion.

As the number of applications and services that can be undertaken by robots increases, so will the need for companies and individuals to employ more robots, in an effort to handle a larger number of tasks, parallelize work and increase efficiency. However, benefits of considering a team of robots as opposed to an individual can only be obtained through coordination and cooperation. While this is true in environments occupied solely by robots, it gains new relevance in environments populated by humans and in tasks that either involve humans or are requested by humans. If a team of robots is not coordinated and is perceived to work at cross-purposes, disputing available tasks, with robots disrupting other robots, or even not sharing relevant information about the tasks and the environment, the perceived intelligence of the Multi-Robot Systems (MRS) by the humans will decrease and so will the predisposition of users to rely more on the robots. Therefore, cooperative human-aware navigation and coordination of multi-robot systems will be a necessity if teams of robots are to operate in real applications in environments shared with humans.

1.2 Relevance of Our Research

Despite the large body of research on human-aware navigation, some challenges of real deployment of social robots in the environment, namely, imperfect data about dynamic human subjects and the impact of uncertainty on social costs are yet to be explored. Moreover, in spite of the potential of multi-robot systems, the research in human-aware navigation area is mostly considering single robot applications and the problem of cooperative human-aware navigation for multi-robot systems -an interesting problem for both multi-robot and human-aware navigation research- is largely unexplored. When available, researchers tend to treat the team of robots solely as the sum of its individual components, neglecting to explore the added benefits of having coordinated actions between teammates.

As a first step towards such human-aware multi-robot system, this thesis will focus on the navigation and mainly coordination aspects of cooperative human-aware robots in stochastic

social environments. Different coordination algorithms will be explored and their performance will be compared with a baseline approach where individual robots execute human-aware navigation from a non-coordinated perspective. The insight obtained from this work will allow us to clarify what coordination strategies work best in different situations that multi-robot systems can be faced with in real-world applications. Moreover, this research will extend the current state-of-the-art of different coordination algorithms by introducing constraints for dealing with the social environments that the robots are placed in. For instance, in market-based coordination mechanisms, social costs will have to be considered as part of the auctioning/bidding schemes. The resulting negotiation must not only produce the most efficient joint plan but also one that respects the social constraints present in the environment as well as the presence, motion and actions of the humans therein.

Our final methodological impact is related to the consideration of uncertainty in both human-aware navigation and multi-robot coordination. Human-aware navigation methods (particularly those focusing on costmaps) will be extended to be adaptable to different degrees of uncertainty associated with the robot's perception of the humans in the environment. Uncertainty in both perception and human motion prediction will also be considered in market-based coordination for influencing the auctioning/bidding strategies.

Another interesting aspect that is currently underinvestigated in human-aware multi-robot systems, is the interactive potential of the robots. Interaction can be an asset to the robots for

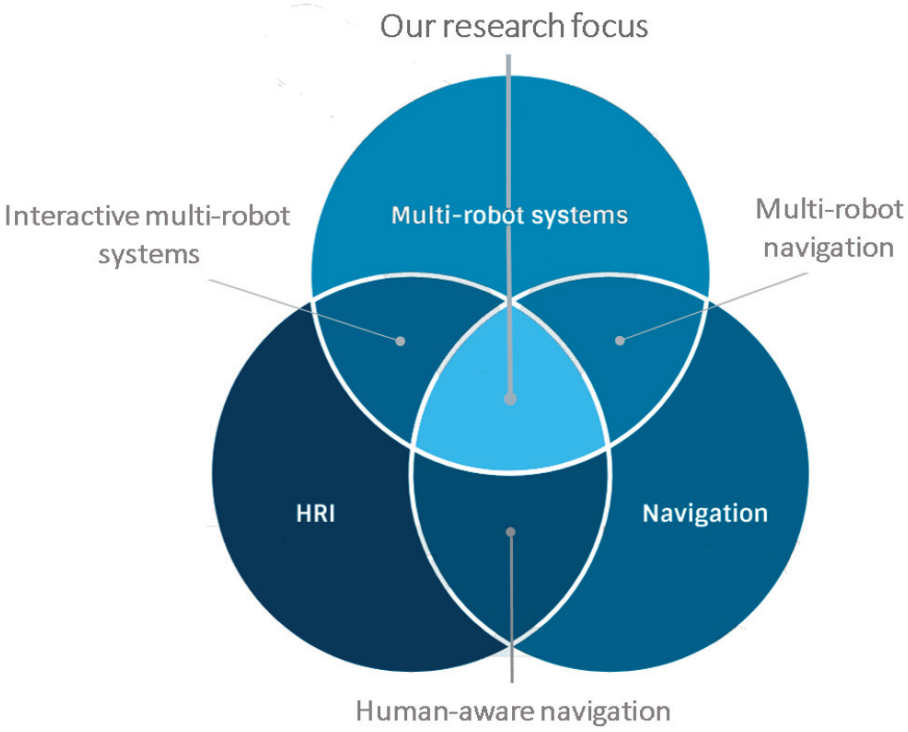


Figure 1.2 – An overview of the main research fields at intersection of which this thesis is situated.

improving their social acceptability as a result of increased legibility and human trust, and for improving the efficiency of their collective mission through actively collaborating with humans. We will target this aspect of the cooperative navigation problem in the final part of our research. Figure 1.2 highlights where this thesis stands with respect to the main underlying research domains.

1.3 Problem Definition

Imagine a group of service robots operating in a hospital. There could be demands for service in various locations at any point of time. The goal of such a team is to provide as many services as possible to the humans while minimizing the waiting times for the requesting users. Additionally, these robots must consider humans and social costs in their decisions in order to be accepted by the people present in the environment. There are two main components to this problem: (i) navigation and (ii) multi-robot coordination and planning, both of which have to ensure human-awareness. Each of these components are well-established fields of research. However, when faced with the dilemma of human-awareness, incorporating the social constraints resulted in from the presence of humans should be added to their classical definitions. In this thesis we define the problem of human-aware coordinated navigation as follows:

> To accomplish a team-level objective involving navigation in a human-populated environment for a group of autonomous robots while including human-awareness in decisions of navigation controllers, coordination strategies and planning schemes.

1.4 Challenges

The main source of challenges in this problem are the humans. As mentioned in a recent article on balancing theory and practice in HRI by Matarić [4], clean formalization for problems involving real people is far from reach. The existing models for human behaviors are all simplifications. As real world encounters of robots and humans are not controlled or carefully calibrated, uncertainty, noise and unpredictable behaviors are inherent parts of the cooperative navigation and planning problem that the robots are faced with. Furthermore, long-term and carefully designed studies are required to fully understand how humans will behave in different situations while the robotic system is in place. This type of study is very difficult to organize and necessitates a long time span.

Achieving accurate human perception is another challenge for robots deployed in real uncontrolled environments, as real world applications limit the infrastructure that can be used by the robotic system. There, the limited on-board resources of the robots should enable real time human detection and tracking and provide the estimated human poses required

for human-aware navigation and planning methods. Although, it is common to assume that one final pose value is given as the tracker output, in reality these pose estimations can be highly uncertain. Furthermore, lack of accurate ground truth data complicates the evaluation process.

The simulation to reality gap is very large in this problem mainly due to the limitations of human behavior modeling. The complexity of the environment is no longer a result of larger sized or more complex structures, but rather that of the large variety of interactions, human behaviors and expectations. This causes the problem space to become very large and systematic tests can no longer be ensured.

For multi-robot coordination and planning, the main challenge is incorporation of highly stochastic social costs and future actions of the humans into the robots' decision making process. As there is no model for the uncertainty caused by human behaviors in uncontrolled environments, the common approaches for dealing with uncertainty in the literature of multi-robot coordination cannot be directly adopted here. Ignoring this factor in coordination and planning can lead to degraded quality of plans as well as social dissatisfaction as robots would simply be blind to humans and will not consider humans as social beings that expect to feel safe, comfortable and able to understand what is happening in their surroundings. Moreover, robots will be ignorant to the changes caused by the humans that can directly affect their plans such as passages being physically blocked by humans.

Summary

Cooperative human-aware navigation is an important problem with many applications in social environments that has remained largely uninvestigated. This research will focus on coordinated human-aware navigation for a team of robots in environments shared with humans. There exist many challenges in this problem, mainly stemming from the presence of humans and the stochastic nature of their behavior.

2 Related Work

THIS chapter of the manuscript provides a brief overview on the methods and algorithms relevant to cooperative multi-robot navigation in social environments. This work has a broad scope and shares connections with a number of different topics in robotics as illustrated in Figure 2.1. In the following sections, we mainly contextualize this research with respect to the state of the art in human-aware navigation, multi-robot coordination and planning, and the intersection of the two topics which is multi-robot human-aware navigation. Nonetheless, there are other areas such as mapping, localization, human detection and tracking, human motion prediction, and human-robot interaction, that we are also interested in, since they need to be exploited as tools in our research.

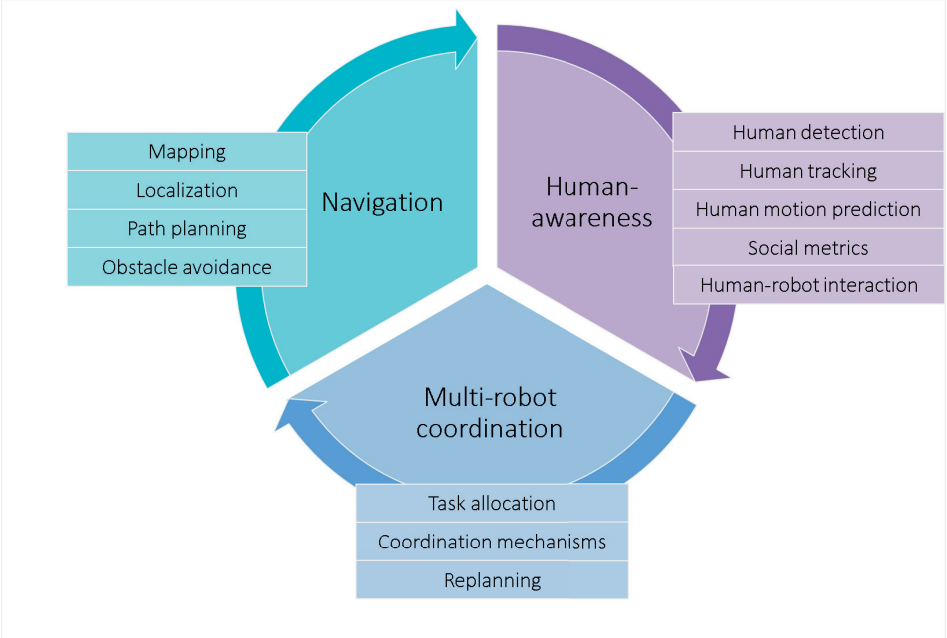


Figure 2.1 – An overview of system components for the proposed multi-robot human-aware navigation framework.

2.1 Human-Aware Navigation

Navigation is one of the main required functionalities for enabling robots to be actively used in real social environments. Robots have to navigate in environments shared with humans and the quality of their movement strongly influences how their intelligence is perceived [5]. Hence, one objective of social robotics research is to develop methods for making robot navigation socially acceptable and human-aware. Human-aware navigation is the crossing between research on robot motion planning and Human Robot Interaction (HRI). It focuses on the interaction dynamics between humans and robots that occur as a result of navigation [1]. Conventionally, safety, comfort, naturalness, and sociability (defined as the adherence to explicit high-level cultural conventions), have been the main focus of human-aware navigation techniques [1]. However, other criteria such as visibility and avoiding the hidden zones created by the surrounding objects [6] have also been proposed for human-aware path planners.

Socially-aware planners that take into account additional social costs while planning for the optimal path, understanding the impact of different social cost models and navigation strategies on social acceptance, and controllers that result in natural robot motion, are among the most investigated topics in this area. In the following sections we will address the most important components of a socially-aware navigation system and detail the challenges of human-aware navigation, focusing on the real deployment aspects of the problem.

2.1.1 Human-Awareness

In HRI literature, there have been numerous definitions of awareness. The shared thread between the definitions is the understanding that participants have of each other in the environment. A list of definitions used for awareness in HRI can be found in [7] along with different types of awareness. Social awareness is another term used to describe the characteristics that human-aware navigation algorithms aim to possess. It is defined as being attentive to the activities and presence of people in a shared environment [7]. Socially-aware navigation is "the strategy exhibited by a social robot which understands social conventions, relative management of space and conforms to them in order to preserve a comfortable interaction with humans. Resulting behavior is predictable, adaptable and easily understood by humans" [8], [9]. Based on this definition, social-awareness subsumes human-awareness.

The key factor we address in our human-aware navigation controllers is *comfort*. In the literature, we can find several strategies for comfort such as having an appropriate approaching strategy [10], maintaining an appropriate distance [11], control strategies to avoid being noisy [12] and use of planning for avoiding interference [13]. Legibility, is another important characteristic for social robots. Several approaches to generate legibility for robot navigation have been reported in [14]. Legibility means that a person intuitively understands the intentions of a robot [15]. It is shown in [16] that legibility and intent-expressiveness should be the main focus of motion planning in the presence of humans as opposed to the predictability of the motion. Predictability is defined as what is expected and not surprising to a human.

One important concept which is used in numerous studies [11], [17]–[19] in this area is the virtual space around a person that is mutually respected by other humans, based on the principle of *proxemics* [20]. Based on this, depending on the relationship and the interaction that exists between humans, people choose different social distances relating to intimate, personal, social or public contexts. Change in the expected distance may indicate dislike if it is too large or cause discomfort if it is too small. Respecting personal spaces, O-spaces and P-spaces [21] are the common social behaviors considered in the literature of human-aware navigation based on the concept of proxemics. However, this could become crucial especially in confined spaces, such as corridors where two agents can navigate only in side-by-side configurations. Some flexibility can be introduced to this proxemics-based approaches to allow reciprocal and interactive motions [22], as seen also in real navigational patterns of humans. A recent comprehensive study on adapting models of proxemics for maximizing social impact can be found in [23].

Social costmaps are a common way to model this principle that have been used in various studies in the field. Many factors can be considered for shaping a social costmap, such as age and gender [10], velocity of the motion [24], etc. The costmap can also be constructed based on learned features. Okal et al. [25], use a learned reward function for navigation comprised of the relative heading of humans with respect to the robot's goal, heading deviation from goal and distance to goal, for defining costs in the costmap. However, the proxemics distance has been the main factor when accounting for comfort in the literature.

There exist a few survey papers on the topic of human-aware navigation that provide a detailed overview on the literature in this topic [1], [9], [26]. Among the most essential elements of a human-aware navigation system, we will address path planning, human motion prediction and the challenges specific to navigation in an environment shared with people, in the following sections.

2.1.2 Path planning

Path planning provides a list of way-points on a map to be followed by the robot which optimize the performance with respect to a global objective function such as the distance traveled. A map of the environment marking the blocked and available spaces is used by navigation algorithms for this purpose. For global path planning, a search method is used to find a set of consecutive states starting from the robot position to the goal position, which, given a specific cost function, constitute the optimal path. Most of these algorithms focus on minimizing the path's length [27], [28]. A general overview of path and trajectory planning can be found in [29]. A human-agnostic navigation method, usually takes the shortest or most energy efficient path in a graph. This is commonly done using a variant of the A* search algorithm. As an example, Luber et al. [8] show the effectiveness of theta* algorithm that is an any-angle A*, for path planning.

Ventura et al. [30] use a different approach based on the Fast Marching Method (FMM) [31]

for path planning, resulting in optimal paths in the absence of unmapped obstacles. FMM is a numerically efficient method to solve the Eikonal equation for a domain discretized as a grid. Its computational complexity is $O(N \log N)$, where N is the total amount of grid cells, which is comparable to Dijkstra's algorithm for sparse graphs. However, the shortest or most energy efficient path is not necessarily the most appropriate and desirable when social factors are taken into account. Factors such as safety, comfort, naturalness, and legibility, are the attributes which are desired for the robot path in social environments. These attributes can be considered in global path planning by means of dedicated cost functions. A list of cost functions used in the literature can be found in [1].

FMM has been proven to be successful in real domestic spaces with high complexity [30]. There are a number of research papers addressing social path planning using FMM [32], [33]. In [32] a theoretical framework for a number of sub-problems of social path planning is presented and an extended model for engaging groups of people is proposed using a special version of fast marching square planning method [34]. However, the information about humans are considered to be deterministically known and noiseless. Moreover, only simulations have been used to show the effectiveness of the method for static people. The same problem needs to be investigated in real world scenarios with the challenges that exist therein.

In local path planning, commands for the immediate future of robot actuators are determined. This is also referred to as "reactive planning" and "collision avoidance" [1]. The key responsibility of the local planner is to ensure safety of the robot and the environment. There exist a number of local planners in general but the more common approach for planners used in human-aware navigation research is the sampling approach (e.g., the Dynamic Window Approach (DWA) [35]). In its conventional applications, local path planning just finds motor responses for the perceived static or dynamic obstacles regardless of their nature and there is nothing specific to human-awareness about this type of planning. Further information about collision-avoidance techniques can be found in this survey [36].

2.1.3 Human Motion Prediction

Human navigation and robot navigation have a mutual influence on each other [1]. In addition, navigation has a communication effect through non-verbal cues and body language. Therefore, prediction is essential for human-aware navigation and having a model of the future actions of a human is extremely helpful. This enables the robot to have plans which are compliant with what the human will do in the near future. The prediction required for navigation is closely linked to human walking behavior. A comprehensive overview of the related publications is given in [8].

There exist two main approaches to prediction, prediction based on geometric reasoning and prediction based on learning [1]. The assumptions of how agents behave in general are used for justifying predictions in reasoning-based approach, e.g., prediction takes into account that obstacles cannot be crossed by other agents in [37]. In learning-based prediction

this justification is done based on the observations of agents' behaviors, notably in special circumstances and environments. In this approach, data of typical trajectories within a given environment are first gathered over time. New samples of the movement of the humans in that environment are then predicted using the data collected in the first phase [13], [38]. In another approach [39], a model of the navigation behavior of cooperatively navigating agents is learned from demonstrations by inferring a model of the underlying decision process, based on observations of their continuous trajectories.

In the human-aware navigation literature, prediction methods based on the Hidden Markov Model (HMM) framework [40], the social force model [41], [42], human motion learning [43], Kalman Filter (KF) [44], [45], and Support Vector Machine (SVM) [46] have been the main adopted approaches (for more information refer to [26]). A very recent work by Rudenko et al. [47], has established a planning-based approach for long-term human motion prediction that accounts for local interactions and can accurately predict joint motion of multiple agents. Although this method has not been applied to human-aware navigation, it shows promising results for this purpose.

2.1.4 Challenges

In this section a number of challenges in human-aware navigation that are most relevant to our research problem are presented.

- We note that, the human motion prediction methods either rely heavily on data collected in known environments or are based on models that simplify the human motion behavior that we would encounter in our target environments. More importantly, the time span provided by the current methods for prediction is much shorter than what we need for our team planning estimation of costs. Since our robots will be planning for tasks in social environments that are usually large such as hospital wards, we need to predict the future state of the world over the course of reaching the tasks which can typically be in the order of minutes. Moreover, highly stochastic nature of uncontrolled social environments in terms of human behavior, make simplified models of the humans inaccurate in reality.
- Human perception in uncontrolled social environments is a challenging problem. Furthermore, identifying and accounting for the uncertainty in perception is of great importance and the assumption of having perfect deterministic human perception for human-aware navigation is too simplistic. Nonetheless, to the best of our knowledge, this has not been the subject of many notable studies. There is a dedicated chapter in [48] on local planning with uncertainty, however, this is not considered in a social context. The sources of uncertainty in [48] are the position of the robot and the obstacles and the partially known *motion* of moving obstacles that are considered to be people. However, perception of people, and the uncertainty in person and group detection and tracking has not been investigated.

- There are a number of limitations in the original proxemics model. Firstly, it only addresses the distance that should be chosen for explicit interaction when standing. This static explicit interaction is certainly not the only situation that needs to be considered. Additionally pose, orientation, gaze and the trajectories of the agents cannot be described using this model. On the other hand, a dynamic proxemic behavior is required for tasks where extrinsic interferences can occur to the human-robot proxemics models. As an example the distance between a speaker and a listener varies in a quiet room compared to a noisy room [49]. It is shown in [38] that intrusion into the personal space appears to be common for walking subjects and proxemics-based strategies for avoidance can lead to conservative and inefficient motion behavior for the robots in some problems. The authors of [38] believe that research in human-aware navigation and manipulation, which currently has mostly taken a model-based approach, is questionable to be effectively employed for complex, real-world applications. They state that socially aware behavior should in fact be learned from real data.

2.2 Multi-Robot Systems in Social Environments

Coordination is of paramount importance in the development and deployment of MRS. In this thesis, we focus on one particular class of MRS coordination mechanisms commonly known as Multi-Robot Task Allocation (MRTA) [50], [51]. In the following sections, we will briefly review the relevant literature for MRTA, replanning in teams of coordinating robots and multi-robot systems in applications involving humans. Lastly, we will present the challenges of cooperative multi-robot systems in social environments focusing on the MRS aspects of the problem.

2.2.1 Multi-Robot Task Allocation (MRTA)

MRTA algorithms vary in design and application [52], [53] but their common objective is to find a mapping between robots in a team and a set of "tasks" that must be accomplished in order for the team "goal" to be completed. The term "task" can have different interpretations in robotics research but herein we assume it to represent a subgoal that is necessary for the overall goal to be achieved, and that can be achieved independently of other subgoals. This subgoal can be at a high abstraction level (e.g., behavioral) or at a lower level (e.g., motion planning).

Among multiple approaches proposed for MRTA, we are mainly interested in distributed approaches that can be executed by a team of robots without the explicit need for an external centralized entity with perfect knowledge of the environment [54]. Furthermore, since our objective is the introduction of such teams in real-world human-populated environments where robots must be aware of the social conventions, we are particularly interested in approaches where cooperation is intentional or explicit.

Market-based multi-robot coordination [55]–[58], is an example of such an MRTA approach. Inspired by market mechanisms and studies in social sciences, researchers have proposed systems like MURDOCH [59], TraderBots [60] and Hoplites [61] to achieve flexible allocation of tasks using auctions among robots. In these systems, robots act as agents trying to maximize their individual profits. Every time a task is auctioned, robots must pay a price to obtain it. Once the task is completed, a payment is done to the robot which won the auction. The underlying assumption is that with every robot trying to maximize its individual profit, the overall team coordination and efficiency will be improved. Market-based systems can be said to have an intentional model of cooperation [62], where different tasks have to be accomplished and robots cooperate explicitly, often through communications, to correctly allocate resources to tasks.

As Gerkey states in [50], "if the robots are deliberately cooperating with each other, then, intuitively, humans can deliberately cooperate with them, which is a long-term research goal of multi-robot research". Moreover, using an intentional model of cooperation, it is more likely that the resulting behavior of the system is easily understood and predictable by the humans present in the environment, a key feature for social robots. There exist a variety of different solutions to MRTA. On the centralized end of the spectrum, most approaches tend to treat MRTA as a combinatorial optimization problem and use standard algorithms to solve it [63]. The Broadcast of Local Eligibility approach [64] and the L-ALLIANCE architecture [62] are examples of solutions that consider MRTA as an optimal assignment problem. Robotic soccer is one of the traditional application of the centralized approach to MRTA [65].

On the other end of the spectrum, the auction algorithm [66], [67] exhibits the distributed nature required for distributed systems. Further research on the auction algorithm [68] has shown that it can be implemented in a distributed fashion. Another distributed yet different approach from market-based coordination is the threshold-based allocation used by [69], [70]. This approach is based on self-organization principles [71], [72], usually taking inspiration from social insects, where cooperation occurs implicitly, often without direct communication. Such approaches cannot be said to have an intentional model of cooperation. A comparison between threshold-based and market-based approaches can be found in [73].

Market-based approaches have been used in search and rescue operations [74], topological navigation [75] and many more applications involving path planning. While most market-based approaches deployed on robots for path planning consider tasks as final locations, a task in the Hoplites framework [61] is composed of a set of locations or way-points. This framework allows for coordinating "plans" encapsulating multiple tasks, instead of tasks. This is the key feature differentiating Hoplites with other methods mentioned so far, except for the Consensus-Based Bundle Algorithm (CBBA) method [68], where planning for sequences of locations is done in a distributed manner. Single or multiple tasks can be auctioned in the context of market-based coordination. Consensus-Based Auction Algorithm (CBAA) proposed in [68] is an example of single task auction algorithm. CBAA assigns the task with the minimum cost to each available robot in a decentralized manner through auction and consensus mechanisms.

The Hoplites framework consists of two concurrent coordination mechanisms: passive and active. A passive coordination quickly produces locally-developed solutions while active coordination produces complex team solutions via negotiation among teammates. Active coordination makes the distinction between this approach and the CBBA proposed in [68]. The active coordination scheme allows for more flexibility in Hoplites compared to CBBA, since robots can modify their plans not just for maximizing their own profit but rather for contributing to achieving a better team performance by means of cooperation, upon request of other team members. This request is formed locally by the robots and targets only a subset of team members that are identified for collaboration.

2.2.2 Stochastic MRTA

As stated in [52], despite the importance of uncertainty in real robotic problems and the potential of stochastic planning for producing sound and robust allocation policies, most MRTA approaches assume a deterministic MRTA model and deal with uncertainty only at execution time by replanning during task execution. In stochastic allocation literature, it is assumed that a model of uncertainty, for instance a probability distribution of robot travel time, task arrival, etc., is available. Such MRTA problems are commonly modeled as Markov Decision Process (MDP) [76], or as pure or mixed stochastic integer programs [77]. In a different approach, approximation of the parametric uncertainties captured by the underlying system model has been investigated in [78] by means of active learning. Despite the research is this area, it is not clear what is the best approach for facing uncertainty and the challenges of MRTA still remain open. It is yet not known if building complex models that incorporate uncertainty is a better approach compared to building less well-informed plans and replanning as often as needed to quickly react to unexpected events [52].

Planning under uncertainty can also be addressed using Partially Observable Markov Decision Process (POMDP). However, this approach is faced with a scalability problem when considering teams of robots. Auctioning of independent local POMDP-based controllers is proposed by [79] to alleviate this problem. Nonetheless, for real uncontrolled environments, the information space get too large to be tractable. In POMDP approaches such as [79], the state and action space of the robots are discretized to be able to computationally solve the problems. This limits the application and impact of such methods when it come to real, spatiotemporally continuous, complex problems.

2.2.3 Replanning

Changes in the environment for a realistic MRTA problems are inevitable. These changes are listed in [80] as having faulty robots, changes in estimated cost due to uncertainties, changes in task definitions, online arrival of tasks, addition of robots to the team, and other changes made by external agents. The planning loop executed by most MRTA methods consists of planning-execution-replanning of tasks [52]. Replanning plays an important role in MRTA as it

is responsible for handling these changes in the environments and maintaining some level of efficiency. Moreover, some temporary failures are shown to be handled by replanning in [81].

Typically, replanning occurs continuously or at predetermined points in time. It can also be triggered based on a set of events such as a robot accomplishing a task or arrival of a new task. However, not all changes or uncertainties in the environments can be captured efficiently using these strategies. In [82], the authors propose a live task modification method that adjusts the composition of teams currently executing tasks, in response to the realities of execution. This is done by means of proactive replanning in a way that the problems or opportunities in a construction assembly scenario are predicted using a centralized planner. This is an interesting approach. However, it is not clear how such proposed method will perform in real environments and more so in the presence of humans.

CBBA with Partial Replanning (CBBA-PR) is introduced in [83] for allocating new tasks that appear online during the solving of the task allocation problem. The authors demonstrate that by resetting the lowest bid tasks from previous rounds of CBBA, the team obtains a fast convergence while still maintaining coordination. The focus of the problem studied in [83] is handling new tasks and not the stochastic changes in the cost of particular tasks or taking into account the new available information for planning. Moreover, there is no human factor involved.

2.2.4 Multi-Robot Human-Aware Navigation

In HRI, applications with multiple robots have not been explored much compared to applications involving a single robot. In this section, we will mention research papers that take into account multiple mobile robots and one or multiple humans in the problem that they address. However, the cooperation aspect is not existent in all of the mentioned works.

Designing an effective user interface for applications involving multiple robots is addressed in [84], [85] for performance improvement. In [86] authors describe a taxonomy for real-time interaction between a human commander and a swarm of robots. In [87] the situation of having several robots guiding several people is investigated. A Social Force Model (SFM) is used to guide a group of humans in a natural way. Moving objects such as robots and humans are modeled as masses under a virtual gravity force in social force model representations. Hence, in [87] attractive forces are defined for pursuing the leader robot and the goal and repulsive forces are defined for obstacles, other humans and robots. There is no consideration of social acceptance by the humans here and humans are only perceived as dynamic obstacles which need to be guided. A centralized methodology is used which relies on the central host for deliberation of robot actions. Higher-level social awareness is considered in a recent paper by Wasik et al. [88] for the formation control problem. In [88], normative aspects are introduced into robot behaviors using institutions to enable robot participation in mixed human-robot societies.

A group of robots guiding a group of people and adapting trajectories to avoid the humans from getting lost or leaving the group is presented in [89]. An optimization process tries to minimize the work of cooperative robots and displacement of humans. Robots change roles accordingly to perform the mission in a more efficient way. Human-aware navigation is not considered in this work and there are no experiments with real robots. Guiding people in an evacuation mission using multiple robots in a closed environment is presented in [90]. The environment is represented by the means of a Laplacian artificial potential field. By estimating the gradient of the field and tracing the gradient descend while keeping a formation, the robot team tries to implement the cooperative exit seeking algorithm. The authors introduce a model for the panic behavior of the humans in emergency situations. In their method one robot is assigned to be the leader and the rest of the robots have the shepherd roles. They pre-calculate the potential function for every point in the environment which makes this method only work for static environments. The concept of human-robot interaction zones has been considered and the robots stay in the social zones of humans. Also the panic model tries to emulate the behavior of humans in such situations but the model seems to be very high level and independent of many factors such as what elements are creating panic. They also assume that when a human is at a certain distance it will be found and guided just by robots moving towards the exit. The experiments have only been conducted in simulation.

A selected work on modeling pedestrian behavior and crowd disaster can be found in [91] that has been ported into the robotic domain by [92] where the local navigation of the robots is based on the mutual avoidance adopted by humans. All of the mentioned works apart from [88], [92] only considered simulated robots. In [92], despite having real robot tests, a simulated model of the human is used.

Very little work has been done in MRTA for social environments. In [93], a general MRS architecture for person search for a team of assistive robots in a retirement home is proposed. Therein, MRTA is considered as a constraint optimization problem and a centralized planning approach based on constraint programming [94] is used to solve it. The focus of this work is on the capabilities of the proposed architecture and social aspects have not been considered. Cavallo et al. [95] introduce socially believable robots that support older adults in urban areas by assisting in delivering groceries and collecting garbage. Their system (RobotEra) adopts a centralized planner for a heterogeneous MRS operating in an elderly care facility. In [96], a real-world multi-robot coordination problem for human guidance, requiring stochastic transitions is shown to be successfully implemented in a centralized fashion, at a scale of five to ten robots. The human behavior is simplified and a discrete state space representing key locations on a map is used for the MDP. Similarly, no social costs have been considered in this study.

In [97], [98] authors introduce the concept of socially invisible robot navigation in the social world using robot entitativity for mobile robots and autonomous vehicle navigation based on prior psychological research. They establish a mapping between emotional reactions and multi-robot trajectories and appearances and generalize their finding across various

Table 2.1 – Comparison of different approaches for human-aware multi-robot systems.

Method	Real robots and humans	Multiple robots	Human-awareness
Guiding groups of humans [87]		✓	
Cooperative robot movements in a guiding mission [89]		✓	
A simple panic model for humans in evacuation [90]		✓	✓
Socially-aware navigation with collision and disturbance risks [8]	✓✓		✓
Local navigation based on human mutual avoidance [92]	✓ -	✓	
MRTA for person search [93]	✓✓	✓	
MRTA for human guidance [96]		✓	
Socially invisible navigation [97], [98]	- ✓	✓	✓
Cooperative human tracking [99]	✓✓	✓	
Our proposed approach	✓✓	✓	✓

environmental conditions. Their method evaluation is based on a web-based study in which simulated videos of the robots are scored by human participants. Entitativity is the extent to which a group resembles a single entity versus collection of individuals. Low entitative groups are used to develop a real-time navigation algorithm that should enhance social invisibility for multi-robot systems. Robots with lower entitativity exhibit more friendliness and comfort and less creepiness and ability to unnerve as reported by authors in [97], [98]. This is an interesting study from the point of view of the impact of navigation of multiple robots on humans.

2.2.5 Challenges

In this section a number of challenges that we are faced with regarding multi-robot systems in social environments are discussed.

- Multi-robot scenarios involving humans have a great capacity to be explored yet, in terms of human-awareness and cooperation. Despite the great body of work in HRI and MRS research, to the best of our knowledge, a cooperative human-aware and distributed navigation method that works efficiently in real social environments is missing in the literature. One possible reason can be the complexity of such systems in terms of design, coordination, deployment and evaluation. Analyzing human response to a single robot is sufficiently difficult in most cases and adding another axis to the problem space can largely increase its difficulty. Table 2.1 shows a comparison between different approaches regarding human-aware multi-robot systems in the state of the art of the field. Our proposed approach will tackle this problem by accounting for all the three aspects indicated, *i.e.*, having a real human-aware multi-robot system.
- MRTA in social environments with human-aware team of robots that account for humans in planning, coordination and navigation has not been considered in the state of the art. Unlike most stochastic MRTA approaches, an uncertainty model for uncontrolled social environments

is not available unless strong assumptions are made or a data-driven approach, targeting a specific environment, is taken. Additionally, the scale and complexity of the problem is too large for applying POMDP-based solutions. Moreover, each encounter of the robots with humans matters and improving the average performance is not the best strategy to gain social acceptance for the robots.

- The interactive potential of a robot team is another aspect that is overlooked in the literature. Namely, how robot-robot and robot-human explicit and implicit interactions should change in the presence of humans and a *team* of robots. Although exploring this aspect adds even more complexity to the problem, it is an essential part for having teams of social robots integrated into environments shared with humans.

Summary

In this chapter we have seen the relevant research in the literature of human-awareness, path planning and human motion prediction in the context of human-aware navigation. Moreover, a number of existing challenges in this domain, namely, limitations of human behavior models and prediction methods, the assumption of deterministic human perception and limitations of proxemics-based social cost models have been detailed. An overview of the state of the art of multi-robot systems focused on MRTA is presented in this chapter. Additionally, a number of relevant research papers for applications involving multiple robots and humans have been introduced. Moreover, a number of challenges for developing cooperative human-aware robot teams have been briefly described in this chapter.

3 Scope of the Thesis

THE purpose of this chapter is to provide an outline of the thesis and to detail the main objectives of our research. Finally, we present our research contributions and the related publications.

3.1 Objectives and Outline

The main objective of this thesis is to develop an end-to-end framework for cooperative human-aware navigation and coordination of multi-robot systems in social environments. We aim to have a realistic, distributed and socially-aware approach by means of taking into consideration the key constituting features (see Figure 3.1) of each of these aspects in our proposed framework. We are interested in understanding how cooperative multi-robot navigation can be achieved in a human-aware manner and how multi-robot navigation, carried out by a number of individually human-aware robots, can benefit from human-awareness at coordination and planning levels.

This thesis is laid out in six parts. A brief description of each part and its corresponding objectives can be found in the following.

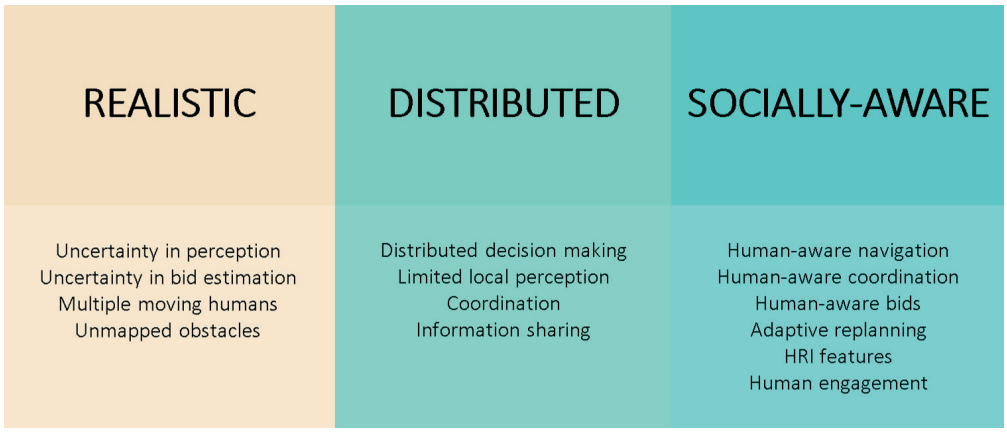


Figure 3.1 – Key characteristics and their corresponding features for the framework proposed in this thesis.

Part I - Introduction and Background In this part, we introduced the motivation and relevance of our research as well as defining our research problem and presenting the main existing challenges therein. Furthermore, we presented a literature review on the main relevant topics to our research problem, *i.e.*, human-aware navigation and multi-robot systems. As these topics have a broad scope, the key aspects of each topic along with the main challenges that are most pertinent to our research problem have been introduced. Additionally, the layout of our manuscript and the main corresponding objectives in each part of this dissertation are presented.

Part II - Platforms and Tools In this part, we introduce our robotic platforms and the main software tools used in this work. Additionally, we describe the basic available functionalities and the interactive capabilities of our main robotic platform. Furthermore, we introduce the high-fidelity simulation tool (Webots) adopted in this research. Subsequently, a number of human detection and tracking methods considered in this work are briefly described and the experimental setups where our real robot experiments take place are introduced.

Part III - Human-Aware Navigation In this part, we introduce the baseline navigation methods for our two robotic platforms. We then explain the different human-aware navigation approaches proposed for each of these robots. Furthermore, we present the expectation-based social costmap model that incorporates perception uncertainty, and finalize the single robot human-aware navigation approach that will be adopted in the consecutive chapters.

Part IV - Cooperative Multi-Robot Navigation in Social Environments In this part, our Hoplite-based multi-robot cooperative navigation method is detailed. We incrementally build our human-aware cooperative navigation framework starting from a deterministic cost model. Human-aware coordination methods using deterministic social costs and risk-based social costs are presented subsequently. Finally, limited local perception and highly stochastic environments are considered by means of our risk-based adaptive replanning method.

Part V - HRI-Agumented Cooperative Multi-Robot Navigation In this part, we detail the integration of the interactive features of our main robotic platform with robot navigation. Furthermore, we introduce an interactive multi-robot task allocation approach by means of adopting interaction in our risk-based replanning method. This is to enable robot collaboration with humans when possible, in order for the robots to improve their individual and team performance by actively asking for human assistance.

Part VI - Conclusion In this final part, we summarize this thesis and detail the core contributions of our work. Moreover, we describe the promising directions for possible continuation of the research presented in this thesis.

3.2 Research Contributions

This work, to the best of our knowledge, is the first work to tackle MRTA in social environments and as far as we know, the topic of human-aware coordination and planning has not been explored in

the literature for multi-robot systems. The work done in the course of this thesis has led to several contributions and related publications detailed in the following.

Part II - Platforms and Tools Our first contributions concerns the use of mobile robots in order to automatize the calibration process with the ultimate purpose of improving UWB-based human localization in a realistic indoor environment. Relevant publications include:

- A. Canepa, **Z. Talebpour** and A. Martinoli. "Automatic Calibration of Ultra Wide Band Tracking Systems Using A Mobile Robot: A Person Localization Case-study." The International Conference on Indoor Positioning and Indoor Navigation (IPIN), Sapporo, Japan, 2017, pp. 1-8.

Part III - Human-Aware Navigation We introduce a framework for a resource-constrained mobile robot platform for human-aware navigation. Additionally, the concept of exception-based social costmaps are introduced for the first time to capture the perception uncertainty in human-aware planners. Relevant publications include:

- **Z. Talebpour**, I. Navarro Oiza and A. Martinoli. "On-Board Human-Aware Navigation for Indoor Resource-Constrained Robots: A Case-Study with the Ranger." IEEE/SICE International Symposium on System Integration (SII), Nagoya, Japan, 2015, pp. 63-68.
- **Z. Talebpour**, D. Viswanathan, R. Ventura, G. Englebienne and A. Martinoli. "Incorporating Perception Uncertainty in Human-Aware Navigation: A Comparative Study." International Symposium on Robot and Human Interactive Communication (RO-MAN), New York, USA, 2016, pp. 570 - 577.

Part IV - Cooperative Multi-Robot Navigation in Social Environments We introduce a flexible multi-robot task allocation framework capable of solving different instances of the MRTA problem. Following an incremental approach, we develop a human-aware coordination method for a team of robots assuming a deterministic cost model. Additionally, we introduce risk-based bids that enable our human-aware coordination to account for the costs in real stochastic social environments. Finally, adaptive risk-based replanning is proposed for dealing with limitations of local perception and unpredicted human behavior. Relevant publications include:

- **Z. Talebpour**, S. Savarè and A. Martinoli. "Market-based Coordination in Dynamic Environments Based on the Hoplites Framework." The 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Vancouver, British Columbia, Canada, 2017, pp. 1105-1112.
- J. M. Palacios-Gasos, **Z. Talebpour**, E. Montijano, C. Sagues and A. Martinoli. "Optimal Path Planning and Coverage Control for Multi-Robot Persistent Coverage in Environments with Obstacles." International Conference on Robotics and Automation (ICRA), Singapore, 2017, pp. 1321-1327.
- **Z. Talebpour** and A. Martinoli. "Multi-Robot Coordination in Dynamic Environments Shared with Humans." IEEE International Conference on Robotics and Automation (ICRA), Brisbane, Queensland, Australia, 2018, pp. 4593-4600.
- **Z. Talebpour** and A. Martinoli. "Risk-Based Human-Aware Multi-Robot Coordination in Dynamic Environments Shared with Humans." IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Madrid, Spain, 2018, pp. 3368-3372.

- **Z. Talebpour** and A. Martinoli. "Adaptive Risk-Based Replanning for Social Robots with Limited Local Perception." IEEE Robotics and Automation Letters (RA-L), 2019. (to be submitted)

Part V - HRI-Augmented MRTA We study the impact of interaction with humans on the MRTA performance and social acceptability of the robots. Currently, we are organizing a user study for an in-depth evaluation of the performance of the proposed approach in this part.

I would like to give credit to my close collaborators who made addressing this diverse set of topics possible. Although I have been directly involved in all of the work presented in this thesis, some parts of this work are the result of a collaborative effort where I have been directly contributing in terms of proposing the idea, supervision and various degrees of development and testing.

Master student Alessio Canepa worked on the Ultra-Wideband (UWB) calibration in Chapter A under my supervision during his master project. I have been collaborating with colleagues form the European project of Multi-Robot Cognitive Systems Operating in Hospitals (MOnarCH), Deepak Viswanathan and Professor Rodrigo Ventura for the work presented in Chapter 11. A first version of the Hoplites-based algorithm presented in Chapter 15 was developed in the course of the semester project of Stefano Savarè under my close supervision, and was subsequently improved and adapted to the human-aware multi-robot coordination and planning framework. A visiting Ph.D. student, José Manuel Palacios-Gasós worked on extending his multi-robot coverage algorithms to real multi-robot systems under my supervision, resulting in the work I partially describe in Chapter 15. Nicolas Talabot has worked on Kinect-based human perception under my supervision during his semester project where he has developed and additional module for extending the 2D position estimations of body joints to 3D poses. Paul Prevel has worked on developing interactive behaviors for the Mbot using existing building blocks during his semester project and has contributed to integration of HRI features with the risk-based replanning and multi-robot task allocation framework under my close supervision. Finally, the credit goes to our fellow members of MOnarCH project for their contributions to providing the basis for this work in terms of single robot navigation and technical support.

Summary

This thesis focuses on developing an end-to-end framework for coordinated human-aware navigation of a team of robots. Throughout this thesis, we aim to have a realistic, distributed and socially-aware approach that takes into account the stochastic nature of social environments and their inherent uncertainty at navigation, coordination and planning levels. Following an incremental approach, single robot human-aware navigation is initially studied. Subsequently, multi-robot coordinated navigation with deterministic and risk-based cost estimations, and adaptive replanning for dealing with limitations of perception and human behavior change are addressed. Finally, interactive capacities of the robots are explored in the context of MRTA by means of incorporating interaction into our risk-based replanning method.

Part II Platforms and Tools

4 Introduction

THE multi-robot system developed in this thesis is composed of a number of physical and software components. The experimental setup providing the necessary modules for successful deployment of the robots and performance evaluation of our methods is another key component in our system. Throughout this research we have conducted experiments with two different robotic platforms. Their perception and localization sensors, actuators, and computational equipment will be detailed in Chapter 5. Implementation and validation of the algorithms presented in this work, have been made possible through several software frameworks, including the Robot Operating System (ROS) and the high-fidelity robotic simulator Webots. These frameworks along with the various human detection and tracking methods evaluated and tested in this research will be presented in Chapter 6. We have conducted experiments in a number of environments in an indoor setting using multiple networked robots. Chapter 7 will describe the specifications and choices made in each environment. As we further advance in the manuscript, more software modules will be added to our system.The system architecture encapsulating all parts with the corresponding connections can be found in the following section.

4.1 System Components

Before going to the details of different modules implementing our proposed methods, we would like to give a preview of the schematics of our framework. Figure 4.1 shows the main system components and building blocks of our system. An incremental approach is taken in developing and extending our proposed framework throughout the manuscript. We start from the already existing basic navigation functionalites and advance to a HRI-augmented social MRTA. Each one of these blocks can be a research topic in itself and we have not focused on contributing to the state of the art in every one of these components. Nonetheless, we have designed our framework in such a way that allows for modifying and improving each block and possibly adding new blocks such as data fusion, on-board human detection and tracking, detecting more types of social interactions and extending the social cost model. Moreover, despite having this framework applied to the Mbots (Figure 4.1 shows the schematics of the system running on the Mbots), none of our methods are platform-dependent and any other mobile robot that adopts a path planner incorporating social costs into the plans, can make use of our proposed framework. The human detection and tracking module shown in Figure 4.1 illustrates the use of an off-board, global human perception system. For an on-board, local human perception, we can imagine to have the human detection and tracking module internal to each robot with an

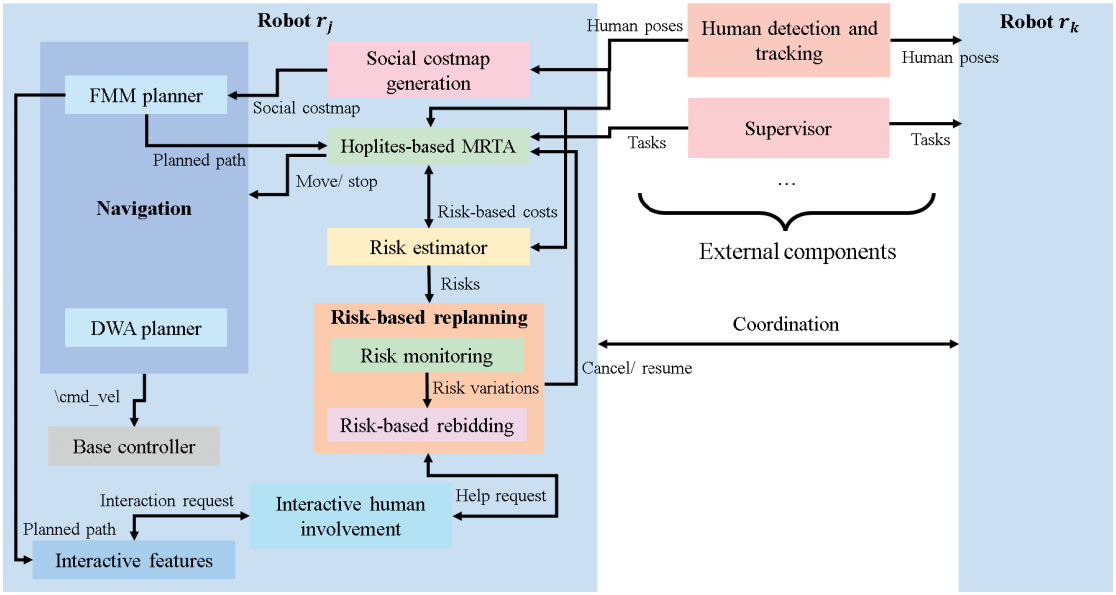


Figure 4.1 – Schematics of the software modules constituting our proposed framework.

additional information sharing block for communicating the perceived human information among team members.

The same code is running on the robots in both simulation and reality. In general, the difference between the real and simulated tests lies in human detection and tracking, ground truth acquiring methods and of course robot perception, actuation, and computational power. We note that the blocks depicted in Figure 4.1, contain internal sub-blocks and this figure shows an abstract view of the main components in our system.

5 Robotic Platforms

THE robotic platforms used in our research will be introduced in this chapter. Apart from the work presented in Chapter 11, the rest of this research has adopted the Mbot as the robotic platform for method validation. Specifications of each robot can be found in the following sections.

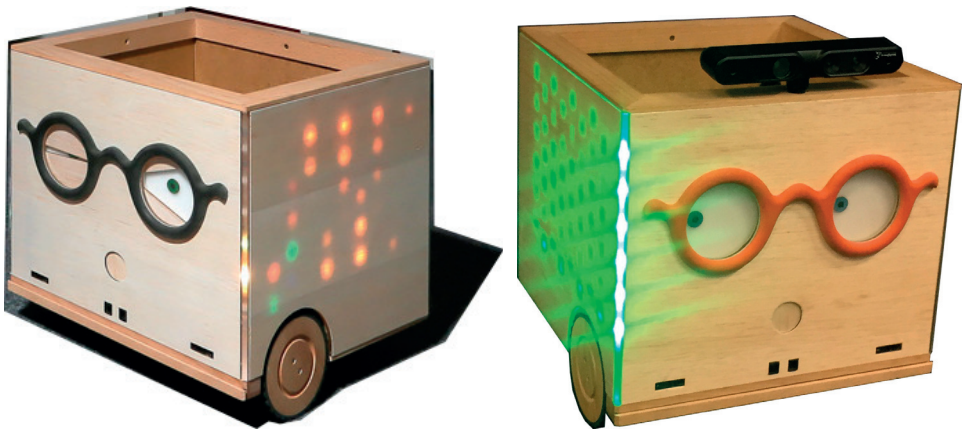
5.1 Ranger Robot

The Ranger [100] is a simple robotic platform designed to interact with children (see Figure 5.1). It is a small robot with limited perception, actuation and computation capabilities. It is inspired by a common object found in many children rooms: a wooden storage box for toys. This box is augmented with robotic capabilities in order to interact with children and motivate them to tidy up their room [101]. This robot has been designed at the Laboratoire de Systèmes Robotiques (LSRO) of EPFL.

The Ranger robot has a body based on a wooden box and is equipped with wheels, mechanical eyes, inertial sensors, three infra-red distance sensors, ground sensors, a bumper, an inside balance sensor, capacitive external touch sensors, LED panels behind the wooden surface, sound, eyeglasses, a detachable pacifier and a relative positioning system to detect other robots. The robot has a square footprint of 30 cm wide and it weighs approximately 3.5 kg.

Because of its customized morphology, autonomous navigation with this platform is a challenging problem for two reasons. First, its kinematic configuration consisting of two differential wheels in the front, two castor wheels in the back, integrated in a squared shape footprint, make the motion planning complicated. Second, the obstacle detection capabilities are limited to the three infra-red sensors placed in the front part of the robot. The detection range of the sensor placed in the middle is $0.2 - 1.50\,m$, while for the other two, it is $0.04 - 0.30\,m$, all of them having a very small Field Of View (FOV). For this reason, we have augmented the sensing capabilities with a depth camera *Primesense Carmine 1.09*, placed in the front upper part of the robot (see Figure 5.1b). This depth sensor has a resolution of $640 \times 480\,pixels$, a horizontal FOV of $1\,rad$ ($\simeq 57°$) and a nominal range of $0.35 - 1.4\,m$.

The Ranger has four processors: three micro-controllers, connected using a CAN bus, which manage the real-time part of the robot, and a full embedded computer running Linux Ubuntu 10.04. The ASEBA framework [102] is used to control the whole system, making the link with the higher-level controllers.



(a) (b)

Figure 5.1 – The Ranger robot. a) Complementary gestures added to the robot for increasing navigation legibility. b) A Ranger augmented with an RGB-D camera to detect humans and obstacles. The picture illustrates the interactive features of the robot. The activated LEDs on the robot's right side and the pupils indicate that the robot is going to make a right turn.

These high-level controllers are executed on a separate laptop placed on top of the robot, running Linux Ubuntu 14.04.

5.2 Mbots

The Mbot [103] is a four-wheeled omni-directional drive robot with an approximately round footprint of 0.65 *m* in diameter and a height of 0.98 *m* (see Figure 5.2). It is endowed with two Laser Range Finders (LRFs) , on both the front and the back for providing 360° coverage. It has been specifically developed within the FP7 European project MOnarCH[1]. Two batteries give it an autonomy of approximately five hours, depending on the usage. The robot has two PCs inside its shell: one manages the sensors and actuators for navigation, while the second one is available for other purposes such as human-robot interaction functionalities. The two on-board PCs run Ubuntu desktop 12.04 and the integration middleware is ROS Hydro. Table 5.1 summarizes the sensors, actuators and other robot design features.

The two LRFs are used for mapping, navigation, and obstacle avoidance. We use both of them simultaneously and the resulting laser point cloud consists of the union between the latest scan of each LRF, with the appropriate coordinate transformation to the common robot body reference frame. There are four of these robots available in total in our two experimental facilities. We note that these robots are networked and are able to communicate over WiFi.

[1] http://monarch-fp7.eu/

Table 5.1 – Mbot robotic platform specifications. DF: Design Features, NLS: Navigation and Localization Sensors, PIS: Perception and Interaction Sensors, ES: Environmental Sensors, LLSS: Low-Level Safety Sensors, LA: Locomotion Actuators, and IA: Interaction Actuators.

Type	Specification
DF	-Robot kinematics: omnidirectional, 4 mecanum wheels -Robot weight: 24 Kg (with batteries) -Payload capacity: 30 Kg -Maximum translational speed: 2.5 m/s -Maximum rotational speed: 600 $^\circ/s$ -Maximum acceleration: 1 m/s^2 (low-level programmed) -Emergency stop deceleration: -3.3 m/s^2 (low-level programmed) -Mini-ITX computer board with CPU, RAM, and SSD -Batteries: supports up to 4 batteries at the same time; Capacity: (12V) 17-20 Ah 5.5 kg each; autonomy: 4 to 6 hours
NLS	-Inertial Measurement Unit (IMU): MPU6050 for orientation estimation -Two 2D laser range-finders: Hokuyo URG-04LX-UG01 for mapping, localization, and obstacle avoidance -12 sonar sensors: Maxbotix EZ4 for obstacle detection -Depth camera: Asus Xtion for obstacle detection, space geometry analysis -Omnidirectional bumper -Four ground sensors -One RFID reader -One Hagisonic StarGazer localization sensor -Eliko's Kio ranging UWB transceivers based on a Decawave chipset
PIS	-Depth camera with microphone (Kinect type): Asus Xtion for interaction, people, and gesture recognition -Microphone array: Asus Xtion for sound feedback for natural user interaction -10" Touchscreen (or tablet) for user feedback on specific contents -Capacitive sensors for user feedback on specific parts of the robot
ES	-Temperature sensor -Humidity sensors
LLSS	-Sonar sensors -Internal temperature sensors -Motor current sensing -Bumper ring switches
LA	-Four Mecanum wheels -Four Maxon RE 35 90W 15V motor with a Maxon GP 32 HP 14:1 gearbox -Encoder HEDS 5540 with 500 pulses for providing an omnidirectional locomotion system
IA	-10" monitor with touchscreen (or tablet) for interaction with displayed contents -Video projector (pico type) for projection of contents -Arms and head servo motors for human-robot interaction -Body LED lights for showing robot expressions -Stereo speakers for content playback and robot communication -Three servo motors to actuate two robot arms and a head

(a) (b) (c)

Figure 5.2 – a) The 4 Mbots available at DISAL. b,c) HRI features of the Mbot.

Summary

In this chapter, we introduced Ranger, a resource constrained mobile robot designed for human-robot interaction studies. Despite its interesting interactive capabilities, limitations of the robot in terms of perception, actuation and computational power convinced us to focus on a more powerful platform for human-aware cooperative navigation. Mbot, the main robotic platform used in this research, was presented in this chapter. This robot has been designed in the context of the MOnarCH project for interacting with children and operating in a social environment, namely, the pediatric ward of the Instituto Portugues de Oncologia de Lisboa (IPOL).

6 Tools

PERFORMANCE evaluation and method validation are integral parts of any research. Methods should be examined by means of repeated tests in different settings while varying different parameters of the problem. For this purpose, we have conducted experiments of increasing complexity to test our developed methods in both simulation and reality. In this chapter, we present the corresponding tools used in our research. We start by describing the software framework and the modules providing a number of basic functionalities for the robots. The simulation framework is subsequently detailed in Section 6.2. Additionally, different human detectors and trackers evaluated and used in the context of this research are introduced. The main reason for detailing such human perception tools is sharing our findings regarding each method in the specific problem of human-aware multi-robot navigation. Only a subset of these methods have been eventually used in our experiments. Nonetheless, listing the strengths and shortcomings of these methods can provide useful information for a reader interested in real human perception solutions for similar problems.

6.1 Software

Several software frameworks have been used in this work in order to facilitate the implementation, validation, and evaluation of the proposed methods. In particular, ROS and the high-fidelity robotic simulator Webots have been used. Moreover, a number of functionalities and behaviors that have been made available through the shared code-base of MOnarCH will be briefly described in the following sections.

6.1.1 ROS

The backbone of all the code running in our system is ROS. ROS is an open-source, meta-operating system for robots providing operating system services such as hardware abstraction, low-level device control, implementation of commonly-used functionalities, message-passing between processes, and package management. It comprises a set of libraries and tools with the aim of facilitating the development of robotic applications and promoting software reuse. The main idea behind ROS is that functionalities in a robotic application can be broken down into separate processes running in parallel. Each process, called a “node” is implemented in a separate program. Nodes within ROS can communicate through the internet protocol suite, commonly UDP/IP. A publisher/subscriber architecture

enables this communication where nodes publish/subscribe to “topics” for sending/receiving data.

6.1.2 MOnarCH Software Stack

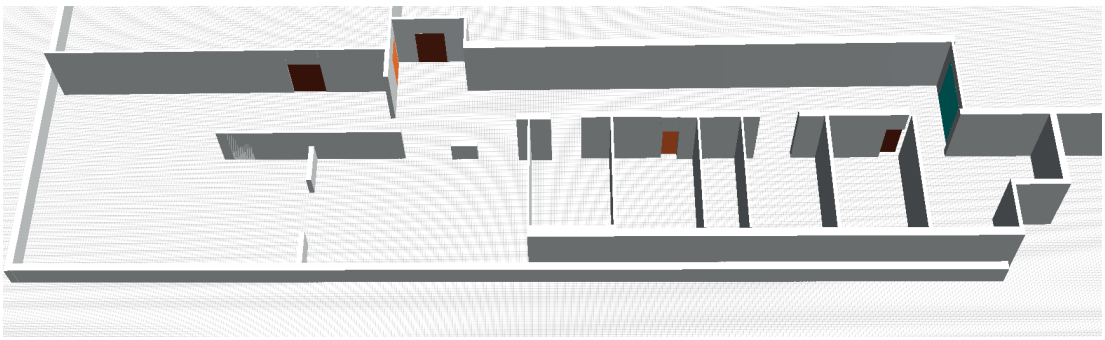
As Mbots were the robotic platform designed for MOnarCH, a number of researchers with different research focuses have been contributing to the functionalities available for these robots. A few of the relevant functionalities will be mentioned in the following:

- **Navigation:** autonomous navigation for a single robot relying on FMM, DWA, and Adaptive Monte Carlo Localization (AMCL) by means of LRFs and odometry.
- **Human perception:** human position estimates provided by a human detector and tracker based on a combination of background-based and Histogram of Oriented Gradients (HOG)-based detectors. The output of multiple detectors are fused to provide a global set of human and robot positions in the environment. This is achieved using an external omni-directional overhead camera.
- **HRI:** the programming interface for low-level interactive features of the robot such as head and arm motion, LEDs for the mouth, the cheeks and the footprint, robot voice, and touch screen are made available through designated topics. The desired interactive behaviors can be achieved when publishing the right data on the corresponding topics. The functionalities implemented for the main features adopted in this research will be detailed in the following.

    1. Lights: the Mbot is equipped with LEDs in the cheeks, eyes and the footprint. We can control these LEDs for showing either (i) a single color, (ii) a smooth transition between different colors, or (iii) blinking between different colors.
    2. Arms: robot arms can be controlled either by setting (i) a desired final position or (ii) an oscillation behavior in order to mimic the arm swing in the human walking pattern.
    3. Head: for creating a gaze gesture, we control the head motion of the robot. This can be done by (i) setting a desired rotation angle relative to the reference point of the head joint or (ii) a rotation relative to the robot’s current position in order to mimic gaze.
    4. Mouth: different patterns can be achieved using the mouth LEDs. A number of sentiments (being sad, happy, scared) have been considered and up to three mouth patterns resembling those sentiments are generated. Upon requesting a desired sentiment, a pattern from the corresponding group is selected randomly.
    5. Speech: the robot can vocalize custom messages that are either generated real-time or pre-recorded, with different voice options in terms of type (robot voice, child voice, etc.), gender, and tone, with an adjustable output volume. The volume of the robot voice is adjusted based on the distance to the human.

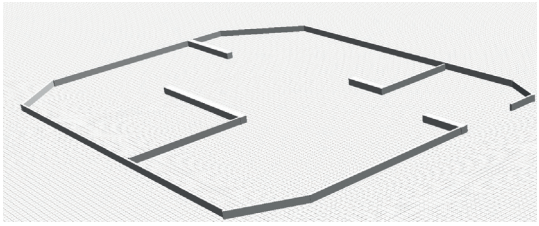
6.2 Simulations

The use of high-fidelity simulators such as Webots [104] is fundamental, especially when considering multi-robot systems. Such a simulator provides a tool to perform repeated experiments under controlled conditions and also perform long-term experiments that are very expensive to have in reality. Webots is a realistic, sub-microscopic, physic-based simulator that allows for perfect integration with ROS,



(a)

(b)



(c)

Figure 6.1 – Snapshot of the Webots world for the environment at a) IPOL, b) DISAL robotic laboratory and c) Jordils experimental arena.

enabling the same algorithms to run both in simulation and on the real robots. As a result, a smooth migration from simulation to reality is made possible using the exact same code.

We have developed models of the environments that we use for our experiments (see Figure 6.1) and a model of the Mbot to match as much as possible the shape and the geometry of the real robot, namely weight, height, center of mass, position of the sensors with respect to the base frame of the robot and the position of the wheels. Additionally, calibrated models of the LRF, mecanum wheels and motors have been added. These data were gathered directly from the real robot. Much effort in terms of optimization of computational costs has been dedicated to dealing with the challenges of faithful and real-time simulations of this rich set of sensors in complex environments, particularly in the presence of multiple robots. As a result, our simulations have similar environmental richness to the real world experiments. This is key, for evaluating the performance of different methods for an MRS targeting social, dynamic and noisy environments.

Regarding simulation speed, all the scenarios with up to two robots can achieve 4-5×real time speed and the scenarios with four robots can run in real time. In all of our scenarios the time step has been set to 64 *ms* for the simulation step and 100 *ms* for the LRFs which represents the actual update frequency of the real device.

We note that one key choice made in this thesis was to not rely on a simplified model of the human

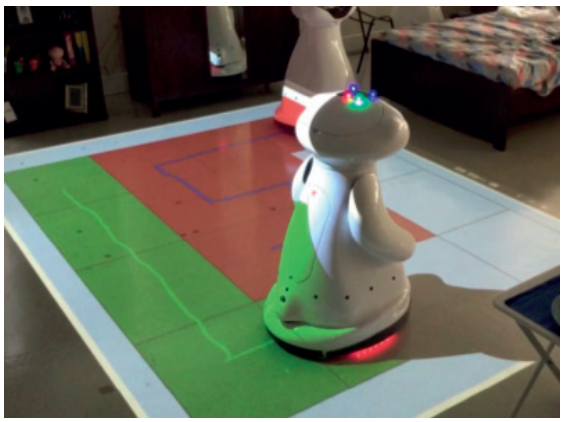


Figure 6.2 – Snapshot of the Mbot being tracked using the active marker.

behavior as we are targeting uncontrolled social environments that can be highly stochastic with sudden behavior change on the human side. Additionally, we did not want to be bound to a limited set of human trajectories collected in a specific environment for our experiments. Instead, we rely on a pool of human trajectories that can be (i) manually generated based on the knowledge of the designer about how humans can move in a given environment (these trajectories can be very diverse as any sequence of way-points can be used to construct them), (ii) automatically reproduced through the possibility of playing back real ROS bags with recorded human trajectories in a specific environment.

6.3 Human Detection and Tracking

Throughout this research, we have tried a number of methods for acquiring human pose estimates using off-board and on-board perception, as this information is a key input for all of our methods and, additionally, for enabling ground truth measurements. A brief description of each method along with its strengths and shortcomings can be found in the following sections.

6.3.1 Off-Board Solutions

In this section we briefly describe the human detection and tracking solutions that are external to the robots.

Omni-Directional Camera

This is the main perception module developed in the MOnarCH project; detailed in Section 12.1. The maximum error across an area of 8×5.5 m^2 corresponding to our robotics laboratory arena (see Section 7.1 for more details) is in the order of 0.2 m for this tracker. However, orientation estimates of the tracker can have large errors, and we observed delays in the tracker output for dynamic human targets. Moreover, as this is an external tracker, it has a limited coverage that will not be sufficient for environments with multiple rooms.

Overhead Camera

We looked into two marker-based tracking solutions for using overhead cameras, Swistrack and OpenCV-based blob detection. SwisTrack [105] is a powerful software for tracking mobile targets in 2D. It is particularly effective for tracking small ground robots. This marker-based approach performs well in controlled environments with a sufficiently large distance between the camera plane and the marker plane. However, it has limited operating area for tracking a human target (1.7 m tall on average) in a standard room, with the marker placed on the human's head. Despite its accurate positioning, we were not able to adopt this solution in our experiments for human tracking as we needed to have more than 5 cameras to cover a volume of $5.5 \times 8 \times 2.8$ m^3 corresponding to our robotics laboratory arena (see Section 7.1 for more details).

In an attempt to overcome the problems of the original SwisTrack, a simple active marker was developed at DISAL (by Lorenzo Sarti), consisting of a 15×20 cm^2 board with four LEDs (see Figure 6.2). This tracking method is based on simple color blob detection functionalities provided by OpenCV. This tracker has been used for acquiring ground truth measurements in Section 17.3. However, this approach has similar shortcomings to SwisTrack. Changes in the lighting conditions affect the performance of this method. Moreover, in rooms with shorter ceilings or with taller targets, the coverage of the cameras for tracking this marker largely decreases making this solution not scalable, as more cameras will be required for providing the same coverage. We note that this marker should be placed on top of the target to be tracked to minimize the likelihood of occlusions.

A 3D version of SwisTrack was necessary to resolve the problems of the previous two approaches. As this step required substantial development and testing effort, we opted for an existing 3D solution, the Motion Capture System (MCS).

Motion Capture System (MCS)

MCS is a commercial tool for recording the movement of designated targets within an experimental arena with very high accuracy. The MCS we use in this research (see Section 7.2 for more information) is a marker-based solution provided by Motion Analysis Inc. that provides us with millimetric scale accuracy. We rely on this system for ground truth measurements, human tracking and emulated local perception of the robots in Chapter 16 onward. However, external perception limits the robots in terms of coverage. Moreover, the very high accuracy of this system comes at a high purchase price, resulting in a solution that is suitable for research purposes only.

UWB Transceivers

UWB is an emerging technology in the field of indoor localization, mainly due to its high performance in indoor scenarios and relatively easy deployment. However, in complex indoor environments, its positioning accuracy may drastically decrease due to the biases introduced when emitters and receivers operate in Non-Line-of-Sight (NLOS) conditions. This undesired phenomenon can be attenuated by creating, a priori, a map of the measurement error in the environment, that can be exploited at a later stage by a localization algorithm. In an attempt to use this technology for human tracking, we proposed the leveraging of mobile robots in order to automatize the UWB calibration process, based on the work of Prorok et al. [106]. The purpose of this method, detailed in Appendix A.1, was to improve the performance of UWB-based people localization in a realistic indoor environment.

In this approach, the error map was the result of a calibration process, which consisted of collecting several measurements of the localization system at different locations in the environment. We demonstrated how the accuracy of an UWB Real Time Location System (RTLS) can be significantly improved through a fingerprinting-based calibration method in complex indoor environments. Moreover, we showed that this method can be effectively implemented using a mobile robot able to autonomously scan the environment. Adopting this method improved the mean localization error of our system by roughly 50%. In particular, this method was shown to be very effective for the rejection of highly inaccurate measurements that have an error greater than 0.5 m. More information about our UWB-based human tracking method can be found in [107].

Despite promising results of this approach, additional experiments performed with multiple human targets showed that as the number of humans in the social environment increases, some measurements can largely differ from the collected error map[1]. Therefore, this method should be further improved for achieving satisfactory levels of accuracy needed in uncontrolled social environments. As a result, we did not adopt this technology in our final experimental setup. Nonetheless, we believe this is an interesting and promising research area with many applications for indoor positioning in human-populated environments.

6.3.2 On-Board Solutions

In this section we briefly describe the human detection and tracking solutions that are performed locally on the robots.

Laser Range Finders for Leg Detection

This is a simple leg detector developed by Arras et al. [108], that uses two dimensional range scans. In their approach, supervised learning is used for creating a classifier that simplifies the detection of people and *Adaptive Boosting (AdaBoost)* is applied for training a strong classifier from simple features of groups of nearby beams corresponding to legs in the range data. Based on our experiments presented at Section 11.2, this is a fast and effective detector. However, it can report false positive detections when faced with objects resembling the shape of human legs such as ripples on a curtain. Orientation estimation using this approach, is based on the velocity vector, assuming a forward walking motion. For this reason, the output of this detector should be fused with that of another detector for having reliable human pose estimates.

Kinect with Low-Lying View Point

This tracker is designed for small-footprint ground robots, developed by Pesenti et al. [109]. To explain their method in short, the point cloud is initially down sampled and after floor detection, potential leg candidates are formed. Thereafter, a trained SVM classifier based on HOG features, calculates the probability of each candidate being a leg. Additionally, tracking is done using a KF. This is a robust tracker with a few false-positive detections that is computationally more expensive than the laser-based leg detector introduced in Section 6.3.2. Thus, we have fused the detections of these two methods for human perception in Section 11.2.

[1] Refer to this `link` for more information

Kinect: Nite

This tracker comes from a ROS meta-package called *cob_people_perception* from the Care-O-Bot Research [2]. It is easy to set up and besides the skeleton tracker, it has a number of other modules for leg detection and data fusion. In this work, we have only used the Nite detector and tracker. Tracking starts by segmenting the scene into background and humans, and then detection of the human skeletons takes place after a calibration phase. We note that any moving object will be detected as a potential human in this approach. This tracker has not proven to be very robust in our tests. It works well when people are facing the camera, in a specific range while holding a basic standing position, but for more complex cases, the speed of tracking can significantly decrease. We believe this is caused by the calibration phase needed to start skeleton tracking for the humans. We observed that the calibration time was variable, and depended a lot on the human's pose (e.g.,standing, walking, etc.). Hence, it might become problematic for cases where the humans are walking, or quickly crossing the robot's FOV.

Kinect: OpenPose

OpenPose is an open source software developed by the CMU Perceptual Computing Laboratory[3], for detecting body, hand, and facial keypoints (a total of 130) of multiple humans on an image. It works on images or videos, and can be used in real-time. It makes use of Deep Neural Networks (DNN) and requires GPU acceleration for improved performance due to its high computational cost. The details and performance comparison of this detector with the Nite tracker can be found in a semester project report following the address in the footnote [4].

OpenPose has shown great performance, accuracy and robustness in our assessments. It works at multiple scales and in cases with partial visibility since it is also able to detect a subset of keypoints. Additionally, a *confidence* score is provided for each keypoint on top of the position estimate, quantifying the detection uncertainty. Compared to the Nite tracker, OpenPose only needs one image for detecting a human, and no calibration is required. Therefore, the time taken to start detecting a human is only dependent on the computation time required to process an image. However, the computation time can be a problem as the forward pass through the deep network is computationally costly, and requires high-end GPU to process images rapidly. Another consideration is that it works on RGB images only (although they have started to include RGB-D support recently) without the depth information given by the Kinect. Thus, the results are 2D position estimates in the image frame, and further processing is required to have 3D detections. Finally, OpenPose only performs human detection without tracking.

Despite our effort for using RGB-D images in OpenPose to get 3D pose estimates for the humans, we did not adopt this method is our experiments in the end, due to the hardware limitations of the Mbot. Since, there is no GPU included in the robot, upgrading the robot hardware, OS and firmware required more time than available. Nonetheless, we believe this is a very effective tool for on-board human pose estimation.

[2] Fraunhofer IPA, Care-O-Bot Research, by Fraunhofer IPA

[3] https://github.com/CMU-Perceptual-Computing-Lab/openpose

[4] https://disal.epfl.ch/teaching/students_projects/AY_2017-2018/DISAL-SP111

6.4 Discussion

Although the tools we introduced in this chapter provide us with satisfying performance, reliability and fidelity to carry out our experiments, like any other tool, they are not perfect. Robot functionalities and more so, on-board human detection and tracking can be further improved to allow for more efficient human-aware navigation and human motion prediction. Moreover, the simulation-to-reality gap inherent to any simulated system, can be further reduced by means of more precise calibrations. Furthermore, larger data sets of human trajectories and human interactions in the real environment can help to build a more diverse pool of human trajectories to be used in simulations.

Summary

This chapter presents the software tools used to validate the work of this thesis. ROS and Webots frameworks are leveraged for implementation, validation, and evaluation of our algorithms. Additionally, the basic available functionalities for the Mbots in the context of the MOnarCH project, and human detection and tracking tools used in this research are described in this chapter. MCS is the final perception tool adopted for ground truth measurements and emulating on-board perception in this thesis due to the millimetric scale accuracy that it provides.

7 Experimental Facilities

REAL-WORLD experiments are fundamental for validation of algorithms in robotics research. This calls for having a reliable setup that enables executions of repeated experiments and provides ground truth for method evaluation. The indoor environments in which our experiments have been performed along with the experimental setup are described in this chapter. The two main facilities where experiments have been conducted, the DISAL robotics laboratory and DISAL motion arena at Jordils, will be detailed in the following sections.

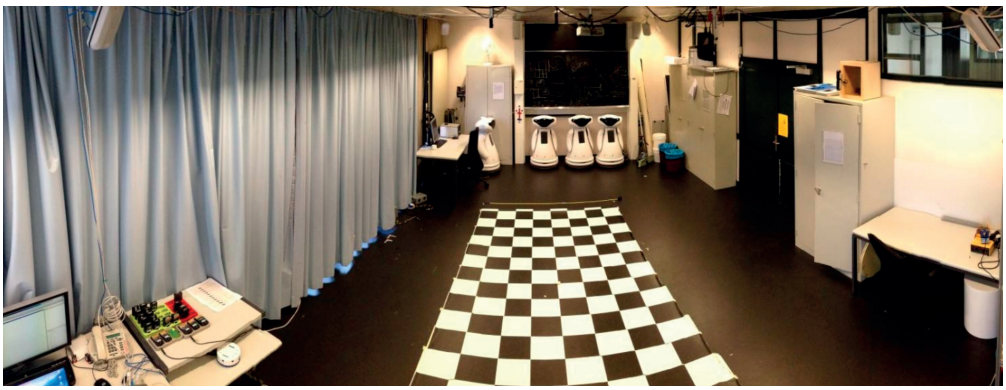
7.1 Robotics Laboratory

Real-robot experiments in Chapter 11-15 have been conducted in the environment depicted in Figure 7.1a. The setup used for our real-world experiments at this facility, composed of an overhead camera and an Ubuntu computer that receives the image stream and processes these data for computing human poses and ground truth, is illustrated in Figure 7.1b. The characteristics of the arena and the camera used for target tracking are given in Table 7.1. We note that the reported size of the arena in the Robotics Laboratory corresponds to the dimension of the whole room since we were targeting a realistic human-populated environment. For the Jordils facility however, we are exclusively reporting the size of the motion arena.

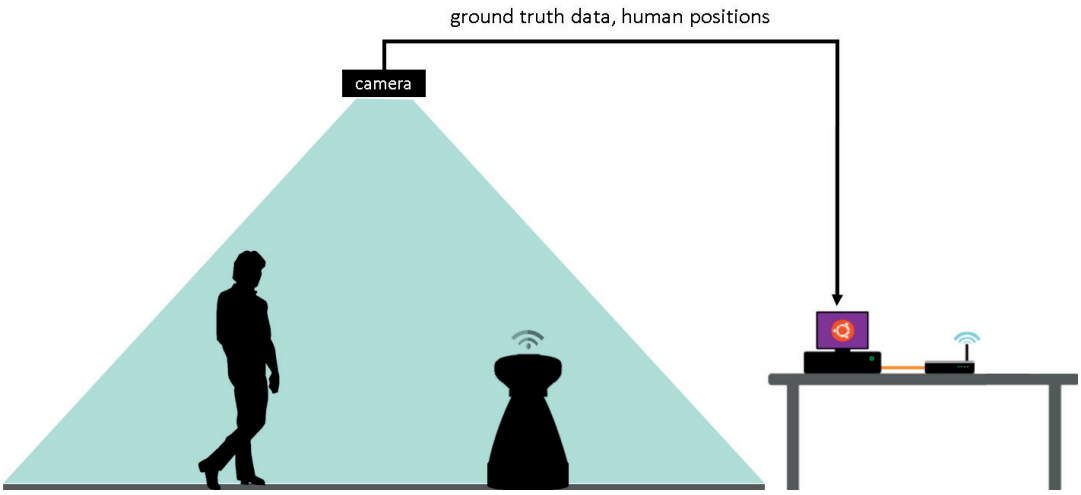
In our setup, we use 1) a GigE color camera (Basler-SCA1000-30GC) which has a standard resolution of 1032×778 pixels and 30 fps, mounted 2.5 m above the robotic arena (used with the OpenCV-based tracker introduced in Section 6.3.1 for experiments in Chapter 11). 2) A PoE omni-directional camera (VIVOTECH Supreme FE8174/74V) with 1080p Full HD resolution and 30 fps, mounted above and in the center of the arena (used with the HOG-based tracker introduced in Section 6.3.1 for experiments in Chapter 12).

Table 7.1 – Characteristics of the experimental arenas.

Name	Length	Width	Height	Number of cameras
Robotics Laboratory	8	5.5	2.8	1
Jordils' motion arena (map I)	11.5	7	5	28
Jordils' motion arena (map II)	9.5	7.5	5	28



(a)



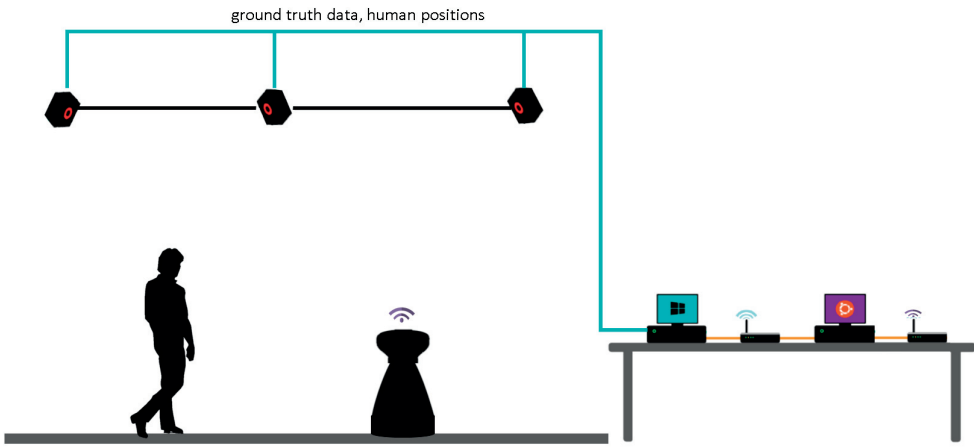
(b)

Figure 7.1 – a) Snapshot of the DISAL robotics laboratory experimental arena. b) Illustration of the experimental setup at DISAL.

7.2 Jordils' Motion Arena

Real experiments in Chapter 16 onward, have been performed in the arena depicted in Figure 7.2. The experimental infrastructure is composed of a control room, a reconfigurable arena consisting of modular building blocks, and a network of cameras anchored to a structure at the ceiling level of the arena to track retro-reflective sphere of 25 mm diameter. In the control room, there exists a Windows computer running the MCS software that performs the pose estimation for the targets, using their corresponding marker sets. Pose estimates are multi-casted into a network at 100 Hz. Another Ubuntu machine, receives these packets of data through the network and using an SDK for communicating with the MCS software, publishes the pose information as a topic in the dedicated MOnarCH network. This time-stamped topic can be recorded in the form of a rosbag or in the log format as the ground truth. The characteristics of the arena and the cameras used for target tracking can be found in Table 7.1.

(a)



(b)

Figure 7.2 – a) Snapshot of the motion arena at Jordils. b) Illustration of the experimental setup at Jordils.

Summary

In this chapter the indoor environments where we have conducted our real-robot experiments have been detailed and the infrastructure and the setup in each facility are described. Initially, we have performed real experiments at the DISAL robotics laboratory. We then moved to the Jordils facility for doing more complex experiments with multiple robots and multiple moving humans. This choice was motivated by the larger area, flexibility of the static map, and more importantly, the accurate human positioning and ground truth that was necessary for our method evaluations.

8 Conclusion

THERE exist a number of key components required for implementation, validation and evaluation of methods proposed in this thesis. These components consist of the hardware platforms, software frameworks, human detection and tracking methods, and the experimental setup, all introduced in this part. Although the current system has satisfying performance, there still exist limitations in terms of hardware and further improvements can be made to the simulation and human positioning tools. Among the various human detection and tracking methods and technologies, we contributed to a UWB-based human localization approach using mobile robots for automatic calibration. Despite promising results in a realistic indoor environment, further studies with multiple humans showed how human bodies can modify the fingerprints due to signal attenuation. This resulted in large errors that were not captured by the underlying error maps, and using a single tag, the current approach could not achieve accurate results. Thus, we refrained from using this method for ground truth acquisition and opted for an MCS instead.

Part III Human-Aware Navigation

9 Introduction

HUMAN-AWARE mobile robots are expected to exhibit behaviors that differentiate them from human-agnostic robots who treat people merely as obstacles. In order to generate human-aware autonomous navigation, we have taken an incremental approach of initially enabling basic autonomous navigation and augmenting it with human-awareness consecutively. Navigation demonstrates human-awareness by modifying the human-agnostic plans given to a robot by means of adopting a socially-aware path planner. Such a planner should take into consideration individual people and possible social interactions taking place between them when computing the optimal path. This information is provided through perceptual data and is translated to social costs using models such as social costmaps.

Since individual human-aware navigation is the first step towards having cooperative human-aware navigation for multi-robot systems, Part III of this thesis will investigate the challenges and solutions for building this functionality. In Chapter 10, the underlying navigation components and the social costmap model are introduced. A case study targeting Ranger, a resource-constrained robotic platform will be presented in Chapter 11. When robots are deployed in real complex environments, having perfect information about the position of the humans, which is a key assumption of many human-aware navigation methods, is too simplistic and not robust to non-negligible perception uncertainty. In Chapter 12, we attempt to model one essential aspect of human-aware navigation that has been overlooked in this area, uncertainty in perception. This is achieved by adopting perception methods with non-deterministic human position estimates and expectation-based social costmaps.

10 Single Robot Human-Aware Navigation

THIS chapter describes the details of single robot navigation and human-aware path planning for our two robotic platforms. There are three main components required for enabling a robot to autonomously navigate in an environment: mapping, localization, and motion planning which is composed of path planning and obstacle avoidance. The principles of each component and their corresponding implementation details will be explained in the next sections.

10.1 Ranger Navigation

In this section basic navigation components of the Ranger will be explained.

10.1.1 Mapping

Map in the context of navigation denotes any one-to-one mapping of the world onto an internal representation for the robot. For this purpose, a spatial model of the environment surrounding the robot is obtained using the Kinect sensor and robot odometry. This spatial model will be used for localization and navigation in later stages of navigation. The map is created by applying a grid-based Simultaneous Localization and Mapping (SLAM) method using Rao-Blackwellized particle filters [110] implemented by the *gmapping* package of ROS.

10.1.2 Localization

Robot localization refers to the robot's ability to establish its own position and orientation within the frame of reference. This is achieved by means of AMCL [111] that is a Particle Filter (PF)-based localization method, implemented by the ROS *amcl* package. It tracks the pose of the robot and provides localization using a given map, odometry and a single row of the depth readings typically obtained by a laser scan. In the Ranger, this laser scan is the result of converting the Kinect's depth image to laser scan data format. We have converted the readings of the depth camera to laser-like readings, by taking the closest depth reading from each column in the 10 central lines of the depth image.

10.1.3 Motion Planning

The motor commands required to take the robot to its final goal without colliding with static or moving obstacles are provided by a global and a local planner. We associate a *costmap* to each of these planners. A costmap assign a cost to each cell of a given spatial map. The details of the navigation approach used here, are explained more extensively in [112]. The global planner also referred to as the path planner, uses an A*-search [1] to find the optimal path to the goal. The local planner responsible for obstacle avoidance is based on the DWA [35].

10.2 Mbot Navigation

The robot navigation is based on the navigation system used in the MOnarCH project [113], detailed in [30]. For understanding the underlying components for autonomous navigation of the Mbot, a brief description of these modules can be found in the following.

10.2.1 Mapping

Similar to Section 10.1.1, we rely on SLAM through the use of ROS *gmapping* package for constructing a static map of the environment based on the laser scan and odometry information.

10.2.2 Localization

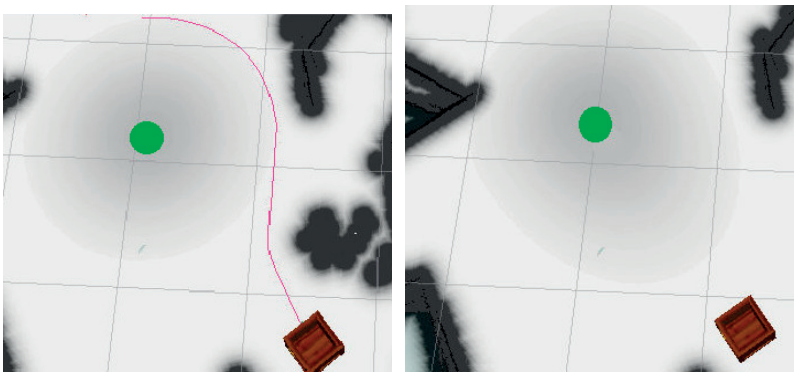
ROS *amcl* package is responsible for providing robot localization based on odometry, laser range finder readings, and a static map.

10.2.3 Motion Planning

Mbot navigation is based on FMM [31] for path planning, together with DWA algorithm for guidance and obstacle avoidance. DWA is essentially a maximization (over a discrete set of feasible velocity candidate commands) of an evaluation function translating three guidance goals: (1) progress towards the goal, (2) clearance from obstacles, and (3) absolute speed. The guidance problem is approached as a two-step process. First, given a goal location, the robot plans its path from the current pose to the goal pose. This is the task of our FMM-based global planner. Second, the plan is executed by the robot, in real time, while avoiding unmapped obstacles. This is done by means of a local path planner based on DWA.

The potential field output by FMM is minima free and yields an optimal path from a given initial to a final goal point. It is optimal in the sense that the integral of a costmap over the path is minimal, given the initial and final points as boundary conditions. The path is the solution of a gradient descend Ordinary Differential Equation (ODE), with the initial point as initial condition. However, we do not explicitly compute a path. Instead, we compute the progress towards the goal directly from the gradient of the potential field. This FMM-based navigation method has been used in MOnarCH for the Mbots considering solely an increased cost near static map obstacles in the robot costmap, keeping the resulting paths away from the obstacles. In this thesis a *social* component is added to this costmap as well.

[1] http://wiki.ros.org/global_planner



(a) (b)

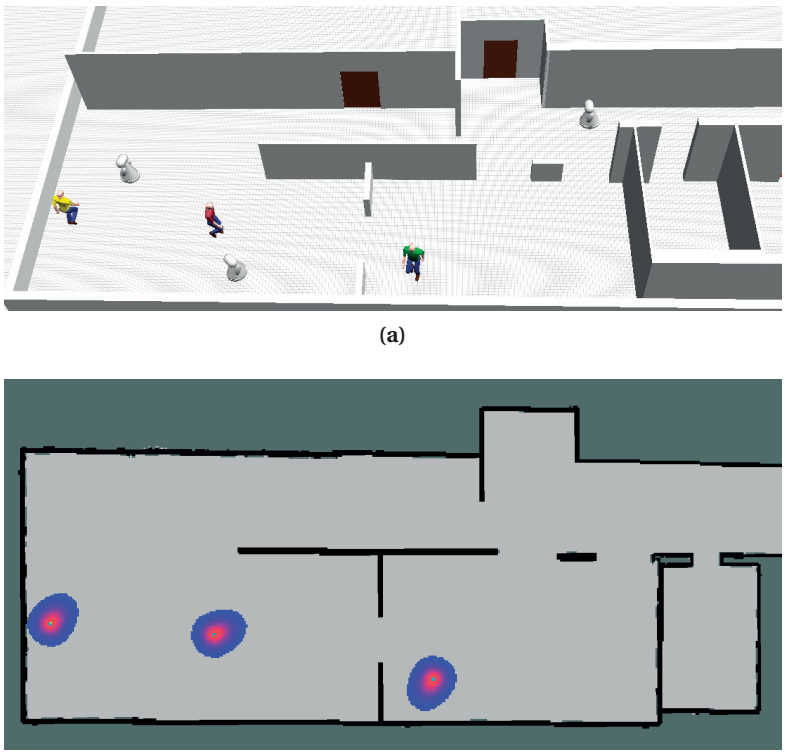
Figure 10.1 – Ranger navigation: (a) The path of the robot given by the planner when encountering a social costmap centered around a static human. (b) The social costmap of a person moving towards the robot.

FMM and DWA run asynchronously. FMM is activated when either a new goal position is given, or when the costmap changes. DWA is running in a closed loop with a fixed rate of 20 *Hz* in our experiments, using the last updated potential field from the FMM.

10.3 Human-Awareness in Navigation

In basic navigation, only the global and local costmaps used for assigning costs to areas corresponding to obstacles are taken into account for path planning. However, when augmented with human detection and tracking modules, the robot is able to have a social consideration for people when performing this task. Once the robot knows where the people are in the environment it should consider them differently from other obstacles and plan accordingly. We rely on proxemics to make this differentiation based on which we assign a social cost to the human personal space that the robot should avoid intruding. This social cost in encoded in an additional costmap that is combined with global and local costmaps for taking the final path planning decision. This social costmap is responsible for making the navigation human-aware. This is based on the concept of layered costmaps proposed by Lu et al. in [114].

We have opted for a proxemics-based cost modeling approach in spite of the shortcomings detailed in Section 2.1.4. There are several reasons that motivate this choice: (i) our proposed approach does not rely on any data collected from a specific environment, (ii) dynamic proxemics requiring the analysis of signals other than the human and robot poses, is not an integral part of our human-aware navigation approach as we only base our navigation on the pose signal, (iii) similar to numerous studies in the field, this spatial modeling allows for having a human-aware navigation behavior in the majority of cases using minimal information, *i.e.*, pose (and velocity). Nonetheless, the proposed framework is not dependent on this choice of social cost modeling and any other model such as the dynamic proxemics model or learned social costs translated into the form of a spatial costmap can easily be employed.



(a)

(b)

Figure 10.2 – Mbot navigation: example of a scenario with three moving people and three robots for: (a) Webots simulator, (b) visualization of the social costmaps in Rviz.

10.3.1 Social Costmap Model

The personal space around a human can be defined as the mixture of two pseudo-Gaussian functions, one for the front and another one for the rear part of the area surrounding the person. The orientation and heading of the person will cause a corresponding rotation in these functions in such a way that the person is always in the center and the absolute orientation of the person matches that of the Gaussian functions. A Gaussian function N, centered on p with covariance matrix Σ, is defined as follows:

$$N(q) = exp\left\{-\tfrac{1}{2}(q-p)\Sigma^{-1}(q-p)\right\} \tag{10.1}$$

q indicates the position of a point and Σ is:

$$\Sigma = \begin{pmatrix} \sigma_x^2 & 0 \\ 0 & \sigma_y^2 \end{pmatrix} \tag{10.2}$$

σ_x and σ_y are used to modulate the shape of the Gaussian and are traditionally chosen in a way to respect the personal space of a person as indicated by the proxemics principle. Various factors such as age and gender [10], and velocity of the motion [24] can influence the size of this area, however, σ_x and σ_y are commonly considered to be constant over time. Getting closer to a person, will cause an increase in the value of this function, and hence the social cost associated to that position will increase.

If the center of the costmap, which indicates the position of the person is not deterministically known,

the costmap can not correctly model the social costs and hence the social path planning could fail in finding an appropriate socially compliant path. This problem becomes much more critical in real applications where perception uncertainty can be non-negligible and robustness is vital for succeeding under different conditions. We believe probabilistic social costs can be a remedy to this problem. An extension of the deterministic social costmap, *i.e.*, what we called an "expectation-based" social costmap is introduced in Chapter 12.

We note that for the experiments done with the Ranger, the social costmap model was chosen to have a variance proportional to the relative velocity of the human, similar to the approach in [24] and the social costs were considered in an additional costmap layer in ROS. Figures 10.1a and 10.1b show how the social costmap looks like around a person in different situations for the Ranger.

As for the Mbots, the social costmap was used to augment the speedmap of the FMM. The update rate of the social costmap is chosen while taking into account the time required for the FMM to compute a plan. Additionally, we chose a constant variance for personal space modeling similar to [33] and relied on frequent replannig for computing human-aware paths (see Figure 10.2). This was mainly motivated by the human positioning errors affecting the velocity estimations in reality. Nonetheless, if sufficiently reliable velocity estimations are available the social cost model can be adapted accordingly. Moreover, despite focusing mainly on personal spaces in social cost modeling, the proposed methods in this thesis are by no means restricted to this type of social costs only. The reason is that we rely on a social costmap that can be realized in the form of a discrete array of measurements or a continuous cost function encapsulating any desired social cost.

Summary

In this chapter, navigation components of the Ranger and Mbot robots were detailed. Additionally, modeling of social costs through the use of social costmaps was explained. These components constitute the basis for our single robot human-aware navigation behavior.

11 On-Board Human-Aware Navigation for Resource-Constrained Robots

In this chapter, we introduce the human-aware navigation solution we devised for the Ranger, a robot with limited sensing and constrained maneuvering, in a structured environment. The goal of this chapter is to achieve autonomous human-aware navigation despite the limitations of the Ranger. Additionally, we aim at utilizing the interactive capabilities of the robot to improve the legibility and intent-expressiveness of its motion.

Human-robot awareness is defined as the understanding that the humans have of the aspects related to the robots such as the status, activities, locations, identities and surroundings of the robots and the certainty associated to this understanding. On the other hand, *robot-human* awareness is the knowledge that the robot has about the commands of the humans that are needed for directing its activities and also any human-delineated constraints which may require a modified course of action and disobeying the original commands. In this chapter, we focus on the combination of both aspects. Social costs form the basis of robot-human-awareness for the Ranger and interactive gestures of the Ranger aim to express the robot's intent to the human to provide human-robot-awareness. The social costmap introduced in Section 10.3.1 is used for modeling the human personal space for the Ranger. Additionally, we also define a set of metrics for human-aware navigation, reflecting comfort, which allow us to experimentally compare our human-aware controller with a human-agnostic counterpart. Systematic experiments are carried out with a real robot in the presence of a human in order to compare our human-aware navigation with a non human-aware approach.

11.1 System Architecture

Our system comprises a number of nodes running within the ROS framework. The role of the main components of this system have been detailed in Chapter 10. For understanding the connections between these components, the system architecture of the robot's software stack is depicted in Figure 11.1. "odometry" and "scan" topics are the main sensory information for the robot navigation. We rely only on the on-board sensors for mapping, localization, human detection and tracking, and navigation.

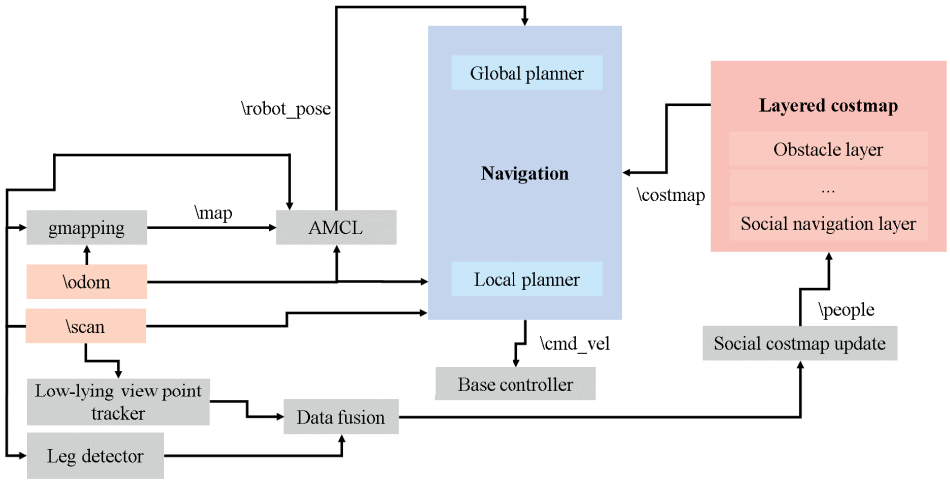


Figure 11.1 – Connections between different system components.

11.2 Human Tracking

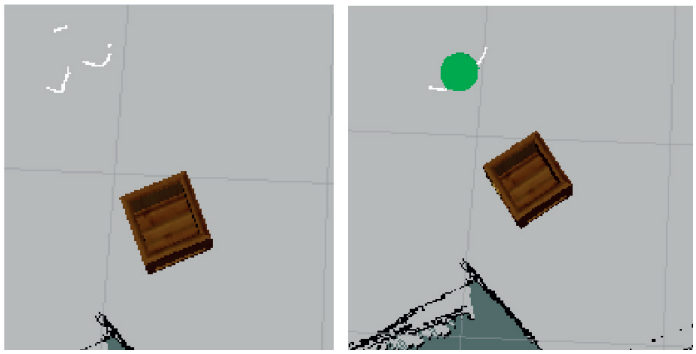
To add human-awareness to a robot capable of autonomous navigation, the robot needs to detect the people present in its surroundings. It is common in the literature of human-aware navigation that detection and tracking of the people is performed or jointly performed by an external tracking system [2]. On-board person detection and tracking becomes an even harder problem for a small-footprint mobile robot with sensors that lie close to the ground due to limited perception. The RGB-D camera of the ranger is positioned such that the robot has a low-lying viewpoint. We perform on-board detection and tracking using two different algorithms (previously mentioned in Section 6.3) that we fuse by means of a Kalman Filter (KF).

The first detection and tracking algorithm introduced by Pesenti et al. [109], is developed for small-footprint ground robots with low-lying view points and has proven to be robust for cluttered indoor environments. This tracker returns the position of the person relative to the robot and its approximate orientation. The second tracker, by Arras et al. [108], is a simple leg detector which uses two dimensional range scans. Figure 11.2a shows a sample visualization of the laser scan for this leg detector and Figure 11.2b is depicting how the final detection of the person is visualized upon leg matching and fusion.

Presenti's tracker detects and tracks people robustly with a few false-positive detections. However, in our experiments the speed of detection for this tracker was lower than that of Arras's tracker. Arras's tracker has a very high speed of detection with the down side of false-positive detections being reported in some cases due to the limited information used for detecting legs. We have fused both trackers using a KF that is initialized with the first detection of Pesenti's tracker and is updated with data coming from both trackers which have different information publishing rates.

11.3 Social Metrics

In order to evaluate the performance of our controllers with respect to social constraints, we introduce three metrics that take the human presence into consideration.



(a) (b)

Figure 11.2 – (a) Robot's laser scan when encountering a person. (b) Final detection and tracking of the person is the result of fusing the two trackers.

11.3.1 Minimum Distance to the Human

This metric (m_1) captures the minimum distance between the robot and the human during the experiment.

$$m_1 = \min_{\forall t \in T} \mathrm{D}(\vec{x}_r[t], \vec{x}_h[t]) \tag{11.1}$$

where D is a function that returns the Euclidean distance between two points. For each measurement $\vec{x}_r[t]$ obtained at time t during an experiment of length T, $\vec{x}_r$ denotes the position of the robot and $\vec{x}_h$ denotes the position of the human.

11.3.2 Time Spent in Areas Associated with Social Costs

The percentage of the time spent in areas with social costs is reported for this metric (m_2). We define the area around a human that should be avoided by the robot as a circle of radius r defined as follows:

$$r = (-2\sigma^2 \cdot log(\frac{C}{A}))^{\frac{1}{2}} \tag{11.2}$$

The choice of having a circle is motivated by the fact that the human is always static in our experiments. A is the amplitude and σ^2 is the variance used for the two-dimensional Gaussian cost function and C is a threshold used for limiting the area of the costmap. This radius is also applied in the implementation to specify the size of the social costmap.

$$f[t] = \begin{cases} 1 & \text{if} \quad \mathrm{D}(\vec{x}_r[t], \vec{x}_h[t]) \leq r \\ 0 & \text{if} \quad \mathrm{D}(\vec{x}_r[t], \vec{x}_h[t]) > r \end{cases} \tag{11.3}$$

$$m_2 = \frac{\sum_{t=1}^{T} f[t]}{T} \tag{11.4}$$

11.3.3 Mean Accumulated Social Cost

For this metric (m_3) the same social cost used by the social path planner is reported for all points in the robot's trajectory over the entire experiment.

$$\begin{cases} m_x = \mathrm{D}(\vec{x}_r[t], \vec{x}_h[t]).\cos(\theta) \\ m_y = \mathrm{D}(\vec{x}_r[t], \vec{x}_h[t]).\sin(\theta) \end{cases} \tag{11.5}$$

$$C[t] = A.exp\left\{-(\frac{m_x^2}{2\sigma_x^2} + \frac{m_y}{2\sigma_y^2})\right\} \tag{11.6}$$

$$m_3 = \frac{\sum_{t=1}^{T} C[t] \cdot f[t]}{T} \tag{11.7}$$

where θ is the bearing of the robot relative to the human, σ_x^2 is the variance of the Gaussian costmap in the x dimension, and σ_y^2 is the variance in the y dimension.

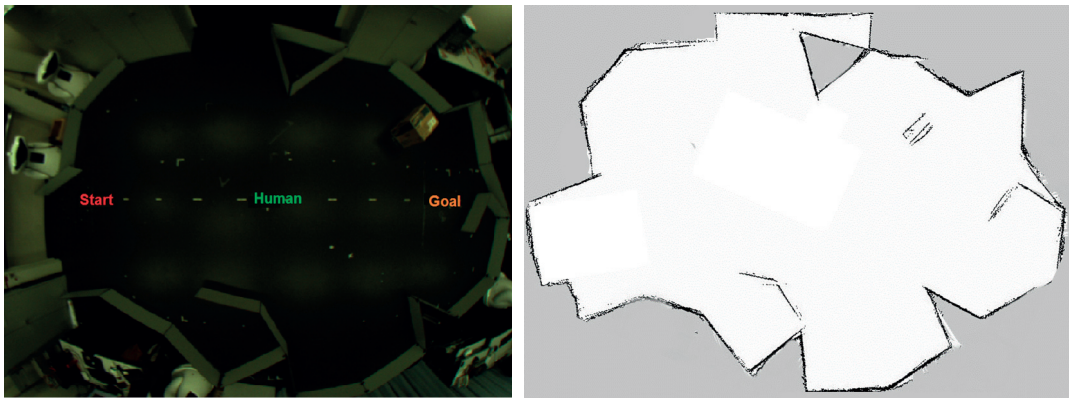
11.4 Experiments

To test our human-aware navigation in a real scenario, we did a number of experiments in the environment depicted in Figure 11.3. We used an OpenCV-based tracking system introduced in Section 6.3.1 as the provider of the ground truth to keep record of the robot pose during our experiments. The positioning error was measured to be 2-3cm in the central area of the arena and 5-7cm further away from the camera on the borders of the environment.

We investigated the performance of our human-aware navigation method in a simple move-to-goal scenario in the presence of a static human. Each experiment has been repeated 10 times for each navigation algorithm. In all experiments, the human is standing in the arena and the trajectory of the robot is recorded by tracking an active marker placed on top of the robot. The human is placed directly under the camera to minimize occlusions. The main reason for having a static human was the difficulties arising from tracking both the robot and the human with the overhead camera, given the small height of the robot that could lead to marker occlusion. The metrics used in our experiments were $m_1 - m_3$ detailed in Section 11.3.

11.5 Results

Figure 11.4 shows two sample trajectories from our experiments. The human-aware approach (blue) is clearly avoiding to enter into the areas close to the human whereas the basic navigation approach (green) favors the shortest path between the starting point to the target position while considering the human only as an obstacle to ensure safety. Figure 11.5 shows the result of our metric evaluation. As expected we can see that there is a large difference between the minimum distance of the robot to the human (m_1) in the two approaches and the human-aware approach keeps a much bigger distance. At the same time the associated costs in terms of the time spent close to the human (m_2) and the mean accumulated social cost (m_3) are also considerably higher for the non human-aware approach showing the effectiveness of our human-aware navigation.



(a) (b)

Figure 11.3 – The experimental arena. a) Snapshot of the experimental arena captured by the overhead camera providing the ground truth for our experiments. The starting position of the robot, position of the human and the goal are marked on the image. b) The map used for navigation.

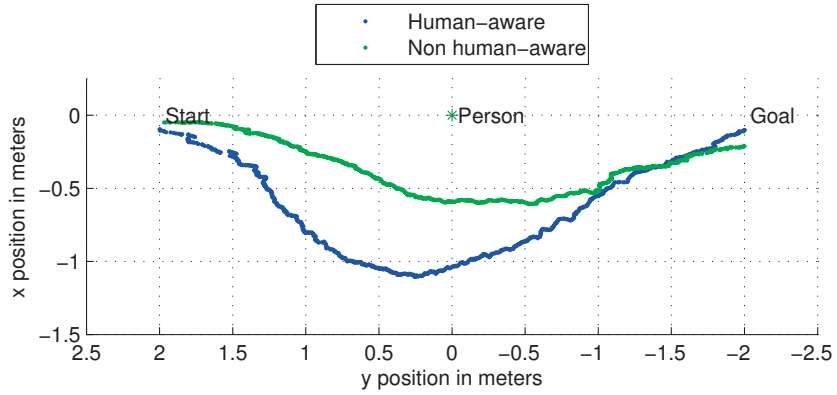


Figure 11.4 – Sample Ranger trajectories captured by the ground truth system for both navigation approaches. Positions are given in the camera coordinate system.

11.6 Discussion

Results show that human-aware navigation that differentiates between humans and obstacles is able to achieve trajectories that respect the personal space of the human and are thus more acceptable for the users. However, there were a number of difficulties for achieving the final human-aware navigation behavior. The limited range and field of view of the depth camera lead to limited perception of the environment. This affects different system components, particularly localization, resulting in a degradation of the overall navigation performance. Additionally, it is not generally very clear how the social costmaps should be preserved over time. Considering the high dynamics of the human movement, a common assumption is that the social cost assigned to the area surrounding a human is only valid as long as the human is detected and is within the perception range of the robot. However, this creates problems for robots with limited perception such as the Ranger, despite the human being static, when the person is no longer perceived. We encountered this problem during our experiments when the

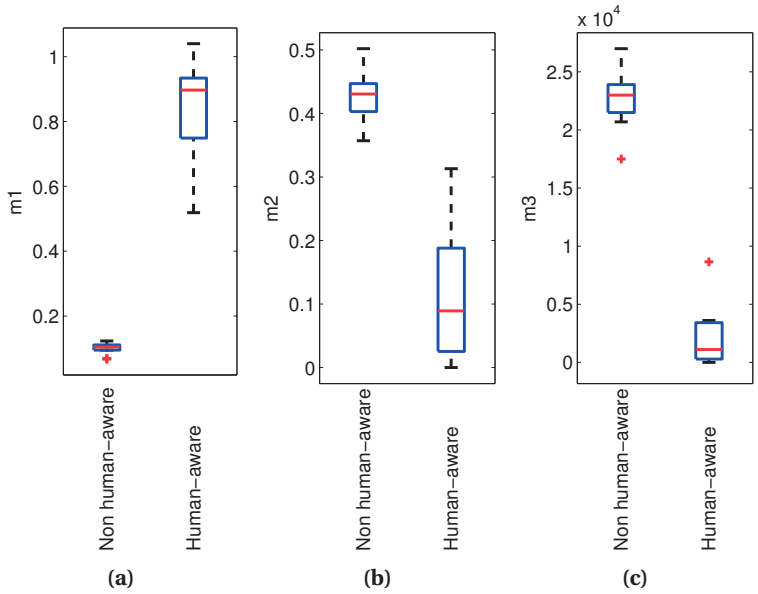


Figure 11.5 – Performance metrics for the experiments obtained from 10 runs for non human-aware and human-aware navigation approaches. a) Minimum distance to the human (m_1 in meters). b) Time spent in areas associated to the social costmap (m_2 in seconds). c) Accumulated social cost (m_1).

robot replanned upon detecting a person for moving in a socially acceptable manner. But the modified robot path caused the robot to lose the human and once the human was outside of its sensing range, it replanned again trying to minimize the distance needed to reach the goal in a human-agnostic manner.

We have chosen to preserve the social costmap for a predefined period of time, if perception updates about the human position are no longer received. In the case of moving humans, a tracker must be used to generate this information for a predefined period of time. This period should be chosen according to the level of dynamics of the human movements in the environment. In our experiments, we update the social costmap using the last received human position for a period of 30 *s* upon receiving no data from the people trackers. After this period, a timeout clears the social costmap and erases all the past data. It is arguable which approach is better in the general case, but as mentioned before we believe this decision of how long to preserve a social costmap is tied to the type of the environment and the perception capabilities of the robot. If the robot is able to observe the people at all times it makes sense not to preserve the social costmaps but if this is not the case, erasing the social costs immediately when the human is out of sensing range, would be questionable. We believe limitations of local perception are among the challenges in human-aware navigation that are commonly overlooked and require further investigation.

To increase the awareness that the human has of the robot and for having a more legible navigation we added complementary gestures to the robot as advised by [14]. This was done for clarifying robot's intention, namely, the direction towards which the robot was going to move. The pupils of the robot's eyes indicated the direction of the movement and the LEDs on the side to which the robot wanted to turn were activated to improve the navigation legibility. Figure 5.1b shows the robot before turning to its right. We note that despite the interactive features running on the robot, we did not truly utilize

them in our experiments due to the limitation of having a static human. Therefore, we did not evaluate the robot-awareness gained by adding features that increase legibility through a user study. In the case of moving humans, this indication of intent can be very useful for the humans for planning their path in the vicinity of the robot. However, we did not pursue this direction with the Ranger.

Although Ranger is a good robot for studying human-robot interactions targeted to children, the limitations of this simple robotic platform result in difficulties for experimenting with more complex algorithms and scenarios. Therefore, the experimental work reported in the remaining chapters of this thesis have been carried out using another robotic platform called the Mbot, already introduced in Section 5.2.

Summary

In this chapter, we demonstrated the effectiveness of autonomous human-aware navigation for a simple domestic robotic platform designed for children, relying only on its on-board sensors and computation capabilities. The system combines human detection and tracking, basic autonomous navigation and the concept of personal space modeled with a social costmap. The final human pose is the result of fusing depth-image-based estimations given by (i) a low-lying view point tracker and (ii) a leg detector. Furthermore, challenges and limitations of this problem have been explained. Results confirm that human-aware paths result in a better performance with respect to social metrics compared to paths generated by a human-agnostic approach.

12 Incorporating Perception Uncertainty in Human-Aware Navigation

One key element of human-aware navigation is human detection and tracking as it provides the data translated to social costs using models such as social costmaps. However, state-of-the-art perception methods are not perfect and are affected by various elements such as the robot movement, humans' movement, lighting conditions, complexity of the environment in terms of occlusions, etc. Due to the approximate nature of the models and the less than perfect human detectors available, we often can only provide estimates of the locations of the humans with an associated uncertainty. Thus, any planning algorithm relying on real perception data, must be able to cope with this inherent problem. Despite this fact, the assumption of having perfect information about the position of people at all times is common in the state-of-the-art research in human-aware navigation and the main research focus is on the planning itself. However, moving to real applications, poses serious challenges in terms of noisy perception information and high uncertainties, that need to be addressed and modeled in a human-aware approach.

In this chapter, we present a novel approach to human-aware navigation that explicitly accounts for perception uncertainty by incorporating the uncertainty of position estimates of humans into the social costmap, using the Mbot robot. We study the effect of this factor on social costs and demonstrate how taking this uncertainty into account in the extended social costmap model, can result in trajectories that are improved in terms of social acceptability. Although we rely on an FMM-based social path planner, it should be emphasized that our approach is planner-independent as long as the planner takes costmaps as input, *e.g.*, it could be used in combination with ROS's navigation stack [114].

12.1 Perception Model

A socially-aware path planner needs to take into consideration individual humans and possible social interactions taking place. This information can be obtained by an external source such as an overhead camera or can be attained using on-board sensors of the robot. Different perception sources for human detection and tracking have different levels of uncertainty and accuracy in their detections, and are affected differently by the environmental factors. As an example, UWB technology can track targets in NLOS conditions while this is not possible for vision-based systems. By taking the uncertainty of perception into account in a human-aware path planner, the same planning method could easily be reused even when the source of perception and consequently perception uncertainty change.

In the proposed framework, a probabilistic approach has been chosen to account for uncertainty in

tracking rather than deterministically reporting positions and dealing with the perception uncertainty inside the tracker module only. We are interested in the underlying *state* of the environment which is the position of the humans. The *detectors* used to estimate these state variables have associated noise due to various factors such as occlusion, lighting conditions, different postures of the people, motion of the robot and the humans, etc. Coupled with this, there is also stochasticity in the state transitions, which makes it hard to compute an exact estimate of the human positions. A principled approach to solve this problem, is to compute a *belief* (posterior distribution) over the states using recursive Bayesian estimation. We first describe the state representation of the system and then explain the tracking model formally.

12.1.1 Detector

The background-based detector proposed in [115] is a very effective probabilistic method, that allows automatic evaluation of the number of people in the scene and detection of their locations. This method has the following advantages. (i) It can incorporate prior knowledge, including which areas in the scene can contain people and how probable it is for people to be in those locations; a probability distribution over the number of people in the scene; a probabilistic model of how close together people tend to walk; etc. (ii) The complexity of the algorithm depends linearly on the number of people in the scene. (iii) The method is robust to changes in illumination, shadows and occlusions. Moreover, it is adapted to adjust to a non-static background automatically.

12.1.2 State Representation

An occupancy grid-based approach is used for tracking the people in the environment in this chapter. The environment is discretized into G cells. Each cell is of size 25 $cm \times 25$ cm. The size of cells has been chosen in such a way that each cell can be occupied at most with one person at any time. The occupancy of each cell is denoted by X_i where $i \in G$. The occupancy of all the cells at time t is the state of the world $\mathbf{X}_t$. At every time instance t, the observations from the detector for each cell i is given by O_i. The set of observations for the whole state is denoted by $\mathbf{O}_t$.

12.1.3 Tracking Model

Let $\mathbf{X}_t$ be the state of the environment at time t. We are interested in computing the current belief over the states $\mathbf{X}_t$ given the observations $\mathbf{O}_{1:t}$, that is, $P(\mathbf{X}_t|O_{1:t})$. Formally, this can be represented by a Bayes filter [116] as,

$$P(\mathbf{X}_t|\mathbf{O}_{1:t}) \propto P(\mathbf{O}_t|\mathbf{X}_t) \sum_{\mathbf{X}_{t-1}} P(\mathbf{X}_t|\mathbf{X}_{t-1}) P(\mathbf{X}_{t-1}|\mathbf{O}_{1:t-1}) \tag{12.1}$$

where the likelihood $P(\mathbf{O}_t|\mathbf{X}_t)$ represents the observation model which expresses the probability that an observation $\mathbf{O}_t$ was made given the state $\mathbf{X}_t$ at time t. Computation of this likelihood is best performed by using a learned model of how the detectors perform in different states.

$P(\mathbf{X}_t|\mathbf{X}_{t-1})$ is the *transition model* which models the evolution of the state variables. For a real multi-person environment, an exact analytical model is intractable. In our system, we use a simple uniform distribution for the transition model. We assume that people move randomly and that there is an equal probability of motion in any direction. $P(\mathbf{X}_{t-1}|\mathbf{O}_{1:t-1})$ is the belief computed at the previous time step.

In a multi-person environment, the state space is extremely complex for computing the exact probability distribution over the states. We use an Markov Chain Monte Carlo (MCMC)-based sampling algorithm to approximately compute the belief. In the next section the implementation details of our probabilistic model for human detection and tracking are explained. Although the detector is modeled probabilistically, it is still needed to learn the distribution $P(\mathbf{O}|\mathbf{X})$ for the detector from data. Given the labeled location of the people in a data set, the uncertainty in the observations is learned for all the locations and configurations of the state space. Since the state space has high dimensionality, we learn the likelihood model over a parametrized state space [117].

12.1.4 MCMC Sampling

MCMC is a widely used sampling algorithm for estimation of complex posterior distribution. It has been gaining popularity in multi-target tracking applications [116]. Compared to traditional particle filters, MCMC-based sampling leads to far less sample impoverishment and thus a much better estimation of the state over time. The core idea of MCMC is to generate samples from a Markov chain. The samples are then evaluated using a *proposal distribution* and accepted or rejected based on an *acceptance ratio*. The proposal distribution should be proportional to the posterior distribution that we are trying to approximate. The MCMC sampler creates hypotheses of the location of the people in the grids and is used to compute the belief over the previous steps. Each sample is an estimate of the occupancy of all the cells taken collectively. In this work, the occupancy of the grids are used as hypotheses. Each cell can either be occupied or not, with initially starting from a random distribution of occupancy and then generating samples using the following moves:

1. *Birth-death proposals:* A cell is randomly selected, and the sample state of the cell is flipped. If the cell was occupied, a proposal which makes the cell unoccupied is generated and vice-versa.

2. *Move proposals:* In this case, an occupied cell is selected and the occupancy is randomly moved to one of the 8 connecting neighbors.

Once the proposal sample is generated, we evaluate the original sample and the proposed sample with reference to a proposal distribution. In our case, we use a learned observation model of the detector output as the proposal distribution. We fold in the detector output $\mathbf{O}_t$ while evaluating the proposals using the proposal distribution. Every proposal is a hypothesis of the state $\mathbf{X}_t$. Evaluating the proposals will give us an acceptance ratio. If the acceptance ratio is greater than 1 the sample is accepted unconditionally. Otherwise, we randomly sample from a uniform distribution and then accept or reject the sample if the acceptance ratio is greater than the sampled value of uniform distribution. Formally, The acceptance ratio is computed as:

$$Acc(x|x^{'}) = \min\left\{\left(\frac{P(\mathbf{O}_t|x)\sum\limits_{x_{t-1}} P(x|x_{t-1})}{P(\mathbf{O}_t|x^{'})\sum\limits_{x_{t-1}} P(x^{'}|x_{t-1})}\right), 1\right\} \tag{12.2}$$

where x is the proposed sample of state $\mathbf{X}_t$, and $x^{'}$ is the initial sample. x_{t-1} is a sample of the state $\mathbf{X}_{t-1}$.

If the sample is accepted, the currently proposed sample will be used as the initial sample for the next step of the MCMC sampling. If rejected, the sample is still added to the set of hypotheses, but we start sampling again from the old sample.

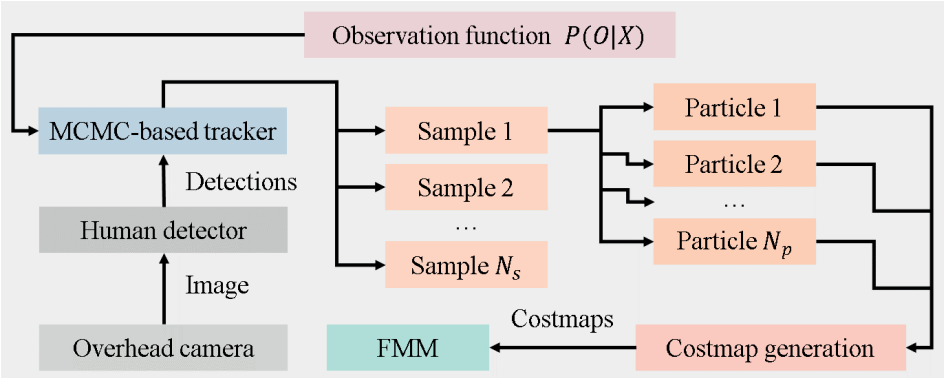


Figure 12.1 – System components diagram.

Sampling is successively repeated until N_s samples are accepted. The threshold for the number of accepted samples are set to be 100 in our experiments. Once N_s samples are accepted, they are collectively represented as the approximate representation of the multi-modal posterior distribution.

For the purpose of social navigation, these samples are converted to a set of particles. Even though the set of samples are fixed, the set of particles can vary, since each sample is a joint sample which represents the whole occupancy grid. When there are multiple people in the environment, each one of the samples can be decomposed into a set of particles, proportional to the number of people (N_p) in the environment.

12.2 Human-Aware Navigation Model

In this section, we will present a number of methods for computing social costs with the particles obtained from the MCMC-based tracker. Using the concept of layered costmaps, an FMM-based planner uses this information for performing the replanning. Figure 12.1 shows the system components and their connections. The social cost model used in this chapter has been previously explained in Section 10.3.1 where the center of the social costmap that indicates the human position is assumed to be deterministically known. If this is not the case, the costmap cannot correctly model the social costs and hence the social path planning could fail in finding an appropriate socially compliant path. This problem becomes much more critical in real applications where robustness is vital for succeeding under different conditions. We believe probabilistic social costs can be a remedy to this problem.

12.2.1 Uncertainty-Based Social Costs

The MCMC-based tracker provides samples of the *union* of the probability distributions for having a person in a given environment. More specifically, a predefined number of joint samples are reported at each time step, containing a set of particles for each person that the tracker detects. In the following, we propose two types of methods for using the particles given by these samples for constructing social costmaps.

Convolution

The core idea we propose for incorporating uncertainty in the costmap is to compute an *expectation*-based costmap. Consider a person at (x_p, y_p), the deterministic costmap at (x, y) is:

$$C(x, y; x_p, y_p) = N(x - x_p, y - y_p) \tag{12.3}$$

N is the 2D Gaussian, modeling the standard social costmaps, *i.e.*, it is N (refer to Equation 10.1) when $q = [x,\ y]^T$ and $p = [x_p,\ y_p]^T$. The probabilistic costmap is given by the expected value of the cost C, given the probability distribution p_h of the human being at $[x_p,\ y_p]^T$:

$$\mathbb{E}\,C(x, y) = \mathbb{E}_{p_h(x_p, y_p)}[C(x, y; x_p, y_p)] = \int\int N(x - x_p, y - y_p)\, p_h(x_p, y_p)\, dx_p\, dy_p \tag{12.4}$$

This is a convolution. We approximate this expectation using a grid of probabilities P, obtained from the tracker particles:

$$\mathbb{E}\,C(i, j) \simeq \sum_k \sum_l N(i - k, j - l) P(k, l) \tag{12.5}$$

By convolving all the particles from the MCMC tracker with the social costmap, we compute an *expected costmap* incorporating all the uncertainty in the environment. This is a principled approach to solve the problem since we are not abstracting away any information provided by the perception system, and hence, in theory, this approach should provide us with a costmap model that would be most informative for uncertainty-based human-aware navigation.

By taking this approach, the conventional 2D Gaussian shape of the social costmap is no longer mandatory, thus this costmap model is more flexible. Additionally, there is no need to know the number of people ahead of time as this information is encoded in the particles (refer to Section 12.1.4).

Clustering

Another approach we propose is to abstract the uncertainty information in the samples by clustering the particles and then computing the uncertainty based on the features of the cluster. Upon receiving the position particles from the tracker we compute the center of the social costmaps, and the σ values for all the people present in the environment by clustering the particles and finding the centroids of the clusters, along with an uncertainty measure based on the spatial cluster scatter. This allows us to adaptively compute the social costmaps at each time step. We will briefly describe the two clustering methods selected for our work in the following.

K-means Clustering K-means clustering [118] method can be adopted for computing the costmap centers and Σ. However, one requirement for using this method is knowing K which is the number of clusters and in our case the number of people, ahead of time. This is not a realistic assumption for dynamic environments with multiple people. Nonetheless, we will test this method as a baseline for comparing the performances of other human-aware navigation methods.

Mean Shift Clustering To overcome the limitation of K-means clustering method for needing to know the number of clusters ahead of time, we chose another clustering method, mean shift clustering

[119], that is able to determine the number of clusters given the particle set automatically. We used the median of all pairwise distances to estimate the bandwidth of the mean shift method.

Deterministic Social Costs

For the purpose of creating a baseline to compare our proposed model, we have also designed a deterministic model for the social costmap. Here, we use a deterministic output from the tracker, without considering the particles, by just using the mode of the distribution as deterministic location of the people, along with the conventional costmap model.

This is the standard approach that is being used for social path planning in human-aware navigation, where it is assumed that the position and orientation of the human is deterministically known at every time step with no model of uncertainty. Once this information is known, the Gaussian-shaped costmap can be used to obtain the social costs. This costmap can be tuned according to the desired parameters. In this chapter, we have taken an approach similar to [33] with $\sigma_x^2 = \sigma_y^2 = 0.255\ m$, for deterministic costmaps.

12.3 Experiments

In this section, we will briefly explain our experimental setup, the experiment scenarios and the metrics used to evaluate the performance of our methods. An extensive suite of experiments both in simulation and reality have been conducted for performance evaluation.

The experimental setup used in this work leverages one Mbot and one or two humans, as illustrated in Figure 7.1b. We used a networked omnidirectional overhead camera with a field of view of 180° to track the positions of the people in the environment. This type of camera was chosen because: (i) it is less obtrusive, and can be left in the environment with less risk of making people feel uncomfortable about being watched; (ii) it provides a global view of the area, with lower risk of occlusion than elevated side-view cameras and with more flexibility as to its positioning; (iii) the number of cameras needed in the environment can be reduced, which has benefits in terms of equipment cost, installation cost and computational load of the perception algorithms. The tracker outputs results at the rate of 3 *Hz*. The ground truth position of the robot is given by AMCL with 5-10 *cm* accuracy, and the person stands and walks on physically marked tracks to get the exact precise ground truth for the purpose of performance evaluation. The control rate of the navigation is 20 Hz while the social costmap generation has a rate of approximately 3 *Hz*. This is to account for the low output frequency of the tracker. The real robot tests were carried out in three different scenarios in the robotics laboratory (see Section 7.1).

12.3.1 Scenarios

We have studied three different scenarios, each having been tested five times. We started with a single static person (Scenario 1) and incrementally increased the complexity to one moving person (Scenario 2) and finally, two static people in the arena (Scenario 3). It should be emphasized that perception uncertainty is affecting the tracking performance and is not evident or quantifiable from just looking at the environment. This means, the person is not aware of what is happening on the tracker side, however, the information given by the tracker greatly affects the behavior of the robot, and therefore its social acceptability. Since we aim to study the effect of perception uncertainty in human-aware

navigation, we chose a task of point to point navigation for the robot in the vicinity of humans, which is the most general and basic navigation task.

In each experiment, the robot starts from a predefined starting point and is sent to one predefined goal. The robot then has to behave appropriately when it encounters people in the arena. For the static case, there is always a person standing between the robot and the straight line to the goal, and for the dynamic case the person moves along this line in the opposite direction, as the robot starts navigating towards the goal, causing a direct encounter with the robot. The following section will explain the metrics used for performance assessment.

12.3.2 Metrics

Five different metrics have been defined for performance evaluation. A subset of these metrics is chosen for each experiment based on the scenario of interest.

- m_1: measures the minimum distance that the robot has kept during the experiment to a human.
- m_2: evaluates how long the robot has been moving in areas associated with social costs, *i.e.*, a position with corresponding non-zero value in the social costmap.
- m_3: quantifies the accumulated social cost, this is to differentiate between being in different positions of the social costmap, which is not reflected in m_2. So if the robot is closer to a person, the social cost will be higher and this metric will increase.
- m_4: evaluates the smoothness of the robot trajectory. This is important from the human observer's point of view when perceiving the robot motion. Humans are known to prefer motion with minimum jerk [120], therefore we took the Root Mean Squared Error (RMSE) of the trajectory jerk in m_4:

$$r_t = \begin{bmatrix} x_t \\ y_t \end{bmatrix}, \quad m_4 = \sqrt{\frac{1}{N}\sum_{t=1}^{N}\left|\frac{\mathrm{d}^3 r_t}{\mathrm{d}^3 t}\right|^2} \tag{12.6}$$

  r_t indicates the position of the robot at time t, and N indicates the number of time steps in the experiment. It should be emphasized that we did not actively try to modify the robot control to get smoother trajectories, we are just interested to see which method results in a more natural and smooth path.
- m_5: is the total number of time steps required to finish the navigation.

12.4 Results

For each of the scenarios described in Section 12.3.1, we have compared the results obtained from the basic navigation (BN), deterministic (DHA), K-means clustering (KHA), mean shift clustering (SHA), and social cost convolution (CHA) human-aware (HA) navigation. We will only report the trajectories and metrics obtained from the results of our real experiments for the sake of conciseness. Larger values for m_1, and smaller values for m_2-m_5 are preferable.

Figure 12.2 shows sample costmaps of the different methods mentioned earlier. It can be seen that the clustering methods can end up with saturated costs when the uncertainty is high due to large σ values. The convolution costmap is shown to be much more flexible and is not limited to a predefined shape whereas all other costmaps have a predefined cut-off distance. Sample trajectories of the robot are depicted in Figure 12.3. It is clear from the plots that HA methods result in trajectories that preserve

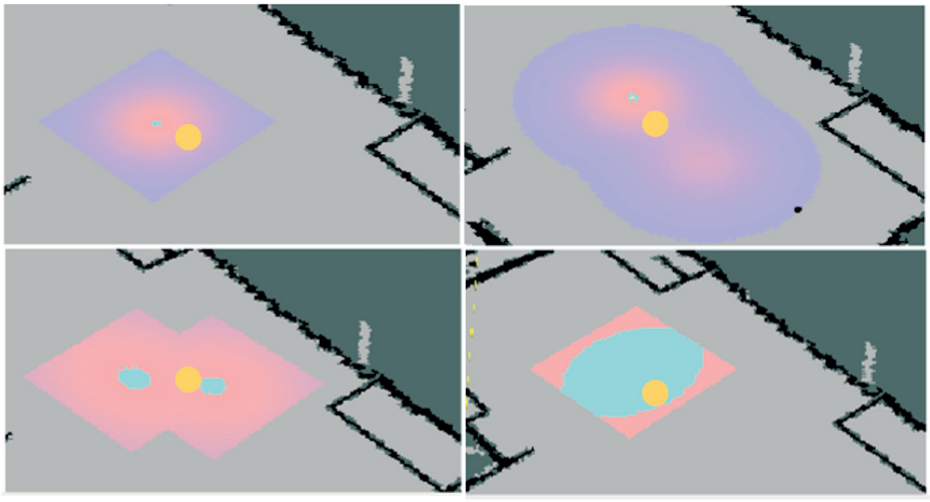


Figure 12.2 – Sample costmap shapes. Top left: standard 2D Gaussian, top right: Convolution method, bottom left: mean shift and bottom right: K-means. The yellow circle indicates the true human position. Areas with red color have a larger cost compared to the blue. The maximum social cost is depicted by the cyan color. This color coding is directly derived from rviz.

larger distances to the people. Additionally, they are smoother and therefore more natural from the point of view of a person, this is supported by Figure 12.4d, 12.5a, and 12.6d. However, this may not be evident from the trajectory plots. This is due to the abrupt movement of non-HA navigation upon encountering a person which considerably affects the smoothness. This abruptness cannot be illustrated using a trajectory plot only.

Clustering methods can cause the robot to modify its plan largely by enforcing a certain cluster shape upon finding cluster centroids: if the new probabilistic data leads to a new centroid that is not very close to the previous one, the costmap could change significantly and therefore the planned path. This is more severe for mean shift clustering due to adaptively modifying the number of clusters as well. This is to be expected given the probabilistic nature of the perception data, however the plan can be modified more smoothly using the convolution method. This method which outperforms all other methods in terms of smoothness in all of our tests, is shown to be a remedy to this problem based on our experimental results. Hence, for the second and third scenario, we only compare the results of BN, DHA, and CHA methods.

When comparing the results of DHA with CHA we can see that the former is a more conservative method in terms of keeping distance to the people when receiving accurate data. If DHA receives a perfect estimate of the person's position it can lead to the desired path, however this is seldom the case. Particularly, in the case of a moving person, the detector could not always keep up with the speed of the person, *i.e.*, the position estimates were reported with delay or the person was lost in some cases, and the robot was faced with the human while considering him an obstacle. This led to abrupt changes and getting too close to the person, see Figure 12.3b. However, by associating larger uncertainty to the estimates in this case, CHA could lead to better plans in terms of proximity and smoothness. Unfortunately, due to our inaccurate ground truth of the moving person, we only rely on m_4 and m_5 for Scenario 2, but we observed the delayed perception and lost person problem during our experiments.

Figures 12.4, 12.5 and 12.6 show the performance metrics for the three scenarios. We will discuss the

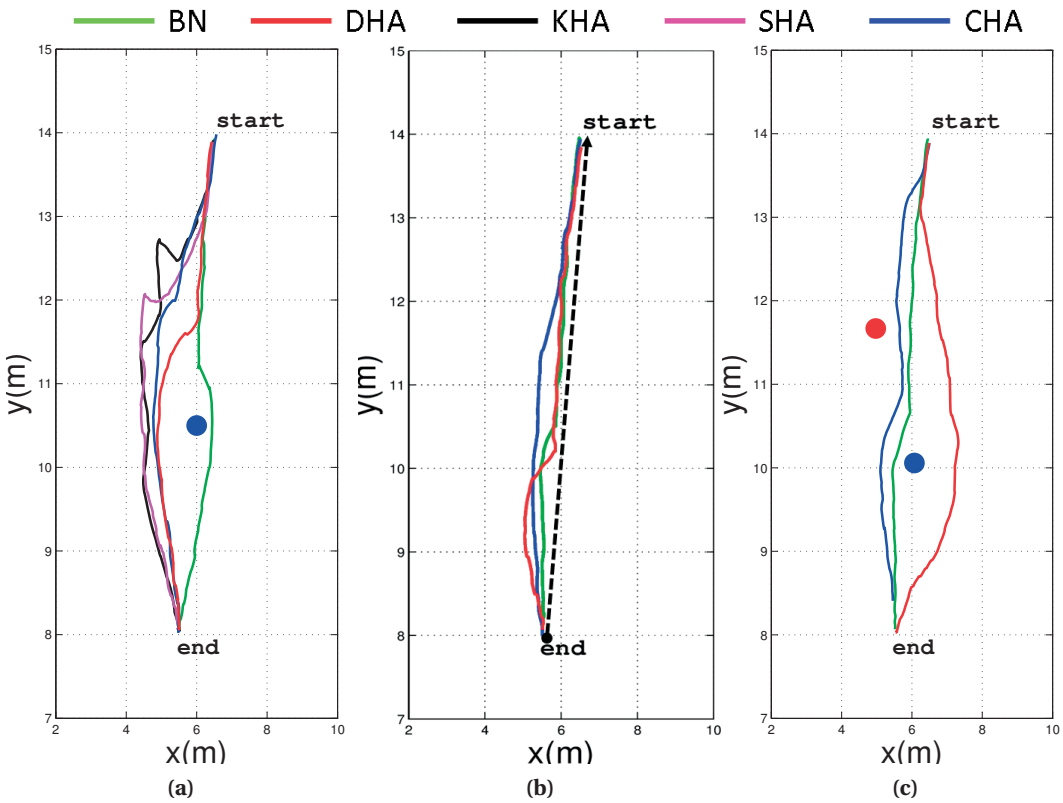


Figure 12.3 – Sample robot trajectories for different methods. Scenarios: (a) Single static person. (b) Single dynamic person. The trajectory of the human depicted by the dashed line, starts from the end point to the starting point of the robot's trajectory indicated by labels on the plots. (c) Two static people.

results of each metric in the following. m_1 has increased for HA methods which shows the effectiveness of our FMM planner in social path planning. However, DHA is more conservative in this regard in the presence of good perception data. m_2 has increased for uncertainty-based methods in the simple scenario as the deterministic tracker is already giving good position estimates, but this is no longer the case when the complexity is increased as seen in Figure 12.6b. m_3 is also reduced for HA methods and more so for uncertainty-based HA methods as the complexity of the problem increases, see Figure12.6c.

m_4 is showing a very interesting result, we can see how uncertainty-based HA methods have managed to introduce smoothness into the trajectories by reducing jerk without deliberately accounting for it. CHA is dominating other methods across all scenarios in this case. Lastly, m_5 which shows the total navigation time is always lower for BN due to optimizing the path length only. The largest values belong to KHA and SHA due to constant modifications of the path, thus taking longer routes. For DHA this metric is lower than CHA for the static person case and comparable to CHA in the moving person scenario, but it increases as the complexity of the environment grows further in the case of multiple people.

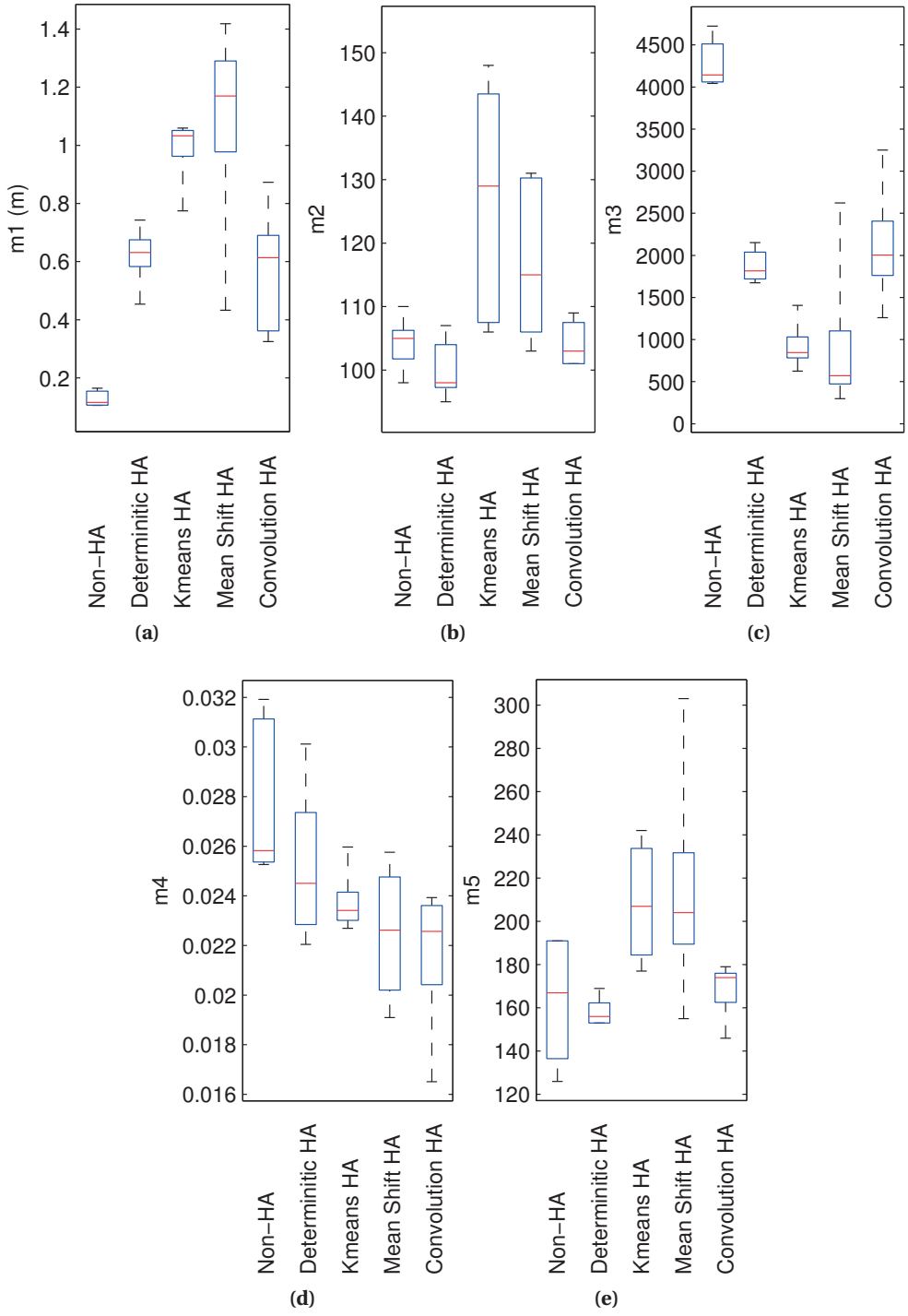


Figure 12.4 – Performance metrics obtained in the single static person scenario. HA stands for Human-Aware in the plot labels.

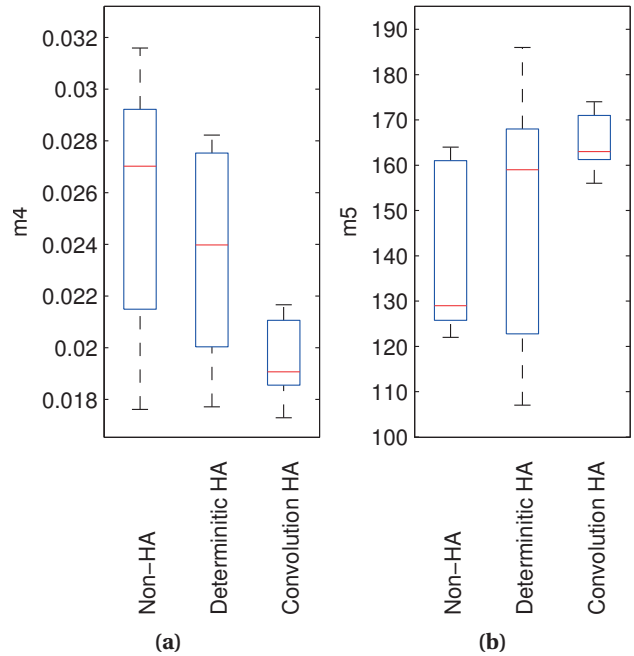


Figure 12.5 – Performance metrics obtained in the dynamic person scenario.

12.5 Discussion

The experimental results show how the extended costmap model can lead to more natural robot trajectories that preserve a social distance from people, as the complexity of the environment grows. Further improvements can be made to the accuracy of robot self-localization and the ground truth of people positions, as they have direct influence on performance evaluations. Moreover, quantifying the uncertainty of perception (investigating the impact of different levels of perception uncertainty on the behavior and performance of each part of our system) can be very useful in analyzing the behavior of expectation-based social costmap computation methods for further in-depth studies. We believe that the expectation-based social costmap will outperform the deterministic approach to a greater extent as the uncertainty of perception grows larger.

Summary

In this chapter, we introduced a principled approach to solve the social path planning problem in real environments with multiple people while explicitly dealing with perception uncertainty by combining the output of a probabilistic MCMC-based tracker with an expectation costmap computation method based on convolution. The proposed approach is implemented in reality and tested in a stochastic environment using an extensive set of experiments. Accounting for uncertainty of perception is shown to result in improved robot trajectories in terms of social distance and naturalness as the complexity of the environment grows. We observed that smoother trajectories with lower jerk that are more natural from the point of view of humans are achieved using expectation-based costmaps.

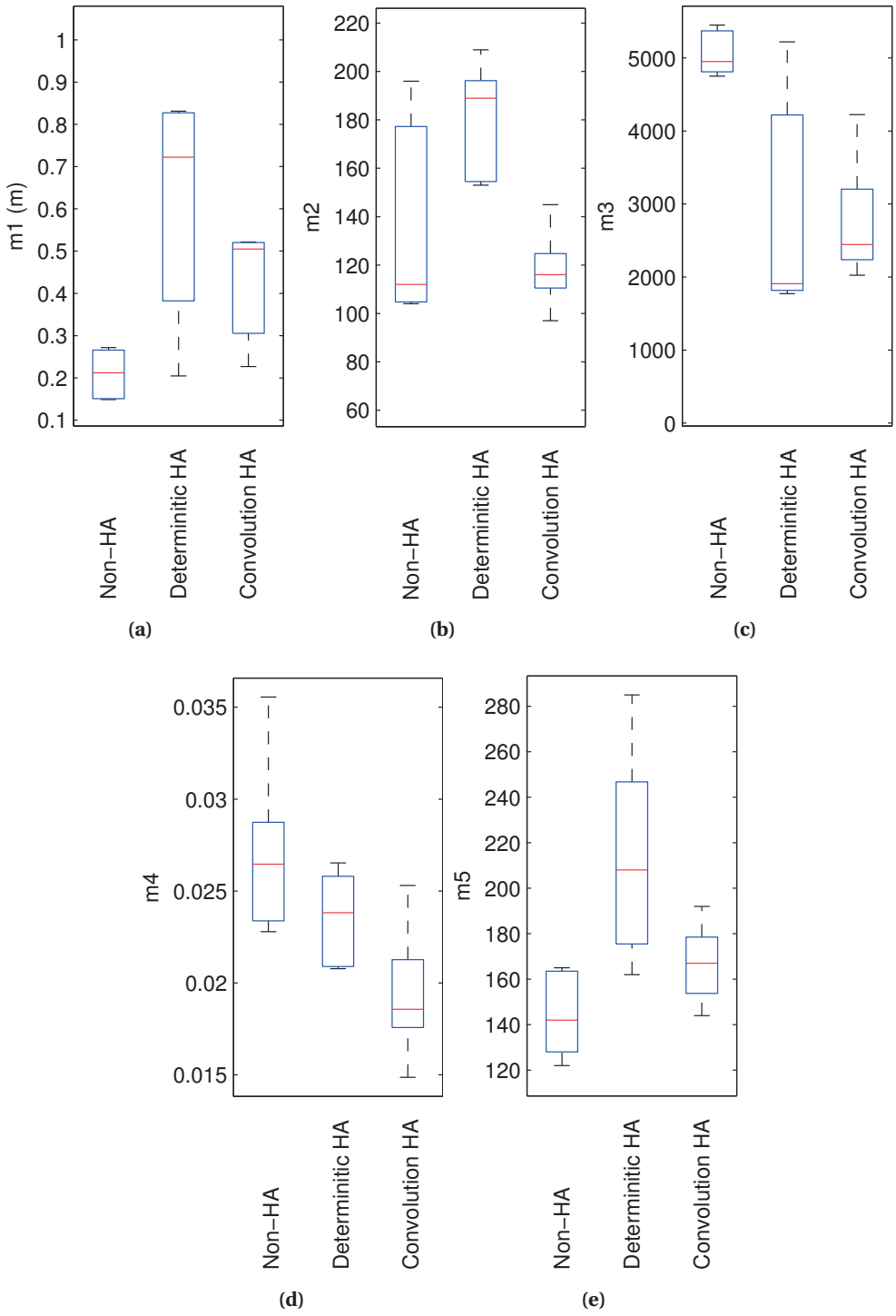


Figure 12.6 – Performance metrics obtained in the two static people scenario.

13 Conclusion

HUMAN-AWARENESS for a mobile robot is accomplished through a human-aware path planner that incorporates the social costs related to humans into its plans. For this purpose, information regarding humans must be acquired and social costs must be modeled and integrated into the planner. In real uncontrolled environments, human perception will not be perfect. As the complexity of the environment and thus the uncertainty in perception grows, methods relying on this key information will not be efficient without accounting for uncertainty. Therefore, deterministic social cost models should revisit their common assumption of having perfect information about the humans and adopt a probabilistic approach. Moreover, in the presence of multiple perception sources of varying uncertainty, the probabilistic social cost modeling and layered costmaps allow for incorporation of all these data directly into the planner's decision.

Another important aspect of human-aware navigation that mainly targets replanning is local perception as mobile robots rely on on-board perception in most real applications. Despite the simplicity of the scenario tested in Chapter 11 with the Ranger, we saw how social costs and thus social paths can change as the perception of the robot changes. In real environments a robot with local perception, will frequently be faced with newly perceived or lost targets as it moves. Social replanning strategies are required to help the robot deal with the highly stochastic environment despite its partial observations.

Providing ground truth data for real experiments is vital for human-aware navigation research as it is required for performance evaluation. However, conducting experiments in real domestic environments make it very difficult to acquire very precise ground truth. In cases that external ground truth is available, we usually have to rely on sensors such as overhead cameras with limited working range and accuracy. Carefully evaluating the uncertainty and performance of on-board perception devices such as the Kinect with accurate ground truth positioning systems such as the MCS in a laboratory can be one remedy to this problem. With this information available, the uncertainty for on-board perception can be quantified and reported for ground truth measurements and social cost computation when robots are operating in real domestic environments.

Part IV
Multi-Robot Cooperation in Social Environments

14 Introduction

DESPITE the numerous applications of robotics in social environments, such as personal assistants at homes, robot tutors at schools, service robots at hospitals and nursing homes, research in the human-aware navigation area focuses mainly on single robots and the problem of cooperative human-aware navigation for multi-robot systems -an interesting problem for both multi-robot and human-aware navigation research- is largely unexplored.

In Part IV, we focus on a particular class of MRS coordination mechanisms commonly known as Multi-Robot Task Allocation (MRTA) [50], [51] in social environments using a market-based approach [55]. In such environments the number of robots are often limited and the number of tasks are usually moderate. The main difficulty for MRTA in such highly dynamic and noisy environments is that plans are likely to change or to be rendered invalid, particularly if the robots are planning for long periods of time. Additionally, robots are required to perform in a socially acceptable manner in terms of navigation and interaction with people and other team members. This adds additional constraints to the planning problem.

Robots can benefit from coordination when faced with social constraints. As an example, when a human chooses to block a passage that a robot needs to traverse for reaching a desired destination to perform a task, what is the best strategy for the team? The human might want to start an interaction with the robot. In this case, the robot will no longer be available to perform the team plan. It could also be that the human has no intention for interaction but simply needs to use that space, for instance because he or she is having a conversation in front of a door. By assigning the already taken task to another robot through coordination, the team-level plan can change on demand to adapt to this situation. Accounting for social factors in task assignment can lead to better team plans in terms of social acceptability and even performance metrics such as time, if team-level planning considers the social costs and handles the unforeseen changes in the environment by means of coordination.

In social environments, when robots take decisions based on the current available information only, they are assuming that their decision remains valid in its period of execution. However, human actions such as walking, and starting or ending an interaction, can largely modify the social costs while a robot is proceeding towards an already allocated task. In such cases, the initial bid estimation for the task allocation is no longer a true representative of the real cost. As a result, the performance and efficiency of team plans can degrade in more complex and dynamic environments. Moreover, having to take decisions based on limited local perception is another challenge that calls for devising strategies for improved coordination and adaptation of plans when initial assumptions no longer hold.

MRTA approaches are required to have enough flexibility for performing efficiently under highly stochastic and uncertain social circumstances. This motivates our decision to base our methods in this part of the manuscript on the Hoplites coordination Framework [61]. We find this framework suitable for the goal of deploying teams of robots in social environments because of its flexibility. In Hoplites, passive coordination can produce local decisions for robots while active coordination can produce joint plans in terms of task allocation for all the robots in the team.

The contributions of this part are as follows. In Chapter 15, we introduce an adapted version of Hoplites for MRTA with deterministic costs. In Chapter 16 our proposed method is extended to explicitly consider humans in its cost formulation and planning. As the next step, costs that have a stochastic nature due to the changing behavior of people are taken into account in Chapter 17. Finally, adaptive risk-based replanning is introduced in Chapter 18 to account for the changes perceived in the environment caused by the limitations of local perception and unpredicted human behavior.

15 Market-Based Multi-Robot Cooperation

THIS chapter focuses on multi-robot coordination for solving the MRTA problem. As the first step towards such a collaboration scheme, a variation of the Hoplites framework is proposed and three instances of increasing complexity of the MRTA problem have been investigated: spatial task allocation based on distance, spatial task allocation based on distance and time, and persistent coverage. The performance of the proposed method is studied in comparison with other state of the art MRTA methods in simulation. Additionally, the problems of spatial task allocation and persistent coverage with real robots are investigated. We note that modifications are made to the original framework in order to both improve and adapt this method for solving MRTA problems. Namely, replanning has been changed to include turn taking and priority planning for avoiding and resolving conflicts. Moreover, instead of estimating the plan time, that is very sensitive to the unpredictable changes in a dynamic environment, we have opted to take the plan length into account. This work is a first step towards adopting MRTA in dynamic human-populated social environments.

15.1 The Original Hoplites Framework

Hoplites [61] is a market-based framework that couples planning with coordination strategies. It is not bound to any particular planning method or in general problem-specific feature and only deals with robot cooperation. Hence, it is a powerful framework to be used in different applications as we aim to show in this chapter. It allows for coordinating plans instead of tasks. A *plan* is a sequence of tasks and computing the cost and the revenue of a plan depends on the problem. This framework consists of two main concurrent coordination mechanisms: *passive coordination* and *active coordination*.

In *passive coordination*, each robot chooses its most profitable plan and broadcasts it to other teammates without any attempt to modify their plans. This information is then used by other robots to reevaluate the expected profitability of their current plans, update and broadcast the changes.

Since robots can affect the actions of one another and change their plans at any point of time, they cannot be very confident about their estimate of the profitability of their actions. Sometimes a robot's best plan can only be marginally profitable and a team plan could result in a higher profit. This indicates modifying the plans of the robot's teammates. This implies that the requesting robot asks its collaborators for compensation price quotes and persuades them into cooperation. If they accept, they will be bound by a contract to complete their portion. This process is ruled by a market-based

approach and constitutes the *active coordination*.

The decision of switching to the active coordination mode (explained in Section 15.3.1) is based on the evaluation of a *balance function*. For robot r_j and a given plan P_k the balance function is generally defined as follows:

$$B_{j,k} = R_{j,k} - C_{j,k} - Z_{j,k} \tag{15.1}$$

where R is a generic revenue function, C a generic cost function and Z the penalty for constraint violations. Note that this is a *local* balance function: the costs, revenues and penalties are related to a single robot. The local balance function is strongly problem-dependent and can contribute to reaching the globally optimal solution on the team level if chosen correctly. Additionally, a problem-dependent *global* balance function is also required for team-level evaluations.

15.2 Proposed Method

Since real social environments are noisy and dynamic, it is required to ensure the validity of plans. Invalid plans can be the result of changes in the environment or other robots changing or stopping their current plans. Therefore, a turn taking mechanism is introduced into the Hoplites framework to avoid computing costly invalid plans. This implies only one robot planning at a given time. To reduce the drawbacks of this choice, two improvements have been made. Firstly, the robots can choose to immediately replan for avoiding conflicts or some particular situations whenever the need arises. This is done by requesting priority from other robots in a distributed manner. Secondly, robots do not wait for their turn to start replanning but rather compute and store a new plan without following it during the other robots' turn. At the start of a robot's turn, if no conflicts were detected, the stored plan is used. Otherwise, the robot will replan again. This leads to speeding up the team performance. Assigning and communicating the turns can be done in a centralized manner using a supervisor or in a distributed fashion by reaching a consensus among robots.

Another difference between our method and Hoplites is basing the planned coordination on the maximum number of tasks rather than a predefined time. This abstraction of time allows for a more robust handling of uncertainties and makes the approach less sensitive to the unpredicted changes that might affect the estimated time for the plans. Although this aspect directly affects the planner, it is a design choice that is far more suitable for dynamic social environments where an accurate estimate of the time to accomplish a plan is not guaranteed. While the work in this chapter has not been tested in such environments, it constitutes a baseline that is intended to be used in environments shared with people in the next steps. Therefore, it is essential to opt for a robust feature that is not very sensitive to uncertainties and noise.

The main drawback of this choice is that robots may decide to take tasks irrespective of their costs in terms of time and, in the case of spatial task assignment, favor further away tasks. However, distance and therefore an estimate of time are accounted for while computing the balance of a plan and by tuning the weights of the balance function appropriately, the desired behavior can be achieved. Moreover, through active coordination an initially suboptimal assignment of a task or plan can be modified by selling it to a robot that is closer, if available. It is important to note that this balance can be computed

Algorithm 1 Passive Coordination for robot r_j with a set of unfinished tasks T

```
1: procedure PassiveCoordination(T, r_j)
2:     T_a ← ∅                                   ▷ T_a is the set of available tasks
3:     for t_i ∈ T do                            ▷ T contains all tasks regardless of their availability
4:         if IsAvail then(t_i)                  ▷ Checks if the task is unassigned
5:             T_a ← Add(t_i)
6:     P ← Planner(T_a)
       return P
```

Algorithm 2 Active Coordination for robot r_j with a set of unfinished tasks T

```
1: procedure ActiveCoordination(T)
2:     T_a ← T                                   ▷ T_a is the set of available tasks
3:     Ω ← ∅                                     ▷ Ω is the set of robots in conflict with r_j
4:     P ← Planner(T_a)                          ▷ Find the plan for the available tasks
5:     Ω ← RobotInConflict(P)                    ▷ Find the conflicting robots
6:     T_P ← GetTasks(P)                         ▷ Find the tasks constituting a given plan
7:     T_a ← T_a − T_P                           ▷ Update the available tasks
8:     for r_ω ∈ Ω do                            ▷ Reset the plans of the conflicting robots
9:         P_ω ← ∅
10:
11:    ▷ Take the most profitable plan for r_j and start replanning for the conflicting robots
12:    for r_ω ∈ Ω do
13:        P_{c,ω} ← PassiveCoordination(T_a, r_ω)   ▷ Find the plans for the robots in conflict
14:        T_{c,ω} ← GetTasks(P_{c,ω})
15:        T_a ← T_a − T_{c,ω}
       return P, Ω, P_c
```

locally by the robots with partial information or in a centralized manner. We have chosen the first approach in this work.

To describe the method in detail, consider a set of N_r robots and N_t tasks, where each task can be assigned to only one robot. The cost of completing a task $\{t_i \in T, i = 1,\ldots,N_t\}$ for robot $\{r_j, j = 1,\ldots,N_r\}$ is denoted by c_{t_i,r_j} and the corresponding revenue is ρ_{t_i,r_j}.

Using a revenue function R, and a cost function C, the robots, who are self-interested agents in pursuit of individual profit, can evaluate each available task and decide whether to take it or sell/buy it to/from another robot. Details of the described method can be found in Algorithms 1, 2, and 3. Note that each robot finds a plan using the coordination mechanisms in a distributed manner. Similar to Hoplites, it is assumed that a group of robots will only accept to cooperate if the requesting robot is able to pay all of them the compensation price.

15.3 Test Cases

To evaluate the effectiveness of the proposed method, we have tested three variations of the MRTA problem with increasing complexity. Initially, the proposed method was tested against a single task greedy algorithm (SGA) and the Consensus-Based Auction Algorithm (CBAA) [68] in simulation, to provide insight about the performance of the methods in a realistic noisy environment. SGA is a single

Algorithm 3 Market-based Coordination for robot r_j with a set of unfinished tasks T

```
1: ▷ This is the main procedure running on each robot
2: procedure MarketBasedCoordination()
3:     P ← ∅                                                    ▷ The current plan
4:     P_s ← ∅                                                  ▷ The stored plan
5:     F ← False                                                ▷ Flag for accepting cooperation
6:     ▷ While there is a task to be assigned
7:     while T ≠ ∅ do
8:         if myTurn() then                                     ▷ Perform task assignment steps only in designated turns
9:             if IsValid(P_s) then                             ▷ Check if the stored plan is valid
10:                P ← P_s
11:            else
12:                P_n ← ∅                                      ▷ The new plan
13:                P_n ← PassiveCoordination(T, r_j)
14:                ρ, c ← GetBalance(P_n)                       ▷ Compute the current revenue and the cost
15:                if ρ ≤ ϱc then                               ▷ ϱ stands for the minimum acceptable gain in revenue
16:                    P_a, Ω, P_c ← ActiveCoordination(T)
17:                    F ← AskForCooperation(Ω, P_c)
18:                    if (F) then
19:                        P_n ← P_a
20:                P ← P_n
21:        else
22:            ▷ Compute and store a plan for the next task allocation
23:            P_s ← PassiveCoordination(T, r_j)
24:            ▷ Checking and responding to potential active coordination requests
25:            if ReceivedCooperationProposal() then
26:                EvaluateProposal()
27:                BroadcastAnswer()
```

task centralized method that assigns the task with the minimum cost to each available robot. CBAA is another single task method that is shown to provide similar solutions to SGA through auction and consensus mechanisms, in a decentralized manner.

Additionally, to show the flexibility of the approach, we tested the proposed method for the problems of spatial task allocation and persistent coverage, in simulation and reality. The two test cases are inherently similar but in the case of persistent coverage, an additional dimension of time is added, making the problem continuously recurring. This section describes the planner that is commonly used in all cases and later on explains the details of each test case.

15.3.1 The Coordination Planner

Since the focus of this work is not optimizing the plans but rather investigating the capabilities of this approach, a simple and computationally inexpensive planner based on single-step optimization has been used. When optimizing a plan consisting of multiple tasks, this planner performs the optimization sequentially. This choice is motivated by the nature of dynamic environments and the fact that a plan can be rendered invalid or suboptimal at any time. Therefore, optimizing the immediate decision is of higher importance and step-by-step optimization is justified.

This planner is used for both types of coordination. However, for active coordination it should addi-

tionally consider team plans. Active coordination occurs when the revenue of the current plan is too small and there exists a task for which the *expected added revenue* is higher for the replanning robot compared to the currently assigned robot. The expected added revenue is the difference between the balance of the robot *with* the task and the balance of the robot *without* the task.

The planning horizon should be chosen to accommodate the dynamics of the environment. In highly dynamic environments long planning horizons are no longer effective and incur unnecessary computational costs without contributing much to a better plan compared to shorter planning horizons. As an example, for a robot that relies on its path planning for computing bids, the changes made to the environment by the movement of people can largely modify the plans. This is exacerbated for longer plans which have a higher probability of change. This is the main reason of not including CBBA in our comparisons. CBBA is most effective in problems with long planning horizons where it can optimize the plan of each robot and the global plan of the team by finding the best sequence of tasks to be performed by each robot in a decentralized manner. If a single task or very short sequences of tasks are to be considered, the superiority of CBBA to single task methods may be overshadowed by its higher computational cost.

Given that short planning horizons are more suited for our target environment, we have set the planning horizon of the proposed method to one task in $H1$. However, to evaluate the result of having a longer planning horizon we have also included $H2$ where the planning horizon is set to two. These are the two variants of the proposed Hoplites-based method that we use throughout this chapter.

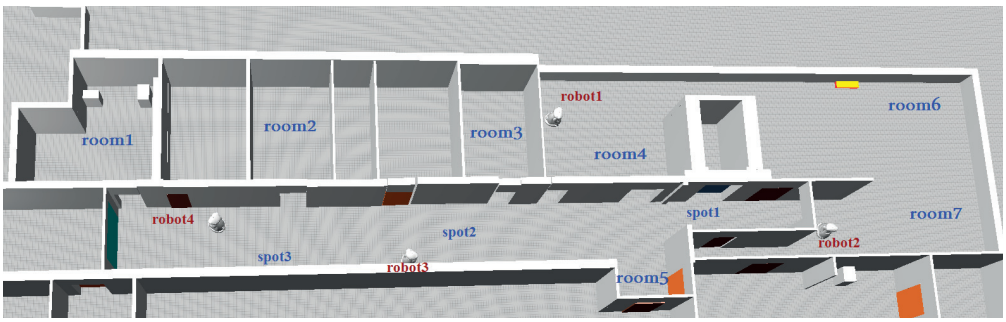
15.3.2 Test Case I: Spatial Task Allocation Based on Distance

Given the formulation in Section 15.2, a team of robots decides how to efficiently subdivide a set of tasks that will induce optimizing a global criterion. This global criterion can be a function of time, distance traveled, etc. In this case, the tasks involve moving to a specific location in the environment. These tasks can be identified locally by the robots through on-board perception or can be broadcasted to all robots by an external source. Many applications such as patrolling, attending service requests, etc., can benefit from this functionality.

In this test case, four robots and 10 tasks constitute the MRTA problem as shown in Figure 15.1. Three metrics of individual robot contribution, total time and total distance of the assignment problem have been considered as the global balance functions to evaluate the performance and the behavior of the MRS. On the local level, each robot tries to maximize a *local balance function* that is inversely proportional to the length of the path planned by the FMM to a given task location.

15.3.3 Test Case II: Spatial Task Allocation Based on Distance and Time

Similar to the previous case, a number of tasks are associated to specific locations in the environment and the robots should find the most appropriate assignment for optimizing the global balance function. The main difference between this test case and the previous one, is the incorporation of time as a factor in the local balance function. The consequence of adding this factor is to encourage robots to reach tasks as early as possible, leading to increased parallelism in the team. The *global balance function* for this problem is defined as the completion time of the team objective. The *local balance function* for robot r_j, given a plan P consisting of tasks t_i, is computed in the following. Note that this function is also problem-specific. It is an instance of Equation 15.1 without the penalty term. This is



(a)

Figure 15.1 – Snapshot of the environment of test case I, with four robots and 10 tasks in a Webots world.

due to constraint violation being prevented on a higher level by replanning and in the lower level by the collision avoidance modules.

$$B_{r_j,P} = \sum_{t_i \in P} (\rho_{t_i,r_j} - \mathrm{D}(l_{t_{i-1}}, l_{t_i})) \tag{15.2}$$

ρ_{t_i,r_j} is the revenue of task t_i for r_j, l_{t_i} is the position of t_i and l_{t_0} is the position of the robot when starting the plan. This function includes a revenue ρ_{t_i,r_j} that is

$$\rho_{t_i,r_j}(k) = \max(0,\ \varrho_{max}(1 - \frac{k - k_{a,t_i}}{\tau})) \tag{15.3}$$

where k is the time in which t_i is reached, ϱ_{max} is the maximum revenue for the task, k_{a,t_i} is the allocation time of t_i and τ is the time after which the positive revenue becomes zero. Note that the arrival time to the task is required for computing the balance function. As mentioned previously, only an estimate of the time to reach a task can be computed in real noisy environments. However, since all robots are faced with the same limitations, this does not affect the team performance considerably in our tests in this chapter.

15.3.4 Test Case III: Persistent Coverage

This problem consists of continuously covering an area with a group of robots. It has many applications such as cleaning, heating, etc. This is a more challenging problem compared to the previous test case. The same approach used in Section 15.3.3 is used here. The robots need to reach designated points in the environment with the purpose of maintaining a desired coverage level over time. Persistent coverage entails a continuous assignment of locations to robots. The coverage level of a point is maximized upon a robot reaching the point.

The *global balance function* in this case is based on the coverage functions of [121], [122] where a set of points discretizing the 2D space is considered. Each point is assigned an initial coverage value and at every time step its coverage level is decayed with a predefined rate of $\delta < 1$. If a point is sufficiently close to a robot its coverage level is increased as shown in Equation 15.4:

$$v(p_{i,k}) = \delta v(p_{i,k-1}) + K \sum_{j=1}^{n_r} f(r_j, p_{i,k}) \tag{15.4}$$

where $v(p_{i,k})$ is the coverage level associated to the point p_i at time k and f is defined as:

$$f(r_j, p_{i,k}) = \begin{cases} 1 & \text{if } d = 0 \\ \frac{R_f - d}{R_f} & \text{if } 0 < d \leq R_f \\ 0 & \text{if } d > R_f \end{cases} \tag{15.5}$$

where $d = \mathrm{D}(l_{r_j}, p_{i,k})$, and l_{r_j} is the position of robot r_j, and the radius that the robot can cover is denoted by R_f. The $f(.)$ function modulates the coverage increase at any point. It assumes the maximum value in the center of the robot and then decreases linearly as depicted in Figure 15.2.

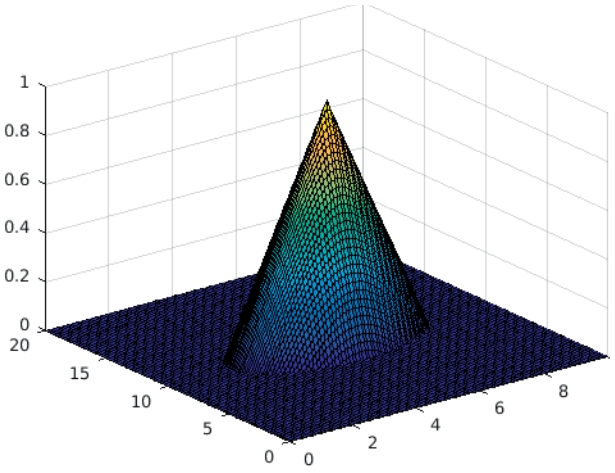


Figure 15.2 – f function for a robot centered on (5,10).

The *local balance function* is composed of two terms: the revenue and the cost. Considering a plan P consisting of tasks t_i, the balance function for robot r_j is defined as:

$$B_{r_j,P} = \sum_{t_i \in P} \left(\rho_{t_i,r_j} - K c_{t_i,r_j} \right) \qquad K > \frac{\dot{\rho}_{t_i}}{v_{max}} \tag{15.6}$$

where

$$\begin{cases} c_{t_i,r_j} = i\ \mathrm{D}(l_{t_{i-1}}, l_{t_i}) \\ \rho_{t_i,r_j}(k) = \min\left(\varrho_{max},\ \varrho_{min} + (\varrho_{max} - \varrho_{min}) \frac{k - k_{a,t_i}}{\tau} \right) \end{cases} \tag{15.7}$$

l_{t_0} is the position of the robot r_j at the start of the plan, k is the time when r_j reaches the task, τ is the time after which the revenue is ϱ_{max} and k_{a,t_i} is the allocation time of t_i to r_j. Note that the cost is dependent on the distance traveled. The revenue is similar to the previous case, but it is increasing with time rather than decreasing. This is because points that have remained uncovered for longer will contribute more to reaching the desired coverage level. Hence, they should have a higher priority.

If the revenue increases with time, the robots will favor longer paths to a given point, to paths that are more optimal in terms of distance and time. This is the reason K is introduced. With a sufficiently large K the increased traveling cost will be higher than the increased revenue for longer paths. Given the maximum speed of the robot v_{max} and the increasing rate per second of the revenue $\dot{\rho}_{t_i}$, each unit of distance must cost more than $\frac{\dot{\rho}_{t_i}}{v_{max}}$ for the robot to move directly to a task rather than taking a longer trip.

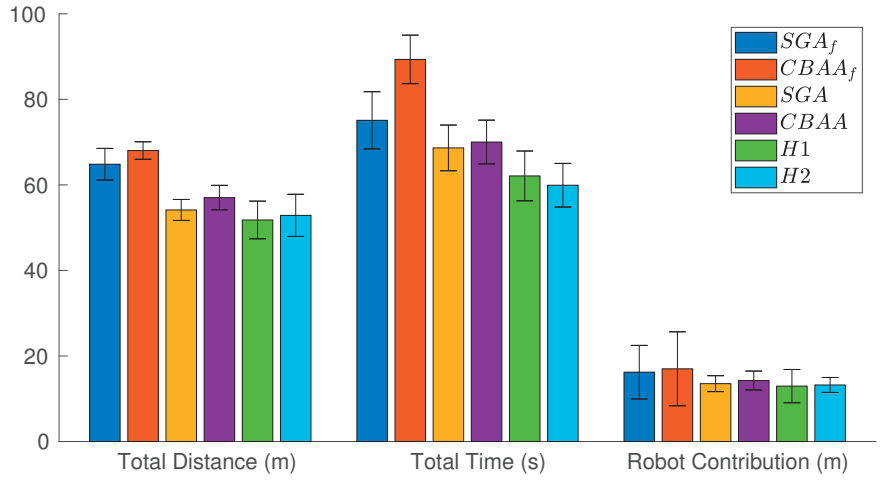


Figure 15.3 – Performance of different MRTA methods for Test Case I.

Additionally, a multiplier factor i is used in the cost function to penalize the tasks of the later steps. This is necessary for avoiding some counter-intuitive side effects, *i.e.*, increasing the revenue causes the robots to give a larger revenue to tasks that are reached later in the plan, since more time has passed. This leads to situations such as a robot taking a task as its second task while it is better to have that task assigned to another robot as the first task. This is an undesired situation since we want to minimize the time for reaching a desired coverage level for the team.

15.4 Results

In this section, we show the results of our tests and quantify the differences in performance between simulation and real robot experiments. Note that the experiments are subject to noise and the robots have an average self-localization accuracy in the order of 0.2 m.

15.4.1 Test Case I: Spatial Task Allocation Based on Distance

For this test case we have tested two variations of SGA and $CBAA$. We have added the constraint of only allocating available and unallocated tasks, similar to the passive coordination in Hoplites. This is to see how local decisions with no coordination for modifying plans perform compared to methods that delegate the tasks to other robots through coordination. This variation of the methods are denoted by SGA_f and $CBAA_f$ as they only consider free tasks. In SGA and $CBAA$ a robot always takes the task that has the lowest cost in a greedy way. This means that when a robot is free, it can request to take an already allocated task. This request will be granted through coordination if the robot can accomplish that task at a lower cost compared to all other team members. This delegation is depicted by red blocks in Figure 15.4a-15.4b.

Looking at Figure 15.3 we can see that SGA_f and $CBAA_f$ are more costly in terms of the total traveled distance and time, compared to their unconstrained counterparts. However, they require less time for communication as there is no coordination done among robots for modifying allocations. The

comparably larger standard deviation of robot contribution for SGA_f and $CBAA_f$ shows how selecting tasks with a local greedy strategy without coordinating with other team members can lead to less evenly distributed plans. The main reason for variability of the solutions for the robots in SGA_f and $CBAA_f$ is the localization error and the time at which a robot finishes a task and becomes available again. Each task is allocated to the closest available robot and if a Robot 1 takes longer to reach a task due to some effort lost in improving its localization, another robot could become available before Robot 1 and take a task that would otherwise be assigned to Robot 1. This is less problematic for the other methods and particularly for the proposed method since the active coordination mechanism negotiates the plans with all other active and inactive robots. The inferior performance of SGA_f and $CBAA_f$ highlights the importance of having a correction mechanism for already allocated tasks as the environment changes.

Based on Figure 15.3, we can see that SGA, $CBAA$, $H1$ and $H2$ have similar performances. However, it can be observed that the proposed method manages to find marginally shorter solutions that take slightly less time compared to both SGA and $CBAA$. This is because of active coordination. Although the centralized SGA suffers the least from communication delays, since a shorter solution is found by $H1$ and $H2$, the total time of the assignment is also less for those methods. There is no significant difference between the traveled distance and the time for $H1$ and $H2$ in this problem. But $H2$ finishes the assignment in slightly less time since it spends less time in the idle state between tasks as a result of two step planning.

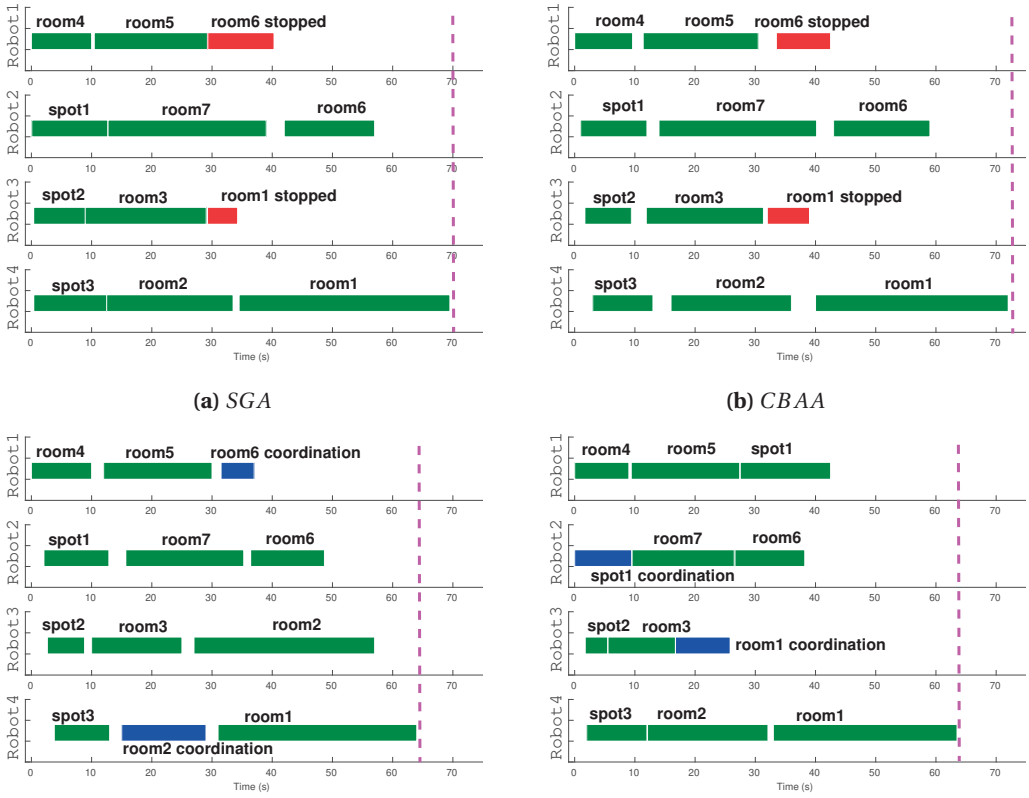
Figure 15.4 shows four different assignment solutions from a sample run for each of the SGA, $CBAA$, $H1$ and $H2$ methods. In Figure 15.4.c, Robot 1 is moving toward “room6” at time 28 but it receives a collaboration request (shown in blue blocks) from Robot 2 upon the completion of “room7” and stops moving and transfers “room6” to Robot 2. This collaboration can also take place for longer plans, as depicted in Figure 15.4.d for “spot1”. By delegating “spot1” to Robot 1, Robot 2 can find a better two-step plan while allowing the next two-step plan of Robot 1 to have a larger local and global balance contribution. The gaps between robot movements relate to the communication delays and the time spent for negotiation.

An interesting difference between the Hoplites-based solutions and SGA, $CBAA$ can be seen in Figure 15.4c. Unlike SGA and $CBAA$, active coordination guides Robot 4 to delegate “room2” to Robot 3 for the benefit of the team despite “room2” being the result of the greedy action selection for Robot 4. In the other two methods, Robot 4 would continue going to “room2” even when Robot 3 is done with “room3”. A similar case can be seen in Figure 15.4d for $H2$, when Robot 2 delegates “spot1” to Robot 1 after “room4” is visited by Robot 1.

15.4.2 Test Case II: Spatial Task Assignment Based on Distance and Time

To see how our method performs in this test case consider Figure 15.5a-b where sample results of real experiments show that active coordination can lead to achieving better results and correcting the decision of passive coordination. Figure 15.5c shows the robot trajectories for a set of 20 tasks with three robots. Results of this scenario can be found in Table 15.1. We note that the number of tasks in the plan are chosen to be two and tasks can be added to the task list dynamically. However, we chose this configuration for the ease of presentation.

Table 15.1 shows that the simulation results follow the real robot test results closely in terms of time. This similarity highlights the strength of our simulation tools. The tests with three robots were only conducted in simulation due to limitations of the available robots. The time gain when adding the third



(a) *SGA* **(b)** *CBAA*

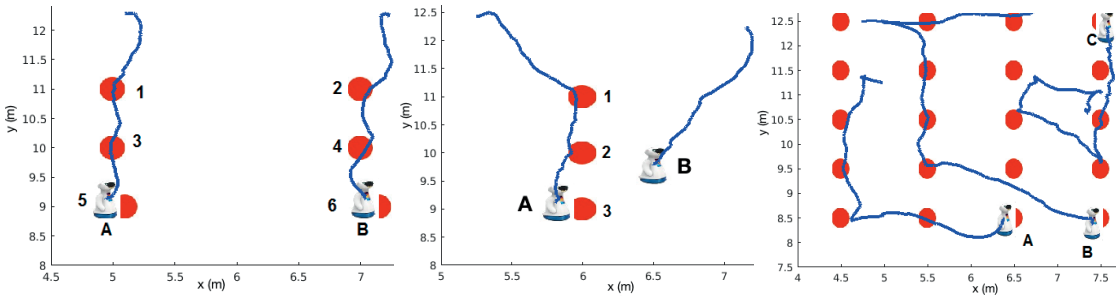
(c) *H*1 **(d)** *H*2

Figure 15.4 – Task assignment per robot over time for a sample run of the first test case, given the following methods: a) *SGA*, b) *CBAA*, c) *H*1, and d) *H*2. Blue blocks indicate collaboration requests and coordination and red blocks represent stopped attempts.

robot is very little. Nonetheless, the mean of the traveled distance is shown to have slightly improved. The σ has increased largely for both distance and time. This could be due to the fact that more robots cause more complex situations and more complicated coordination. We can see an even distribution of number of tasks between the robots (see Figure 15.5c). This confirms that the time is spent in coordination rather than moving.

Table 15.1 – Results of the Spatial Task Assignment problem. μ is the mean, σ the standard deviation, T the time to completion in seconds and D the total distance traveled by the robots in meters.

-	#Robots	μ_T	σ_T	μ_D	σ_D
Simulator	2	44.36	3.45	9.78	1.44
Real robots	2	43.12	5.31	8.47	2.6
Simulator	3	43.57	9.1	6.9	2.07



(a) (b) (c)

Figure 15.5 – a) Using passive coordination Robot A would take Tasks (1,3) and Robot B (2,4), then Robot A would take (5,6). However, active coordination leads to distributing the last pair of tasks rather than assigning it to one robot. b) Passive coordination would assign (1,2) to Robot A and (3) to Robot B. However, (3) is more suited for Robot A and is later assigned to it by active coordination, once Robot A is free. c) Robot trajectories and task distribution for the case of three robots. Robots are shown in the map coordinates with meter scale. The position of the robots in the figures indicates their final location in the experiment.

15.4.3 Test Case III: Persistent Coverage

Two sets of tests have been conducted for investigating the performance of the proposed method for the persistent coverage problem with two robots. For the sake of conciseness only real robot results are reported. Robots operate in the same environment of the previous experiment and tasks are created by means of a spatial grid. The list of parameters used in our implementation can be found in Table 15.2.

To understand the effect of R_f two values representing the radius of the robot footprint and its double are tested. Clearly for some applications such as cleaning, R_f should be the radius of the robot, but for some other cases e.g., heating it could be sufficient to assume a larger radius. Figure 15.6a shows the mean coverage level for different R_f values.

Three scenarios have been tested with varying number of tasks. Mean and variance of the coverage function along with the histogram of coverage levels are reported. The steady state coverage levels are shown in Figure 15.6a. The numerical results of all the cases are presented in Table 15.3. μ indicates the mean, σ the standard deviation, and c_v is an indicator of variation defined as:

$$c_v = \frac{\sigma}{\mu} \tag{15.8}$$

As the number of points are increased so does the resolution of the coverage (see Figure 15.7a-c) and this gives a better result from the variance and mean point of view. We observe a large increase in σ when increasing R_f, despite seeing a much better coverage in terms of μ. This can be observed in Figure 15.7d-f. This increased σ is the reason for introducing c_v because looking at the variance alone can be misleading here. c_v captures the mutual effect of both factors and is preferred to be smaller.

Table 15.2 – Parameters used in the persistent coverage problem

Parameter	δ	K	Small R_f	Large R_f
Value	0.99	20	0.325 (m)	0.65 (m)

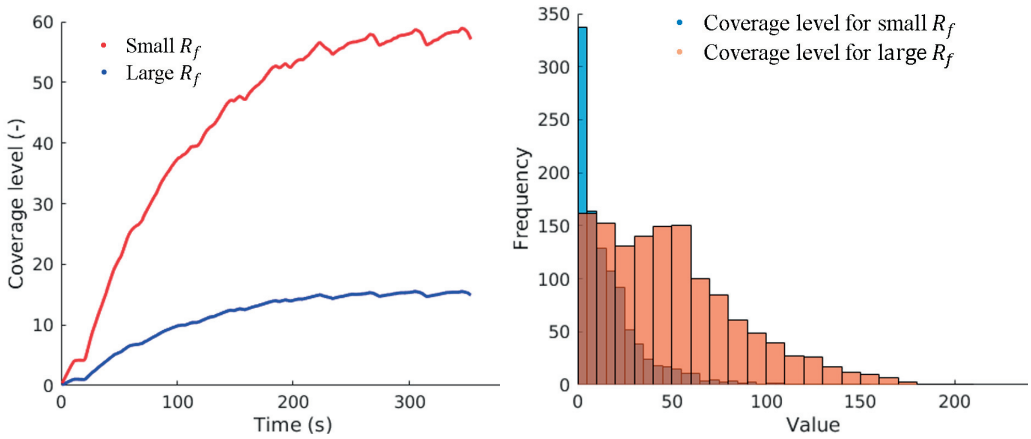


Figure 15.6 – a) Mean coverage level over time for small R_f and large R_f for 20 tasks. b) Histogram of the coverage values. Blue indicates small R_f and orange indicates large R_f.

Table 15.3 – Coverage levels in the 6 cases studied in the persistent coverage problem. μ is the mean, σ the variance and c_v the coefficient of variation.

#Tasks	R_f	μ	σ	c_v
12	Small	14.53	25.1	1.72
20	Small	14.93	18.41	1.23
48	Small	16.01	16.01	1.43
12	Large	55.25	48.72	0.88
20	Large	57.13	35.98	0.63
48	Large	63.5	39.39	0.62

The reason for the increased variance can be seen from a different angle by looking at Figure 15.6b. The histogram of large R_f is more distributed and has larger values for the majority of points compared to the histogram of small R_f. The distribution of coverage values for large R_f has a higher mean but also a higher variance since the values are farther apart. For small R_f it can be observed that most points have low coverage levels and the distribution is pushed towards the lower end. Nonetheless, we can see how increasing R_f has led to a better coverage with a comparably larger μ and smaller c_v.

We note that the problem of persistent coverage can be tackled by means of other approaches as well. In [122] we proposed a different approach where optimal paths in terms of coverage quality have been computed locally using an FMM-based planner that takes into account a safety distance to the obstacles and an improvement measure for the coverage.

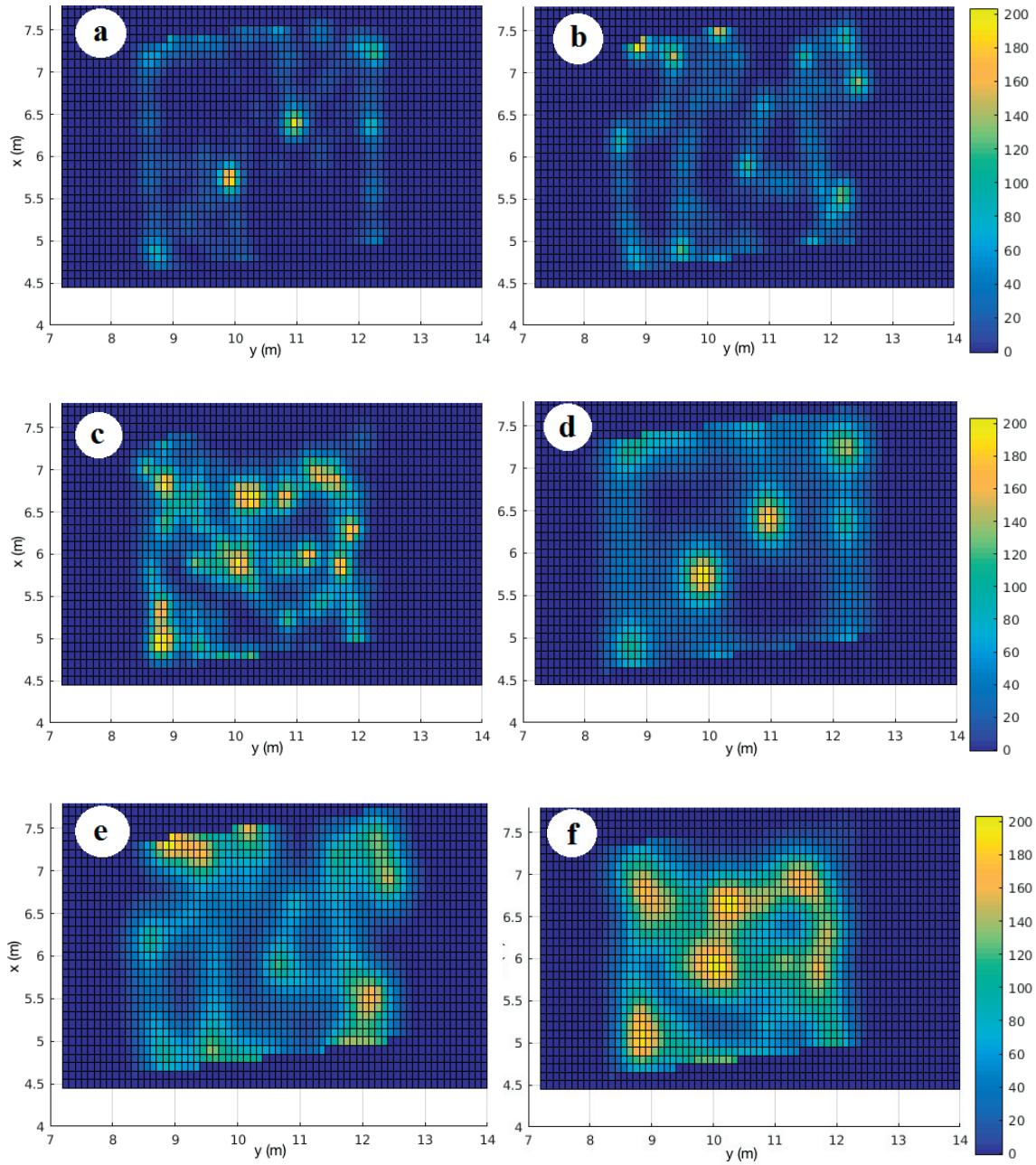


Figure 15.7 – Steady state coverage values for small R_f and a) 12 b) 20 and c) 48 tasks. Steady state coverage values for large R_f and d) 12 e) 20 and f) 48 tasks.

15.5 Discussion

Results confirm that the proposed approach is able to solve the spatial task allocation and persistent coverage problems in general. The use of this framework for solving the persistent coverage problem provides interesting insights by taking a high-level approach that is different from the commonly used

solutions for this problem such as computing robot trajectories to keep the desired coverage level. However, there exist some limitations. Particularly, in the case of persistent coverage, this method is suitable for applications where moderate spatial resolutions are sufficient such as patrolling. For the next steps, we will introduce humans into the problem and improve the bid estimations by explicitly accounting for social factors and the presence of humans in a more dynamic social environment.

Summary

In this chapter, we proposed a method based on the Hoplites framework for solving the MRTA problem. Adaptations and customizations have been made to Hoplites for our targeted social MRTA application, by modifying how the replanning is done and basing the planned coordination on the maximum plan length as opposed to time. We have demonstrated the flexibility of this market-based framework by applying it to different scenarios of increasing complexity. Results confirm the effectiveness and flexibility of this method for solving MRTA problems.

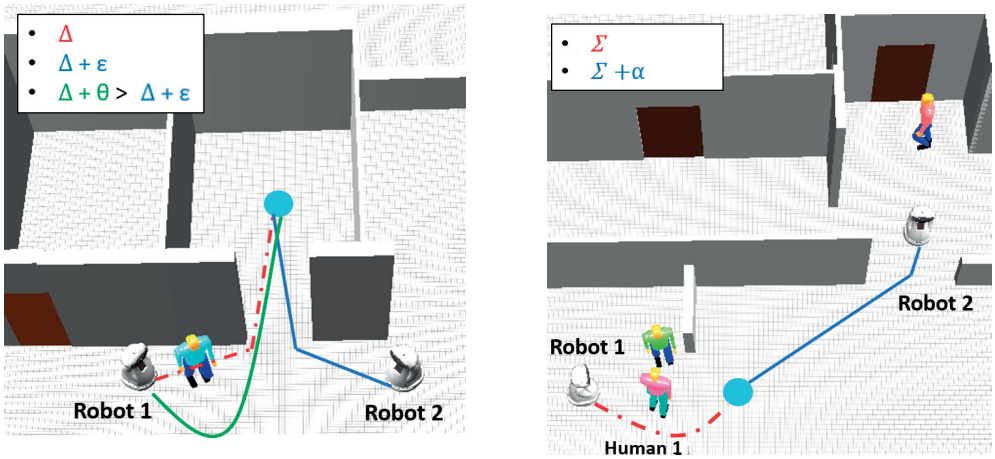
16 Multi-Robot Cooperation in Dynamic Environments Shared with Humans

In this chapter a human-aware MRTA method is introduced that considers humans not only as social entities in individual path planning but also at the task planning level. The main extensions made to the method proposed in Chapter 15 driving this strategy are: accounting for social costs in bid evaluations, and requesting cooperation in socially blocking situations.

A team of cooperative robots that rely on coordination and joint planning cannot be agnostic to the humans when operating in a social environment. Humans can largely change the state of the environment causing the previously computed plans to be invalid or suboptimal. To understand this effect, studies involving both MRTA and social performance metrics are required in environments with different levels of complexity in terms of noise and dynamics of humans. Therefore, in this chapter we aim at providing insights on human-agnostic MRTA in social environments of varying complexity. Additionally, we propose a social MRTA method that accounts for humans in both individual and team-level plans. Furthermore, we conduct a comparative study in simulation and with real robots for MRTA instances with: (1) no humans present, (2) humans considered only as unmapped obstacles, (3) humans considered as social entities by individual robots at navigation level only, and (4) humans considered both at task planning and individual navigation levels. To the best of our knowledge, human-aware robot coordination for team-level planning in MRTA has not been investigated in the literature.

16.1 Social Multi-Robot Cooperation

We propose a system that builds upon two main components of (i) Hoplites-based multi-robot coordination introduced in Chapter 15 and (ii) human-aware navigation presented in Chapter 12. We will revisit these components and highlight the extensions and improvements motivated by social human-populated environments leading to our proposed method. In this chapter, the same MRTA problem of Chapter 15 is considered with the addition of social costs. For succeeding in a social environment, robots are required to ensure that social constraints, namely personal and interaction spaces of humans, are not violated. If the environment is changing in terms of human positions and interactions, how should a robot estimate the plan cost? What should be the strategy if a plan is rendered invalid due to the changes in the environment? How often or when should the robots replan and recompute the costs? These are interesting and challenging questions that need to be considered in MRTA for human-populated environments. In the following, we detail how the previous market-based coordination method is extended to answer some of these questions.



(a) (b)

Figure 16.1 – Two cases where accounting for the presence of humans lead to a modified team plan. a) Accounting for the added distance caused by human-aware navigation in the local balance function. b) Social time-out occurring as the result of humans forming a blocking passage. The cyan circle represents the task.

16.1.1 Socially-Aware Balance Functions

Since a distributed auction process requiring bid computations governs the market, accounting for social costs entails improving and extending the estimation of the balance function to include a social term. Hence, social factors should also be aggregated into the estimated value for bidding on the tasks. Designing the balance functions and aggregating the social terms are problem-specific and may vary across different environments. Compared to a human-agnostic bidding, the bid estimates containing social terms can result in modified team plans. As an example, in Figure 16.1a, given a local balance function that scores tasks inversely proportional to their distance to the robot, Robot 1 will take the task since its path length (Δ) to the task is inferior to that of the other robot ($\Delta + \epsilon$), with $\epsilon > 0$. However, by taking into account the added distance θ that Robot 1 has to travel to avoid intruding the personal space of the human, Robot 2 will take the task since $\Delta + \theta > \Delta + \epsilon$, with $\theta > \epsilon > 0$.

16.1.2 Social Time-Outs

If humans start interacting with a robot that has been assigned a task, assuming that attending to immediate requests of humans has a higher priority, the robot should stop. In a human-agnostic planning approach other team members assume that the task is taken and carry on with the remaining tasks. If a time-out is foreseen for unfinished tasks, after waiting for that period of time the task would be available on the market again and otherwise other robots will assume it will eventually be reached by the responsible robot. A similar situation can happen if the cost of a task changes due to the movement of the humans, as illustrated in Figure 16.1b. When robots are scoring the task while considering the social costs, Robot 1 will be the best candidate given the smaller distance to the task location since $\Sigma < \Sigma + \alpha$, with $\alpha > 0$.

However, if Human 1 walks back while maintaining the interaction, the path of Robot 1 can be blocked by the larger interaction space formed by the two humans (O-space). This way, despite Robot 1 having

Algorithm 4 Social time-out detection for robot r_j with a set of unfinished tasks T and a set of humans H

```
1: procedure SocialTimeOut(T, H)
2:     ▷ The flag indicating social time-out
3:     F_t ← False
4:     ▷ While the robot is assigned a task and is moving towards it
5:     while IsActive() do
6:         ▷ If at least one human is present close to the robot
7:         if InHumanVicinity(H) then
8:             ▷ If at least one human has started interaction with the robot
9:             if HumanInteractionRequest(H) then
10:                F_t ← True
11:            ▷ If the robot has not moved enough during a time window
12:            if NavigationProgressed(T) ≠ True then
13:                if HumanInteractionRequest(H) ≠ True then
14:                    F_t ← True
   return F_t
```

enough free space to pass, it will have to cross the O-space and thus, incur a larger social cost or it might have to wait for a long time trying to find a new path as people move. Given the formulation chosen for the local balance function and the weight of the social term, Robot 2 can be in such situation a better candidate for taking the task.

We have introduced a priority planning caused by social time-outs in our proposed coordination algorithm. In this chapter (see Algorithm 4), social time-outs are triggered when immediate human interactions start with an active robot or in case an active robot is not advancing in its planned path in the vicinity of humans for a predefined period of time. If a robot detects a social time-out it sends cooperation requests through active coordination.

Without going into the details of active and passive coordination detailed previously in Chapter 15, Algorithm 5 describes our proposed market-based coordination method. Accounting for social costs in bid computation and requesting cooperation upon social time-out detection are the main extensions and improvements made to our previous work to meet the requirements of a social environment. These changes are highlighted in blue in Algorithm 5.

16.2 Experiments

In this section we will describe the evaluation metrics and the case studies constituting our experiments. We note that due to the highly dynamic and noisy nature of social environments, planning for long horizons will not be effective and could incur unnecessary computational costs without contributing much to a better plan compared to shorter planning horizons. This motivates our choice of having a single task planning horizon for the rest of this thesis.

16.2.1 Evaluation Metrics

The local balance function of robot r_j, for each task t_i belonging to a plan P is defined in the following. This function is inversely proportional to the length of the path planned by the FMM to the desired

Algorithm 5 Market-based Coordination for robot r_j with a set of unfinished tasks T and a set of humans H

```
1: ▷ This is the main procedure running on each robot
2: procedure MarketBasedCoordination
3:     P ← ∅                                                    ▷ The current plan
4:     P_s ← ∅                                                  ▷ The stored plan
5:     F ← False                                                ▷ Flag for accepting cooperation
6:     ▷ While there is a task to be assigned
7:     while T ≠ ∅ do
8:         if myTurn() then                                     ▷ Perform task assignment steps only in designated turns
9:             if IsValid(P_s) then                             ▷ Check if the stored plan is valid
10:                P ← P_s
11:            else
12:                P_n ← ∅                                      ▷ The new plan
13:                P_n ← PassiveCoordination(T, r_j)
14:                ▷ Compute the current revenue and the cost considering the humans
15:                ρ, c ← GetBalanceSocial(P_n, H)
16:                ▷ ϱ stands for the minimum acceptable gain in revenue
17:                if ρ ≤ ϱc ∨ SocialTimeOut(T, H) then
18:                    P_a, Ω, P_c ← ActiveCoordination(T)
19:                    F ← AskForCooperation(Ω, P_c)
20:                    if (F) then
21:                        P_n ← P_a
22:                P ← P_n
23:        else
24:            ▷ Compute and store a plan for the next task allocation
25:            P_s ← PassiveCoordination(T, r_j)
26:            ▷ Checking and responding to potential active coordination requests
27:            if ReceivedCooperationProposal() then
28:                EvaluateProposal()
29:                BroadcastAnswer()
```

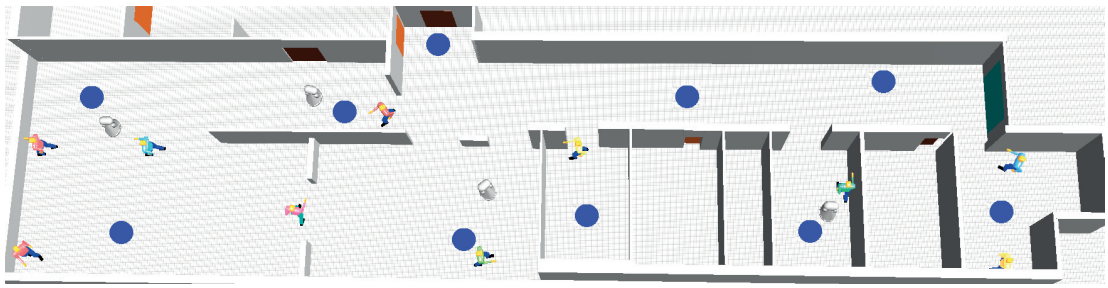
task. We note that for social task planning, the path planned by the FMM already takes into account the social costmaps representing the human-centric Gaussian cost functions.

$$B_{r_j,P} = \sum_{t_i \in P} (\rho_{t_i,r_j} - \mathrm{D_s}(l_{t_{i-1}}, l_{t_i})) \tag{16.1}$$

ρ_{t_i,r_j} is the revenue of task t_i, l_{t_i} is the position of t_i and $l_{t_{i-1}}$ is the position of the task that appears before t_i in P. This function includes a revenue ρ_{t_i,r_j} that is decreasing with time as shown below.

$$\rho_{t_i,r_j}(k) = \max(0, \varrho_{max}(1 - \frac{k - k_{a,i}}{\tau})) \tag{16.2}$$

where D_s is the length of the human-aware path computed by the FMM planner, k is the time in which t_i is reached, ϱ_{max} is the maximum revenue for the task, $k_{a,i}$ is the allocation time of t_i and τ is the time after which the positive revenue becomes zero. This utility function is added to reinforce reaching the tasks as early as possible. In real noisy social environments, only an estimate of the time to reach a task can be computed. Additionally, due to the presence of people, because of the random interactions they form among themselves and with robots, estimations of the task times can be far from reality for some of the robots and acceptable for those that are not faced with humans. Therefore, for the remaining of



(a)

Figure 16.2 – Snapshot of a sample test for ten moving humans in Webots. There are ten tasks depicted by blue circles representing the locations that should be visited by the robots.

this thesis, we have decided to only include the distance term in the local balance functions.

As for the global balance function concerning MRTA, the total traveled distance (M_1) and the mission time (M_2) are reported for all experiments. For evaluating the performance of the MRS in terms of social-awareness, the maximum accumulated social cost (M_3), the maximum time steps spent in areas associated with social costs (M_4) and the minimum distance to any human throughout the experiment (M_5) are reported among all robots. Three case studies of varying complexity have been addressed in this chapter. The following sections describe them in detail.

16.2.2 Case Study I: Human-Agnostic MRTA in Social Environments

To get a better understanding of how the existing MRTA methods perform in human-populated environments without considering any additional social elements, a series of experiments have been conducted. We have tested five different scenarios in simulation with the following configuration in terms of presence of people, *Scenario 1A:* no humans, *Scenario 1B:* five static humans, *Scenario 1C:* five moving humans, *Scenario 1D:* ten static humans, and *Scenario 1E:* ten moving humans.

Static humans are randomly placed on positions that the robots would have to traverse. Dynamic humans are randomly moving in all parts of the environment while avoiding collisions with other agents. Figure 16.2 shows the faithful simulated model of the oncological ward of the IPOL hospital in Lisbon used in our simulations as a representative complex indoor environment. Robots are relying on their self-localization for computing the local balance functions and the evaluation metrics have been obtained from ground truth values provided by the simulation. Initial positions of robots and humans for all experiment sets are the same. The reason for this choice is, firstly, to compare the same MRTA problem instance across all scenarios and, secondly, to ensure that humans are located at positions that are likely to create social costs for the robots.

In this case study, a total of 10 tasks represented as desired locations in the environment must be visited by four robots. Each scenario has been tested 10 times in simulation and only MRTA metrics, i.e., M_1 and M_2 have been considered. The goal of this study is to provide insight into the MRTA performance and behavior when humans exist in the environment. It is clear that without accounting for the human presence, social constraints have been frequently violated especially as the number of people increased. For the following two case studies, social metrics as well as the standard MRTA metrics have been considered.

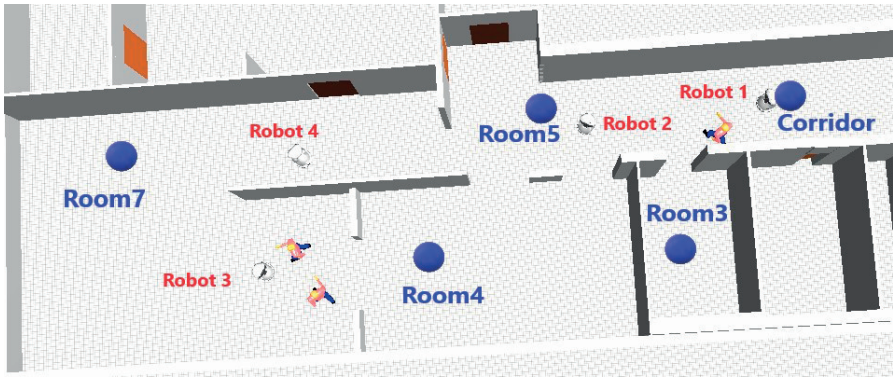


Figure 16.3 – Snapshot of a sample test for one static and two moving humans who are about to start interaction with Robot 3. There are five tasks depicted by blue circles representing the locations that should be visited by the robots.

16.2.3 Case Study II: Comparative Evaluation in Simulation

In this case study, a problem consisting of five tasks, four robots and three humans has been considered. The aim of this assessment is to study the performance of the socially-aware task planner in comparison to the other approaches. This case study demonstrates how team plans can change if social constraints are to be considered and how a robot that is stopped due to social interactions with people, can affect the team performance if a team cooperation strategy is not foreseen for such circumstances. As depicted in Figure 16.3, there exist a static human and two dynamic humans who start moving towards Robot 3, when the simulation begins. They initiate an interaction with the robot and therefore, the robot will be interrupted while moving to its assigned task.

To compare the performance of different MRTA strategies, four series of tests across two scenarios have been conducted. In *Scenario 2A* there are no humans present, whereas in *Scenario 2B* three humans exist in the environment as explained in the case study description. This latter scenario adopts three different algorithms: *Scenario 2B-AG:* human-agnostic robots, in this scenario humans are not considered in the team plan and are only considered as obstacles in robot navigation. *Scenario 2B-AWI:* human-aware robots as individuals, in this scenario humans are not considered in the team plan but navigation considers their presence as described in Section 10.3.1. *Scenario 2B-AWT:* human-aware robots as a team, here humans are considered at both planning and individual navigation levels.

We note that three types of timeouts have been implemented for the following situations, (i) time-out due to social interactions, (ii) time-out due to no progression in the assigned task, and (iii) time-out due to large localization errors and lost robot. We have added the time-out for case (ii) for enabling the robots to finish their mission in case one robot takes too long to accomplish a task due to unforeseen reasons in scenarios other than *Scenario 2B-AWT*. The values considered for these time-outs are 10, 30, and 30 seconds, respectively. The larger values are chosen due to the impact of the time-out; for instance, abandoning a task without coordination with teammates is costly and the algorithm should be more conservative in triggering such action. Furthermore, the time required to identify an interaction with a human is significantly less compared to the time required to conclude that a robot is stuck, lost or down.

The social metrics are obtained excluding the robot stopped due to engaging in interactions with humans. The concept of layered social costmaps in ROS has been used for implementing the social

Figure 16.4 – Snapshot of the real robot experiment.

costs for human-aware navigation. The value of social cost for a given position varies between 0 and 255. The radius of the social costmap is 1 m. Similar to [33], $\sigma_x^2 = 0.255\ m$ and $\sigma_y^2 = 1.5 \times \sigma_x^2$. Intrusion of the areas in front of a human is more severely penalized using this formulation. All four sets of tests have been repeated for ten simulation runs. Robots are relying on their self-localization for computing the local balance functions and the evaluation metrics ($M_1 - M_5$) have been obtained from ground truth values provided by the simulation.

16.2.4 Case Study III: Comparative Evaluation with Real Robots

A similar comparative study of Section 16.2.3 has been conducted in reality. While methodologically this study is exactly the same as that of Section 16.2.3, here the problem instance consists of three tasks, two robots and two humans, as depicted in Figure 16.4. In this test, two static humans are having a conversation while standing in a rather narrow passage. The goal of this assessment is to examine how the social costs associated to humans and their interaction area can change the performance and plans of different MRTA approaches. The placement of tasks and the social costmaps are illustrated in Figure 16.5. Robots are relying on their self-localization for computing the local balance functions. The experiments are carried out in the Jordils motion arena (see Section 7.2) and therefore, the position and orientation of the humans can be captured by the available MCS with millimetric accuracy and broadcasted to the robots. Again, $\sigma_x^2 = 0.255\ m$ and $\sigma_y^2 = 1.5 \times \sigma_x^2$. Experiments have been conducted in an arena corresponding to map I (see Table 7.1).

16.3 Results

In this section the results of the three case studies explained in the previous section will be discussed.

16.3.1 Case Study I: Human-agnostic MRTA in Social Environments

Figure 16.6 shows the performance of this test across different scenarios. The metrics used in this study are M_1 and M_2, similar to Section 15.3.2. It can be seen that both metrics are increased when humans are present. Since the environments are all exactly the same from the point of view of the task planning

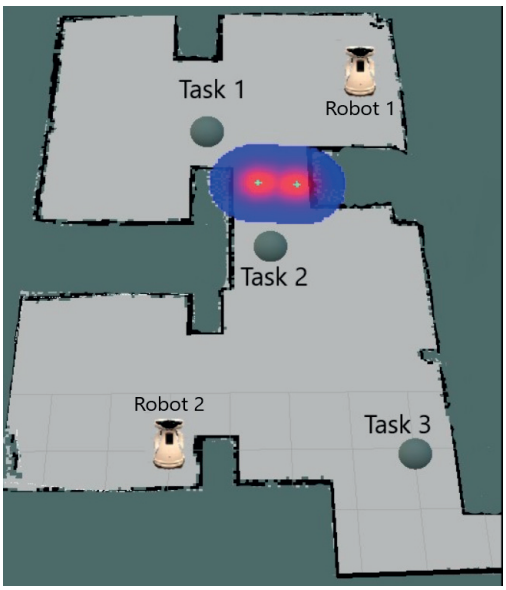
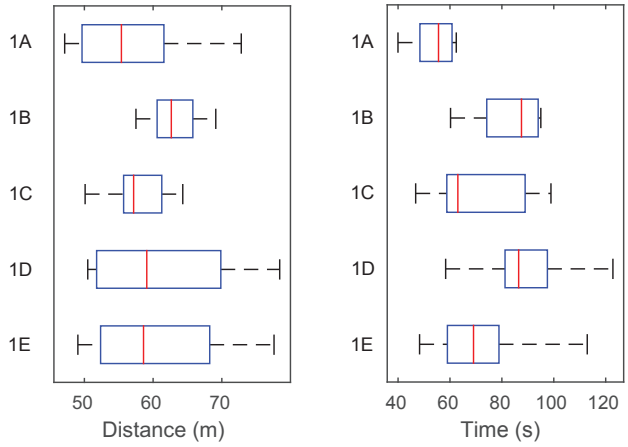


Figure 16.5 – Placement of the tasks, robots and humans at the beginning of the experiment. Humans are facing each other. They are positioned in the center of the costmap. The areas closer to the humans and their interaction space are more costly (red) and the further away we get from the humans the cost will decrease (blue).

method, allocation of tasks remains similar to that of Scenario 1-A and humans affect the timing (M_2) of the tasks mostly as observed in Figure 16.6. This is due to humans appearing as (dynamic) unmapped obstacles that could result in longer modified paths, blocked passages or interfere with the localization of the robots.

Based on Figure 16.6b, the mission time for Scenario 1B and 1D with static people is higher compared to Scenario 1C and 1E with dynamic humans. This may seem counter-intuitive since human motion adds more complexity to the scenario. However, static humans cause a larger increase in the localization error of the robots. Indeed, throughout the runs we observed robots getting stuck in a close proximity of static humans. This happens since the robot thinks it is too close to an obstacle and stops moving, due to the localization error having jumps from time to time. Since static humans do not move, long periods of blockage that can only be broken by means of time-outs occur. M_2 is increased in such cases since the task of the blocked robot is not reassigned to any other team member upon long waits. This is one reason that necessitates modification of team plans in social environments. In the case of moving humans, the robot is able to retrieve its correct position once it receives new information from the laser scans and when it senses a large enough distance to the surrounding obstacles. Thus, the increase in M_1 and more so M_2, is less than the static case.

In our tests, increasing the number of humans (e.g., 1B vs. 1D and 1C vs. 1E) led to larger variations in M_1. The particular cases which lead to blockage, narrow passages and long waits, are the main cause of the larger traveled distances and longer mission times and do not necessarily increase as the number of people increases. Since we positioned the static humans on challenging spots already with five humans (Scenario 1B), the overall difficulty of this scenario was similar to the case of ten static humans (Scenario 1D) in terms of problematic situations. As a result, the mean value of the two metrics are not very different as the number of humans are increased. This result has been certainly biased by the choice of human positions. However, this choice was made to ensure problematic situations exist in all



(a) M_1 **(b)** M_2

Figure 16.6 – The total distance and time for the five different scenarios for human-agnostic MRTA obtained from 10 runs.

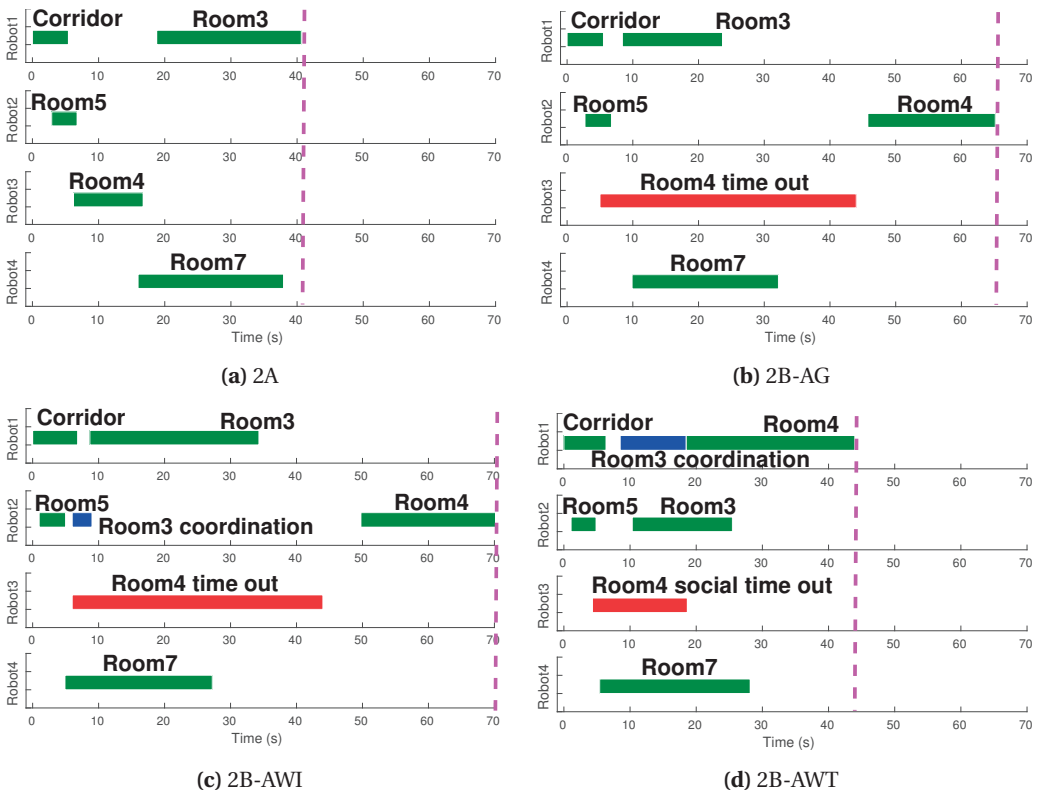
scenarios. Nonetheless, in uncontrolled social environments more problematic situations are likely to happen as the number of humans increases.

16.3.2 Case Study II: Comparative Evaluation in Simulation

To understand the task assignment depicted in Figure 16.7, consider Figure 16.3 again and assume Robot 1 has reached the Corridor task and Robot 2 is at Room 5. For deciding who is going to accomplish the next task, *i.e.*, Room 3, without accounting for the human, the distance between Robot 1 at the Corridor is smaller than that of Robot 2 at Room 5. Therefore, in Scenario 3A and 3B-AG Robot 1 takes Room 3. In Scenario 3B-AWI, Robot 2 delegates Room 3 to Robot 1 through collaboration, despite becoming available earlier in time. In Scenario 3B-AWT on the other hand, Robot 2 takes Room 3 as the result of active coordination between Robot 1 and Robot 2. Note that the team plan changes when social costs are taken into account.

For the second problematic situation leading to time-outs, in Scenario 2B-AG and 2B-AWI, Room 4 will be taken by the closest available robot which is Robot 2 but only after the time-out period. In Scenario 2B-AWT, the moment the social time-out for Robot 3 is issued, replanning through active coordination takes place and Robot 1 which is the closest available robot, will take the task (Robot 2 has been sent to Room 3). This significantly reduces the mission time (M_2) compared to 2B-AWI as seen in Figure 16.8b.

Figure 16.8 demonstrates M_1-M_5 for Scenario 2A to 2B-AWT. For Scenario 2A, we expect to have the smallest M_1 and M_2 due to no humans existing in the environment. This is confirmed by Figure 16.8a-b. In the human-agnostic method, robots do not modify their paths to respect the social spaces around humans and will therefore, travel a smaller distance compared to a human-aware robot for the same plan. Thus, M_1 for Scenario 2B-AG will be smaller compared to 2B-AWI and 2B-AWT, as confirmed by Figure 16.8a. However, the human-agnostic approach leads to large social costs due to not respecting social constraints. This can be observed in Figure 16.8c-e where larger $M_3 - M_4$ are recorded for the human-agnostic method. On the other hand, this method achieve the smallest values on M_5 due to humans being considered merely as obstacles.



(a) 2A

(b) 2B-AG

(c) 2B-AWI

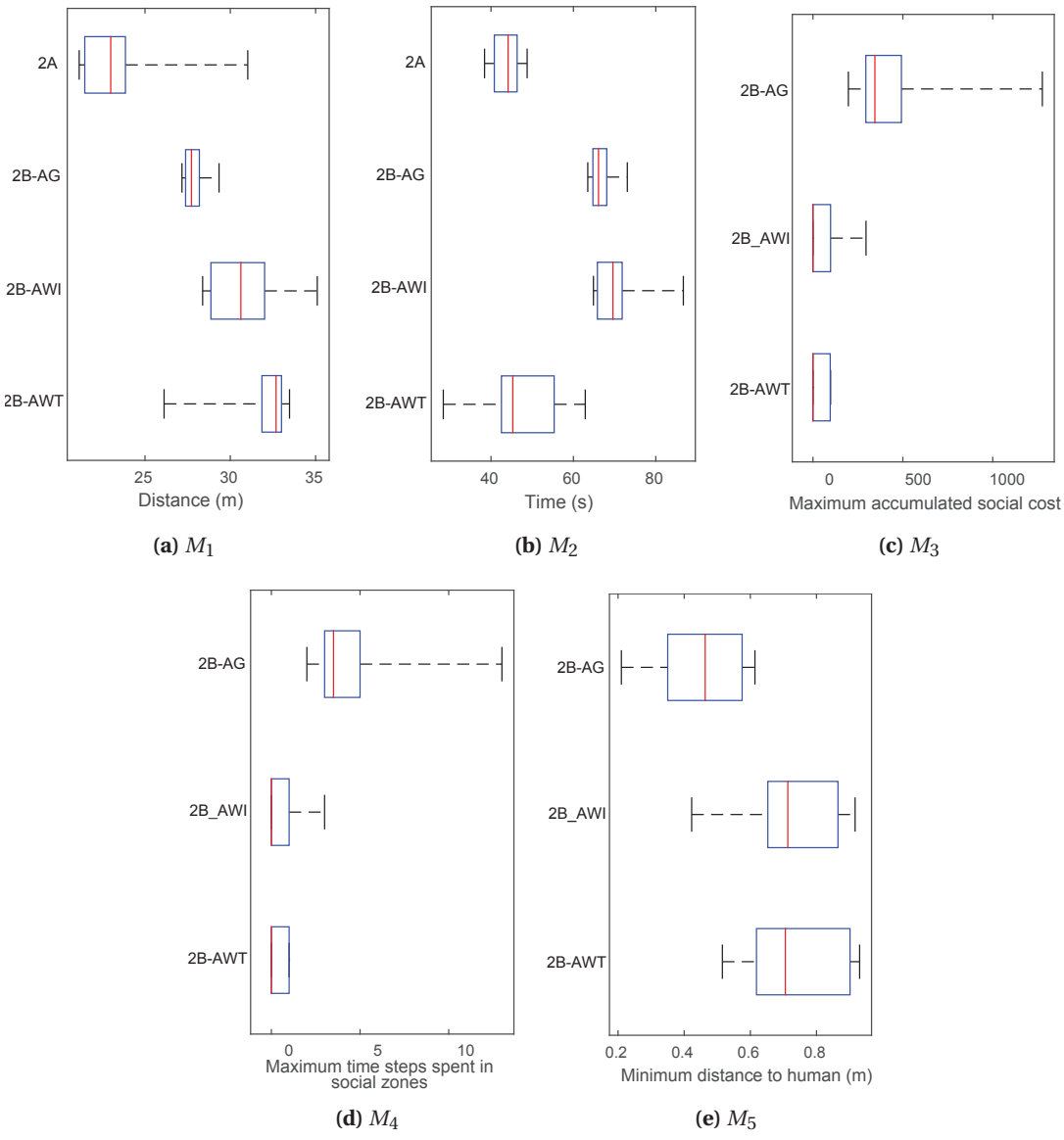
(d) 2B-AWT

Figure 16.7 – Task assignment per robot over time for a sample run of the second set of experiments for scenarios 2A, 2B-AG, 2B-AWI, and 2B-AWT respectively. The blue blocks indicate cooperation requests and coordination and the red blocks represent long waits leading to time-out. End of mission (M_2) is marked by the vertical line.

We can see that both Scenario 2B-AWI and 2B-AWT have negligible social costs and social zone violations. In principle, they should have zero M_3 and M_4, however, small localization errors can lead to occasionally moving into areas with non-zero social costs or generating the false positive assumption of being there. M_5 for both Scenario 2B-AWI and 2B-AWT is in the recommended range and larger compared to that of Scenario 2B-AG. We can see that on average, the robots navigate as close as the edge of the costmap (0.7 m) in Scenario 2B-AWI and 2B-AWT.

16.3.3 Case Study III: Comparative Evaluation with Real Robots

Similar to the previous section, sample team plans for all four scenarios can be found in Figure 16.10. We can see how the team plan changes across different scenarios. Figure 16.10a displays the team plan in a human-free environment where tasks are scored inversely proportional to their distance to the robot. In this experiment, the passage leading from Task 1 to Task 2 is populated by two interacting humans. Thus, large social costs are associated to that area. This causes Task 2 to be very costly for Robot 1 from the point of view of a social coordination mechanism. Additionally, this cost is large enough to stop the robot from moving and hence, causing a time-out. In Scenario 3B-AG and 3B-AWI, Task 2 is assigned to Robot 1 as it is closer to that location (see Figure 16.5). However, since the passage can not be traversed, Robot 2 will take the task only after a progress time-out has been broadcasted.



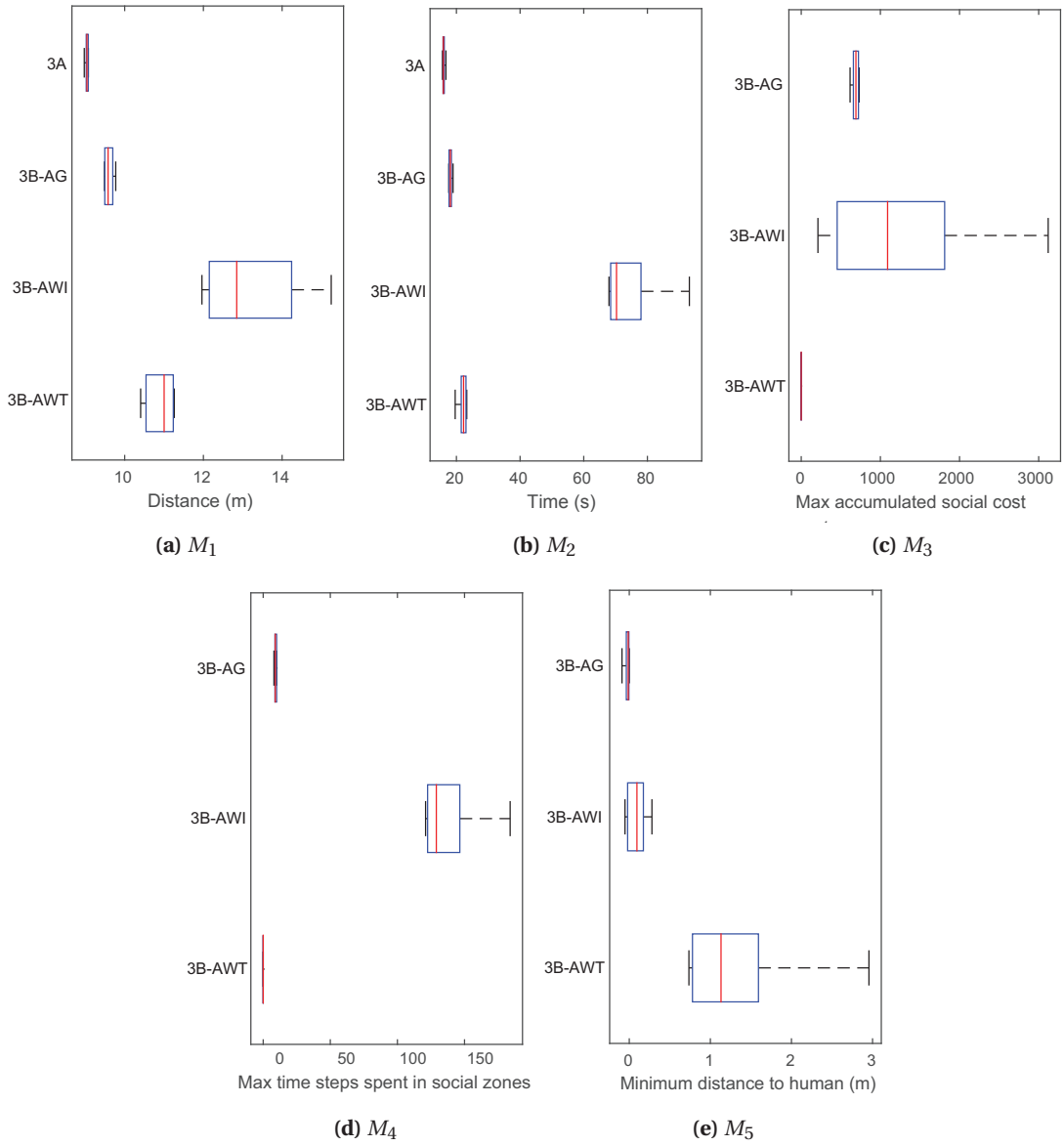
(a) M_1 **(b)** M_2 **(c)** M_3

(d) M_4 **(e)** M_5

Figure 16.8 – Performance metrics for the second set of experiments in simulation obtained from 10 runs. Note that for metric M_3 to M_5, no performance for scenario 2A is plotted since no human was present.

For Scenario 3B-AWI, we observed Robot 1 constantly moving to find a way to Task 2 or slightly moving in social zones and stopping there in some runs. This explains the larger M_1 and M_3 in Figure 16.9.

In Scenario 3B-AWT, Robot 1 will initially take Task 2 since it becomes available before Robot 2. However, upon computing the value for the task a collaboration request is sent and Task 2 is delegated to Robot 2 due to its large social cost. Scenario 3B-AWT is shown to have superior performance in terms of M_1 - M_5 compared to Scenario 3B-AWI while ensuring no social constraints are violated and maintaining a large enough M_5 as shown in Figure 16.8c-e. Not modifying robot trajectories to avoid social zones causes M_1 and M_2 to be smaller for Scenario 3A and 3B-AG at the cost of having large social costs. A video of



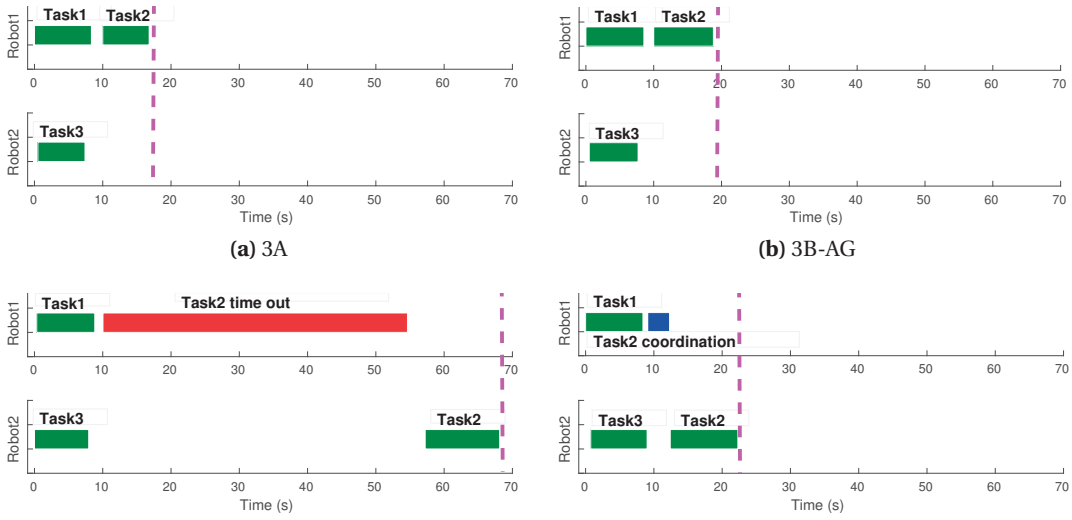
(a) M_1 **(b)** M_2 **(c)** M_3

(d) M_4 **(e)** M_5

Figure 16.9 – Performance metrics for the third set of experiments for real robots obtained from five runs. Note that for metric M_3 to M_5, no performance for scenario 3A is plotted since no human was present.

sample runs of this test as well as further material related to this project can be found following the link indicated in the footnote[1].

[1] http://disal.epfl.ch/research/SocialRoboticsNavigation



(a) 3A **(b)** 3B-AG

(c) 3B-AWI **(d)** 3B-AWT

Figure 16.10 – Task assignment per robot over time for a sample run of the real robot experiment, for scenarios: 3A, 3B-AG, 3B-AWI, and 3B-AWT respectively. The blue block indicates collaboration requests and coordination and the red block represents long waits leading to time-out. End of mission (M_2) is marked by the vertical line.

16.4 Discussion

Evaluating the performance of a human-agnostic MRTA approach in the presence of multiple static and moving people showed how the mission time can considerably increase if humans are present in the environment. Simulation and real robot results confirm that accounting for social costs and time-outs that are the products of social encounters, at both distributed planning and individual navigation levels have superior results in social environments. Human-awareness on an individual level alone is not sufficient for the robots to ensure appropriate social behaviors and in socially blocking situations non-social metrics such as mission time can also be significantly affected if humans are not taken into account on the task planning level.

Social environments can be very dynamic and unpredictable. Therefore, assuming that task costs can be correctly estimated for a given period of time is not always correct. Thus, effective bid estimation for incorporating future actions of the humans and their uncertainty will be studied in the next chapter.

Summary

To understand the effect of humans on MRTA, the impact of a realistic environment with varying number of static and moving humans on the behavior and performance of human-agnostic MRTA was studied in this chapter, using an extensive suite of experiments in simulation. Results show that the total traveled distance and time are increased when humans are present in the environments. Localization noise was also increased particularly in the case of static people. Subsequently, a human-aware MRTA method was proposed in this chapter by means of accounting for social costs in bid evaluations and requesting collaboration in socially blocking situations. For evaluating this method, a number of problematic cases resulting in longer modified paths, blocked passages, and long waits were investigated through a comparative study. Both simulated and real robot experiments confirm the effectiveness of accounting for humans at both team and individual levels. This leads to respecting social constraints as well as achieving a better performance based on MRTA metrics for socially blocking situations.

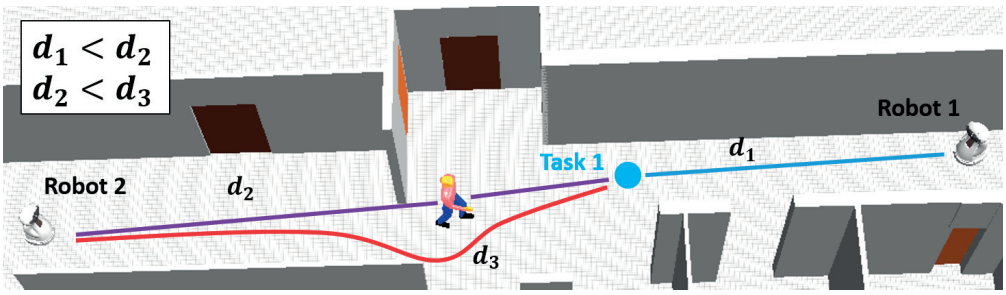
17 Risk-Based Human-Aware Multi-Robot Cooperation in Social Environments

In this chapter, we study MRTA in dynamic social environments with costs that have a stochastic nature due to the changing behavior of people. We introduce risk-based bids that incorporate human trajectory prediction uncertainties and furthermore, social costs in their formulation. In Chapter 15, a study of MRTA in dynamic and noisy environments for spatial task allocation confirmed the effectiveness of our Hoplites-based method as a first step towards deployment of MRTA methods in social environments. In Chapter 16, we proposed a human-aware coordination method for MRTA that accounted for the distance overhead of human-aware paths in the local balance function of the robots and allowed for instant cooperation in socially blocking situations. However, robots could only see a snapshot of the environment and costs were assumed to be deterministic.

To illustrate the problem, consider the example in Figure 17.1, where two robots are coordinating to find the best team plan for taking Task 1. In a social MRTA approach that only considers the current available information when bidding on a task, and given a local balance function that scores tasks inversely proportional to their distance to the robot, Robot 1 will take Task 1. This is because the distance that Robot 1 has to travel (d_1) is smaller than that to be traveled by the other robot (d_2). Additionally, Robot 2 will have to travel an even larger distance (d_3) to avoid the social costs associated to the personal space of the human. However, in reality the human is moving towards Robot 1. This means there will be no social costs associated to the path of Robot 2. Additionally, Robot 1 will have to modify its path in order to respect the personal space of the human and will have to travel a larger distance than planned. If robots consider the future positions of the human while estimating their bids, they will know that Robot 2 is a better candidate for taking Task 1 in this situation.

Accurate estimation of future positions of humans in uncontrolled environments is not possible in reality. Nonetheless, prediction of human motion despite being error prone, can still provide valuable information about the changes that are likely to occur. Taking a decision based on uncertain information can be seen as taking a risk. In the context of MRTA, a risk measure that adopts predictions and captures their errors as uncertainties can be a useful extension to the deterministic estimation of costs. Furthermore, accounting for the added social costs corresponding to risky situations, can help the robots to take more informed and socially-aware decisions.

Unlike most stochastic MRTA approaches, an uncertainty model for uncontrolled social environments is not available unless strong assumptions are made or a data-driven approach, targeting a specific environment, is taken. Additionally, the scale and complexity of the problem is too large for applying POMDP-based solutions. Moreover, each encounter of the robots with humans matters and improving



(a)

Figure 17.1 – Two robots bidding on a task that will have a changing cost over time. The human is initially static. He then decides to move towards Robot 1. This snapshot shows the initial position of the robots and the human for Case Study I in Webots.

the average performance is not the best strategy to gain social acceptance for the robots. As a result, we opt for an approach that uses risk as a heuristic for estimating stochastic costs.

Risk-based navigation (Risk-RRT [8], [21]) has been adopted in human-aware navigation for single robots. Therein, risk, or the probability of collision with objects or intrusion in socially costly areas, along any candidate trajectory is taken into account for selecting an appropriate human-aware path. Inspired by this idea, we propose a number of risk formulations for estimation of stochastic costs for social MRTA in the next sections. These estimations form the basis of the local balance functions for the robots.

The contributions of this chapter include proposing a concept of risk-based human-aware bids that account for changing costs in human-populated environments, evaluating the effect of human trajectory prediction error on risk-based bid estimation and team performance, and proposing different risk formulations to account for the prediction error, risk estimation accuracy, and social costs. To the best of our knowledge, MRTA with stochastic human-aware costs in social environments has not been investigated in the literature.

17.1 Stochastic Risk-Based Bids

Consider a robot bidding on a task in the vicinity of a human. In a case where a static human previously considered by the robot starts moving and clears the robot's path, the initial bid on the task has been an over-estimation. On the contrary, if a moving human suddenly occupies parts of a robot's path or decides to interact with the robot, the initial bid has been an under-estimation. In other words, costs associated with tasks are uncertain and can vary over time. On the other hand, accurate prediction of the future for humans is not possible because of the uncertainties inherent to uncontrolled social environments. Therefore, we propose an abstraction that can extract higher level information from perceptual data, by introducing risks. “Risk” is defined as the probability of occupation of an area with social costs by the robot.

Consider a reformulation of local balance function B introduced in Section 15.1 for a robot r_j and a given plan P at time k shown in the following:

$$\begin{aligned} B_{r_j,P(k)} &= R_{r_j,P(k)} - C'_{r_j,P(k)} \\ C'_{r_j,P(k)} &= C_{r_j,P(k)} + Z_{r_j,P(k)} \end{aligned} \tag{17.1}$$

where R is a generic revenue function, C a generic cost function and Z the penalty for constraint violations. In a human-populated environment, the presence of people can lead to an increase in the cost for a plan compared to the estimated cost of that plan in an empty deterministic and noise-free environment. Similar to the way humans assess situations by means of evaluating the risks against the benefits when taking a decision, the robot will compute a risk-based cost C'' that will be aggregated with a revenue R for a given plan P.

Another term Q is introduced to capture this stochastic cost which is proportional to the risk γ associated with the plan. $f_m(.)$ is a user-defined function used to aggregate the risk with revenue, cost and penalty terms, based on the risk formulation method m.

$$\begin{aligned} B_{r_j,P(k)} &= R_{r_j,P(k)} - C''_{r_j,P(k)} \\ C''_{r_j,P(k)} &= C'_{r_j,P(k)} + Q_{r_j,P(k)} \\ Q_{r_j,P(k)} &= \textstyle\sum_{t_i \in P(k)} f_m(\gamma_{r_j,t_i,k}) \end{aligned} \tag{17.2}$$

When evaluating B_{r_j}, for each task t_i in P the risk of being subjected to additional costs due to human actions must be determined throughout the mission. This risk is computed on the basis of the distance D between the robot and any human present in a predefined vicinity of the robot at any point of time. Let's assume that at time k, t_i will be started in k_1 seconds and will be reached in k_2 seconds. The risk associated with this part of the plan is defined as follows:

$$\gamma(r_j, t_i, k) = \int_{k+k_1}^{k+k_2} \sum_{h \in H} g_m(D(l_{r_j,k'}, l_{h,k'}))\, dk' \tag{17.3}$$

Here, $g_m(.)$ is a function that is inversely proportional to D, *i.e.*, the distance measure between the robot position l_{r_j} and the position of the human l_h. Since risk is defined as the probability of occupation of an area with social costs by the robot, g is chosen to normalize the distance in a predefined vicinity of the human with radius ϵ_R:

$$g_m = \frac{\max\{0, \epsilon_R - D_m\}}{\epsilon_R + D_m} \tag{17.4}$$

where D_m indicates the distance between the robot and a human h, using method m.

The key information for computing risk is the future positions of people. This calls for a human trajectory predictor to make the required information available. Throughout this chapter, we assume that the output of the human trajectory predictor is a Gaussian distribution. $M_{h,k}$ is the mean and $\Sigma_{h,k}$ is the covariance matrix of the distribution. For time k and human h, we will have a predictor that provides us with the following information:

$$l_{h,k} = \mathcal{N}(M_{h,k}, \Sigma_{h,k}) \tag{17.5}$$

The robot position on the other hand, can be given by the path planner. For any given goal (l_{t_i}), we obtain a sequence of way-points $w \in W_{r_j,t_i} = \{w_1, w_2, ..., w_N\}$ that constitute the robot path, with w_0 being the initial position of the r_j. An estimate of the corresponding reaching time for each w is also known $k_w \in \{k_{w_1}, k_{w_2}, ..., k_{w_N}\}$. The robot trajectory is then discretized and k is ignored as it is implicitly accounted for through the use of k_w. The new risk formulation can be written as:

$$\gamma(r_j, t_i) = \sum_{w \in W_{r_j,t_i}} \sum_{h \in H} g_m(D(w, l_{h,k_w})) \tag{17.6}$$

H represents the set of all humans perceived by the robot. Depending on the risk modeling approach, different distance and risk formulations can be chosen. We are interested to know how this choice can affect the performance of the team plan in terms of both MRTA and social metrics. Therefore, we propose the following risk formulations. Details of our proposed generic bid estimation algorithm can be found in Algorithm 6.

17.1.1 Euclidean Distance (D_E)

By choosing this metric, using a simple formulation that requires only the end points without the uncertainty associated with them, we compute the straight-line distance between two points, *i.e.*, way-point and human positions, in Euclidean space. The expected value of the human prediction distribution will indicate the position of the human.

$$\gamma(r_j, t_i) = \sum_{w \in W_{r_j,t_i}} \sum_{h \in H} g_m(D_E(w, \mathbb{E}(l_{h,k_w}))) \tag{17.7}$$

17.1.2 Mahalanobis Distance (D_M)

Since human trajectory predictions provide a Gaussian distribution with a covariance matrix, we do know about the uncertainty associated with every prediction. Thus, a more accurate distance measure can be extracted if this uncertainty is taken into account by means of Mahalanobis distance. The Mahalanobis distance between robot r_j and human h can be written as follows. Note that $w = l_{r_j,k_w}$.

$$D_M(l_{r_j}, l_{h,k}) = \sqrt{(l_{r_j} - M_{h,k_w})^\top \Sigma^{-1}_{h,w_k} (l_{r_j} - M_{h,k_w})} \tag{17.8}$$

$$\gamma(r_j, t_i) = \sum_{w \in W_{r_j,t_i}} \sum_{h \in H} g_m(D_M(w, l_{h,k_w})) \tag{17.9}$$

17.1.3 Integrated Distances

In Sections 17.1.1 and 17.1.2, only way-points from the planner have been considered for risk computation. However, if the granularity of robot path is not fine enough, there are cases where important events can be missed; for instance, robot and human collision or social zone intrusion can happen in between two way-points without the two ends of the trajectory segments being affected by it. Therefore, a piece-wise linear breakdown of the segment between way-points can be considered with a predefined resolution. An example of this situation can be seen in Figure 17.2. The formulation of g_m for D_E and D_M for integrated distances is as follows:

$$g_m(D_E(w, \mathbb{E}(l_{h,k_w}))) = \max_{w_p \in PWL(w_{-1}, w)} g_m(D_E(w_p, \mathbb{E}(l_{h,k_{w_p}}))) \tag{17.10}$$

$PWL(w_{-1}, w)$ represents the piece-wise linear breakdown of the segment between w and its previous way-point w_{-1}. Instead of max operator, summation can also be used to account for the accumulated social costs. Similarly, for D_M integrated risk will make use of the following

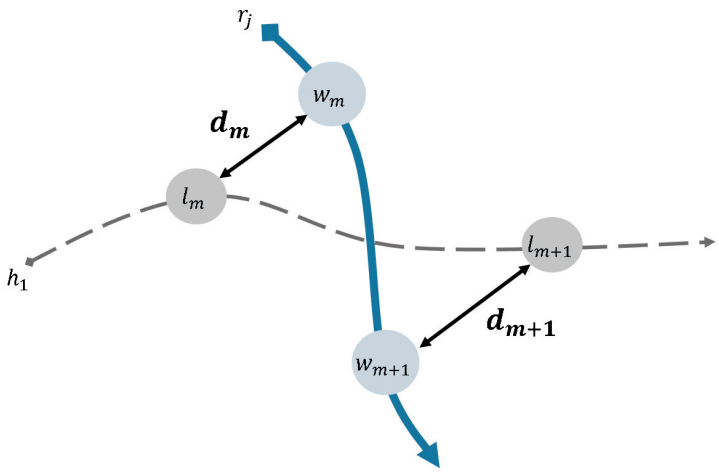


Figure 17.2 – An example of collision between r_j and a human h_1 in between way-points. If r_j relies only on the distances between each way-point w_m and its corresponding human position estimate l_m, it will find that both d_m and d_{m+1} are large enough and the trajectory is not risky. However, when integrating the risk over the entire trajectory segment, a collision is detected.

distance formulation.

$$g_m(D_M(w, l_{h,k_w})) = \max_{w_p \in PWL(w_{-1}, w)} g_m(D_M(w_p, l_{h,k_{w_p}})) \tag{17.11}$$

17.1.4 Social Cost Incorporation

Human presence can affect the cost for a task in a number of ways. The distance traveled will increase, risk of interactions and therefore, incomplete missions are introduced and additionally costs concerning human discomfort or inconvenience can be incurred. By assigning costs on the basis of social costmaps to the risk formulation, social factors are further reinforced.

Expectation-based social costmaps are used to incorporate the uncertainty in the human positions reported by the human trajectory predictor. Consider a human at (x_p, y_p), the deterministic costmap at (x, y) is:

$$S(x, y; x_p, y_p) = N(x - x_p, y - y_p) \tag{17.12}$$

N is the 2D Gaussian modeling the standard social costmaps. The probabilistic costmap is given by the expected value of the social cost S, given the probability distribution of the human's position $p_h(x_p, y_p)$ which in this case belongs to a normal distribution. For any point $w = [x, y]^\top$ in the vicinity of this human, cost can be written as follows:

$$\mathbb{S}(w) = \mathbb{E}(p(x_p, y_p)[S(x, y; x_p, y_p)]) = \int\int N(x - x_p, y - y_p) p_h(x_p, y_p) dx_p dy_p \tag{17.13}$$

This is in fact the convolution of the normal function modeling the human position l_{h,k_w} and a Gaussian social costmap model S centered on the mean of l_{h,k_w} which is $[x_p, y_p]^\top$. Using this approach, we compute an *expected costmap* incorporating all the uncertainties in the environment. Based on this social costmap we now can have a risk value that incorporates social costs directly. The risk formulation for the end point and integrated social costs can be written as the following respectively:

$$\gamma(r_j, t_i) = \sum_{w \in W_{r_j,t_i}} \sum_{h \in H} \mathbb{S}(w) \tag{17.14}$$

$$\gamma(r_j, t_i) = \sum_{w \in W_{r_j,t_i}} \sum_{h \in H} \max_{w_p \in PWL(w_{-1}, w)} \mathbb{S}(w_p) \tag{17.15}$$

We note that our assumption of having a normal distribution for human predictions does not constrain this social risk formulation. Any other distribution or particle-based output can be considered as $p_h(x_p, y_p)$. In case of predictors with discrete output, the same formulation seen in Equation 12.5 can be used.

The potential additional social costs only occur in the vicinity of humans. This means for scenarios with many tasks and many robots, only a subset of robots and tasks which are subjected to social costs are making the difference with non risk-based approaches. The building block of such scenarios, is bidding on one task affected by one human. If robots could improve their estimates of the cost for such a task, they will consequently improve their team level performance since more grounded decisions will be taken. Additionally, by means of an aggregated formulation of risk that accounts for all the humans perceived by the robot, every human is considered when computing the bid and areas containing more humans will be associated a higher risk.

There are two main components to MRTA performing well in dynamic social environments: 1) improved bid estimation, 2) adaptive replanning. In this chapter, we focus on the former since we believe providing a detailed evaluation of different methods for risk-based bid computation is the first essential step for understanding how to approach stochastic social costs for acquiring better team plans. Moreover, reliable risk estimation is the basis for devising adaptive replannig strategies that can accommodate the high dynamics of social environments. Currently, replanning is done when a task is accomplished or for verifying the validity of a stored plan when a robot is on its way towards a task (refer to Section 18.1 for more details).

17.2 Experiments

Two case studies of increasing complexity will be described in this section. In Section 17.2.1, we will investigate the performance of different risk formulations as well as the effect of human trajectory prediction error on the method performance. In Section 17.2.2, a comparative study

Algorithm 6 Risk-based bid estimation of robot r_j for task t_i with a set of humans H

procedure BIDESTIMATION(l_{t_i}, H)
  ▷ Compute the path to t_i assuming an empty map
  $(W_{t_i}, K_{t_i}, d_{t_i}) \leftarrow$ `PathPlanning`(l_{t_i})
  ▷ Initial bid for the task is the distance traveled in an empty map
  $b_{t_i} \leftarrow d_{t_i}$
  ▷ Initially, the last way-point in the robot path is set to its current position
  $w_{-1} \leftarrow l_{r_j}$
  for $w \in W_{t_i}$ **do**
    for $h \in H$ **do**
      ▷ Find the human position at time k_w
      $l_{h,k_w} \leftarrow$ `PredictHumanPosition`$(l_{h,k_{w_{-1}}}, k_w)$
      ▷ Estimate the risk of h for r_j between w_{-1} and w
      $\gamma_{h,w} \leftarrow$ `EstimateRisk`$(w_{-1}, w, l_{h,k_{w_{-1}}}, l_{h,k_w})$
      ▷ User-defined function for aggregating risks with the distance-based bid
      $b_{t_i} \leftarrow$ `AggregateRisk`$(\gamma_{h,w}, b_{t_i})$
    ▷ Store the current way-point as the last visited way-point for the next segment
    $w_{-1} \leftarrow w$
  return b_{t_i}

targeting 1) human-agnostic navigation and planning, 2) human-aware navigation without considering humans in the planning phase, 3) human-aware navigation and planning based on deterministic costs, and 4) human-aware planning based on stochastic costs without any individual human-aware navigation will be explained.

We note that in each case study, human trajectories are the same across all runs with different methods. This choice has been made to ensure that we are comparing the same MRTA problem instance. Each scenario has been repeated for ten simulation runs. Robots are relying on their self-localization for computing the local balance functions and the evaluation metrics ($M_1 - M_5$) have been obtained from ground truth values provided by the simulation. Similar to the Chapter 16, for the global balance function concerning MRTA, the total traveled distance (M_1) and the mission time (M_2) are reported for all experiments. For evaluating the performance of the MRS in terms of social-awareness, the maximum accumulated social cost (M_3), the maximum time steps spent in areas associated with social costs (M_4) and the minimum distance to any human throughout the experiment (M_5) are reported among all robots.

For computing social metrics, we rely on Gaussian social costmaps. Similar to [33], we have chosen $\sigma_x^2 = 0.255\ m$, and $\sigma_y^2 = \sigma_x^2$ since we don't include orientation in our predictions. The value of social cost for a given position varies between 0 and 100 and the radius of the social costmap is 1 m.

17.2.1 Case Study I: Risk Formulation and Trajectory Prediction Analysis

This case study consists of two robots, one task and a dynamic human that causes the cost of the task to change over time (see Figure 17.1). The goal of this case study is to investigate

the main challenging aspect of changing costs, *i.e.*, impact of the human. This case study serves as the building block for more generalized cases with more robots and more humans. In the presence of multiple people, each robot will perform the same operation to compute an accumulative risk accounting for every perceived human.

For this case study, we have only included the point-based risk methods, since with 0.25 m granularity of the way-points in robot path planning, there is no need for opting for an integrated risk model in our problem.

To study the effect of prediction error on the performance of our method, different levels of noise have been tested for each risk formulation. This is done by means of adding noise to the ground truth trajectory of the human, *i.e.*, sampling a point l_h from a Gaussian distribution $\mathcal{N}(M_{GT}, \Sigma_{Noise})$ with:

$$M_{GT} = [x_{GT}, y_{GT}]^\top \quad , \quad \Sigma_{Noise} = \begin{bmatrix} \sigma^2 & 0 \\ 0 & \sigma^2 \end{bmatrix} \tag{17.16}$$

x_{GT} is the x component and y_{GT} is the y component of the true human position. The output of each prediction is another Gaussian distribution $\mathcal{N}(l_h, \Sigma_{Noise})$. When adopting a KF for human position prediction, a state consisting of $[x, y, v_x, v_y]^\top$ is tracked and $[x, y]^\top$ is observed. x and y are the x and y components of the human position in the world frame, respectively, and v_x, v_y are the corresponding velocity vector components.

A constant velocity dynamics model is assumed for the human motion. The observation noise is assumed to be 0.05 m for both x and y. The process noise is considered to be 0.1 m for x and y, and 0.5 m/s for v_x and v_y. As a result, predicting further into the future is subject to larger errors and the uncertainty grows dynamically when using a KF. For the cases where noise

Table 17.1 – Details of different scenarios in case study I. Noise stands for the human trajectory prediction noise.

Scenario	No. of Humans	Risk Method	Noise σ (m)
1A	0	-	-
1B-AG	1	-	-
1B-E0	1	Euclidean	0.0
1B-E0.5	1	Euclidean	0.5
1B-E2	1	Euclidean	2.0
1B-E-KF	1	Euclidean	KF
1B-M0.5	1	Mahalanobis	0.5
1B-M2	1	Mahalanobis	2.0
1B-M-KF	1	Mahalanobis	KF
1B-S0	1	Social costmap	0.0
1B-S0.5	1	Social costmap	0.5
1B-S2	1	Social costmap	2.0
1B-S-KF	1	Social costmap	KF

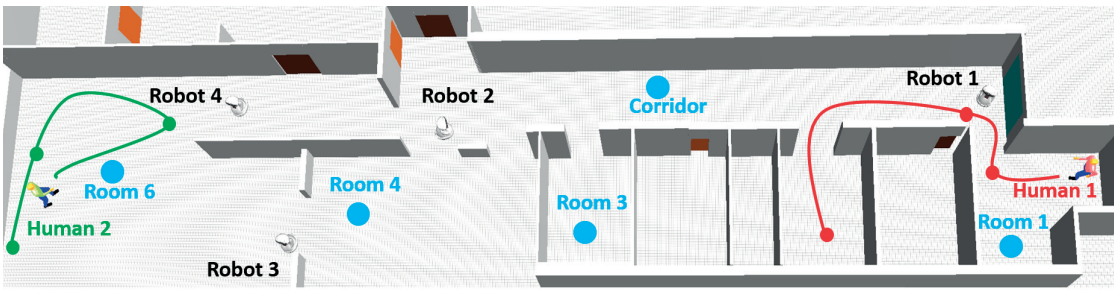


Figure 17.3 – Initial position of the robots and humans in case study II in Webots. Blue circles depict the location of tasks and smaller circles on human trajectories represent brief pauses.

is added to the ground truth however, the uncertainty is bounded to the noise distribution. Table 17.1 lists the details of each scenario in terms of noise-level and risk method. We note that the human performs noise-free and perfectly repeatable actions over different runs, by means of replaying a recorded rosbag. Therefore, the noise in the human position prediction is only caused by the prediction error of the estimator.

17.2.2 Case Study II: Comparative Evaluation of Different MRTA Strategies

In this case study, a problem consisting of five tasks, four robots and two dynamic humans has been considered. Fig 17.3 illustrates the initial position of the robots, task placement and human trajectories. The aim of this assessment is to study the performance of the risk-based social task planner in comparison with the other approaches in a more complex setting. This case study demonstrates how team plans can change if future risks are to be considered and how social costs can be significantly reduced by means of team plans that avoid social risks.

To compare the performance of different MRTA strategies, four series of tests across two scenarios have been conducted. In *Scenario 2A,* there are no humans present, whereas in *Scenario 2B,* two moving humans exist in the environment as explained in the case study description. This latter scenario adopts three different algorithms. *Scenario 2B-AG,* human-agnostic robots; in this scenario humans are not considered in the team plan and are only considered as obstacles in robot navigation. *Scenario 2B-SD,* social deterministic costs and individual human-aware navigation; here humans are considered in task planning but through a social planner that takes decisions based on currently available information only. *Scenario 2B-SR-KF,* risk-based social planning without individual human-aware navigation; in this scenario a KF predictor is used for bid estimation. Among different risk formulations, S-KF was shown to be the most effective (see Section 17.3.1), hence, we selected this method for further evaluation. We chose to decouple the risk-based social planner from individual human-aware navigation to highlight the strengths of this planning approach and assess the contribution of risk-based bids in finding better plans and reducing social costs.

17.3 Results

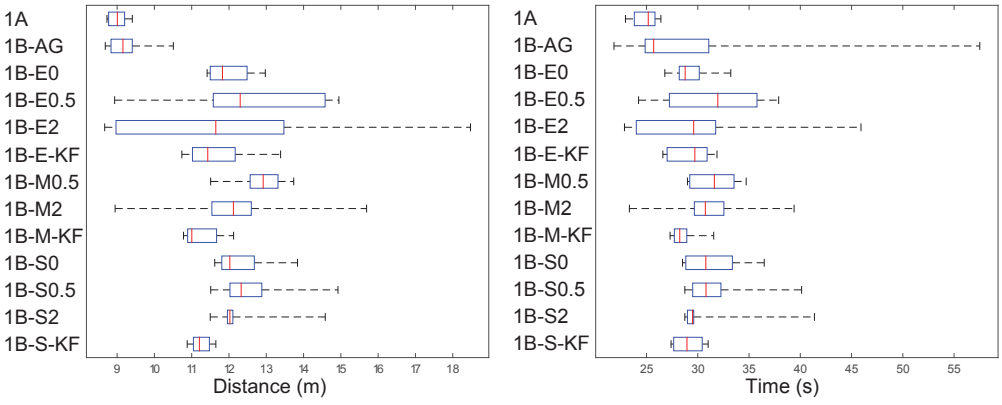
In this section the results of the two case studies explained in the previous section will be discussed.

17.3.1 Case Study I: Risk Formulation and Trajectory Prediction Analysis

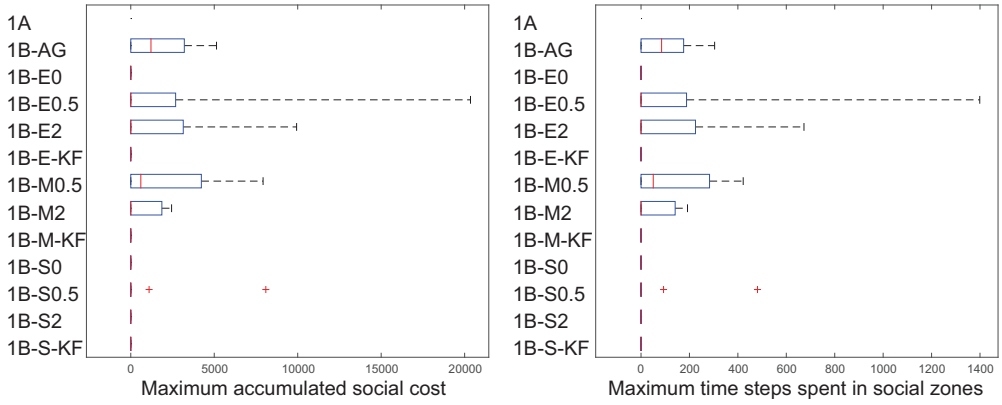
Consider Figure 17.1 again. Upon initial biding of both robots, Robot 1 is the closest to "Task 1" and will therefore be allocated to this task in scenario 1A and 1B-AG. However, when future risks are taken into account in task planning, in all other scenarios Robot 2 will request a cooperation and take "Task 1". Social task planning based on deterministic costs has not been included in this case study since it is clear that with the added distance overhead of social paths (see d_3 in Figure 17.1), Robot 2 will not be able to overbid Robot 1. For this case study, plans are not included for brevity and we focus on the extracted metrics instead. Figure 17.4 shows the performance of this case study across different scenarios. For scenario 2A, we expect to have the smallest M_1 and M_2 due to no humans existing in the environment. This is confirmed by Figure 17.4a-b. Scenario 1B-AG has similar performance to 1A in terms of $M_1 - M_2$ on average. However, without individual human-aware navigation we observed Robot 1 having a mild or major collision with the human every few runs. This resulted in pauses and localization errors for the robot and hence the increased variability in M_1 and M_2 compared to scenario 1A, despite following the same task assignment. Lack of human-awareness has led to large social costs for scenario 1B-AG as seen in Figure 17.4c-e.

For risk-based bidding methods, we can observe zero socials cost on average, and they maintain an appropriate distance to the human across all tests despite noisy predictions. The only exception here is 1B-M0.5 where Robot 2 was able to persuade Robot 1 later than other scenarios in five of the runs. As a result, Robot 1 stopped at a position that would be sometimes too close to the human trajectory and without activating human-aware navigation or social collision avoidance, Robot 1 ended up being in the human's social zone. Similar behavior was observed for fewer runs in scenario 1B-E0.5 and scenario 1B-E2.

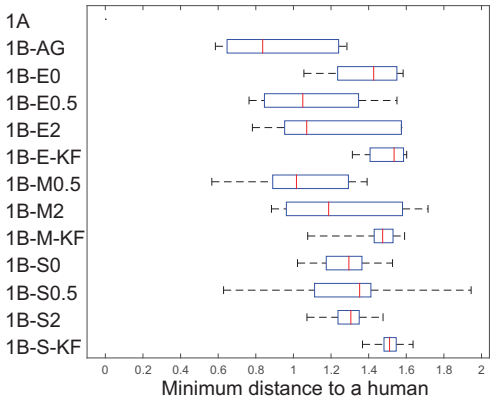
All scenarios with zero prediction error, KF predictor, and risks that are based on social costmaps, have resulted in zero social costs. We believe the reason is that social costs reinforce the risk in areas closer to the human, *i.e.*, a larger penalty (based on the social costmap model) is assigned to all areas associated with possible human presence. Despite large prediction errors, particularly for KF, we observe that the correct decision has been taken by the robots. We believe that extracting a direction from predictions is the key for improving the bid estimates here. This is an abstraction compared to accurate position estimation. However, this information can be used to improve team plans in many cases. Based on the results of this case study, we choose the risk formulation that was based on social costmaps with a KF human trajectory predictor for our risk-based bid estimations in the next case study. We note that other scenarios with frequent sudden changes in human trajectories or more complex dynamics can be found where this solution alone will not be sufficient. This is where monitoring risk variations and



(a) M_1 **(b)** M_2



(c) M_3 **(d)** M_4



(e) M_5

Figure 17.4 – Performance metrics for the first set of experiments obtained from 10 runs. Note that for metric M_3 to M_5, no performance for scenario 1A is plotted since no human was present.

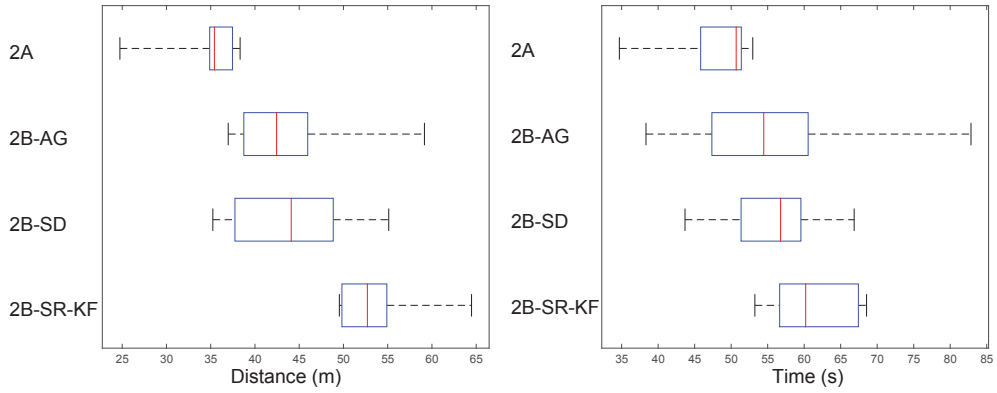
adaptive replanning can help. We will pursue this further in the next chapter.

17.3.2 Case Study II: Comparative Evaluation of Different MRTA Strategies

Looking at Figure 17.3, for a non risk-based task planning strategy, movement of Human 1 and Human 2 causes change to the initial estimated costs of "Room 1" and "Room 6", respectively. Sample team plans for all four scenarios are shown in Figure 17.6. We can see how the team plan changes for scenario 2B-SR-KF compared to the other scenarios. For a human-free environment where tasks are scored inversely proportional to their distance to the robot and given the proximity of robots to tasks, "Room 1" is the best choice for Robot 1 and "Room 6" is the best option for Robot 4. This is confirmed by the team plans (see Figure 17.6a and 17.6b) for both scenario 2A and scenario 2B-AG. In scenario 2B-SD humans are taken into account when computing bids by means of including the distance overhead of the social paths. We can see that despite this consideration, the team plan does not change (see Figure 17.6c) and instant human-aware decisions are not able to find the more appropriate task assignment. Scenario 2B-SR-KF on the other hand, computes risk-based bids and includes future estimates of human motion in decision making. Thus, a different and less socially intrusive plan is found for the team as depicted in Figure 17.6d.

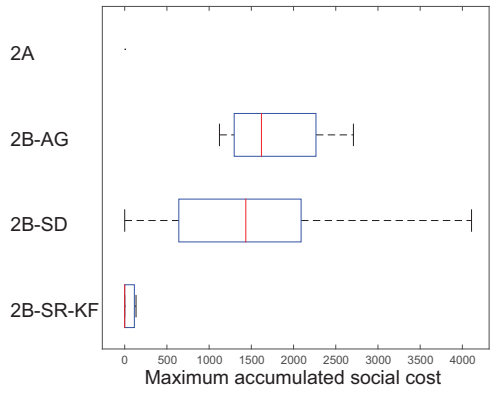
Figure 17.5 demonstrates M_1-M_5 for scenario 2A to 2B-SR-KF. It can be seen that scenario 2A has the smallest M_1-M_2 due to lack of human presence. Slightly larger M_1-M_2 can be seen for 2B-AG and 2B-SD with large variations in M_2. This is again caused by the minor or major collisions with humans. We observed that despite having human-aware navigation for robots in 2B-SD, robots could not fully eliminate social costs, although $M_3 - M_5$ has been improved for 2B-SD compared to 2B-AG on average. We observed two problematic situations for the robots in term of human-aware navigation. Firstly, when Robot 1 is moving to "Room 1", Human 1 is about to exit the room. The doorway is a very narrow passage that does not allow for both of them to progress. This is a hard situation for human-aware navigation, and we observed that Robot 1 partially intruded the social space of the human with variable severity in different runs. Secondly, there were a few cases of collision when a human was moving too fast for the robots to be able to adjust their paths and our individual human-aware navigation was not able to ensure respecting the social constraints in more difficult and dynamic situations.

Contrarily, in scenario 2B-SR-KF because of avoiding risky areas and despite having no individual human-aware navigation, robots were able to ensure social-awareness. Risk-based planning does not promise less traveled distance or less time (see Figure 17.5a and 17.5b). Depending on the human behavior it might find a plan that is longer or shorter in terms of distance and time, but it ensures that decisions will be made considering all aspects (including social costs) together.

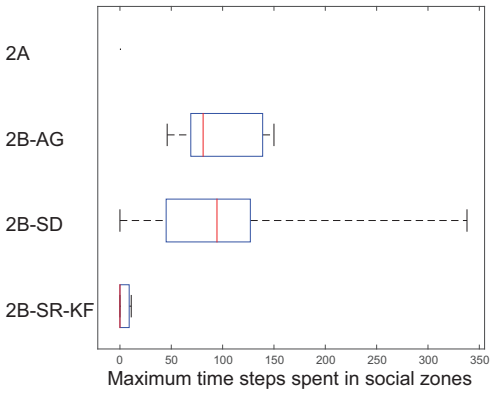


(a) M_1

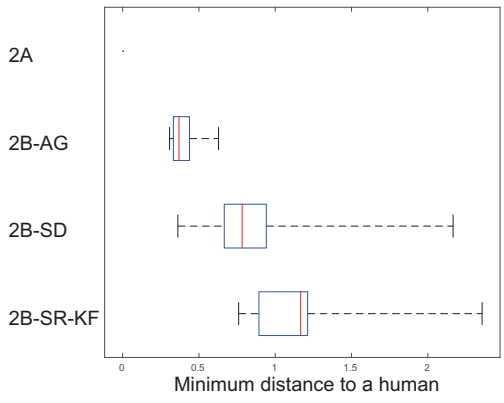
(b) M_2



(c) M_3

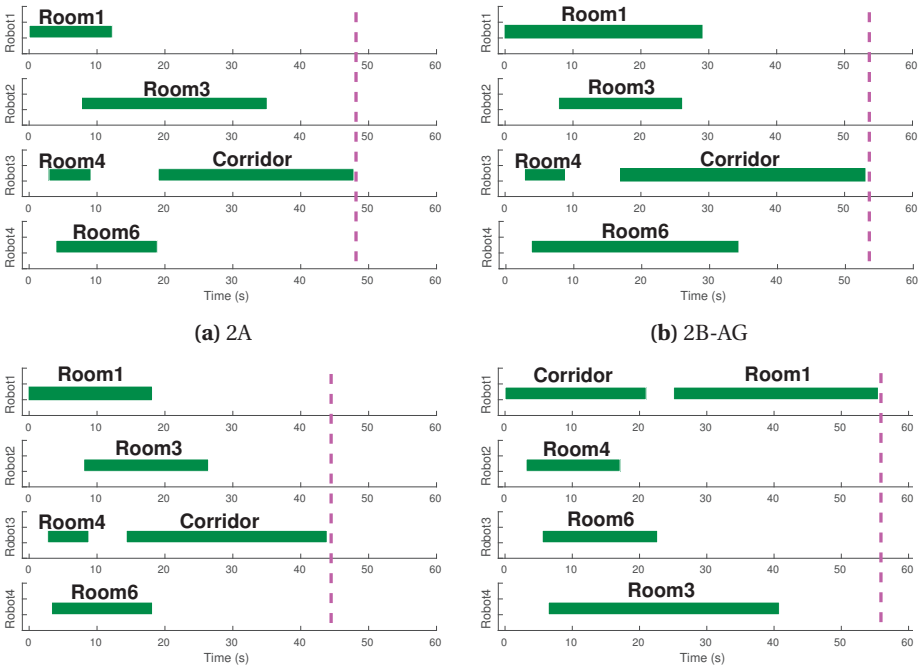


(d) M_4



(e) M_5

Figure 17.5 – Performance metrics for the second set of experiments obtained from 10 runs. Note that for metric M_3 to M_5, no performance for scenario 2A is plotted since no human was present.



(a) 2A **(b)** 2B-AG

(c) 2B-SD **(d)** 2B-SR-KF

Figure 17.6 – Task assignment per robot over time for a sample run of the second set of experiments for scenarios 2A, 2B-AG, 2B-SD, and 2B-SR-KF respectively. End of mission (M_2) is marked by the vertical line.

17.4 Discussion

Results confirm that risk-based plans that account for social costs lead to better team plans in term of social metrics, prevent difficult social situations, and reduce the need for the lower level human-aware navigation to be activated. Although risk-based planning alone was able to achieve socially acceptable results in our case studies, the combination of risk-based human-aware coordination and planning, and human-aware individual navigation ensures that social constraints will be respected even if higher level plans incur some social costs due to yet unpredictable changes in the environment or other sources of uncertainty. In the next chapter, we will focus on adaptive risk-based replanning that can improve the team performance by correcting inaccurate or invalid estimates that can occur as a result of sudden changes in the environment. Additionally, we will look into MRTA with limited local perception and information sharing among robots.

Summary

In this chapter, we proposed a risk-based bid estimation method for MRTA with stochastic costs in dynamic social environments. We investigated the effect of prediction error on the performance of different risk formulations and demonstrated the effectiveness of including a predictive component in the risk formulation despite the lack of accurate position estimation for humans by means of testing different levels of prediction error for known human trajectories and in a separate approach, using a Kalman filter for human trajectory estimation. Furthermore, we proposed different risk formulations accounting for prediction error, risk estimation accuracy, and social costs. Additionally, a comparative study targeting human-awareness on individual and task planning levels was conducted. Results confirm that risk-based bids lead to more socially acceptable team plans that account for social costs and prevent difficult social situations that can lead to less effective human-aware navigation, such as traversing narrow passages occupied by humans.

18 Adaptive Risk-Based Replanning for Social Robots with Limited Local Perception

One key element of dynamic social environments is change. In such uncontrolled environments human behavior can change at any time after a robot has taken a decision. This can lead to degraded or invalid plans. Taking predictions into account can alleviate this problem but cannot eliminate it fully. Particularly for robots with limited perception new information can be perceived at any point in time after a decision has been made and while a robot is executing its plan. In this chapter, MRTA for robots with limited local perception is addressed. Additionally, we target the problem of dealing with unpredictable changes in the environment. The questions we are aiming to answer in this chapter are the following: (i) how should a robot or a team of robots deal with the new or lost perceptual information that is inherent to limited local perception? (ii) how should a robot react to unforeseen behavior changes of humans at the task planning level?

We believe both of these problems have a similar solution. As behavior change induces unpredicted motion and limited perception provides new information or lack of previously present information, what we in fact need to answer is how should MRTA methods employ the new information about humans in a highly stochastic social environment. In other words, how should a team of robots perform replanning when faced with highly stochastic human-related information?

This is among the challenges of MRTA that still remain open and the best approach for facing uncertainty in this context is not currently known. In [52] such dilemma is summarized as follows: “Is it more beneficial to build a complex model that incorporates uncertainty, or is it enough to build less well-informed plans and replan as often as needed to quickly react to unexpected events?” We believe that a hybrid approach can also be taken for tackling this problem. Leveraging the concept of risk-based bid estimation for human-aware coordination introduced in Chapter 17, we propose an Adaptive Risk-Based Replanning (ARBR) strategy for handling new information and unpredicted human behavior. ARBR enables the robots to modify their active plans by incorporating the new relevant updated information about humans in a distributed fashion. We perform an extensive suite of experiments in simulation and reality to evaluate the performance of the MRS employing this strategy.

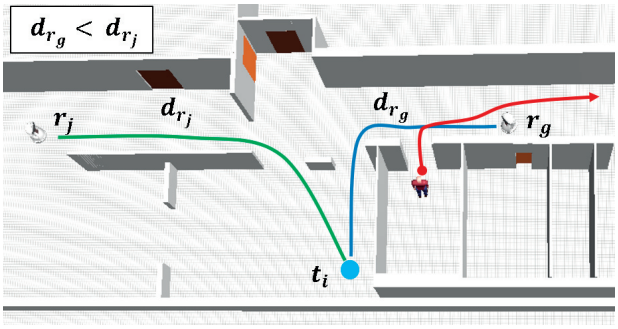


Figure 18.1 – A scenario where multiple active coordination attempts are required to find the best plan.

18.1 Replanning

The strategy currently adopted by the robots is to replan when (i) a task is accomplished and the robot is ready to take its next task, and (ii) for verifying the validity of a stored plan when a robot is on its way towards a designated task. As long as an unfinished task t_i exists, robots perform replanning even if the remaining task is assigned to another robot r_j. If at any point in time replanning results in another robot r_g to be the best candidate for accomplishing t_i, active coordination is executed and t_i is delegated to the newly chosen robot r_g. This can be seen in Figure 18.1 where t_i is initially assigned to r_j at time k as r_j plans first. However, as $d_{r_g} < d_{r_j}$ and $c_{t_i,r_g} < c_{t_i,r_j}$, robot r_g will request active coordination in its turn.

Regardless of the frequency at which replanning takes place, there is a constraint imposed by active coordination that prevents a robot from modifying its plan in some cases. Active coordination can only be done once for a given task t_i between two robots r_j and r_g. When a robot accepts to participate in active coordination, it is bound by contract to do as promised. This is part of the Hoplites framework design to ensure that when the requesting robot r_g pays a compensation price to a robot r_j engaged in active coordination, an agreement is made based on which the desired task t_i will be assigned to the requesting robot r_g. In an environment with deterministic costs, this choice does not limit the robots.

However, in a highly stochastic social environment, the state of the environment in terms of social costs can change at any time. Therefore, the initial estimation of costs based on which active coordination has taken place can be incorrect.

Corollary 1. Given a task t_i and robots r_j and r_g, with $s_{t_i,r_j,k} \in \{0,\ 1\}$ indicating the assignment of t_i to r_j at time k, limiting active coordination to one attempt leads to suboptimal plans if:
$s_{t_i,r_j,k} = 1, \quad c_{t_i,r_g,k_b} < c_{t_i,r_j,k_b}, \quad c_{t_i,r_j,k_{b'}} < c_{t_i,r_g,k_{b'}}, \quad k < k_b < k_{b'}$

Proof. This is a proof by contradiction using the example depicted in Figure 18.1. We assume that one attempt of active coordination is enough for finding the best candidate among two

robots when the above conditions hold. When r_g performs task allocation at time k_b the human is static and $c_{t_i,r_g,k_b} < c_{t_i,r_j,k_b}$. Thus, t_i is assigned to r_g through active coordination as explained previously. However, after r_g starts its plan the human starts moving. Meanwhile, r_j is replanning repeatedly as it has no tasks to accomplish.

Using a risk-based bid estimation, at time $k_{b'}$, robot r_j realizes that with the moving human $c_{t_i,r_j,k_{b'}} < c_{t_i,r_g,k_{b'}}$ and therefore, r_j should request active coordination and take the task. However, the currently active robot r_g will never delegate t_i to r_j and will continue moving to t_i since r_g has already paid r_j a compensation price. This will lead to a suboptimal plan by imposing the constraint of having only one active coordination attempt. ■

This is a limitation that must be addressed for stochastic environments. In the context of our risk-based replanning, adopting this constraint can lead to a deadlock since when r_g senses the human motion, ARBR is triggered and r_g will stop and cancel t_i. If r_g is stopped and knows r_j should take t_i but does not allow it, none of the robots will progress and thus a deadlock occurs for as long as $c_{t_i,r_j} < c_{t_i,r_g}$.

We will address this problem by means of identifying cases in which active coordination must be permitted through risk monitoring. Our proposed risk-based replanning procedure will be working alongside the replanning method previously available.

18.2 Adaptive Risk-Based Replanning

Imagine a crowded corridor at a hospital and a team of robots with limited local perception. As the robots moves, they can observe many people. There may be humans who are newly perceived. Some of the previously tracked human targets may no longer be observed. For the perceived humans who are still observed by the robots, there can be a subset of people who are behaving as expected and a number of them who are behaving in an unexpected way. All these humans pose some degree of risk to the robots. This risk is highly stochastic and must be monitored to ensure that the robots are aware of changes in the environment.

As these social environments are very dynamic, the appropriate use of information updates is key in having a good performance. However, not every new update requires the robots to modify their plans and only a subset of these updates are relevant to the robot team. Having a replanning strategy that is activated upon every arrival of new information is very suboptimal. Replanning with a specified frequency is also prone to low performance: if the frequency is chosen to be too low with respect to the changing dynamics of the environment, the robots might not be able to react to the changes in a timely fashion; if the frequency is too high, this will result in a very resource consuming operation.

We would like to devise an adaptive replanning strategy that avoids replanning when unnecessary while being able to correctly identify when team plans should be revisited. Our goal is to determine when replanning is needed based on risk and human motion prediction

uncertainty. This strategy is realized by means of (i) information sharing, (ii) monitoring the social risks, (iii) risk-based rebidding when necessary, and (iv) active coordination.

18.2.1 Information Sharing

As most robotic systems only have limited and local perception in reality, they also have only a partial view of the environment. One advantage of multi-robot systems is the increased coverage of the environment in terms of gathering perceptual data. When robots are distributed in different parts of a large social environment, they can provide important information to other team members about areas that may be out of their reach in term of perception. This can be a valuable asset to the robot team that helps to take better decisions for task assignment.

For estimating the stochastic component of costs in social MRTA, the main information to be communicated to team members is related to humans. For every perceived human h this information must include the human pose l_h. Other relevant information such as human velocity v_h, the interactions that the human is involved in I_h, etc., can all be reconstructed based on l_h with sufficiently fast perception updates. Hence, this information is adequate for risk monitoring and decision making of the robots. Moreover, when multiple robots can directly perceive a human, the shared data can be made more accurate through data fusion. In our experiments, we implemented information sharing for a robot as sending pose information of directly perceived human targets to all team members and receiving pose information of human targets only perceived by other team members.

Extending the perceptual domain of the team is not the only reason why information sharing is vital. If robots are taking decisions based on different assumptions to fulfill a collective goal, they must make sure there are no discrepancies between the information that form the basis of their decisions. If such discrepancies exist, suboptimal decisions will be taken and specifically in our implementation of risk-based replanning, deadlock can occur.

Corollary 2. ARBR for robots with limited local perception is prone to deadlock with no information sharing.

Proof. This is a proof by contradiction using the example depicted in Figure 18.1. We assume that for robots r_j and r_g with limited local perception and no information sharing deadlock cannot occur. We note that in Corollary 1, robots were assumed to have a global view of the environment. At time k_b, task t_i is assigned to r_g as $c_{t_i,r_g,k_b} < c_{t_i,r_j,k_b}$. At time $k_{b'}$, human h is detected only by r_g causing an increase in $c_{t_i,r_g,k_{b'}}$. This increase is exclusive to $c_{t_i,r_g,k_{b'}}$ and can only be observed by r_g directly. We assume that the increased cost satisfies $c_{t_i,r_g,k_{b'}} > c_{t_i,r_j k_{b'}}$. As a result, ARBR will trigger replanning and r_g stops as long as $c_{t_i,r_g,k_{b'}} > c_{t_i,r_i,k_{b'}}$.

While r_j is not assigned to any tasks, it replans frequently for the only remaining unfinished task t_i. This robot will only take t_i if $c_{t_i,r_g,k_{b'}} > c_{t_i,r_j,k_{b'}}$. Given that the local social cost induced by h is only known to r_g and this information is not communicated to r_j, from the point of view

of r_j task t_i is better suited for r_g since $c_{t_i,r_g,k_{b'}} < c_{t_i,r_j,k_{b'}}$. Thus, r_j will remain idle. This leads to a deadlock for as long as $c_{t_i,r_g} > c_{t_i,r_j}$ and h is only observed (and not communicated) by r_g. ■

18.2.2 Risk Monitoring

How do we decide if we should stop, step aside or continue walking when moving towards a destination in a crowded environment? We observe, predict and take an action. If something unexpected happens we are ready to adapt to the situation either by adjusting our path, waiting or completely changing our route. What motivates us to modify our action at any point in time is the cost we estimate in an uncertain situation.

In the context of MRTA, robots can take a similar approach by monitoring the social risk of every unfinished task in the environment locally. If risk variations are large, robots would be ready to revisit their decisions and adapt to the new state of the environment. In other words, a past decision made by a robot is only revisited if the risk has significantly changed since then. In this case, the robot will evaluate the impact of this risk variation. Should the changes modify the task assignment for the robot, it will stop executing its current plan since it is no longer as profitable as estimated. The robot will then replan by means of performing active coordination with a subset of team members that may be interested.

Without considering the prediction uncertainty of human behavior this strategy can be wrong. To explain this we revisit the definition of social risk γ for robot r_j and task t_i introduced in Section 17.1.4.

$$\gamma(r_j, t_i) = \sum_{w \in W_{r_j,t_i}} \sum_{h \in H} \int\int N(x_w - x, y_w - y)\, p_h(x, y)\, dx dy \tag{18.1}$$

Imagine a robot approaching a static human while moving towards its destination (location of t_i) at time k. As that robot gets closer to the human, the prediction horizon of the social risk associated to t_i will be smaller. As a result, the position of the human will be predicted with higher certainty at this time k'. Thus, for any given point (x, y) in the vicinity of the human we have $p_h(x, y)_{k'} > p_h(x, y)_k$. Consequently, the social risk for (x, y) increases as well and a larger cost will be associated to the areas around the human compared to when the robot was further away. This means that the social risk of an already taken task has increased and the robot will stop. This will cause the task to never be accomplished as long as there is at least one static human in its close vicinity.

The risk increase in this case is an artifact of increased certainty that correctly assigns larger social costs to areas that should have a large penalty in case of intrusion. However, since robots primarily should react to changes in the environment, the correct criterion for ARBR based on which a robot should consider revisiting its decision must ensure that a degree of uncertainty exists in the current estimation.

Based on this, the risk monitoring strategy that we adopt, only considers risk variations if they

have a minimal uncertainty level. Uncertainty of human behavior mainly comes from the human motion. Therefore, static people are assumed to have significantly smaller uncertainty in their motion prediction. We note that for including static people in robots risk monitoring, we only consider their social risk when they are newly perceived. Once they have been accounted for, their associated risk will not be monitored as long as they remain static. We will describe the details of our ARBR algorithm in the next section. We note that we perform tracking and motion prediction for lost human targets who are no longer perceived by the robots in order to have a smooth transition in the social risk. Since we need a realistic estimation of costs induced by lost human targets despite not being able to perceive them. There is a limited time for lost target tracking after which the target is ignored since the uncertainty of prediction grows too large to be meaningful given the lack of information updates.

18.2.3 Risk-Based Rebidding

As only a subset of information updates require the robots to revisit their plans, for identifying when a robot should change its current plan, the corresponding conditions must be defined. These conditions will be checked for triggering a rebidding command. In general, a robot should reconsider its plan upon arrival of new information if risk estimation has a minimal level of uncertainty and i) the risk of accomplishing the currently active task is increasing, or ii) the risk of accomplishing another task has largely decreased making it more profitable than the currently active task for the robot.

To formulate these conditions, we consider how the risk associated to a task can vary over time and how this change can affect the robot plan. For a task t_i and a robot r_j, we denote the last time that a bid has been placed for t_i prior to task allocation by k_{b,t_i}, the risk of t_i for r_j by γ_{t_i,r_j}, the gradient of risk indicating the rate of risk variation by $\dot{\gamma}_{t_i,r_j}$, the risk associated to t_i in the last bidding attempt by γ_{b,t_i}, and the cost of accomplishing t_i for r_j by c_{t_i,r_j}.

As the basis of risk monitoring is to react to change, we must ensure that risk variation is large enough to truly indicate a change. Additionally, with noisy perception and abrupt changes in the environment we must make sure that the variation in the risk trend is meaningful. Moreover, to avoid being too reactive to risk variations and repeatedly triggering rebidding, there must be a sufficiently large time window between rebidding attempts. These constraints constitute the first set of conditions written as follows:

$$|\gamma_{f,t_i} - \gamma_{b,t_i}| \geq \max(\alpha\gamma_{b,t_i}, \Gamma_{min}) \tag{18.2}$$

$$(k - k_{b,t_i}) \geq K \tag{18.3}$$

where γ_{f,t_i} is the filtered risk signal and K is a parameter indicating the minimum time interval between rebids. We implemented a median filter and and average filter to remove outliers and small local variations. Γ_{min} is a minimum risk value and α is introduced to adapt the minimum risk value threshold in proportion to the risk magnitude.

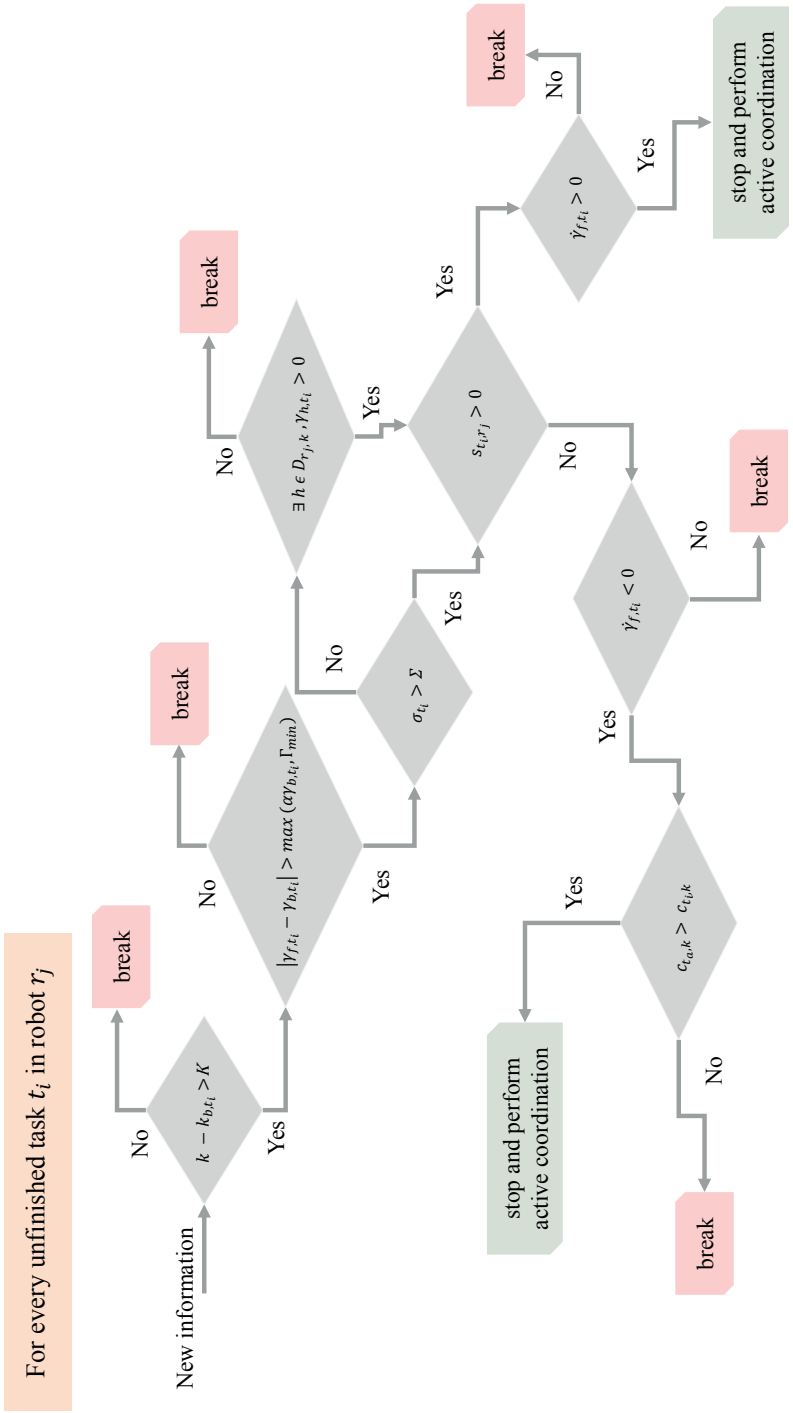


Figure 18.2 – Flow chart of the adaptive risk-based replanning (ARBR) algorithm.

Algorithm 7 Adaptive Risk-Based Replanning (ARBR) for robot r_j with a set of unfinished tasks T and a set of perceived humans H

procedure ARBR(T, H)
  for $t_i \in T$ **do**
    ▷ Compute the path to t_i assuming an empty map
    $(W_{t_i}, K_{t_i}, d_{t_i}) \leftarrow$ PathPlanning(l_{r_j}, l_{t_i})
    ▷ Compute the risk γ_{t_i} and uncertainty U_{H,t_i} of human motion for t_i
    $(\gamma_{t_i}, U_{H,t_i}) \leftarrow$ RiskEstimation(W_{t_i}, K_{t_i}, l_{t_i}, H)
    $\Gamma_{t_i,k} \leftarrow \gamma_{t_i}$ ▷ Γ indicates the vector of risks up to time k
    $c_{t_i,k} \leftarrow$ ComputeCost(γ_{t_i}, d_{t_i})
    if $B_{t_i} \neq \emptyset$ and $t_a \neq \emptyset$ **then**
      $\Gamma_{f,t_i} \leftarrow$ FilterRisk(Γ_{t_i})
      if RBRT(Γ_{f,t_i}, U_{H,t_i}, $c_{t_i,k}$, t_i) **then**
        Stop()
        ActiveCoordination(t_i)

Algorithm 8 Risk-Based Rebid Triggering (RBRT) for robot r_j and task t_i given the vector of risks containing the filtered risk signal Γ_{f,t_i}, perceived human uncertainties U_{H,t_i}, and the task cost $c_{t_i,k}$ at time k

procedure RBRT(Γ_{f,t_i}, U_{H,t_i}, $c_{t_i,k}$, t_i)
  $f_v \leftarrow$ *False*, $f_i \leftarrow$ *False*, $f_a \leftarrow$ *False*, $f_c \leftarrow$ *False*, $f_h \leftarrow$ *False*
  $\gamma_{f,t_i} \leftarrow \Gamma_{f,t_i,k}$
  $\dot{\Gamma}_{f,t_i} \leftarrow$ ComputeRiskTrend(Γ_{f,t_i})
  $\dot{\gamma}_{f,t_i} \leftarrow \dot{\Gamma}_{f,t_i,-1}$ ▷ $\dot{\Gamma}_{f,t_i,-1}$ indicates the latest filtered risk variation recorded for t_i
  $\sigma_{t_i} \leftarrow max(U_{H,t_i})$
  ▷ Note any newly perceived static human that results in risk for t_i
  if $(\exists h \in D_k \wedge \ \gamma_{h,t_i} > 0)$ **then**
    $f_h \leftarrow$ *True* ▷ Human first encounter flag
  if $|\gamma_{f,t_i} - \gamma_{b,t_i}| \geq \max(\alpha\gamma_{b,t_i}, \Gamma_{min}) \wedge (k - k_{b,t_i}) \geq K$ **then**
    $f_v \leftarrow$ *True* ▷ Risk variation flag
  else
    return *False*
  if $\neg s_{t_i} \wedge (\dot{\gamma}_{f,t_i} < 0) \wedge (\sigma_{t_i} > \Sigma \vee f_h)$ **then**
    $f_i \leftarrow$ *True* ▷ Inactive task reconsideration flag
  if $s_{t_j} \wedge (\dot{\gamma}_{f,t_i} > 0) \wedge (\sigma_{t_i} > \Sigma \vee f_h)$ **then**
    $f_a \leftarrow$ *True* ▷ Active task reconsideration flag
  if $\neg s_{t_i}$ **then**
    $f_c \leftarrow (c_{t_a,k} > c_{t_i,k})$ ▷ Decreased cost for the inactive task flag
  else
    $f_c \leftarrow$ *True*
  return $f_v \wedge (f_i \vee f_a) \wedge f_c$ ▷ Final reevaluation decision

Another key factor in risk-based rebidding is uncertainty. If human behavior can be estimated with sufficient certainty due to lack of motion, then variation of social risk would only be due to the increased social cost of approaching a static human as explained in Section 18.2.2.

In this case, γ_{t_i,r_j} is only considered for rebidding if a static human affecting c_{t_i,r_j} has been detected for the first time. We define σ_{t_i,r_j} as the social uncertainty associated to t_i for r_j. It indicates the maximum velocity among all human targets who impact the social risk γ_{t_i,r_j}. The minimum uncertainty level for ARBR is denoted as Σ. The second set of constraints are as follows:

$$\sigma_{t_i,r_j} > \Sigma \tag{18.4}$$

$$\exists h \in D_k \wedge \gamma_{h,t_i,r_j} > 0 \tag{18.5}$$

where D_k is the set of newly perceived static humans for t_i at time k and γ_{h,t_i,r_j} is the γ_{t_i,r_j} considering only h. The purpose of this condition is to include the risk of a static human only once in rebidding computations.

The active state of r_j and the assignment of t_i to r_j at time k can be defined as follows respectively:

$$s_{r_j} \in \{1,0\} \tag{18.6}$$

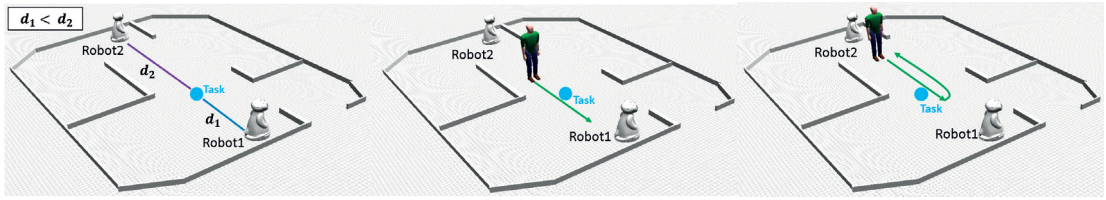
$$s_{t_i,r_j} \in \{1,0\} \tag{18.7}$$

If no task is allocated to a robot ($s_{r_j} = 0$) or there is no prior bid estimation ($B_{t_i,r_j} = \emptyset$) for t_i, there is no need for r_j to perform ARBR as there is no decision to be reconsidered. For an active robot ($s_{r_j} = 1$) however, the replanning decision for t_i depends on s_{t_i,r_j}. In general, each robot tries to find a plan that minimizes c_{t_i,r_j}. For a given t_i this translates to minimizing γ_{t_i,r_j}. Therefore, an increasing risk trend ($\dot{\gamma}_{t_i,r_j} > 0$) is problematic if t_i is the currently active task ($s_{t_i,r_j} = 1$) and a decreasing risk trend ($\dot{\gamma}_{t_i,r_j} < 0$) is interesting if t_i is not assigned to the robot ($s_{t_i,r_j} = 0$). In this case, t_i will replace the currently active task only if $c_{t_i,r_j} < c_{t_a,r_j}$, where t_a is the currently active task. The last set of constraints considered for triggering rebidding can be written as follows:

$$s_{t_i,r_j} \wedge (\dot{\gamma}_{f,t_i} > 0) \tag{18.8}$$

$$\neg s_{t_i,r_j} \wedge (\dot{\gamma}_{f,t_i} < 0) \wedge (c_{t_i,r_j} < c_{t_a,r_j}) \tag{18.9}$$

Figure 18.2 illustrates the flowchart of our risk-based replanning method. The adaptive replanning and rebid triggering algorithms are detailed in Algorithm 7-8. We note that the subscripts indicating the robot are omitted for brevity as each algorithm is running locally on one robot.



(a) (b) (c)

Figure 18.3 – The simulated arena for test case S-I. a) Task and initial robot positions, b) the trajectory of walking for *NG-W* and *AG-W* scenarios and c) the trajectory of the human with an unexpected behavior for *NG-C* and *AG-C* scenarios.

We note that if the output of the human prediction is a normal distribution, the final risk-based social cost resulted in from the convolution of N and p_h functions, will be another Gaussian function with $M_c = M_N + M_{p_h}$ and $\Sigma_c = \Sigma_N + \Sigma_{p_h}$ [123].

18.3 Experiments

To evaluate the performance of our proposed method we have conducted an extensive suite of experiments with increasing complexity in simulation and reality. This section details the set of test cases used for our experiments. The first set of test cases denoted by *S* are performed in simulation and test cases denoted by *R* are conducted with real robots. We note that *A* indicates adaptive replanning, *N* indicates non-adaptive replanning, *G* indicates global perception, *Ci* indicates circular FOV and *C* indicates conic FOV in the labeling of the scenarios.

Each scenario has been repeated for ten runs. Robots are relying on their self-localization for computing the local balance functions. The evaluation metrics ($M_1 - M_5$) have been obtained from ground truth values provided by the simulation or MCS in real robot tests. The social metrics ($M_3 - M_5$) have been computed for the moving robots to avoid penalizing a static robot when a human decides to approach it. Throughout runs we have introduced randomness in human behavior by adding a random starting delay to the motion of each human. The algorithmic parameters of RBRT are shown in Table 18.1.

Table 18.1 – RBRT algorithmic parameters.

Parameter	α	Γ_{min}	K	Σ
Value	0.1	5	2	0.05

18.3.1 Test Case S-I: Global Perception and Human Behavior Change

This test case is designed to show how despite having global perception human behavior change can lead to a suboptimal plan for the robots. This test case is conducted in an arena depicted in Figure 18.3 and consists of one task, two robots and one human. Four scenarios

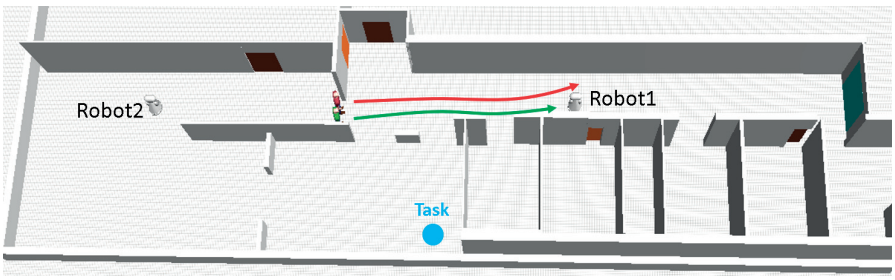


Figure 18.4 – The simulated arena of test case S-II. This snapshot shows the initial position of the robots and humans along with human trajectories.

have been considered for this test case. *Scenario NG-W*, non-adaptive replanning with global perception for a walking human, *Scenario AG-W*, adaptive risk-based replanning with global perception for a walking human, *Scenario NG-C*, non-adaptive replanning with global perception for a human with unexpected behavior change, and *Scenario AG-C*, adaptive risk-based replanning with global perception for a human with unexpected behavior change.

18.3.2 Test Case S-II: Local Perception in a Partially Observable Environment

This test case focuses on the impact of local perception on team plans and consists of one task, two robots and two humans in an environment depicted in Figure 18.4. Here, a team of two robots needs to accomplish one task in the presence of two walking humans that cannot be initially seen. Two scenarios have been considered for this test case. *Scenario NCi-W*, non-adaptive replanning with local perception, and *Scenario ACi-W*, adaptive risk-based replanning with local perception. In this test case robots have a circular field of view with a radius of 4 m. Although having a conic filed of view is more common with vision-based sensors, we can imagine robots relying on information obtained by fish-eye cameras that can be considered to have a circular field of view. A larger and more complex environment has been chosen for this test case to better highlight the impact of local perception where the arena size is significantly larger than the range of local perception.

18.3.3 Test Case S-III: Global vs Local Perception Around a Static Human

This test case consists of one task, two robots and a static human blocking a passage in an environment depicted in Figure 18.5. The following four scenarios have been considered in this test case. *Scenario NG-S*, non-adaptive replanning with global perception, *Scenario AG-S*, adaptive risk-based replanning with global perception, *Scenario NCi-S*, non-adaptive replanning with local perception, and *Scenario ACi-S*, adaptive risk-based replanning with local perception. Similar to test case S-II, robots have a circular field of view with a radius of 4 m. In this test case, the non-adaptive replanning method has the advantage of replanning frequently on just one task. This allows the robots to be able to react to changes quite fast as there is only one task to be considered. Any other number of tasks will put the non-adaptive

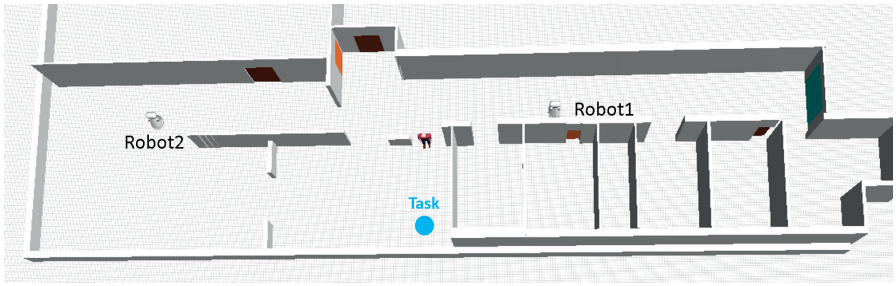


Figure 18.5 – The simulated arena of test case S-III. This snapshot shows the initial position of the robots and the static human.

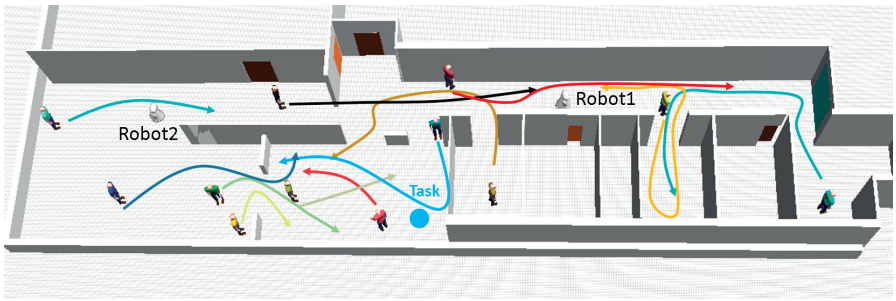


Figure 18.6 – The simulated arena of test case S-IV. This snapshot shows the initial position of the robots and the trajectory of the humans.

replanning method in a less favorable position. The reason is that with multiple tasks the second robot finds another task to accomplish and only replans again when its currently allocated task is done.

18.3.4 Test Case S-IV: Global vs Local Perception in a Highly Dynamic and Populated Environment

In this test case, we evaluate the behavior of robots adopting different replanning strategies with local and global perception in a crowded social environment. This test case consists of one task, two robots and 12 humans in an environment depicted in Figure 18.6. The random delays introduced in deterministic human trajectories led to differences as large as 8 m in terms of relative positions among runs between the robots and the humans in this test case. Hence, each run can be considered as a different problem instance for the robots in terms of social costs. Nevertheless, similar human behaviors are seen across runs on average.

Eight scenarios have been considered in this test case, *Scenario NG*, non-adaptive replanning with global perception, *Scenario AG*, adaptive risk-based replanning with global perception, *Scenario NCi*, non-adaptive replanning with circular FOV, *Scenario ACi*, adaptive risk-based replanning with circular FOV. In the next four scenarios robots have a more restricted local perception with a conic FOV and a range of 4 m: *Scenario NC-120*, non-adaptive replanning with 120° FOV, *Scenario AC-120*, adaptive risk-based replanning with with 120° FOV, *Scenario*

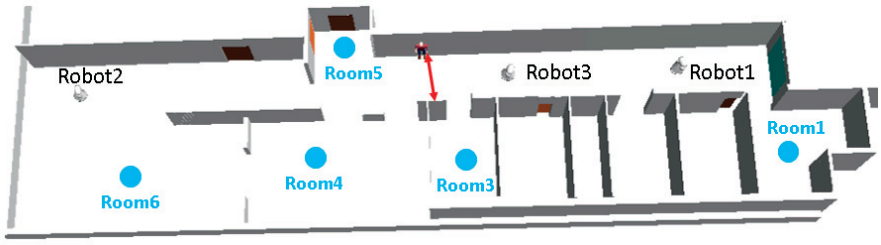


Figure 18.7 – The simulated arena of test case S-V. This snapshot shows the initial position of the robots and the trajectory of the human. The human constantly walks the width of the corridor.

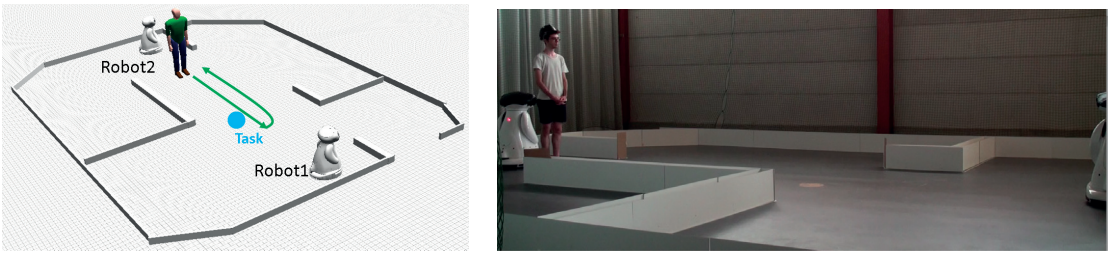
NC-65, non-adaptive replanning with with 65° FOV, and *Scenario AC-65*, adaptive risk-based replanning with with 65° FOV.

18.3.5 Test Case S-V: Limited Field of View and Human Behavior Change

This test case consists of five tasks, three robots and one human in an environment depicted in Figure 18.7. This is a challenging scenario since the human motion is highly stochastic and the human behavior is rapidly changing as he constantly walks the width of the corridor. Thus, the prediction that a robot makes may or may not stay valid by the time it gets to a close vicinity of the human. The following four scenarios have been considered in this test case. *Scenario NG*, non-adaptive replanning with global perception, *Scenario AG*, adaptive risk-based replanning with global perception, *Scenario NC*, non-adaptive replanning with local perception, and *Scenario AC*, adaptive risk-based replanning with local perception. Local perception of the robots consists of a conic 65° FOV with 4 m of range. This choice is motivated by the specifications of the Kinect and characterizes our local perception for the remaining test cases (S-V to R-II). As a result, we only indicate conic local perception using C from now on, in the scenario labeling.

18.3.6 Test Case R-I: Human Behavior Change

This test case is similar to test case S-I for a human with changing behavior. Figure 18.8 shows the simulated and real test environments of this test case. We note that for the real test cases in this chapter, the map II motion arena is used for experiments (see Section 7.2). Four scenarios are considered in this test case, *Scenario NG*, non-adaptive replanning with global perception, *Scenario AG*, adaptive risk-based replanning with global perception, *Scenario NC*, non-adaptive replanning with local perception, and *Scenario AC*, adaptive risk-based replanning with local perception. The perception available to the robots is local with a conic FOV of 65° and range of 4 m.



(a) (b)

Figure 18.8 – Placement of the robots and the task in the arena for test case R-I. a) human walking trajectory, and b) snapshot of the initial state of the real robot experiment.

(a) (b)

Figure 18.9 – Placement of the robots and the tasks for test case R-II. a) human walking trajectories, b) snapshot of the initial state of the real robot experiment.

18.3.7 Test Case R-II: Multi-Human Partially Observable Environment

In this test case three tasks must be accomplished by a team of two robots in the presence of two walking humans that cannot be initially perceived by the robots. As a result, the initial risk-based bid estimations will not remain valid throughout the experiment. This test case highlights how information updates can be used to find better team plans. Similar to test case R-I, limited local perception with a range of 4 m and FOV of 65° is used. Figure 18.9 shows the simulated and real test environments of this test case. Four scenarios are considered in this test case. *Scenario NG*, non-adaptive replanning with global perception, *Scenario AG*, adaptive risk-based replanning with global perception, *Scenario NC*, non-adaptive replanning with local perception, and *Scenario AC*, adaptive risk-based replanning with local perception.

18.4 Results

In this section the results of the test cases explained previously will be discussed. Before going to the details of the results, one emergent behavior observed during runs should be explained. Robots displayed a waiting behavior when confronted with increasing social costs for their

currently active task. What drives this behavior for a robot r_j and an active task t_a is c_{t_a,r_j}:

$$c_{t_a,r_j} = \hat{c}_{t_a,r_j} + f(\gamma_{t_a,r_j}) \tag{18.10}$$

We note that f is a monotonically increasing function. As r_j progresses towards t_a the deterministic part of the cost $\hat{c}_{t_i,r_j}$, is reduced and any increase in c_{t_a,r_j} would be due to the increase in γ_{t_a,r_j}. For $\dot{\gamma}_{t_a,r_j} > 0$ there can be cases where despite ARBR stopping the robot, r_j would still be the best candidate for t_a as $r_j = argmin_r(c_{t_a,r})$. Thus, t_a will be assigned to r_j again. Nonetheless, ARBR will stop r_j once again and the robot will be prevented from moving towards the assigned task. This results in a waiting behavior for as long as:

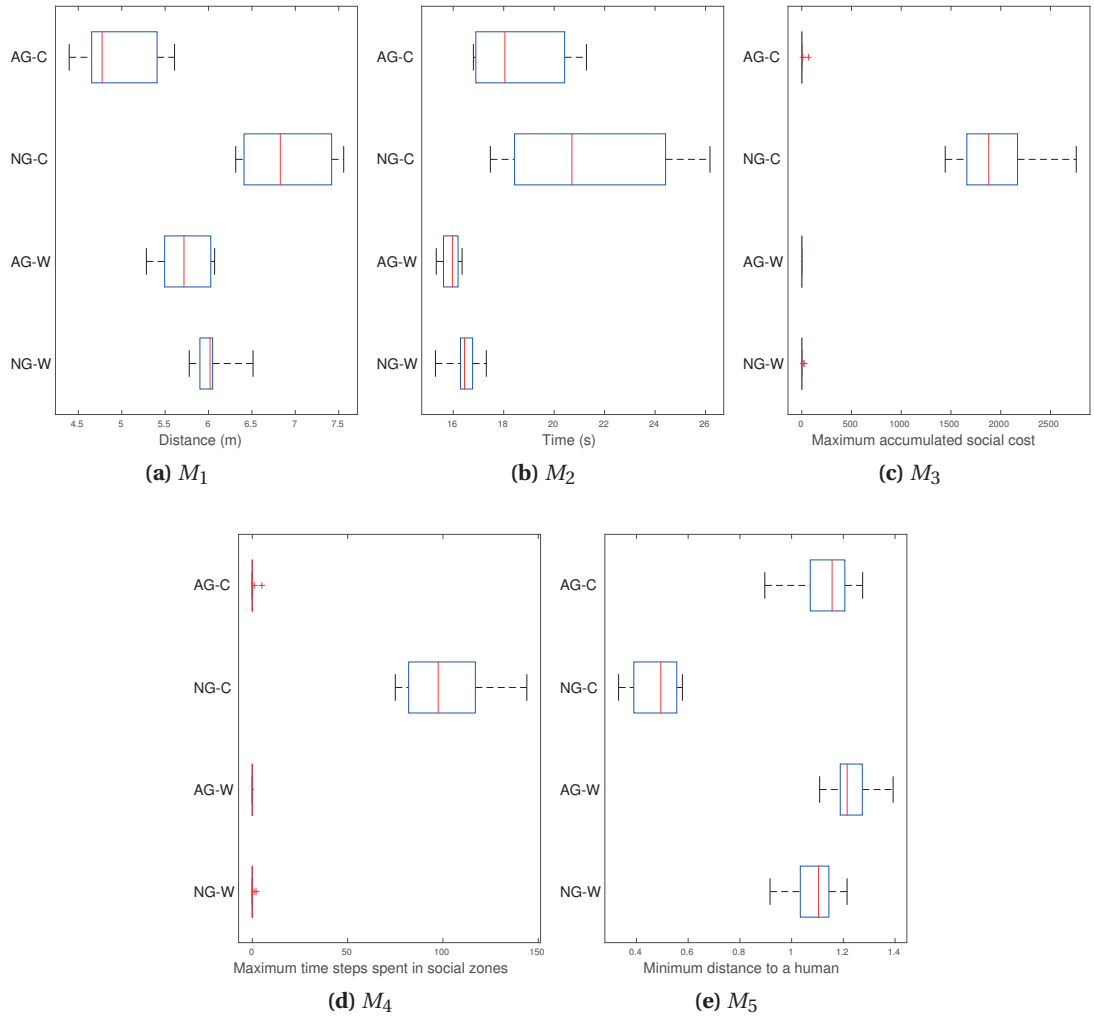
$$s_{r_j,t_a} = 1 \quad \wedge \quad \dot{\gamma}_{t_a,r_j} > 0 \quad \wedge \quad r_j = argmin_r(c_{t_a,r}) \tag{18.11}$$

The duration of the waiting period depends on how the environment changes. These waiting periods increase the total mission time by stopping the robot from performing a socially risky motion. Consequently, this prevents accumulating social costs and leads to a better performance with respect to social metrics at the price of a longer execution time, as shown in the rest of the results section.

18.4.1 Test Case S-I: Global Perception and Human Behavior Change

Consider Figure 18.3a again. In an empty arena, Task will be assigned to Robot1 as $d_1 < d_2$. However, when a human is walking towards Robot1, Robot2 will be the best candidate for Task. This can be seen for scenarios *NG-W* and *AG-W* in Figures 18.11a and 18.11b respectively. As the walking human suddenly changes his behavior and turns around, the previous trajectory estimation made by the robots is no longer valid. Nonetheless, in Scenario *NG-C*, Robot2 will continue moving towards Task despite the human getting closer to the robot, making Task very costly. We know that in this situation Robot1 should take Task. However, due to the constraint of single attempt of active coordination, Robot2 will not delegate Task to Robot1 as seen in Figure 18.11c.

For scenarios with non-adaptive replanning, we note that had there been more tasks in the environment, Robot1 would probably not have replanned for Task as it would have been assigned to another task by the time the change had occurred. An adaptive risk-based strategy on the other hand, can spot the risk change in the environment and will allow active coordination to be executed for the second time. Consequently, Robot1 will take Task as depicted in Figure 18.11d. Looking at Figure 18.10, we can observe similar performances across scenarios with the walking human but for the scenarios with behavior change, there is a clear difference between the performance of *NG-C* and *AG-C*, particularly, regarding social metrics. This is due to keeping an outdated plan in the strategy taken by *NG-C* that leads to more traveled distance and larger social costs.



(a) M_1 **(b)** M_2 **(c)** M_3

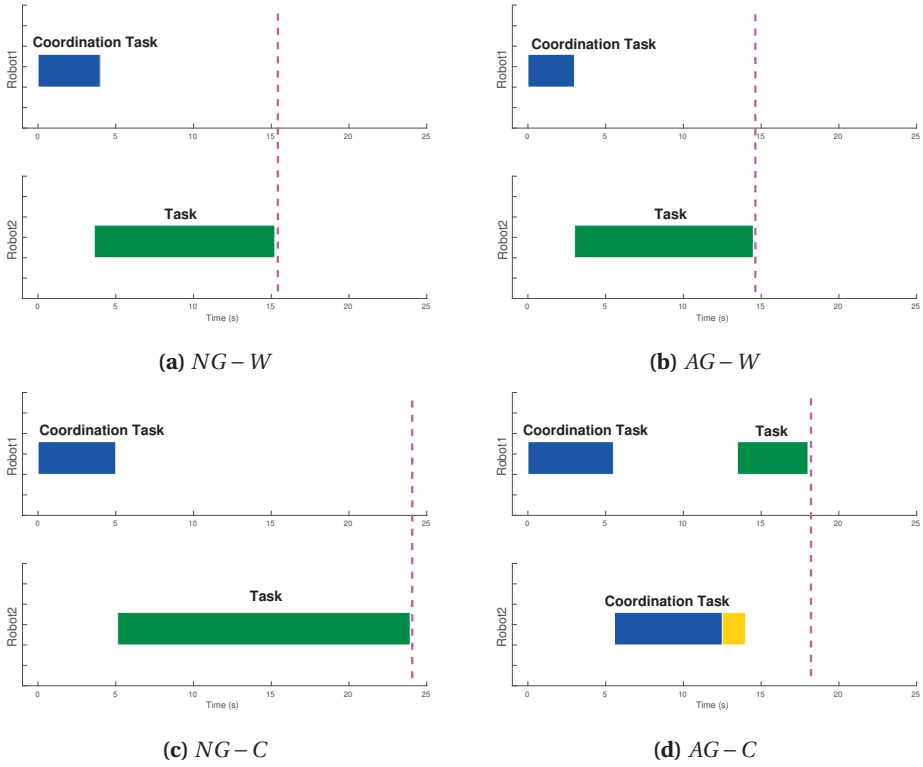
(d) M_4 **(e)** M_5

Figure 18.10 – Performance metrics for test case S-I obtained from 10 runs.

18.4.2 Test Case S-II: Local Perception with Circular and Conic Field of Views

In the environment depicted in Figure 18.4, without the presence of humans Task will be assigned to Robot1. However, with the two humans walking towards Robot1, Robot2 is a better candidate to be assigned to Task. This is not known to the robots unless they have a global view of the environment. In this test case, we evaluate the performance of ARBR and a non-adaptive replanning strategy for robots endowed with limited local perception.

In scenario *NCi-W*, Task will initially be assigned to Robot1. Meanwhile, Robot2 is idle and frequently replanning. As a result, when the humans are first detected, with the next replanning of Robot2 active coordination is triggered and Task is assigned to Robot2. We note that if Robot2 was launched before Robot1, a problem similar to scenario *S-I:NG-W* would happen where the second active coordination attempt could not be permitted. Despite similar plans of the



(a) $NG-W$ **(b)** $AG-W$

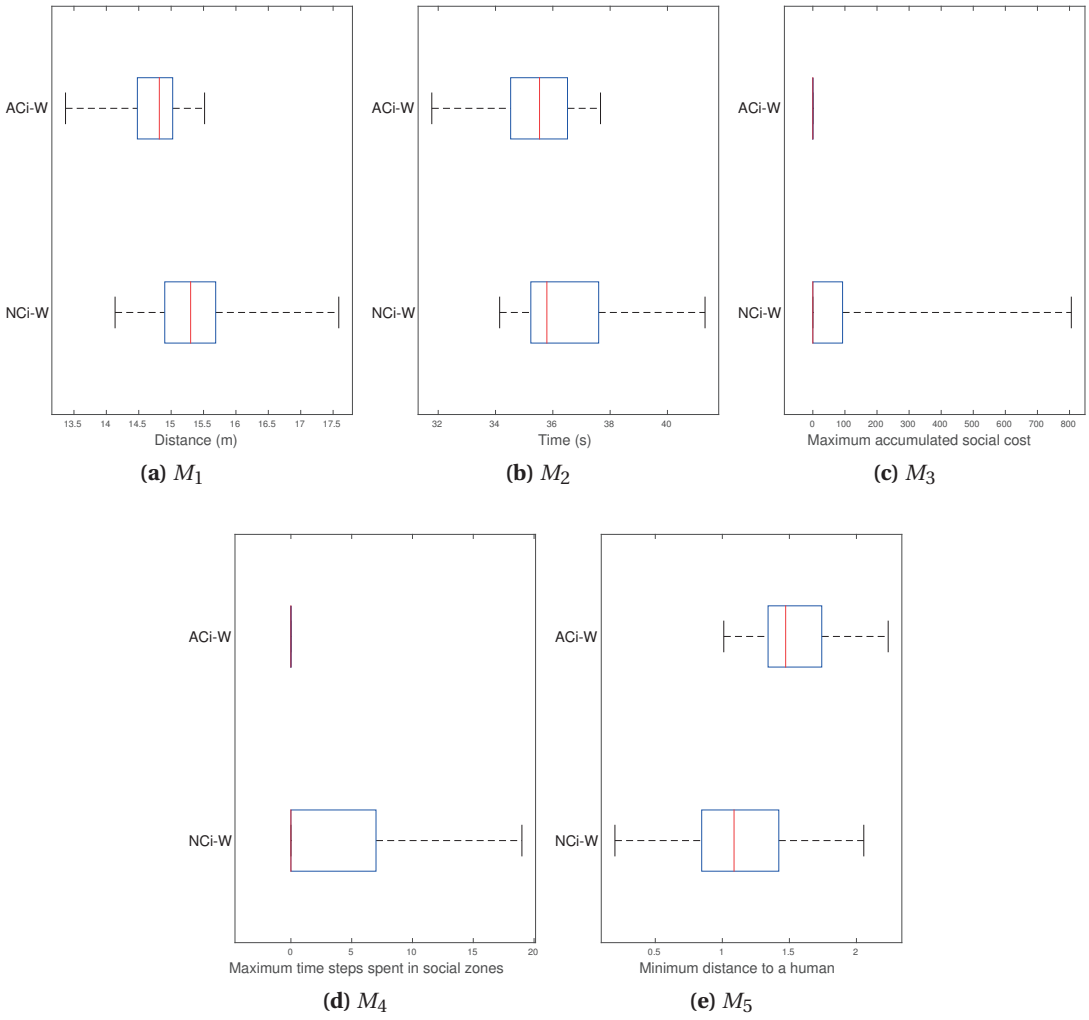
(c) $NG-C$ **(d)** $AG-C$

Figure 18.11 – Task assignment per robot over time for a sample run of test case S-I for scenarios *NG-W, AG-W, NG-C,* and *AG-C* respectively. End of mission (M_2) is marked by the vertical line.

two scenarios depicted in Figure 18.13, better performance is reported for scenario *ACi-W* as shown in Figure 18.12. The reason is that ARBR can react to the new incoming information faster. This strategy can identify that Robot1's decision for accomplishing Task should be reevaluated given the increasing social cost caused by the newly perceived humans. Hence, Robot1 is stopped and does not further advance into the socially costly areas.

18.4.3 Test Case S-III: Global vs Local Perception Around a Static Human

With no humans present in the environment, Task will be assigned to Robot1 in this test case. However, with one static human positioned as shown in Figure 18.5, Robot1 is no longer able to reach the task as expected and Task should be assigned to Robot2. Similar to the previous scenario, Robot1 plans first and Robot2 replans frequently. In scenario *NG-S*, when Robot2 replans it finds that it is better suited to take the task, thus, active coordination is triggered and Task is assigned to Robot2. The performance with respect to $M_1 - M_2$ and the team plan of scenario *AG-S* are similar to those of scenario *NG-S* as shown in Figures 18.14 and 18.15 respectively. This is because in this setting there are no information updates that can change the optimal team plan for the robots, therefore, ARBR and non-adaptive replanning behave similarly in terms of task assignment. However, unlike scenarios adopting ARBR, scenario



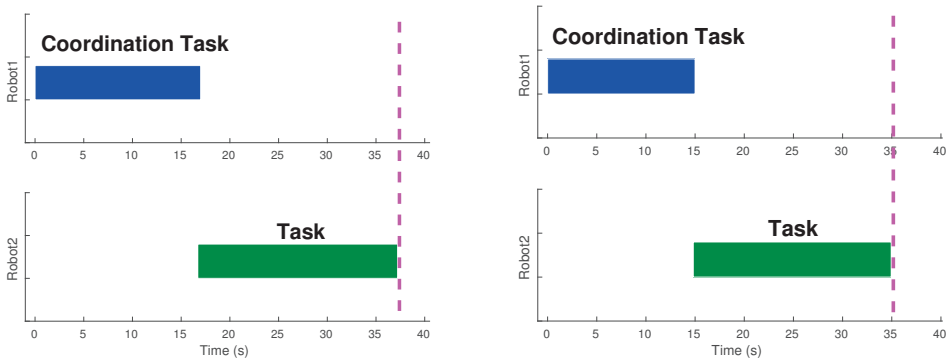
(a) M_1 **(b)** M_2 **(c)** M_3

(d) M_4 **(e)** M_5

Figure 18.12 – Performance metrics for test case S-II obtained from 10 runs.

NG-S has non-zero M_3. This is because before Robot2 takes over, Robot1 continues to move towards the task despite the increasing social cost.

When robots reply on local perception, they perceive the human only when he or she is within their perception range. For Robot1 this means it would have to travel a long distance before considering the human in its bid estimations. Once the human is perceived by Robot1, it will communicate the pose information of the human to Robot2. In scenario *NCi-S* the next time Robot2 plans, it will trigger active coordination and Task will be assigned to Robot2. Similar to scenario *AG-S*, Robot1 continues to move towards the task despite the increasing social cost before delegating the task to Robot2. In scenario *ACi-S* however, upon detecting the human, ARBR notifies Robot1 to stop as the social risk is increasing and replannig is triggered. Consequently, Task will be assigned to Robot2. Therefore, in spite of having similar plans as shown in Figure 18.15, scenario *ACi-S* has a significantly better performance in terms of social



(a) $NCi-W$ **(b)** $ACi-W$

Figure 18.13 – Task assignment per robot over time for a sample run of test case S-II for scenarios *NCi-W*, and *ACi-W* respectively. End of mission (M_2) is marked by the vertical line

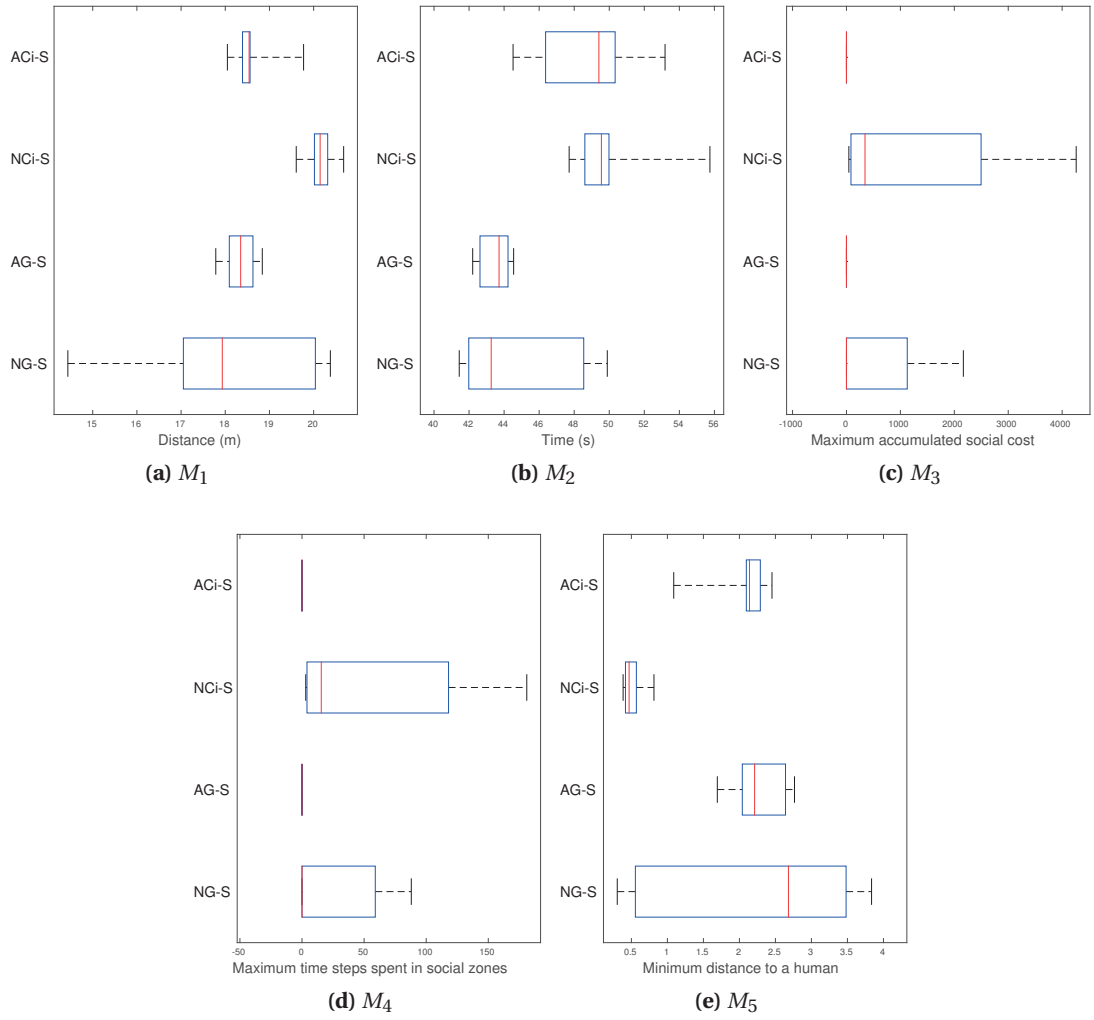
metrics ($M_3 - M_5$) compared to scenario *NCi-S*.

We can observe that the global perception has helped the non-adaptive strategy in achieving a better performance compared to local perception. This is true as long as the information provided by global perception remain valid during the course of task execution, otherwise, global perception can mislead the robots with outdated information. We will explain this is more detail in the next test cases.

18.4.4 Test Case S-IV: Adaptive Risk-Based Replanning in a Highly Dynamic and Populated Environment

Given an empty environment, Task will be assigned to Robot1 in this test case. However, when humans are added to this setting the cost of accomplishing the task can largely change. Depending on how the environment changes the best candidate for the task can vary. This is a test case with highly stochastic social risks caused by the 12 moving humans depicted in Figure 18.6. The results of this test case are organized in two parts: (i) scenarios with global and circular perception, and (ii) scenarios with limited conic FOV.

Looking at Figure 18.16 we can see how ARBR has resulted in better performance in terms of social metrics compared to a non-adaptive replanning strategy. The distance traveled in scenario *AG* and *ACi* are also less compared to their non-adaptive counterparts. However, the time taken by scenario *AG* and *ACi* is longer due to the emergent waiting behavior explained previously. We can observe how having full knowledge of the human poses has lead to more efficient plans in terms of all metrics for scenario *AG* compared to scenario *ACi*. Relying on global perception leads to a more conservative task assignment approach as scenarios with global perception account for all relevant changes in the environment even if those changes are prone to further variation since they are caused by humans who are far away. For the non-adaptive strategy however, global perception cannot compensate for the limitations of



(a) M_1 **(b)** M_2 **(c)** M_3

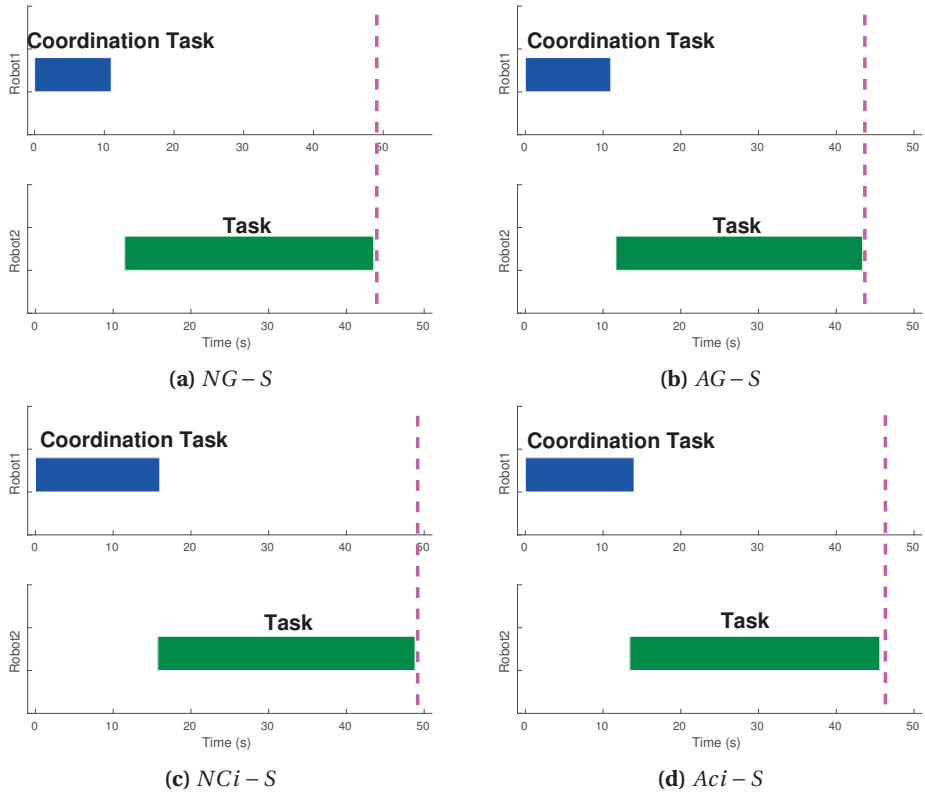
(d) M_4 **(e)** M_5

Figure 18.14 – Performance metrics for test case S-III obtained from 10 runs.

the decision making process and the plan will not be reevaluated when the social cost changes even if the robot can observe the changes.

Similar to scenario *SI: NG-C* the limitation of active coordination attempts causes the plans of scenario *NG* and *NCi* to be unable to adapt to the new information despite performing replannig. We can see how ARBR has led to earlier identification of rising risks in Figures 18.17b-18.17d. The periods in which robots are waiting are depicted by yellow blocks in the task assignment figures. Notice how more behavior switches can be seen for scenario *ACi* compared to scenario *AG* in Figure 18.17. This is caused by the new information that robots receive when detecting new human targets nearby at different points in time given the limited local perception.

As local perception gets more limited, the performance of the robots changes as well. Looking



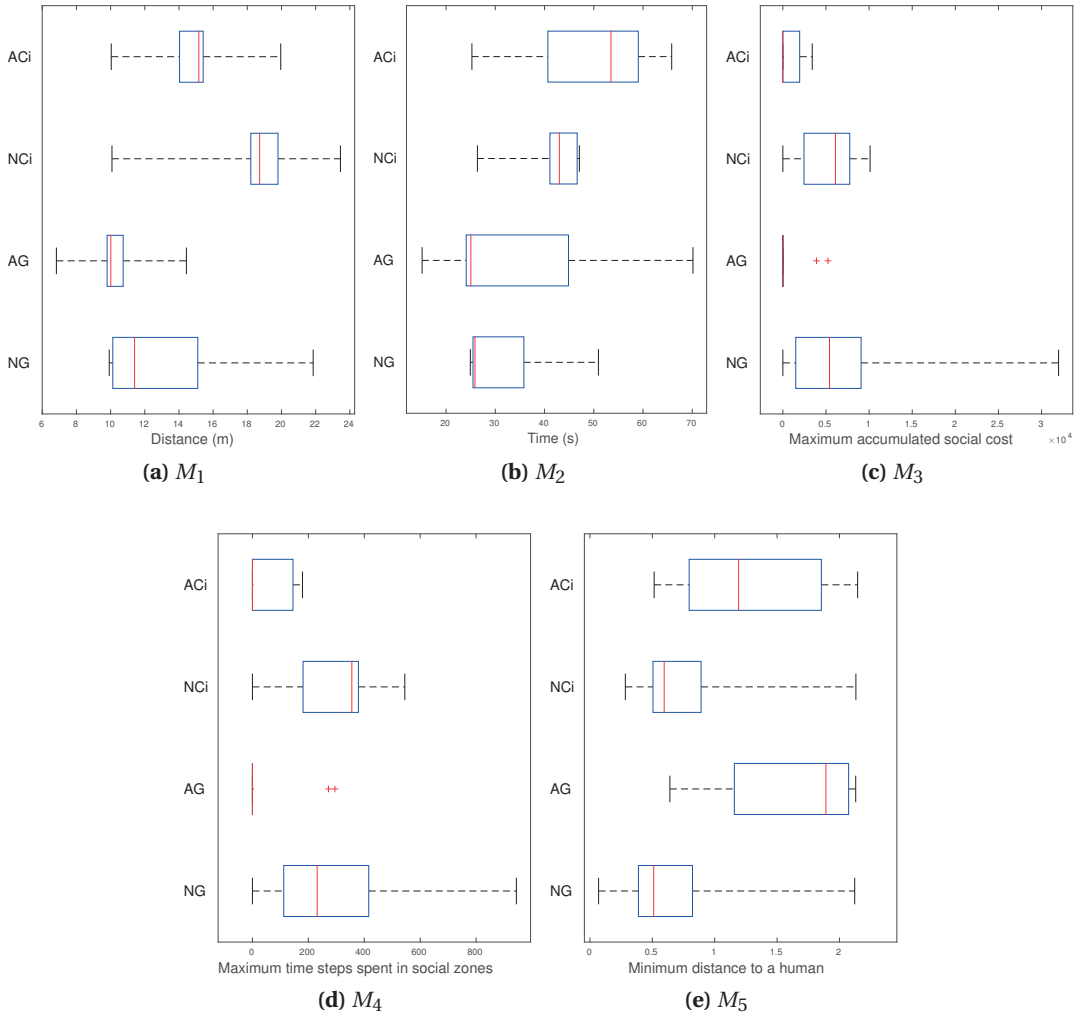
(a) $NG-S$ **(b)** $AG-S$

(c) $NCi-S$ **(d)** $Aci-S$

Figure 18.15 – Task assignment per robot over time for a sample run of test case S-III for scenarios *NG-S, AG-S, NCi-S,* and *ACi-S* respectively. End of mission (M_2) is marked by the vertical line.

at Figure 18.16, we see that despite a moderately degraded performance, local circular perception with ARBR has comparable performance with its global perception counterpart with respect to $M_3 - M_5$. However, when conic FOVs of 120° and 65° are introduced, a meaningful difference can be seen in the team performance. Figure 18.18 shows the performance of the second set of scenarios derived from five runs. We can see that ARBR is superior to the non-adaptive replanning strategy across all scenarios. However, as the FOV gets more restricted we can observe that the robots show a degraded performance with respect to the social metrics. Additionally, we can see an increased number of behavior switches in Figures 18.19b-18.19d compared to Figures 18.17b-18.17d.

As the perception gets more restricted in a highly stochastic and populated environment, the plans alone cannot ensure social constraints to be respected at all times. This calls for the help of the human-aware navigation running locally on each robot to prevent social intrusions from happening. Nonetheless, a human-aware team plan is necessary as it assists in finding less difficult situations for accomplishing the tasks in terms of social risks. As a result, the combination of human-aware plans and single robot human-aware navigation has a higher chance of succeeding in maintaining a social behavior.



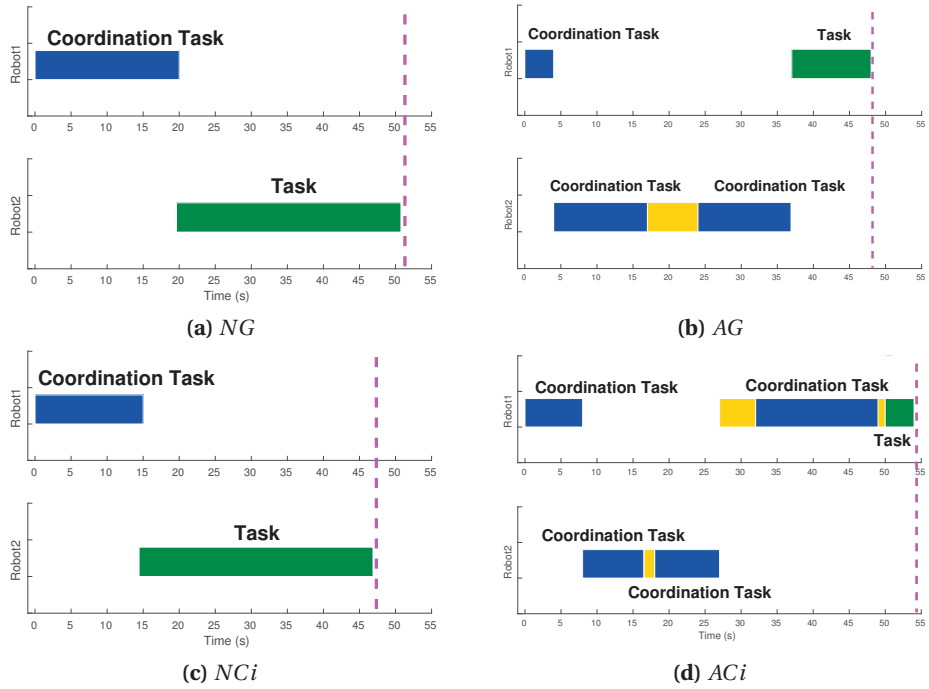
(a) M_1 **(b)** M_2 **(c)** M_3

(d) M_4 **(e)** M_5

Figure 18.16 – Performance metrics for test case S-IV with global and circular perception obtained from 10 runs.

Aside from the fact the robots are faced with limited perception in reality, another reason to study the effect of local perception on MRTA performance is to understand the drawbacks and benefits of having a global perception of the world compared to having a local view. Can knowing less actually be helpful when making a decision?

Having local perception is shown to have its advantages and drawbacks. Comparing the social performance of non-adaptive replanning strategies across all scenarios, we can observe that a more restricted perception has better performance if the replanning strategy of the robots does not adapt to the social changes. In such a strategy, taking decisions based on outdated data can lead to suboptimal decisions. This is exacerbated with global perception since there can be even more inaccurate information used for decision making. However, if the robots can adapt to the new incoming information, having a broader perception can help to take



(a) NG **(b)** AG

(c) NCi **(d)** ACi

Figure 18.17 – Task assignment per robot over time for a sample run of test case S-IV for scenarios *NG*, *AG*, *NCi*, and *ACi* respectively. End of mission (M_2) is marked by the vertical line.

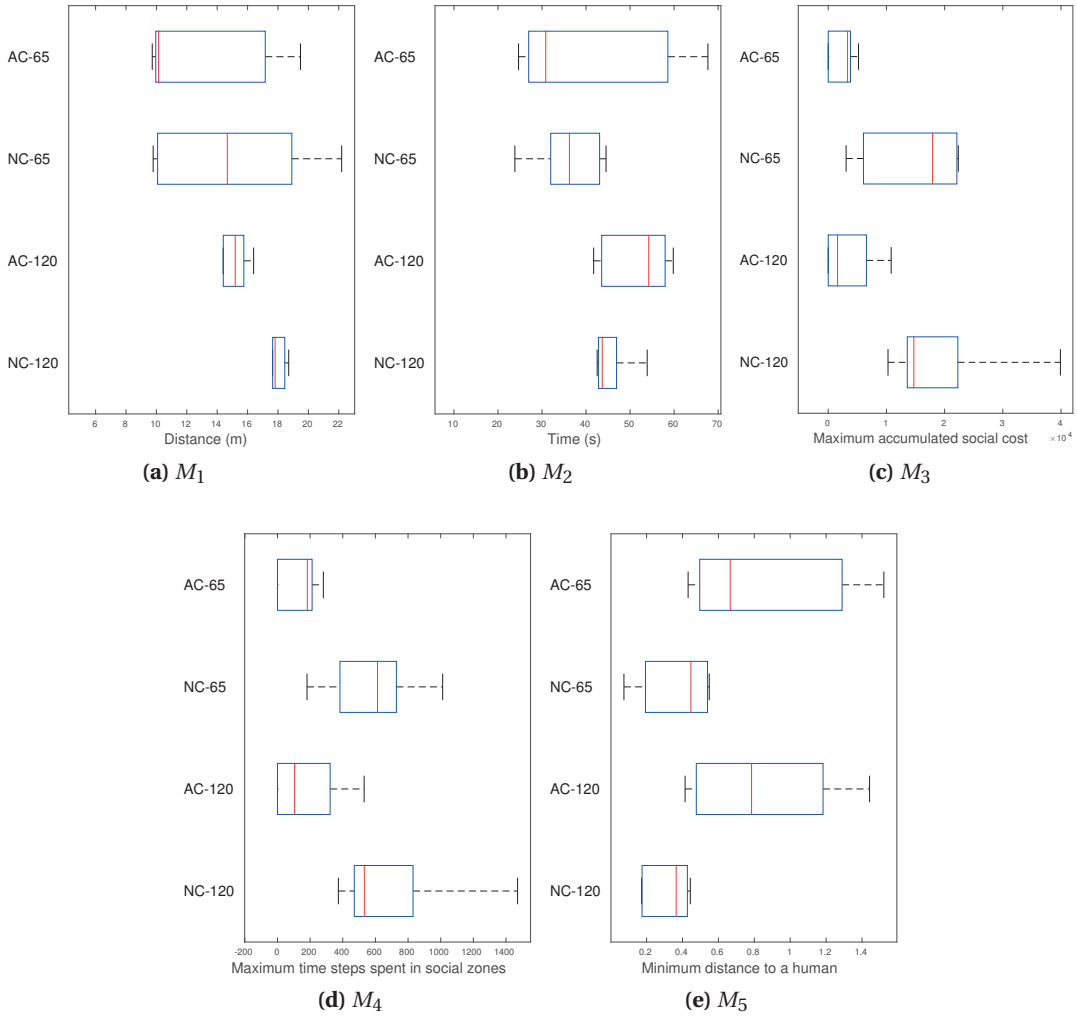
better decisions although it will be more expensive and more sensitive. The reason is that the robot will consider all relevant updates in its decision making including the ones that may be less/not relevant to its decision over time.

As an example, monitoring the risk of people behind the robot can be computationally expensive while being less probable to have an effect on its decision. Nonetheless, it can ensure that the robot will mitigate risks as mush as possible given the extent of available information using a conservative approach. This can also be seen when comparing circular and conic field of views.

18.4.5 Test Case S-V: Limited Field of View and Human Behavior Change

This test case, combines the two challenges of human behavior change and limited perception in a more complex MRTA problem with three robots and five tasks. In an empty arena, the task assignment for the robots is as follows: Robot1 goes to Room1, Robot2 to Room6 and Robot3 to Room3. Robot1 then moves to Room5 and meanwhile receives an active coordination request from Robot3. Consequently, it delegates Room5 to Robot3. Room4 is assigned to Robot2.

As the human is constantly changing his walking behavior, the initial human motion prediction done by a non-adaptive approach can be incorrect. Thus, it will lead to a decision that is likely to be poor by the time the robot gets to a close vicinity of the human. This explains why ARBR

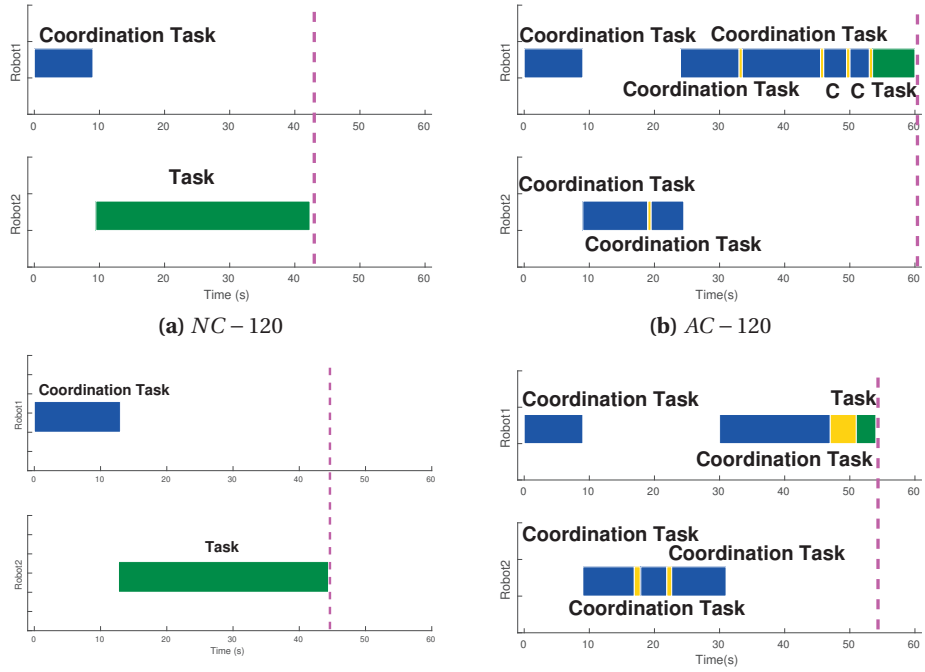


(a) M_1 **(b)** M_2 **(c)** M_3

(d) M_4 **(e)** M_5

Figure 18.18 – Performance metrics for test case S-IV with conic field of view obtained from 5 runs.

has a better performance in terms of all metrics compared to its non-adaptive counterparts as seen in Figure 18.20. Once more, scenario *AG* has a better performance compared to scenario *AC*, due to having a more accurate prediction of the human trajectory based on updated perceptual information. Robots with limited perception can only react to the human once the human is observed within the FOV of one of the robots. This typically, leads to later plan modification and occasionally, minor violations of social constraints. Moreover, looking at Figure 18.20c we can see how a non-adaptive replanning strategy performs better with limited perception in terms of social metrics. The reason is that the replanning strategy of scenario *NG* relies on incorrect predictions in most cases, since the human behavior changes within the execution period of the plan and is no longer what was expected to be at the decision making time. Furthermore, due to having global perception, *NG* is more reactive to change. Figure 18.21 shows how scenarios adopting ARBR partition tasks into two sets given where the



(a) $NC-120$ **(b)** $AC-120$

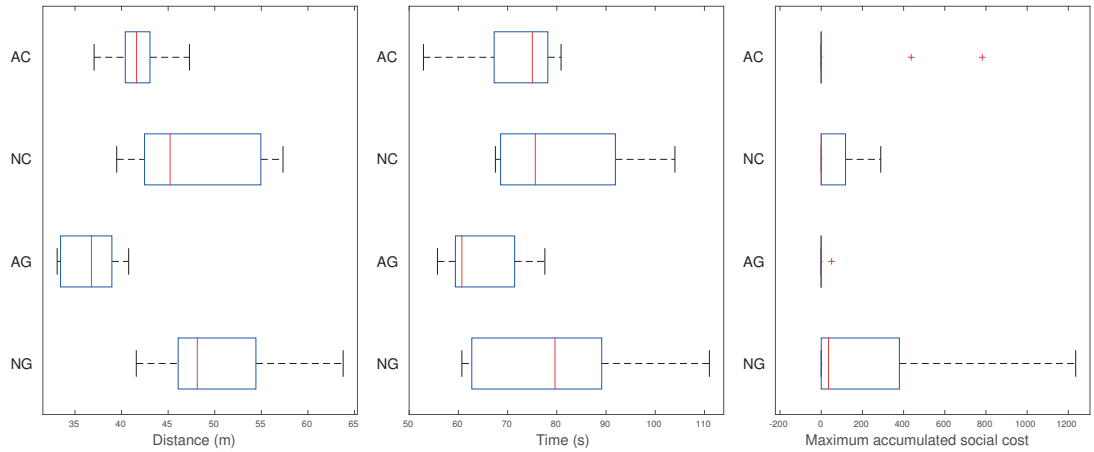
(c) $NC-65$ **(d)** $AC-65$

Figure 18.19 – Task assignment per robot over time for a sample run of test case S-IV for scenarios *NC-120, AC-120, NC-65,* and *AC-65* respectively. End of mission (M_2) is marked by the vertical line.

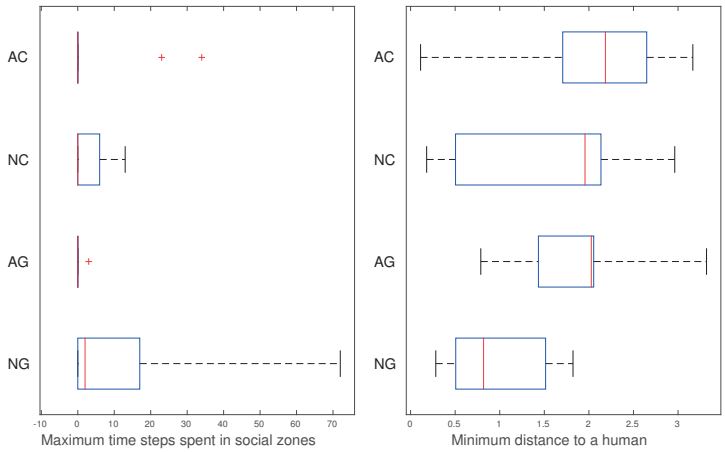
human is located. Robot2 takes care of all the tasks on one side and the other two robots cover the side with Room1 and Room3.

To see how risk monitoring and adaptive rebidding work, consider Figure 18.22, where the risk value associated to the Room5 is plotted over time up to the last rebidding event for the robots. The oscillations in human walking behavior can be seen in the risk signal in Figure 18.22a. We have plotted the risk for Robot1 and Robot3 in scenario *AG*, and for Robot3 in scenario *AC*, since these robots were the only ones affected by the risk-based rebidding process. As Room5 is the one task mainly affected by the human walking behavior and the variation of cost for other tasks is not significant we have only plotted the risk for Room5.

In scenario *AG*, Robot1 initially takes Room5 and rebids once in reaction to the risk variation. When Robot3 reaches Room3 it takes over Room5 by means of active coordination. The risk trend changes as the human walking behavior changes, consequently, Robot3 stops and rebids, but as long as Robot2 is not free, Robot3 would still be the best candidate. As a result, multiple rebidding attempts occur while Robot3 moves toward Room5. Finally, Robot3 rebids and finds Robot2 responding to its active coordination request. As a result, Robot2 moves to Room5.



(a) M_1 **(b)** M_2 **(c)** M_3



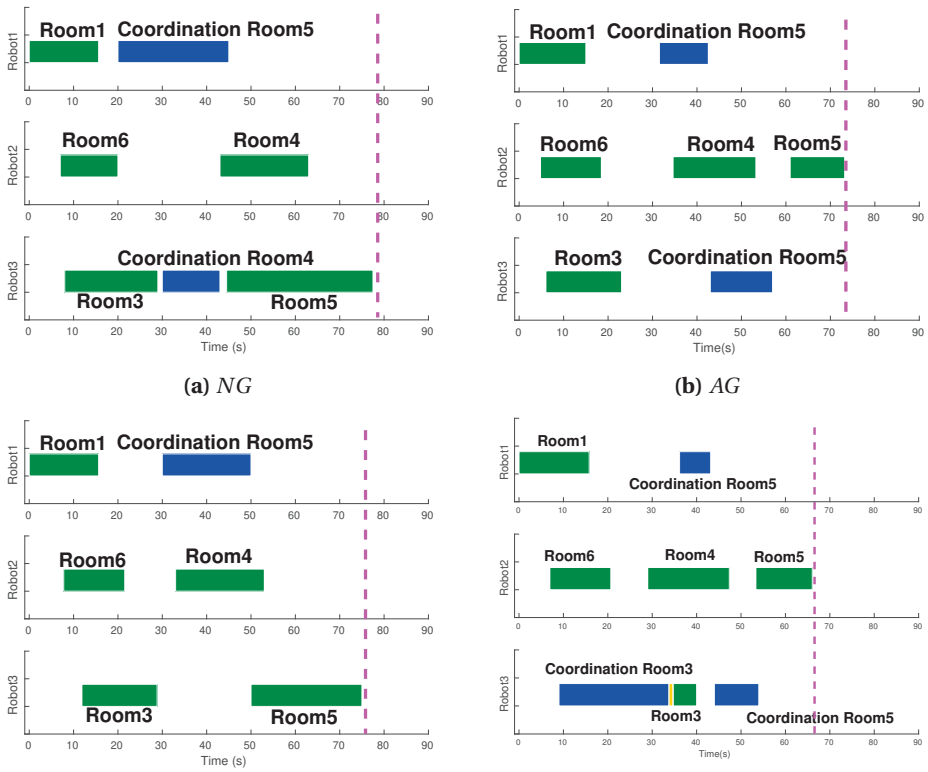
(d) M_4 **(e)** M_5

Figure 18.20 – Performance metrics for test case S-V obtained from 10 runs.

18.4.6 Test Case R-I: Human Behavior Change

This test case, has the same configuration as test case S-I. In a human-free environment Robot1 initially takes the task. In the presence of a walking human with behavior change, non-adaptive replanning strategies assign the task to Robot2 through active coordination and cannot reassign it to Robot1 despite the replanning attempts. This happens while Robot1 is aware that it is the best candidate for the task.

Figure 18.23 shows the performance of the four different scenarios tested in reality. We can observe similar performances comparing the simulated and real robot experiments for this test case (see Figure 18.23 and Figure 18.10). ARBR has superior performance compared to its non-adaptive counterpart across all scenarios. Similar social performance can be seen for the scenarios adopting the adaptive replanning strategy. However, the distance and time ($M_1 - M_2$)



(a) *NG* **(b)** *AG*

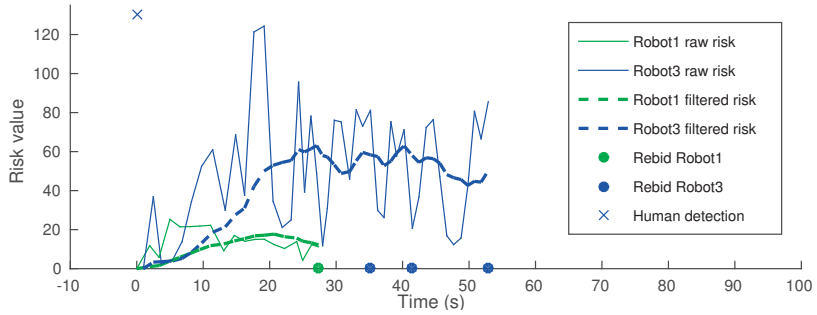
(c) *NC* **(d)** *AC*

Figure 18.21 – Task assignment per robot over time for a sample run of test case S-V for scenarios *NG*, *AG*, *NC*, and *AC* respectively. End of mission (M_2) is marked by the vertical line.

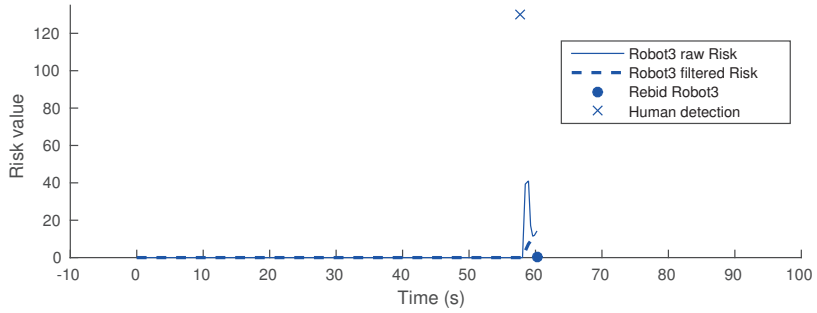
for scenario *AG* are longer compared to scenario *AC*. This is because global perception is faster to detect the human and as a result, Robot2 is dispatched to the task at an earlier time and travels a larger distance before it is notified of the change and stops. As Robot1 has also stopped earlier, the remaining distance between Robot1 and the task is larger and requires more time to traverse. With local perception however, Robot1 is stopped later and as a result, has traveled a larger segment of its path to the task before resuming its motion for the second time.

The task plan executed in each scenario can be seen in Figure 18.24. We can observe that a different task assignment is done for scenarios with adaptive replanning compared to their non-adaptive counterparts. The waiting period depicted in the task plots for Robot2 occurs since despite the increasing risk of the task that is caused by the human behavior change, Robot2 is still the best candidate to take it. Nonetheless, the increasing risk forces the robot to stop and thus, this waiting behavior emerges. Finally, the risk gets large enough to notify Robot2 that Robot1 is the best candidate for taking the task.

The risk variation for scenarios *AG* and *AC* is depicted in Figure 18.25. We can see how the risk for the task initially increases for Robot1 in both plots. Consequently, a rebid for Robot1 happens and the task is delegated to Robot2. Robot2 predicts no risk for the task as long as



(a) *AG*



(b) *AC*

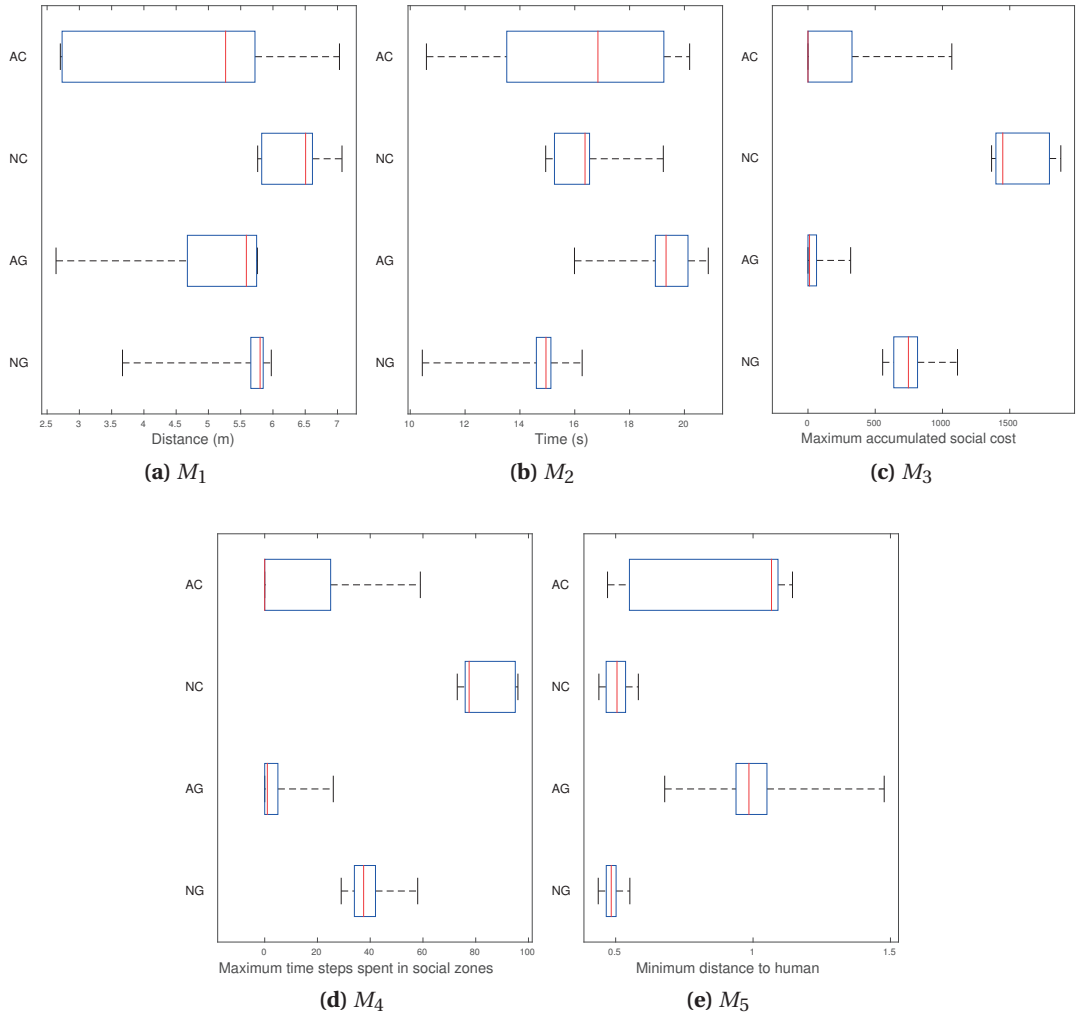
Figure 18.22 – Risk plots over time for Room5. We note that the risk signal is only plotted up to the last rebidding point. a) Robot1 and Robot3 have been affected by risk in the global perception mode, b) with local perception only Robot3 acted upon risk variations during its planning.

the human is moving towards Robot1. Once the human changes his walking direction, the risk rises and rebidding occurs. As mentioned previously, multiple rebids prevent Robot2 from moving to the task. Finally, the risk is large enough to delegate the task to Robot1 by means of active coordination.

18.4.7 Test Case R-II: Multi-Human Partially Observable Environment

In this test case, given the problem configuration in an empty arena, robot plans are as follows: Robot1 first takes Task1 and then Task3, and Robot2 takes Task2. The challenging part of this test case is that Robot1 cannot observe the two humans that are about to start walking in the arena. Given the different delays introduced in the starting time of each human's walking motion, robots are faced with different social costs. Additionally, different perceptual information is available for decision making with local perception throughout runs depending on how the humans are relatively positioned with respect to the robots.

Figure 18.26 shows the performance of the four different scenarios tested. Similar to the previous test cases, adaptive replanning has lead to better performances with respect to social metrics ($M_3 - M_5$). For ($M_1 - M_2$) however, scenario *NG* and *NC* are performing better. This is due to the waiting periods introduced in the plans and also the modified task allocation



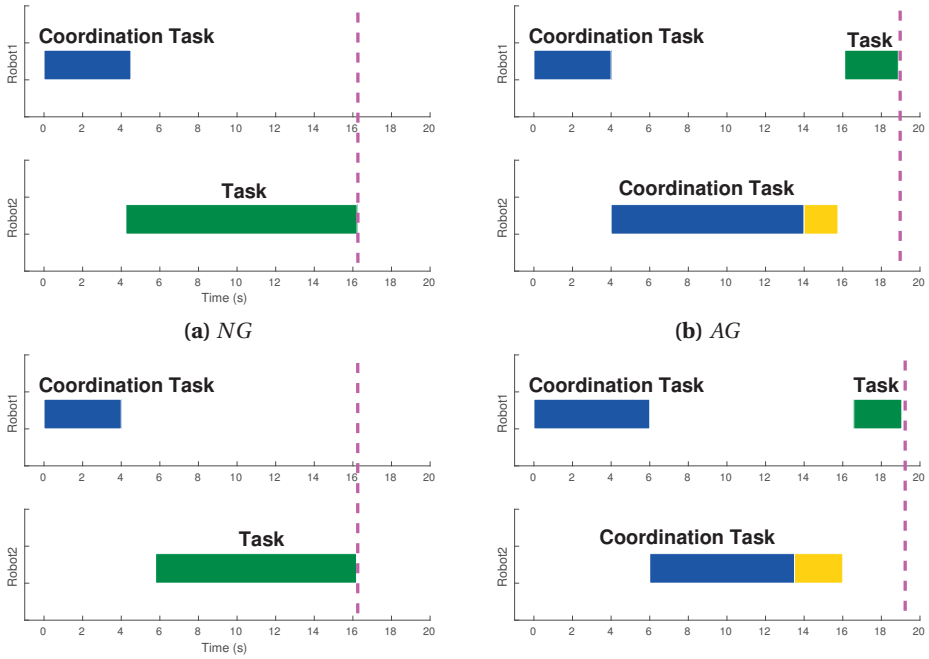
(a) M_1 **(b)** M_2 **(c)** M_3

(d) M_4 **(e)** M_5

Figure 18.23 – Performance metrics for test case R-I obtained from 10 runs.

resulted in from adapting to social costs. We can also observe that global perception has led to slightly better performance in scenarios with adaptive replanning and contrarily, slightly worse performances in scenarios with non-adaptive replanning.

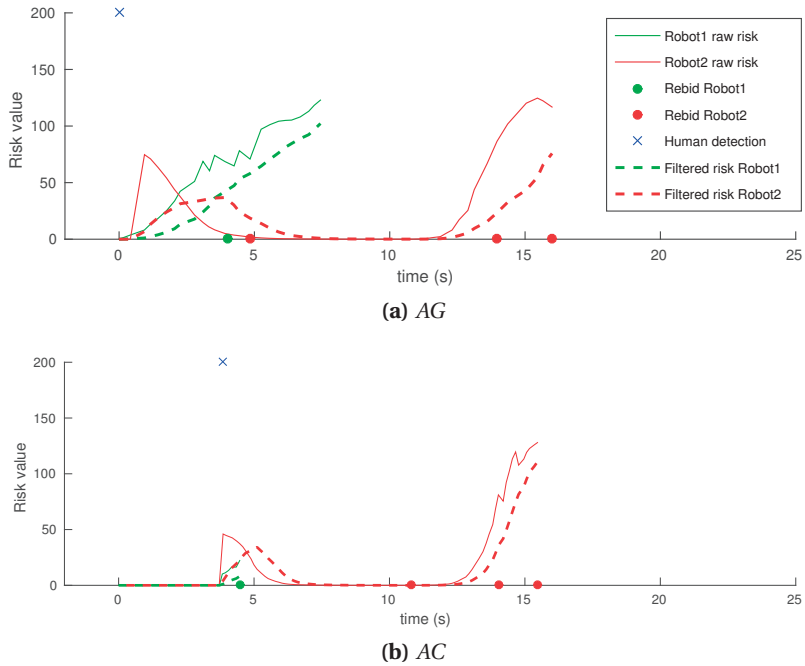
Figure 18.27 shows the robot plans for a sample run of each scenario. We can see how the plans have changed when adopting ARBR. Moreover, we can see how waiting periods have been introduced to deal with increasing social costs. In this test case, in scenarios *AG* and *AC*, Robot1 initially moves to Task1 it then stops when sensing the increased risk for all tasks. Later on it moves to Task3 and then to Task1. In some runs where at least one human started to move after a larger delay, we observed either Robot2 initially taking Task1 and then delegating it through active coordination, or completing Task1. This explains the larger variation in $M_1 - M_2$ for scenario *AG* and *AC* compared to scenarios *NG* and *NC*.



(a) *NG* **(b)** *AG*

(c) *NC* **(d)** *AC*

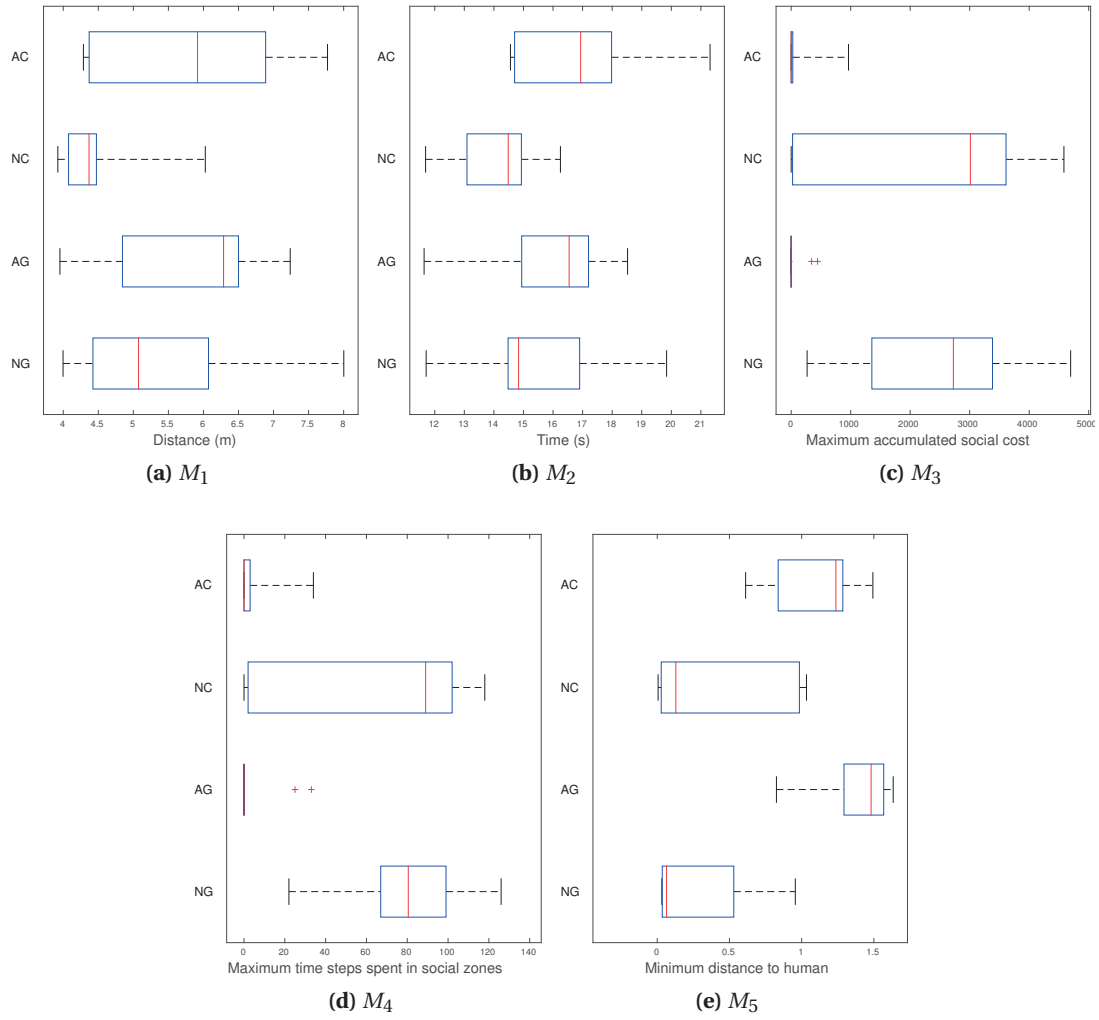
Figure 18.24 – Task assignment per robot over time for a sample run of test case R-I for scenarios *NG*, *AG*, *NC*, and *AC* respectively. End of mission (M_2) is marked by the vertical line.



(a) *AG*

(b) *AC*

Figure 18.25 – Risk plots over time for Robot1 and Robot2 in test case R-I for a) scenario *AG*, and b) scenario *AC*.

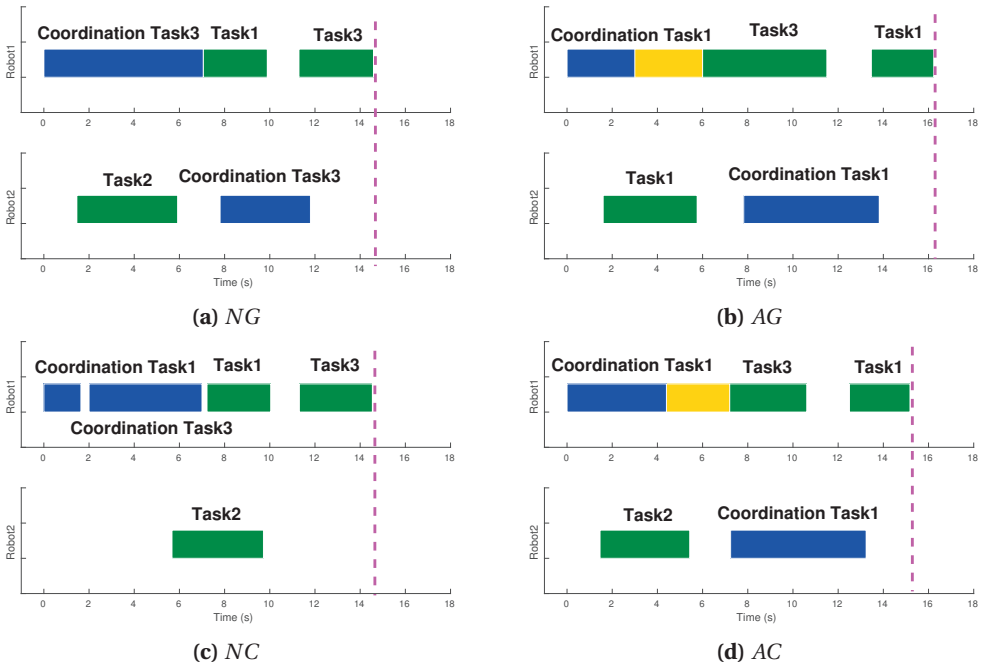


(a) M_1 **(b)** M_2 **(c)** M_3

(d) M_4 **(e)** M_5

Figure 18.26 – Performance metrics for test case R-II obtained from 10 runs.

Risk plots for all tasks in scenario *AG* and scenario *AC* are shown in Figure 18.28 and Figure 18.29 respectively. The plots for global perception show smoother risk signals due to more frequent updates and knowing the pose of all humans. Risk monitoring for Robot2 does not trigger rebidding with global perception as humans are not posing any significant risks for the task assigned to Robot2. With local perception on the other hand, both robots have had rebidding attempts although Robot1 is the robot that is mainly affected by the social risks. Figure 18.30 depicts the risk signal for all tasks and both robots for scenario *AC*. We can see how Robot1 rebids with every major change in the risk trend (when concavity of the signal changes). The rebidding attempt of Robot2 happens when it has been assigned Task1 and finds an increasing risk trend upon detection of the Human2. This risk plot belongs to a run where Human2 has had a longer delay and therefore, has not been detected by the robots earlier on. Robot2 rebids and since Human2 is moving away from Task1 at that time, Robot1 takes over the task through active coordination.



(a) *NG* **(b)** *AG*

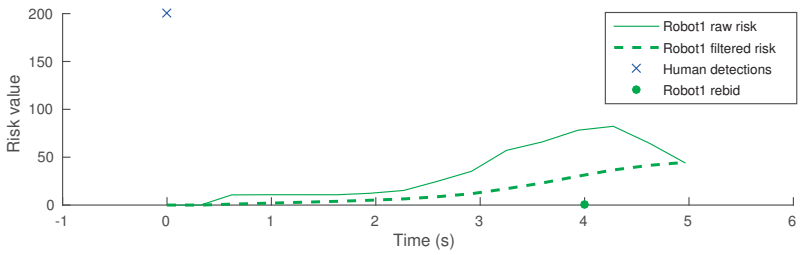
(c) *NC* **(d)** *AC*

Figure 18.27 – Task assignment per robot over time for a sample run of test case R-II for scenarios *NG*, *AG*, *NC*, and *AC* respectively. End of mission (M_2) is marked by the vertical line.

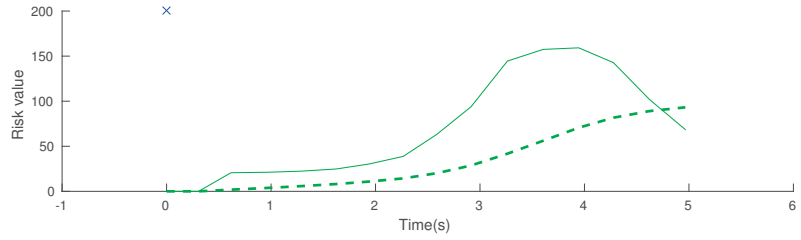
18.5 Discussion

Adaptive risk-based replanning has shown to have superior performance in terms of social metrics in all test cases studied in this chapter. This comes with the price of longer plans in terms of traveled distance and time in some cases. Global perception has shown to improve the performance of the robot team that has an adaptive replanning strategy. Without adaptation to social risks and changes in the environment, global perception can lead to worse performances compared to local perception. As the local perception gets more restricted in a highly dynamic and stochastic environment, the performance of ARBR strategy degrades. However, a realistic conic FOV of $4m$ range and 65° is shown to do very well in experiments with moderately stochastic human behavior. As the the environment gets more dynamic and stochastic, ARBR still performs significantly better compared to its non-adaptive counterpart but we can observe that social constraints are violated from time to time.

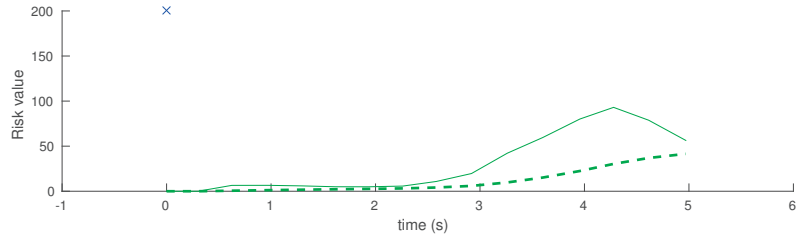
As the purpose of this chapter was to propose a replanning strategy to deal with local perception and unpredicted human behavior, we focused on the role of coordination in improving the plans. We can think of cases where the individual human-aware navigation of the robots can modify the robot trajectories to mitigate risk locally. We note that firstly, not all socially risky cases can be resolved using single robot human-aware navigation since some socially challenging situations can only be avoided on the higher level. As an example, when a person wants to exit a room and robot wants to enter the room, finding enough maneuvering space in



(a) *AG Task*1



(b) *AG Task*2



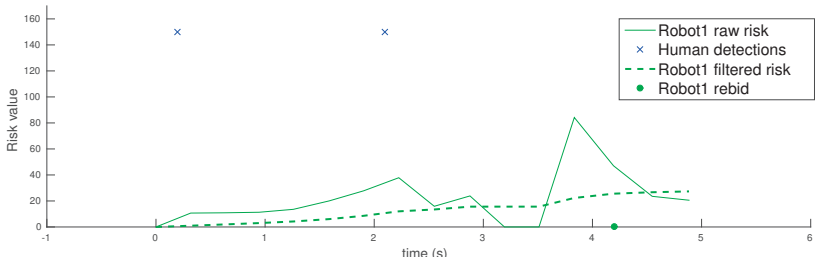
(c) *AG Task*3

Figure 18.28 – Risk plots over time for Robot1. Note that the decision to rebid depends on the active task and the uncertainty of human motion prediction and is not only based on the risk trend. a) Task1 risks, b) Task2 risks, c) Task3 risks

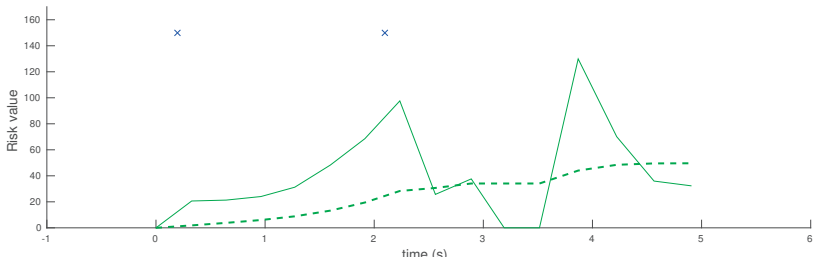
real time given the large relative speed of the human motion can be difficult. If more than one human is present in this situation things get even harder. The strategy we adopt in this case is to rely on the higher level task planner to find the most appropriate plan for the robots that eliminates such difficult social situations, and have the individual human-aware navigation resolve the local problems that might occur.

Secondly, to evaluate the impact of adaptive risk-based replanning on finding socially-aware plans we must decouple the contribution of the local human-aware path planner from that of the human-aware task planner. Thus, we have conducted all of our experiments with solely human-aware coordination. In the final integrated system comprised of all the components introduced in this thesis, we will have the adaptive risk-based replanning strategy working along with the human-aware navigation. This should further improve the social score of the robots.

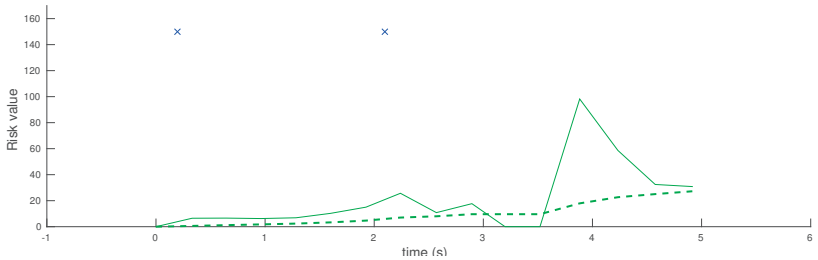
For integration of ARBR and human-aware navigation, it is important to decide how the risk should be considered with respect to the deterministic cost associated with a task. In other



(a) *AC Task*1



(b) *AC Task*2



(c) *AC Task*3

Figure 18.29 – Risk plots over time for Robot1. Notice the sharp changes in the risk signal caused by less frequent updates of local perception. a) Task1 risks, b) Task2 risks, c) Task3 risks

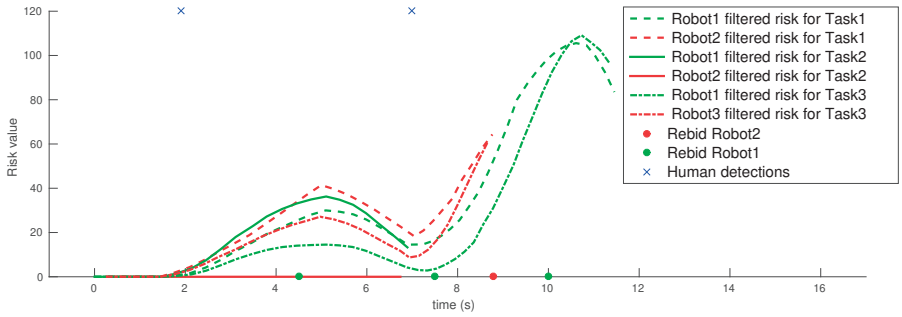


Figure 18.30 – Risk plots over time for all tasks monitored by the robots in scenario *AC* up to the last rebidding attempt.

words, how much of the problem should fall on the task planner side and how much of it should be resolved locally? We can think of robots with different risk taking preferences and thus, different characters. If a robot is too sensitive to the changes in the environment and therefore, to the risk, it has a conservative character. On the other hand, a robot might give a

lower weight to the social risks and have a risk-taking character. We can imagine a robot that is neither too conservative nor too risk-taking. This robot reacts to significant changes in risk at the task planning level and resolves local issues by means of its human-aware navigation. This is a choice that should be made based on the specific environment that the robots are operating in.

For further improvement of the work presented in this chapter the following points can be considered. Currently, map-based information is not incorporated in social risk estimation. We can think of including information such as the probability of finding an area without social costs in the vicinity of a human in addition to the distance from the human in our risk formulation. Moreover, risk can also be seen as a utility measure if we aimed at increasing robot encounter with humans. For example, if the goal of the robot team is to accomplish tasks in a way that increases the probability of human interaction, risk can be seen as utility. In conclusion, this concept of risk can be further improved by including spatio-temporal information. It can also be further explored for other applications that require a stochastic utility measure related to human motion.

Summary

In this chapter, we proposed an adaptive risk-based replanning strategy for dealing with limitations of local perception and unpredicted human behavior based on variations of social risk and human motion prediction uncertainty. Results confirm that this strategy outperforms the non-adaptive replanning strategy in all cases with respect to social metrics. The overall performance of the team depends firstly on its replanning strategy and secondly on the available information about the humans. Although an adaptive replanning strategy with global perception leads to the best performance, it is computationally expensive and infeasible in some real applications. Local perception shows comparable results as long as updates of relevant human poses affecting the risk for a task are available within the execution time of that task. Conversely, the non-adaptive replannig strategy is shown to have degraded results with global perception as decisions in this case can be based on outdated information that lead to invalid plans.

19 Conclusion

HUMAN-AWARE MRTA is an essential part of future multi-robot systems that adopt service robots in human-populated social environments. The main challenges in these environments include dealing with uncertainty of human behavior and limited robot perception. The highly stochastic nature of such environments causes robot plans to be rendered invalid or suboptimal and socially unacceptable if humans are not considered at the planning level. Accounting for humans as social beings at the individual navigation level can help to alleviate this problem in robot encounters with people but cannot fully resolve the problem at the team-level since robots are human-agnostic when planning their tasks.

Moreover, uncertainties of human behavior as well as human motion prediction are necessary components that need to be considered in MRTA for achieving plans that can adapt to the changing environment. As robots have limited local perception in reality, social MRTA methods must be able to identify situations in which past decisions should be reconsidered either due to arrival of new information that has not been available before, or due to unpredicted changes in human behavior.

For this purpose, we have based our task allocation and coordination framework on Hoplites, a market-based framework that couples planning with both passive and active coordination strategies. We modified and adapted this framework for MRTA in dynamic noisy environments and demonstrated its effectiveness in a number of problems. Subsequently, the former framework was extended to explicitly incorporate humans in its plans. Humans were considered in the proposed coordination mechanism by means of accounting for social costs in bid evaluations and requesting collaboration in socially blocking situations. As the costs of tasks are constantly changing in dynamic social environments, the concept of risk-based bids was introduced to incorporate human trajectory prediction uncertainties and furthermore, social costs in bid formulation. Finally, we introduced an adaptive risk-based replanning strategy for dealing with the limited local perception and sudden behavior changes inherent to real uncontrolled social environments.

Results confirm that risk-based bidding and adaptive risk-based replanning have superior social performance compared to their human-agnostic and risk-agnostic counterparts. This is achieved at the expense of increased traveled distance and mission time in some cases. Nonetheless, the adopted strategy ensures that all relevant aspects for socially acceptable and effective MRTA are considered by the robots while computing and executing their plans.

We note that the contributions of this part are mainly targeted to coordination and local balance estimation. Although Hoplites has shown to be very effective and particularly flexible for MRTA in social environments, we can imagine other underlying frameworks making use of the same proposed notions, mainly risk-based bids and adaptive risk-based replanning. Moreover, although the KF predictor showed to have good performance in our experiments, more advanced human trajectory predictors can easily be integrated into our framework. Similarly, more advanced social constraints can be incorporated in the proposed framework through modification of the social cost function.

Part V

HRI-Augmented Cooperative Multi-Robot Navigation

20 Introduction and Preliminaries

THIS chapter highlights the role of HRI in human-aware MRTA. Several fundamental questions are to be asked when considering integration of HRI in MRTA, and those questions form the basis of our studies in this part. Can difficult and risky social situations be resolved by means of interaction? If so, how should the decision for starting an interaction be made? Can HRI-augmented human-aware MRTA perform better than non-interactive human-aware MRTA not only with respect to social metrics but also traveled distance and time?

We note that this part of the manuscript aims at integrating all the previously presented components with the addition of HRI. Many interesting questions can be asked for understanding Human-Multi-Robot Interaction (HMRI). An interactive team of robots has enormous potential and despite this fact very few studies have been done on the subject. An in-depth study of the important questions regarding HMRI is outside the scope of this thesis. However, we have tried to focus on a few relevant questions for social MRTA. Our experiments and user studies are limited to a small group of subjects and further tests are required for drawing more statistically significant conclusions. Nonetheless, we believe our proposed solutions and the presented studies help to shed some light on this currently unexplored subject.

We start this chapter by presenting a brief introduction. As the Mbot has various interaction capacities, including speech generation, animated LEDs, and head and arm motion, it is an ideal platform for exploring the link between HRI and navigation through explicit interaction. As the first step towards creating a baseline for interactive MRTA, the integration of Mbot HRI features with robot navigation is detailed in this chapter. In Chapter 21, leveraging the concepts of social risk and adaptive risk-based replanning, we propose a HRI-augmented MRTA method that utilizes the interactive capabilities of the robots for improving both team performance and social acceptance.

20.1 Interactive Teams of Robots and Their Potentials

Although multi-robot systems are harder to interpret and understand, they offer robustness through the redundancy of their individuals. Moreover, they offer performance improvement through parallelism as individuals act collectively and solve tasks more rapidly. If we add interactive capabilities to these systems, they have the potential to be more than just mobile objects providing services, whose actions can be hard to interpret and predict. They can coordinate not only implicitly with their peers, but also explicitly with the humans. Explicit coordination of robots with the team members can also be explored as it can improve their social acceptance due to the increased legibility.

To the best of our knowledge, HRI for MRS has not been explored in the context of MRTA in the literature and very few studies exist on MRS with direct human interaction. The majority of MRS research targeting human collaboration focuses on humans commanding a team of robots. What we are instead interested in, is to know whether multi-robot teams can rely on human assistance for having a better performance and whether social robots should be treated as social beings and not solely as service providers.

In the HRI literature, it is common for social robots to make most of the effort in human encounters. Balancing the shared effort between a human and a robot for cooperative navigation has been discussed in the work of Khambhaita et al. [22]. Although robot's share in terms of effort to avoid collisions can be adapted in [22], it is considered to be unacceptable for the robot to expect more effort from the human. This is an example of the robot having a rigid service provider role in the peer vs service provider debate mentioned earlier. Nonetheless, we can imagine cases where a robot can *request* the human for some assistance effort. Is this necessarily unacceptable behavior for a robot given that this is a natural social behavior among humans? Requesting the humans for help has been studied in the literature of HRI for single robots. For more information refer to [124] and [125].

Human-robot cooperation in HRI studies, commonly targets joint goals. However, we are interested in the kind of human cooperation that would assist a robot in achieving its own goal by means of interaction. Moreover, we are mainly focusing on direct interaction with the human, unlike approaches where the cooperation takes place at the planning level and the robot takes "most of the load" [22]. So far, our proposed approach always gives priority to humans and avoids socially costly situations. However, given the interactive capabilities of the robots, this is not the only solution. Robots can influence the humans and make requests in order to modify their surroundings. This can make the initial socially costly situation evolve, allowing the robots to improve their performance. An interesting study on the effect of robot cognitive and behavior skills on the human trust [126] indicates that participants show willingness to comply with robot instructions even in the case of unusual requests despite erratic robot behavior, as long as the task requested is revocable or harmless. This supports our hypothesis of increased robot performance when adopting interactive human-aware MRTA.

We aim for improved naturalness, intent expressiveness and human involvement by means

of augmenting MRTA with interactive capabilities. Another reason for exploring the HRI capabilities of the robots in this context is to improve legibility. We observed in our previous experiments that without any signaling of robot's intention, the human subject was confused at times and could not anticipate the next actions of the robot. We believe that the importance of legibility should be further studied for HMRI as humans are faced with a more complex and possibly harder to interpret situation compared to single robot encounters.

20.2 Challenges

In this section, a number of challenges that are most relevant to interactive multi-robot systems are presented. There is an interesting article on balancing the theory and practice in HRI [4] that further details a number of these points.

- Formalism: real-world problems involving real people are very difficult to formalize as they are highly stochastic, noisy and messy [4].
- Fieldwork and long term studies: lack of long term studies with multiple robots is an evident problem for drawing meaningful conclusions in this area of research. Conducting such studies is sufficiently hard with a single robot and performing tests with multiple robots require even more effort in terms of time, scenario and metric design, result evaluation, credit assignment, etc. Moreover, multi-robot tests are technically more challenging due to the complexity of multi-robot systems. Nonetheless, long-term commitment to perform real studies and interact with users outside of the controlled laboratory environments is necessary for the advancement of research in this area [4].
- Social cost models: proxemics studies have been formulated for one robot only. Imagine a team of robots (instead of a single robot) approaching a human. It is very likely for the human to have a different level of comfort if all the robots got as close as a single robot would.
- Highly complex and difficult to predict human behavior: a lesson learned from the MOnarCH project was that introducing a second robot in the hospital ward creates a feeling of overwhelm for some people when they can see both robots at the same time. Furthermore, different reactions were observed in kids compared to the doctors and hospital staff in this situation. We note that the space available for multi-robot experiments was constrained at the hospital and the team of robots may have been perceived differently in a larger environment. This is an example showing how field studies can reveal interesting, yet surprising, findings about social environments and human preferences towards multi-robot systems that system designers and engineers may not be able to foresee. Further studies are required to understand the human assessment of multi-robot teams in real social environments.

20.3 Mbot Interactive Features

There exist a number of HRI features available on the Mbot that enable verbal and non-verbal communication (see Section 6.1.2). Mbot's non-verbal communication mainly focuses on visual cues through gestures, multimedia contents and lights. Verbal communication for the Mbot is realized by means of speech.

Gestures can either be used as a fully contained communication means, such as waving to greet someone, or as visual cues linked to a behavior or a state of mind, *e.g.*, gazing at a point of interest. Robot gaze is an important gesture that has been the subject of many HRI studies [127]. In human locomotion, gaze is used in a top down hierarchy, preceding the body motion [128]. Mimicking this behavior can be a way to increase the legibility of the robot navigation, as humans will see similar visual signs to those of the navigation behavior of other humans. In [129], the authors emphasize on the importance of gaze and show that the robot gaze affects the participants' perception of its motion, and the robot's motion affects the perception of its gaze. This dependency implies that robots should control their gaze and body motion jointly. Based on this, we implemented an anticipatory gaze gesture for indicating where the robot is planning to go by rotating the robot head toward the upcoming navigation way-points.

Multi-media content can be presented to humans by means of the robot projector or the touch-screen display. However, the proper use of these modalities requires being in an appropriate distance from the human and having access to a projection surface for the projector. Therefore, we did not include these modalities in our interactions.

Communication via light is not a natural human interaction method. Thus, it should be designed in a way that can be easily interpreted by someone that is not aware of the color scheme that the robots employ. For this purpose, a set of meaningful colors have been tested in [130]. The temporal dimension of the light signal can also be used in order to improve the interpretation of colors, as seen in [131]. We used similar schemes to [130] when adopting light in our interactions.

Speech may be the most powerful communication tool, as the misinterpretation risks are very low in comparison to the previously mentioned communication means. However, it must be ensured that verbalized information are understandable by the humans. As shown in [132], this is done by adding abstraction levels to the information. Inspired by these mentioned works, we designed a number of interactions that were integrated with individual navigation of the robots and in a later stage with the social MRTA.

20.4 Single Robot Interactive Navigation

Integration of HRI features with robot navigation in this work aims at introducing appropriate interactions on top of the existing (human-aware) navigation. Therefore, interaction modalities are utilized for demonstrating the robot's internal navigation state (see Figure 20.1). These

(a) Moving state **(b)** Stuck state

Figure 20.1 – Displaying the internal state

states include (i) idle: the robot does not have a navigation goal, (ii) moving: the robot is moving towards a specific goal and (iii) stuck: the robot has moved towards a goal, but is no longer progressing. The design of these interactions has been based on the work of [130] and [131]. Table 20.1 summarizes the description of the non-verbal interactions used for expressing different internal states of the robot.

Furthermore, we designed an interactive behavior for requesting human assistance in blocking situations by means of speech. This behavior is used when the robot is in the "stuck" state. This state is caused either by a physical obstacle blocking the robot's path or a human imposing social costs. In the former case the robot will ask the human to help with removing the obstacle, and in the latter case, the robot will ask the human to move and clear its path. Details of the gaze, greeting, asking the human for help, and asking the human to move behaviors can be found in Appendix B.

Table 20.1 – Expressing internal state with different non-verbal interaction modalities.

State	Cheeks	Eyes	Footprint	Arms	Head	Mouth
Idle	Blue pulse ($T = 10\ s$)	-	-	Rest	Rest	-
Moving	Pink pulse ($T = 2\ s$)	Green	Green	Oscillating	Gaze at way-point	Happy
Stuck	-	Red	Red blinking ($T = 1\ s$)	Rest	Gaze at the closest human	Scared

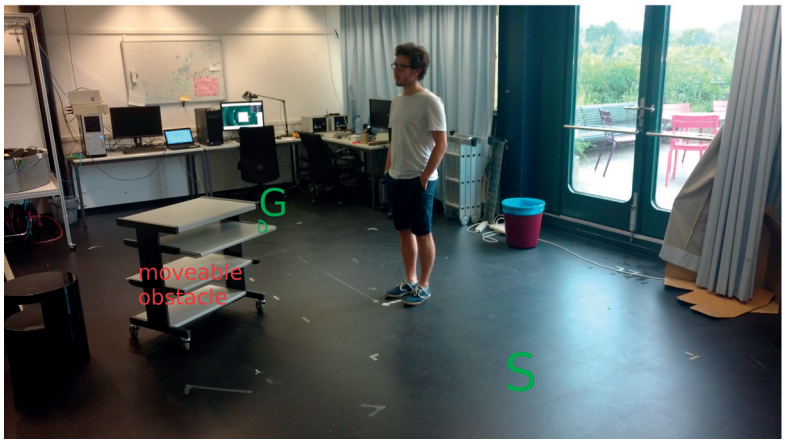


Figure 20.2 – The navigation scenario layout. S indicates the start and G indicates the goal position for the robot.

20.4.1 Experiments

In this section we will describe the test case used for our experiments. Our tests are conducted in an arena depicted in Figure 20.2. The test case consists of one robot and one human. The robot is initially in an idle state and is later dispatched to a goal. Upon receiving the navigation goal, the robot plans the path to the goal and starts moving. Subsequently, it enters the greeting zone of the static human subject. The robot is then faced with two changes compared to the initial map. It can perceive (i) a human with a corresponding social costmap, and (ii) an unmapped obstacle. This causes the robot to replan. However, it cannot find an obstacle-free path that respects the social constraints. As a result, it will get stuck. The robot can only reach its goal when the blocked path is cleared. Thus, it will ask the human for help. When the human clears enough space for the robot to pass, the scenario can be completed. We note that a human-agnostic robot will not get stuck in this case since the robot can handle the unmapped obstacle when ignoring the human.

We have conducted preliminary studies with a limited number of subjects (4 people) for the following four scenarios: (i) no interaction, (ii) non-verbal interaction only, (iii) verbal interaction only, and (iv) both types of interactions. Similarly to the work of Steinfeld et al. [133], participants were asked to give scores for their trust in the robot [0-10], engagement/attachment towards the robot [0-10], and robot legibility [0-3].

20.4.2 Results

The results of our experiments demonstrated in Tables 20.2-20.3, show the positive impact of interaction in reducing the time needed for human subjects to respond positively to the help request elicited by the robot. Moreover, interaction, and particularly verbal interaction, have shown to improve the social score of the robot. The bold entries in Table 20.2 indicate the first trial (the first encounter with the robot) for a human subject, since we randomized the order

Table 20.2 – Time needed for the completion of each scenario.

Subject	No interaction	Only non-verbal	Only verbal	Both
1	-	20s	16s	18s
2	20s	**30s**	20s	20s
3	24s	22s	**24s**	20s
4	20s	20s	24s	**18s**

Table 20.3 – Social scores.

	Before interactions	After interactions
Trust (mean %)	57.5	72.5
Engagement and acceptability (mean %)	50	80
Legibility (mean %)	40	75

of testing the different interaction scenarios for having unbiased testing. However, we note that our trials have been very limited and our experiments are only a proof of concept for the required baseline to achieve a better performance with HRI-augmented MRTA.

Summary

In this chapter, we presented a number of interactive behaviors used in single robot navigation. These behaviors serve as the building blocks for an interactive team of robots performing human-aware MRTA. Preliminary results show the positive impact of adding interaction to the robot navigation in terms of engaging humans in assisting the robot and also increased trust, acceptance and legibility.

21 HRI Assisted Cooperative Navigation

ROBOT teams can benefit from the presence of humans in social environments. Although the common approach for social robots is to find plans that minimize social costs while always giving priority to the humans, in this chapter we study a team of robots that actively ask for human assistance for achieving their goals by means of interaction. We hope to move towards the goal of having natural collaborative interactions between robots and humans by using the potential interaction capabilities of the robots, while considering the robots not only as service providers, but also as social beings. This would allow the team of robots to improve their performance, while accounting for social acceptability. We emphasize that the role of the robots will primarily be to serve the people and give higher priorities to humans. Nonetheless, we believe there is room to explore what can happen if we allowed a more natural mixing of robots with humans through interactions that already exist in our society, such as the tendency to help someone if we can. This must of course come at a reasonable price for the human, which in the case of social MRTA, could be as low as moving a few steps.

The HRI-augmented MRTA we are aiming for, calls for a number of capabilities similar to the ones identified in the work of Dias et al. [134]. The authors of [134] focus on adjusting the level of autonomy for peer-to-peer human-robot teams. Six important capabilities for performance optimization including requesting help, maintaining coordination, establishing situational awareness, enabling interactions at different levels of granularity, prioritizing team members, and learning from interactions, are proposed in [134]. Among the aforementioned capabilities, the first three are discussed by Sellner et al. [135] as major issues that affect human-awareness in multi-agent teams.

Although we are not focusing on a mixed human-robot team in this research, similar capabilities can be useful for cooperation between robots and humans in our problem of interest despite humans and robots not sharing a common goal. This is because the interest of the human in assisting the robot can be interpreted as teaming up with the robot momentarily for the robot to achieve its goal. In [134], human intervention is categorized into two primary forms of (i) physical intervention and (ii) intervention through sending direct low-granularity

commands to the robot. In both cases, the decision to intervene is made by the human. HRI-augmented MRTA can be seen in a way as a human intervention directly requested by the robots. The type of intervention we are looking for in our research is the physical type. Examples of this type of intervention are humans clearing the space for a robot to go through a socially blocked passage or removing a piece of furniture blocking the robot's path.

The interactive behaviors introduced in the previous chapter enable the robots to make their behaviors and goals understandable and legible. Moreover, direct communication of information to the humans is made possible by means of speech. This along with the human-aware MRTA method proposed in Chapter 18 creates a baseline for the interactive risk-based human-aware MRTA approach introduced in this chapter.

In the following sections, we will detail the proposed interactive MRTA method along with a number of questions we encountered while working on this problem. Real robot experiments adopting this method will be presented in Section 21.2 and the results will be discussed in the following sections.

We have focused on the feasibility and technical aspects of interactive MRTA in this chapter. Further in-depth user studies and research are required for gaining a better understanding of this problem. The insights reported in this chapter open up multiple avenues for future work in the field of HRI with multiple robots. Although long term studies with different human subjects are needed to draw solid conclusions from tests, we believe studies like this can be interesting for the researchers working at the intersection of the MRS and HRI research areas. Currently, to the best of our knowledge, there does not exist any other work with real robot experiments that studies the MRTA in social environments and particularly, studies the mutual robot human interaction aspect of the problem.

21.1 Human Assistance for Resolving Socially Costly Situations

While interactions corresponding to single robot navigation are active on each individual robot, robots can also benefit from interaction for achieving a better performance as a team. For this purpose, we consider two main aspects of multi-robot cooperative navigation in social environments. Firstly, we focus on the team efficiency for carrying out the designated tasks, by means of non-social performance metrics, namely, total traveled distance and mission time. Secondly, the social awareness aspect of the team performance is considered in our proposed method. We note that in this chapter, we mainly focus on the non-social metrics for evaluating the performance of our proposed solution. We plan on conducting a user study for having subjective assessments of legibility, naturalness and positive perception from the human point of view, to evaluate the social acceptability of our work.

Our approach for adopting interactions in human-aware MRTA is to tackle the cases where human assistance can make a large difference in terms of the non-social metrics. We target situations in which accomplishing a designated task can be very costly for the robot due to

the presence of the humans. Taking large detours for avoiding a socially blocking situation or waiting a long time for these situations to be resolved are examples of such cases. Moreover, we make sure that interactions are designed and initiated in a way that conforms to the social constraints. As an example, a robot cannot start a request while it is not in the social zone of the human who is causing the socially costly situation.

The strategy proposed for interactive MRTA is similar to the human-aware MRTA with adaptive risk-based replanning. The initial bid estimation of robots considers the future risks of each task without the possibility of interaction. The main reason for this choice is that socially costly plans should be avoided at the team-level and inevitable and unforeseen changes in the environment should be resolved locally. Previously, this was done by means of adaptation using risk-based replanning. In the interactive approach, however, instead of changing the team plans when identifying an increasing risk for the active task, the robot will resort to interaction. It will assess the risk of the situation, and if a successful interaction could result in a significantly lower risk and an acceptable cost, it will take the risk of continuing its current plan and it will request help when appropriate.

The outcome of this interaction can change the social risk trend as a result of modifying the state of the environment. In case of a positive human response and thus, low risk, there will be no need to perform active coordination to change the team plans. However, if the social risk continues to have an increasing trend because of a non-cooperative human, the requesting robot will ask for active coordination. In other words, it either delegates the task to another team member or waits until the situation changes. We note that interaction is only considered for reducing the social risk trend. Therefore, only when a robot senses increasing social risk for its active task, will it consider interaction.

Algorithm 9 details the Interactive Risk-Based Rebid Triggering (IRBRT) that implements this functionality. Lines 15-17 are modified compared to Algorithm 8. Uncertainty in human motion for increasing risk is no longer required to activate the risk-based interaction/replanning (removed from line 15), as approaching a static person and requesting for assistance are among the interactions added to MRTA. In this algorithm, upon perceiving a rising risk trend for the active task, instead of replanning, we consider an interactive request (line 16) for resolving the socially costly situation.

When a robot considers to request interaction using Algorithm 10, it initially checks whether interaction can be useful in the current situation. If so, it will find the human(s) (h_t) it should interact with. The robot then moves toward the location of the active task, while monitoring its risk. The robot periodically checks if the conditions of starting an interaction with h_t are met (Algorithm 13) and the social risk of the task is on the rise. If the risk of the active task is significantly decreased at any time, the robot will decide to not initiate the interaction and considers the risk to be resolved. However, if the interaction occurs, the robot will wait for the human response. In case of a granted request, it will move to the task location and unblock the active task. But if the request is not accepted, the robot will decide to activate replanning

Algorithm 9 Interactive Risk-Based Rebid Triggering (IRBRT) for robot r_j and task t_i given the filtered risk signal Γ_{f,t_i}, perceived human uncertainties U_{H,t_i}, and the task cost $c_{t_i,k}$ at time k

1: **procedure** IRBRT(Γ_{f,t_i}, U_{H,t_i}, $c_{t_i,k}$, t_i)
2: $f_v \leftarrow False$, $f_i \leftarrow False$, $f_a \leftarrow False$, $f_c \leftarrow False$, $f_h \leftarrow False$
3: $\gamma_{f,t_i} \leftarrow \Gamma_{f,t_i,k}$
4: $\dot{\Gamma}_{f,t_i} \leftarrow$ ComputeRiskTrend(Γ_{f,t_i})
5: $\dot{\gamma}_{f,t_i} \leftarrow \dot{\Gamma}_{f,t_i,-1}$ ▷ Take the latest risk variation
6: $\sigma_{t_i} \leftarrow max(U_{H,t_i})$
7: ▷ Note any newly perceived human that results in risk for t_i
8: **if** $(\exists h \in D_k \wedge \ \gamma_{h,t_i} > 0)$ **then**
9: $f_h \leftarrow True$ ▷ Human first encounter flag
10: **if** $|\gamma_{f,t_i} - \gamma_{b,t_i}| \geq \max(\alpha\gamma_{b,t_i}, \Gamma_{min}) \wedge (k - k_{b,t_i}) \geq K$ **then**
11: $f_v \leftarrow True$ ▷ Risk variation flag
12: **else**
return *False*
13: **if** $\neg s_{t_i} \wedge (\dot{\gamma}_{f,t_i} < 0) \wedge (\sigma_{t_i} > \Sigma \vee f_h)$ **then**
14: $f_i \leftarrow True$ ▷ Inactive task reconsideration flag
15: **if** $s_{t_j} \wedge (\dot{\gamma}_{f,t_i} > 0)$ **then**
16: **if** InteractiveRequest(Γ_{f,t_i}, U_{H,t_i}) **then**
17: $f_a \leftarrow False$ ▷ Interaction resolved the risky situation. No need for replanning
18: **else**
19: $f_a \leftarrow True$ ▷ Active task reconsideration flag
20: **if** $\neg s_{t_i}$ **then**
21: $f_c \leftarrow (c_{t_a,k} > c_{t_i,k})$ ▷ Decreased cost for the inactive task flag
22: **else**
23: $f_c \leftarrow True$
24: **return** $f_v \wedge (f_i \vee f_a) \wedge f_c$ ▷ Final reevaluation decision

and consider the social risk to be unresolved.

For understanding the details of IRBRT, we list the main questions that need to be answered in the context of this problem and describe the corresponding solutions adopted in our approach. We propose a solution (in a corresponding algorithm) to answer each of these questions. We note that each of these questions can be a separate topic of research in HRI and we do not claim to have the best answers. Nevertheless, we propose solutions that meet our needs and make an improvement compared to our previous approach.

1- How does interaction impact the team of robots in terms of planning?
In our proposed method, adopting interaction has the consequence of blocking the task for other robots. Upon deciding to interactively resolve a situation, the robot will remove the task from the list of tasks that can be considered by other robots (BlockTask function at line 5 of Algorithm 11). Without this constraint, while a robot is interacting with a human for progressing, another robot that is a better candidate for accomplishing the task may be freed. If that robot takes the task, the interaction started by the first robot will be pointless. Moreover, an interacting robot should not halt its interaction if a more appropriate task becomes available. Robots should keep their word otherwise we expect that they will lose the trust of the humans

Algorithm 10 Interactive request for human engagement in HRI-augmented MRTA for robot r_j and task t_i given the filtered risk signal Γ_{f,t_i} and perceived humans H

procedure InteractiveRequest(Γ_{f,t_i}, U_{H,t_i}, t_i)
  $f_i \leftarrow$ *False* ▷ Flag indicating if robot's request has been granted by the human
  $f_r \leftarrow$ *False* ▷ Flag indicating resolved risk
  ▷ Check if the interaction should be discarded due to the complexity of the social cost
  if DiscardInteraction(Γ_{f,t_i}, H) **then**
    return *False*
  $(h_t, p_t, v_t, \gamma_t) \leftarrow$ FindInteractionTarget(Γ_{f,t_i}, H)
  while RiskTrend(H, t_i) $> 0 \ \wedge \ \neg$StartInteraction(h_t, p_t, v_t) **do**
    ContinueToGoal(t_i)
    $(h_t, p_t, v_t, \gamma_t) \leftarrow$ FindInteractionTarget(Γ_{f,t_i}, H)
  ▷ $0 \leq \zeta < 1$ is used to check the variation in risk magnitude for detecting resolved risks
  if Risk(H, t_i) $< \zeta . \Gamma_{f,t_i,-1}$ **then**
    $f_r \leftarrow$ *True* ▷ Risk has lowered. No need to rebid
  else
    $k_i \leftarrow$ InteractAndRequest(h_t, p_t, v_t, γ_t)
    $f_i \leftarrow$ ReceiveRequestResponse(h_t, k_i, Γ_{f,t_i})
    if f_i **then** ▷ Request granted by the human. No need to rebid
      SuccessfulCollaborationInteraction(h_t, p_t)
      ContinueToGoal(t_i)
      UnblockTask(t_i)
      $f_r \leftarrow$ *True*
    else ▷ Request not granted by the human. Replanning is required
      UnsuccessfulCollaborationInteraction(h_t, p_t)
      $f_r \leftarrow$ *False*
return f_r

Algorithm 11 Identify whether interaction can resolve the socially risky situation for robot r_j given the filtered risk signal Γ_{f,t_i} and perceived humans H

procedure DiscardInteraction(Γ_{f,t_i}, H)
  $c_{t_i} \leftarrow$ ComputeCostForAllRobots(t_i)
  ▷ Consider interaction only if r_j has the lowest overall cost for t_i at this time and the situation is considered to be socially blocking
  if $r_j == argmin_{r \in R}(c_{t_i,r}) \wedge$ SociallyBlockingSituation(t_i, H) **then**
    BlockTask(t_i)
    return *True*
  else ▷ Cancel interaction. This will lead to replanning through active coordination eventually and another robot will take t_i
    return *False*

Algorithm 12 Identify human target for interaction for robot r_j given the filtered risk signal Γ_{f,t_i} and perceived humans H

procedure FindInteractionTarget(Γ_{f,t_i}, H)
  ▷ The interaction target is either the human causing the highest social cost or the center of the containing group
  $(h_t, p_t, v_t, \gamma_t) \leftarrow$FindTheHumanWithHighestCost(Γ_{f,t_i}, H)
return $(h_t, p_t, v_t, \gamma_t)$

Algorithm 13 Check whether interaction with the target should be started for robot r_j given the human target h_t, his pose p_t, and his velocity v_t

procedure StartInteraction(h_t, p_t, v_t)
  $f_s \leftarrow$ *False* ▷ Flag indicating whether interaction can be started
  $f_s \leftarrow$ CheckRelativePose(h_t, p_t, v_t)
return f_s

Algorithm 14 Interact with the human target h_t, given his pose p_t, and his velocity v_t

procedure InteractAndRequest(h_t, p_t, v_t)
  $msg \leftarrow$ ConstructMsg(h_t, p_t, v_t)
  $k_i \leftarrow 0$
  if $v_t < V$ **then** ▷ Identify static targets, V is the minimum velocity for moving people
    ApproachStopAsk(msg, h_t, p_t)
    $k_i \leftarrow k$
  else ▷ For moving targets
    ▷ Try making the request while moving but stop if the relative distance leads to social costs
    ApproachAsk(msg, h_t, p_t, v_t)
    $k_i \leftarrow k$
return k

Algorithm 15 Check if the human responds positively to the request for robot r_j given the human target h_t, and the filtered risk signal Γ_{f,t_i}

procedure ReceiveRequestResponse(h_t, k_i, Γ_{f,t_i})
  $mode \leftarrow$ "risk-based" ▷ Other modes include "distance-based" and "interactive-dialogue"
  $f_r \leftarrow$ *False* ▷ Flag indicating resolved risk
  ▷ Wait for up to K_i seconds for a decreased risk. $0 \leq \zeta < 1$ is to detect a positive human response leading to a reduced risk
  while Risk(H, t_i) $> \zeta.\Gamma_{f,t_i,-1} \wedge (k - k_i) < K_i$ **do**
    InteractiveWait(h_t, p_t)
  if Risk(H, t_i) $\leq \zeta.\Gamma_{f,t_i,-1}$ **then**
    $f_r \leftarrow$ *True*
  else if Risk(H, t_i) $> \zeta.\Gamma_{f,t_i,-1} \wedge (k - k_i) > K_i$ **then**
    $f_r \leftarrow$ *False*
  return f_r

and seem rude and unthoughtful. This is likely to negatively impact the user engagement in helping the robots.

2- In which socially risky cases should a robot consider interaction instead of replanning through active coordination?

This is a among the most important questions in this problem. Although some increasing risks can be resolved by means of interaction, there will still be cases where delegating the task to another team member will result in a better plan. For example, Figure 21.1 depicts a case where r_i encounters multiple people and avoiding social costs through interaction will be more costly despite $d_{r_i} < d_{r_j}$. If all the humans in this example responded positively to r_i and perceived the robot as socially acceptable, r_i will still need more time to accomplish the task compared to r_j. Moreover, there can be cases where the task that requires interaction could be correctly

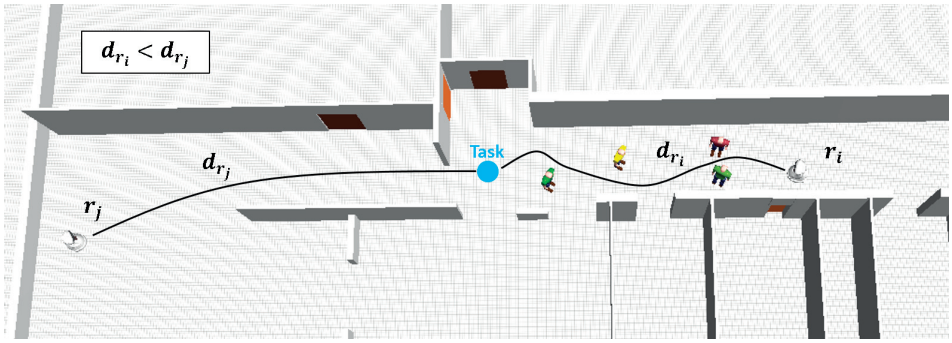


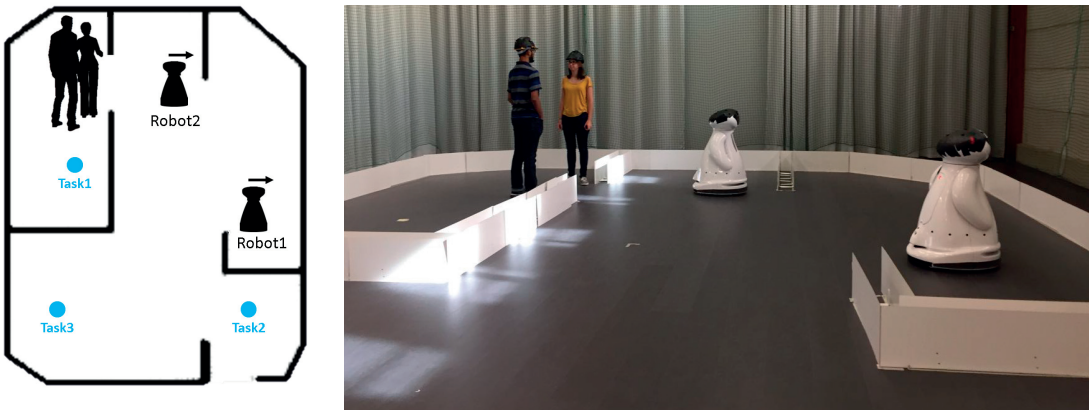
Figure 21.1 – An example where multiple interactions will be more costly than changing the plan and delegating the task to another robot.

delegated to another robot r_j through active coordination, since $c_{t_i,r_j} = min_{r \in R}(c_{t_i,r})$. If r_i blocks this task then r_j cannot take it anymore leading to a degraded assignment. This should be avoided as the decision to allow an interaction can eventually affect the whole team. `DiscardInteraction` function (Algorithm 11) is used to identify cases where interaction cannot be beneficial to the team.

3- Who should the robot interact with?
Based on the social risk associated to each human in the vicinity of the robot for the designated task, an interaction point is chosen (Algorithm 12). In the case of humans not forming a group, the human with the largest social cost is chosen as the interaction target. If the human with the largest cost is in a group, the center of that group is chosen as the interaction point. Addressing the active member of the group or each group member alternatively is a better approach for directing gaze [129], however, since detecting the active member of the group was not possible with the current sensing system, we target the center of the group instead.

4- How should the robot interact with the subject(s) in terms of message content, addressing the humans, and requesting for help?
The robot should first inform the human of its presence. For this purpose, the robot will utter a greeting message when in the vicinity of the human regardless of the heading. It will then approach the human and wait until the human faces the robot. The robot will then vocalize a message while directly gazing at the interaction point. The message will communicate a general statement for requesting help. If the interaction target is static, the robot will stop before talking to the human. For dynamic humans however, it will start interaction while it is moving but it will stop before intruding the personal space of the humans. Refer to Algorithm 14 for details of this step.

5- How should the robot analyze the human response?
There can be different ways of analyzing the human response. For instance, through the changes in the distance, by analyzing speech if the human says something back and in our implementation, by means of evaluating the social risk variation. Based on this risk, the human response to the robot request is evaluated in Algorithm 15, where the robot waits



(a) (b)

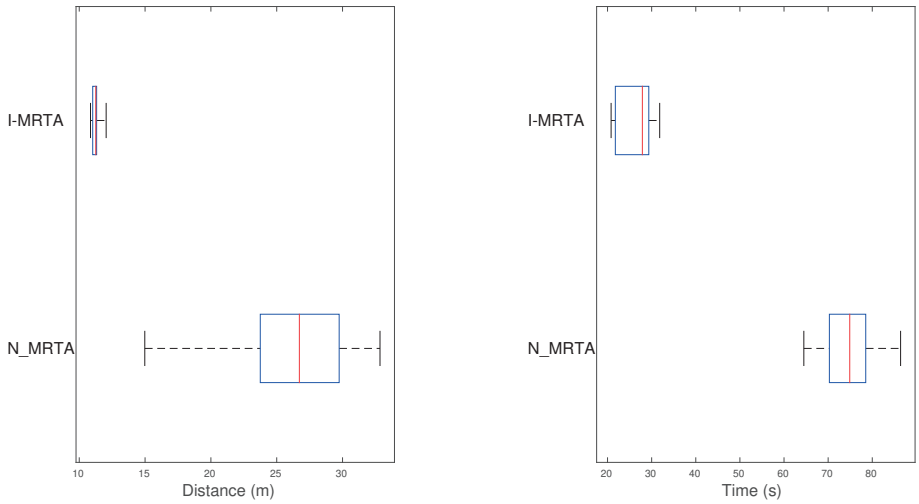
Figure 21.2 – The initial position of the robots, humans and the placement of tasks for the experiments. a) Schematics of the scenario. Arrows indicate robot orientation at initial time. b) Sample snapshot of the real scenario.

for a predefined period of time while monitoring the risk. If at any point in time the social risk of the target and the overall risk decreases significantly, the robot will consider this a positive response. Otherwise, upon time-out the robot will assume that the request has not been granted.

6- How should the robot react based on the received response?
Upon receiving a positive response, the robot will thank the human and resume moving to the designated task. In case of dynamic humans and narrow spaces, the robot will adapt its social costmap to a less restrictive model upon detecting a positive response based on risk monitoring. This is because through interaction, the positive human response is an indicator of robot-awareness on the human side and thus allows for adapting the social cost model and using the space less rigidly. If a negative response is detected, the robot will say a polite statement and perform replanning. The robot will remember to not ask this human for assistance anymore. The robot can only interact with this human in another encounter after accomplishing at least one other task. This is to avoid showing the annoying behavior of constantly returning and asking a human for help. Refer to lines 16-23 Algorithm 10 for more details.

21.2 Experiments

We have conducted a series of experiments to evaluate the performance of the proposed interactive MRTA. The scenario we tested consisted of two robots, two humans, and three tasks. The humans are initially engaged in a conversation, forming a socially blocking situation for the robots. Figure 21.2 shows the initial state of the experiment. Two algorithms have been adopted for MRTA in this setting, *I-MRTA*: interactive MRTA, and *N-MRTA*: non-interactive MRTA. The non-interactive algorithm implements the social MRTA method with adaptive



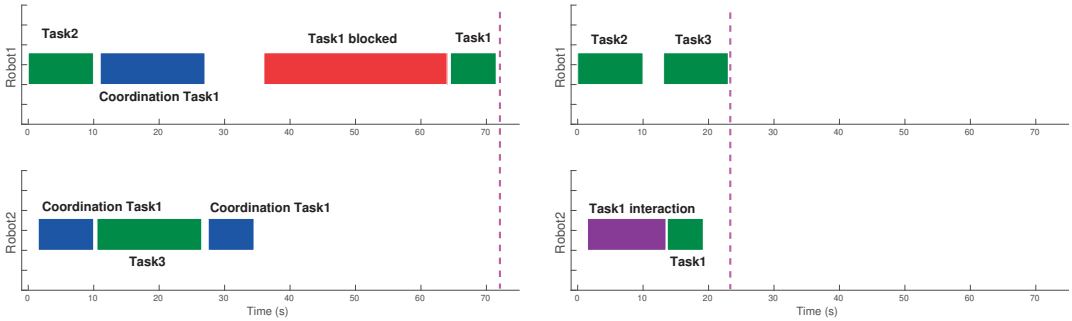
(a) M_1 **(b)** M_2

Figure 21.3 – Performance metrics obtained from 10 runs.

risk-based replanning introduced in Chapter 18. Each algorithm has been repeated for ten runs. Robots have a restricted local perception with a range of 4 m and 65° conic FOV. Robots are relying on their self-localization for computing the local balance functions.

The MRTA evaluation metrics ($M_1 - M_2$) have been obtained from ground truth values provided by the MCS. We did do not include ($M_3 - M_5$) in our evaluations since more elaborate social metrics should be used to measure the performance of the robots in terms of social acceptability as robots are no longer just implicitly interacting with humans. The social costmap changes across experiments based on human motion and interaction response. As a result, the mapping between space and social costs previously used in our evaluations is not the same with the I-MRTA method. Social metrics in this case should be subjective human assessment of the robots in terms of comfort, legibility, trust and ability to engage the human user. Those aspects are not considered in our current evaluations and are left for a future study. However, using ($M_1 - M_2$) we can still compare how these algorithms perform in terms of traveled distance and time. For more information about evaluation metrics definitions refer to Section 16.2.1.

In a human-free environment, Robot1 will take Task2 and Robot2 will take Task1. If Robot1 accomplishes its task before Robot2, it will take Task3. Otherwise, Robot2 will take Task3 and later delegate it to Robot1 through active coordination. We note that robots are positioned in such a way that they cannot see the humans initially. As a result, their initial plan does not consider the presence of the humans.



(a) $N-MRTA$ **(b)** $I-MRTA$

Figure 21.4 – Task assignment per robot over time for a sample run of the experiments. End of mission (M_2) is marked by the vertical line.

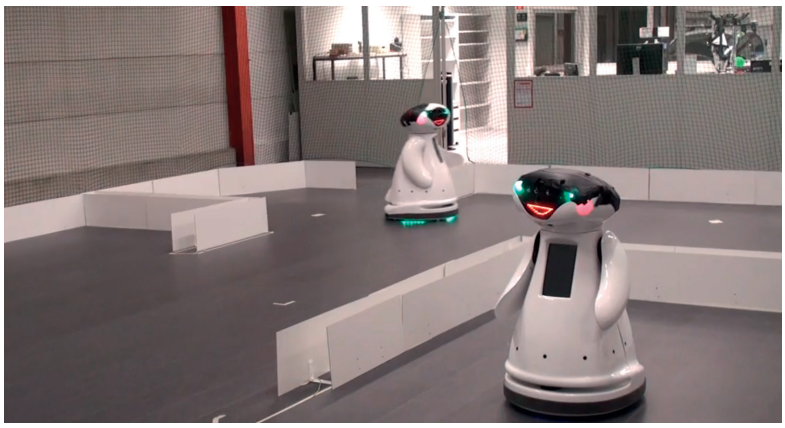
21.3 Results

The two algorithms used in our experiments lead to different team plans as they adopt different strategies. This results in a different performance with respect to M_1 and M_2 as seen in Figure 21.3. In the non-interactive algorithm (N-MRTA), when Robot2 perceives the two humans, an increased social risk is perceived and adaptive replanning is triggered. As a result, Robot2 will stop moving towards Task1 and take Task3. Then Robot1 will move to Task1 and eventually stop due to the increased risk, Robot2 will then move towards Task1 and stop when approaching the humans. Both robots will remain there due to the high social cost of the blocked passage for as long as the humans do not move. This can be seen in Figure 21.4.

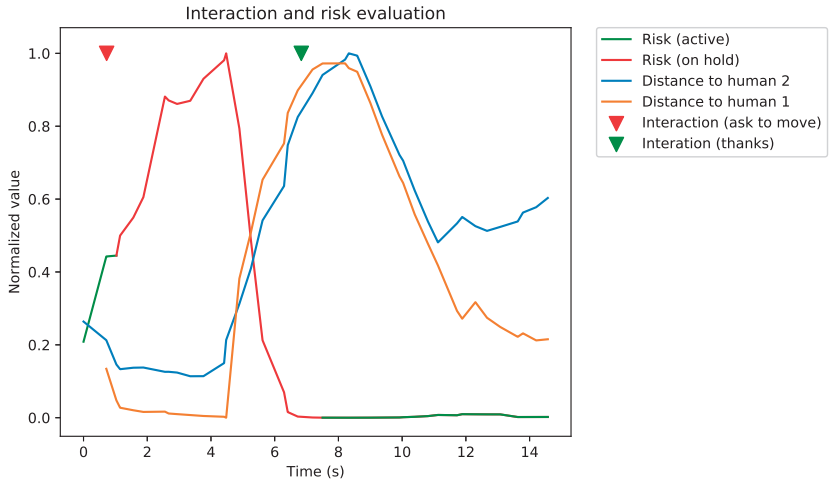
The interactive algorithm (I-MRTA) has a different strategy when encountered with the humans. Similar to the previous method, Robot1 will take Task2 and Robot will take Task1 initially. Upon detecting the humans, Robot2 will approach the humans, greet them and stop. It will then ask the humans to move. It will wait for a predefined period of time during which it monitors the current risk of Task1. When humans respond positively to the robot by clearing the path, the magnitude of the risk signal will decrease. When the risk is measured to be sufficiently low, Robot2 will consider this a successful collaboration with the humans. Subsequently, it will thank the humans and move to Task1. In the meantime, Robot1 will move to Task3 after accomplishing Task2. This leads to a more efficient plan in terms of both distance and time metrics as seen in Figure 21.3.

The task assignment for the robots using the two algorithms can be found in Figure 21.4. In this figure, the blue blocks indicate an attempt of moving to a task stopped by active coordination, the red block indicates an attempt stopped due to no progress in human-aware navigation and the purple block indicates an attempt leading to direct interaction with the humans.

Figure 21.5a shows a snapshot of the two interactive robots moving in the environment. For understanding the details of the interaction timeline in the I-MRTA algorithm, consider Figure 21.5b, where the social risk and distance to the humans are plotted for the interaction over



(a)



(b)

Figure 21.5 – a) Multiple interactive robots. b) Interaction timeline for Robot2 given the social risk and human-robot distance.

time. The social risk and distance to humans are normalized for the purpose of visualization. We can see how an interaction starts with the increasing risk (marked in red). This causes the robot to move from “active” state to “on hold”. Upon interaction, the robot verbally asks the humans to move. It then waits for their response by monitoring the social risk signal. During this period, the robot and the designated task are on hold. When the social risk has sufficiently decreased (the distance has sufficiently increased between the robot and the humans along the robot path), a successful collaboration is detected. The robot will thank the humans in this case and move towards the task, changing its state to “active” (marked in green).

21.4 Discussion

The results seen in Section 21.3 confirm the effectiveness of interactive MRTA in finding better solutions by means of actively requesting human collaboration. For N-MRTA, we simply asked the targets to keep on their conversation and move whenever they wanted. As the targets were members of our group, there was a tendency to not help the robot if not asked and this lead to long waiting times for the robot. Eventually, humans cleared the passage for the robot and the robots accomplished all the tasks. In general, if humans block a passage for a sufficiency long time (30 seconds or more), a time-out is activated and the robot is considered to be stuck.

For I-MRTA however, the humans always responded positively to the robot's request. As a result, the scenario we tested was inclined to support I-MRTA. We note that if humans had responded negatively to the robot, the traveled distance and mission time for the team would have increased for I-MRTA as well. Nonetheless, I-MRTA will lead to a modified task assignment compared to N-MRTA when encountered with social risk. In case of successful human collaboration, I-MRTA will lead to better team plans since it actively tries to change the state of the environment in favor of the robot team.

When performing experiments in the interactive mode, we observed that the limited perception of the robot was a major challenge for legibility. From the point of view of the human subjects, when the robot was facing a human it was assumed that the human is accounted for by the robot, however, this was not always the case. This was due to the limited FOV of the robots and also the human detection error. Although human tracking is performed with millimetric accuracy using the MCS, our emulated limited perception was subject to error caused by the robot's self localization. We only allowed the robot to perceive human targets that were within its sensing range and FOV given the robot position reported by its self-localization. This resulted in missed targets and delayed actions on the robot side in a number of encounters. We believe making humans aware of the robots perceptual capabilities is important in shaping their expectation of the robots and thus social acceptability of the robot team.

The social risk measured by the robots forms the basis of our interaction triggering and human response detection. Initially, we relied on the risk trend to detect whether or not a request had been granted. In other words, a negative trend in the risk signal ($\dot{\gamma} < 0$) was interpreted as a positive response. However, in most cases, the way that the humans moved led to a number of changes in the risk trend and $\dot{\gamma} < 0$ was not kept consistently for a positive human response. As this measure was too sensitive to human motion and was not robustly indicating a granted request, we opted for taking a more conservative measure of ensuring a proportionally significant decrease in the social risk value. Thus, $\texttt{Risk}(H,\ t_i) < \zeta.\Gamma_{f,t_i,-1}$ was chosen as the indicator of a positive response in our experiments, with $\zeta = 0.2$. The drawback of this choice is the longer time that it takes to detect a positive response compared to checking $\dot{\gamma} < 0$. However, this measure was shown to be much more robust for correctly detecting successful collaboration in our tests.

We note that in general, the decision to allow or discard an interaction should be based

on higher-level reasoning and not only based on the social risk signal. This requires more advanced situational awareness. As an example, a robot should not ask an injured person in a wheelchair to move, whereas this request can be reasonable for a person who is just waiting in a corridor. Knowing the state of the interaction target requires further information processing. In our experiments, interaction occurs in socially blocking situations where humans are blocking a passage or occupying a narrow space that will make social navigation costs inevitable, without considering further aspects of the problem such as the human state.

Summary

In this chapter, we proposed an interactive social MRTA method that actively requests human collaboration in socially blocking situations. Experiments with real robots confirmed that appropriate use of explicit interactions can lead to a better team performance in terms of total traveled distance and time. Interactive robots were observed to be more socially acceptable and legible as well. However, user studies with a sufficiently large human subject group are necessary for confirming the hypotheses formed in the course of our experiments. We came across a number of relevant HMRI questions in our trials that will be detailed in the next chapter.

22 Conclusion

INTERACTION can be a powerful asset for individual robots and teams of robots in social environments. Our preliminary tests showed the effectiveness of adding interactive features to the robots as individuals. Furthermore, utilizing interaction to modify the socially costly situations through requesting human help has shown to improve the traveled distance and time for the robot team if interactions are adopted appropriately. Additionally, social acceptability was observed to be improved as robots were more legible and natural. Nonetheless, interacting with the human instead of delegating the task to another robot can lead to longer plans and many user interventions if not adopted in the right situations. We emphasize that these aspect must be verified by means of a user study for concluding such observations.

The real robot experiments conducted with single and multiple interactive robots around humans, led us to a number of questions that need to be answered for enabling teams of robots to properly operate in everyday applications around humans. We list a few of these questions in the following.

1. What role should be considered for cooperative service robots in social environments with respect to humans? Should robots be merely service providers, or peers, or something in between? How does this differ for a single robot compared to a robot that is a member of a team?

2. Should explicit robot communication be directed to humans or to robot teammates (when possible)? In other words, should a robot directly address the humans or should it explicitly interact with its teammates with the goal of influencing a collaborative human behavior in an indirect manner?

3. How does the importance or urgency of a task, explicitly mentioned in the interaction, affect the human response to a robot request?

4. Should robots have an identity that distinguishes them as an individual in a team? Should they exhibit slight differences in terms of appearance, tone of voice, greeting

manner, etc.? Will people respond differently to such robots compared to robots with no identity?

5. How would a human respond to the team of robots if one or more robots exhibited faulty actions, confusing behavior or impolite conduct such as not staying engaged in an ongoing conversation with a human? Would they associate the undesired behavior to the whole team and form a judgment? What would be the long term effect of this behavior in terms of social acceptability for the team?

6. How do humans react to a specific encounter with a robot, if multiple robots repeat the same behavior at the same time, for example, being approached by multiple robots at the same time or being called by multiple robots at the same time?

Conclusion Part VI

23 Conclusion

OVER the course of this thesis, we examined the problem of cooperative human-aware navigation and coordination of multi-robot systems in social environments. As the use of mobile robots increases in environments shared with humans, applications requiring multiple robots also arise. The goal of having teams of robots in social environments is to benefit from the advantages of multi-robot systems such as providing better and faster services through increased work force, cooperation, coordination and information sharing. Furthermore, new applications such as escorting and emergency evacuation, can be enabled when adopting multiple robots. In such cases, robots are required to navigate and plan as a team while being aware of the humans and acting in a socially acceptable manner.

Human-awareness is a fundamental topic for the inclusion of multi-robot systems in real social environments as it impacts both social acceptability and performance of the team in terms of non-social metrics. This topic falls on the intersection of navigation, MRS and HRI research fields, and despite its importance, is largely unexplored from the MRS and HRI aspects. This motivated our endeavor for studying the human-aware cooperative navigation problem to understand how the presence of humans can affect the performance of a team of robots and how robots should take humans into consideration at individual and team levels when operating around people.

Overall, we proposed an end-to-end framework for cooperative navigation of human-aware multi-robot systems considering the challenges of the problem in real, uncontrolled and stochastic social environments. This problem has a broad scope and an enormous capacity for research. We hope that the framework and the methods proposed in this dissertation will contribute to providing a basis for the researchers in MRS and HRI communities. However, achieving such an outcome presents many challenges in the following research domains: (i) human-aware navigation for single robots, (ii) human-aware coordination and planning for multi-robot systems, and (iii) HRI for a team of cooperative robots. Each of these aspects, constitute an essential part of our research problem and are studied and addressed in a dedicated part of this manuscript.

23.1 Summary of Contributions

The work in this dissertation provides core contributions along the lines of the three main aforementioned research domains. We revisit these contributions in the following.

Our work on platforms and tools, detailed in Part II of this manuscript, provided the following contributions:

- Two experimental facilities of increasing complexity were established for our real robot experiments. We reproduced our experimental environments in the high-fidelity simulator Webots. This allows for testing different scenarios in terms of complexity with varying number of robots, humans, tasks and navigation difficulty as well as conducting repeated experiments under controlled conditions and also performing long-term experiments that are very expensive to have in reality.
- We investigated the problem of automatizing the calibration process of UWB-based people localization using a mobile robot in complex real world environments. This approach showed promising results for tracking a human in a number of complex settings. However, further studies with multiple humans showed that the adopted method should be further improved and expanded for crowded social environments as the collected fingerprints can largely change due to the presence of multiple people.

The contributions regarding single robot human-aware navigation detailed in Part III of the manuscript, are as follows:

- We investigated the problem of human-aware navigation for a simple and resource-constrained robotic platform with limited sensing and maneuvering. We verified the effectiveness of the human-aware navigation method based on social costmaps through repeated real robot experiments. The social costmaps were generated using the fused information coming from a leg detector and a tracker for low-lying viewpoints, adopting Kinect RGB-D information.
- The novel concept of expectation-based social costmap was introduced to capture the perception uncertainty inherent to real noisy applications. This approach was able to achieve trajectories capable of keeping a more appropriate social distance to the people, compared to those of the human-aware navigation approach that rely solely on perfect perception. Especially, when the complexity of the environment was increased. Accounting for uncertainty of perception resulted in smoother trajectories with lower jerk that are more natural from the point of view of humans.

Part IV focuses on the topic of multi-robot cooperative navigation in social environments. Our multi-robot cooperative navigation used the Hoplites framework as the coordination

baseline, and we focused on MRTA among different classes of MRS coordination problems. We summarize the contributions of this part in the following:

- As a first step towards adopting MRTA in dynamic human-populated social environments, we introduced a flexible Hoplites-based coordination framework that was shown to effectively solve the MRTA problem for three variations of increasing complexity, spatial task allocation based on distance, spatial task allocation based on time and distance, and persistent coverage.
- Human-aware deterministic bid estimation and requesting team collaboration in socially blocking situations were proposed. Thanks to these new methods, humans were considered not only at individual navigation level but also at the team-level planning. Both simulated and real robot experiments confirmed that accounting for humans at these two levels can lead to respecting social constraints as well as achieving a better performance based on MRTA metrics. However, bid estimations in this method only relied on the currently available information.
- To deal with the highly stochastic nature of social environments, we proposed the concept of risk-based bids that incorporate human trajectory prediction uncertainties and social costs in their formulation. We demonstrated the effectiveness of including a predictive component in the risk formulation despite the lack of accurate position estimation for humans. This approach was able to find more socially acceptable team plans that reduce the need for the lower level individual human-aware navigation to be activated. Risk-based plans that account for social costs, prevent difficult social situations that can lead to less effective human-aware navigation, such as traversing narrow passages occupied by humans.
- As real robot applications rely on limited local perception and human behavior can change at any time, an adaptive risk-based replanning strategy was proposed for dealing with limitations of local perception and unpredicted human behavior. This replanning method is based on the variations of social risk and human motion prediction uncertainty. Our simulated and real robot experiments confirmed that this strategy outperforms the non-adaptive replanning strategy with respect to social metrics and in some scenarios with respect to MRTA metrics as well. Furthermore, the non-adaptive replanning strategy was shown to have degraded results with global perception compared to local perception as decisions in this case can be based on outdated information that lead to invalid plans.

Finally, we explored the interactive potential of a team of robots in Part V. Our contributions are summarized as follows:

- Leveraging the basic interactive features of our robotic platform, we designed interactions with the aim of increasing robot legibility and human trust. These interactions

were integrated with individual navigation. Moreover, an interaction was designed and implemented for cases in which the navigation progress of the robot was stopped, either physically or socially. This interaction was designed to be actively initiated by the robot to ask the human for assistance. We targeted physical human assistance such as moving a furniture or stepping away from the robot's path in order to resolve the blocked navigation progress.

- Resolving socially blocking situations through explicit interaction with humans and requesting for help was integrated into our human-aware MRTA. Adaptive risk-based replanning was extended to resort to HRI, in cases where interaction has the potential to modify the environment through human actions, for the robot to achieve its goal with minimized social costs. This approach can deal with socially blocking situations that cannot be resolved by means of delegating tasks to other team members. As an example, if the only entrance to a room is blocked, no robot can reach a task located in that area, so the robot that is responsible for this task should either wait for the entrance to be cleared or navigate through the socially blocked entrance, ignoring the humans. With the interactive solution however, the robot can communicate a request for human assistance and collaboration. This interactive attempt can lead to solving the problem in many cases.

23.2 Discussion and Outlook

Among the many challenges of the problem studied in this thesis, perception, human motion and behavior prediction, estimation of highly stochastic costs, and human-robot interaction in the presence of multiple robots have been the most essential in our experience. As real social environments are highly stochastic, noisy and uncontrolled, cooperative navigation and planning methods therein, must be able to deal with the limitations of perception and its inherent uncertainty. Perception provides the key information that constitutes the basis of social cost modeling, human motion prediction and path planning. Therefore, it largely impacts the overall performance of the robot team. Multi-robot systems have a great potential for overcoming the intrinsic limitations of single robot perception by means of enhanced coverage, information sharing, cooperation, and coordination. Nonetheless, cooperative perception should consider robust solutions and perception uncertainty in particular, for human detection and tracking.

Furthermore, the highly stochastic problem of human motion and behavior predication is one key element in social MRTA that must be considered with the same approach of ensuring robustness. Similarly, opting for a stochastic approach for cost estimations that form the basis of coordination and planning for the robots in a team, can significantly improve the performance of the robots in real social environments.

Although the main social interaction addressed in this work has been the use of the common space between robots and humans, we believe that single and multi-robot cooperative naviga-

tion can largely benefit from explicit interaction for increased robot legibility and human trust as well as improved team plans. We note that we currently have limited evidence due to the lack of in-depth user studies to reliably claim these hypotheses; nevertheless, we describe a number of findings based on our observations in the following.

Our observations made it clear that appropriate use of the interactive features could help to improve the understanding of the robots' intentions as well as improving their social acceptability. Furthermore, initiating an interaction to ask for assistance, can help to associate more intelligence to the robot and additionally improve the MRTA metrics such as the total traveled distance and time. We observed the social-awareness of the robots should be further reinforced in multi-robot scenarios given the increased complexity of the encounters that humans have with the robots, since it is increasingly more likely to create confusion and discomfort with multiple robots that lack legibility.

Multiple robots enable scenarios that pose new questions to HRI researchers such as, when and how should a robot consider human assistance for improving team performance? Should robots be visibly (or vocally) identifiable as individuals given that in multi-robot scenarios, there exist a number of robots and we no longer have "the" robot as the sole non-human actor? How does the social conduct of an individual robot impact the subjective assessment of the humans about the other robots as individuals and as a team?

The framework proposed in this work comprises a number of key components, each of which can be further extended and improved. Moreover, additional plug-ins can be added to the current system. We will detail a number of such potential improvements in the following.

Although human-aware navigation is among the main components of our framework, we mainly focused on incorporation of human perception uncertainty in the social cost models used in this problem. Despite its importance in real environment, this aspect is overlooked in the majority of the research in this area. The current individual human-aware navigation method can be further improved by means of considering reciprocal and cooperative navigation of humans and robots similar to [22].

The social cost model used in this thesis relied heavily on proxemics and we focused on the costs corresponding to intrusion of personal and group interaction spaces. Nonetheless, our framework is not limited to these types of social interactions and the social cost models can easily be extended to incorporate more types of interactions. Moreover, the concept of proxemics should be further studied for multiple robots to modify and extend the existing model used in the literature for cases where more than one robot could be approaching a human.

In the course of our real robot experiments, we emulated the limited local perception for our robots, despite developing a 3D human detector based on OpenPose. This was mainly due to the Mbot hardware limitations. Local perception allows robot deployment in real human populated environments where global perception may no longer be available. Not depending

on an external tracker, largely extends the application of the robots as they will no longer be constrained to a specific area. However, local perception should be accurately characterized before such deployments as performance evaluations will have to rely on the data gathered by the local perception as ground truth, in cases where no other means are available.

We believe there are many interesting research aspects in the problem of social MRTA that are yet to be explored. Although changes happening in a highly stochastic social environment can make long plans invalid, bundle algorithms incorporating multiple tasks in their planning can still be useful in some cases. Heterogeneous teams of robots and other types of task allocation problems such as time-extended tasks, multi-task robots, multi-robot tasks, and multi-human-robot tasks can also be considered in the context of social MRTA.

Other further improvements that can be considered include using more accurate human motions prediction methods, adopting other types of human detection and tracking technologies such as laser-based leg detectors and UWB-based human trackers, performing data fusion, improving the situational awareness of the robots by means of gesture recognition, and learning the bid estimation parameters based on the data gathered in the target environment. Last but not least, long term user studies in real environments should be conducted for understanding the challenges of social MRTA, proper evaluation of the proposed solutions, and formulating the social acceptability criteria for the robots as individuals and as a team.

A Case Study: UWB-Based Person Localization

As human localization is among the main elements of human-aware navigation, we present an affordable and easy human positioning solution based on Ultra-Wideband (UWB) technology developed at the initial stages of our research in this appendix. UWB is an emerging technology in the field of indoor localization, mainly due to its high performances in indoor scenarios and relatively easy deployment. However, in complex indoor environments, its positioning accuracy may drastically decrease due to biases introduced when emitters and receivers operate in Non-Line-of-Sight (NLOS) conditions. This undesired phenomenon can be attenuated by creating, a priori, a map of the measurement error in the environment, that can be exploited at a later stage by a localization algorithm.

In this work, the error map is the result of a calibration process, which consists of collecting several measurements of the localization system at different locations in the environment. We leverage mobile robots in order to automatize the calibration process with the ultimate purpose of improving UWB-based people localization in a realistic indoor environment. The process exploits existing algorithms in the field of robot localization, conveniently adapted to the available technology for addressing our test cases.

A.1 UWB Real Time Localization System

The UWB Real Time Location System (RTLS) we used in our work is the Eliko's Kio Ranging. This solution which is based on a Decawave chipset, makes use of Unsynchronized Ranging (UR) and requires four anchors. This choice has come after a thorough comparison with the Ubisense Series 7000 system which, despite better performance in Line-of-Sight (LOS) conditions, had a worse performance in multi-room environments [136]. The same result has been obtained by Jiménez et al. in [137], who compared in a very large industrial environment the Decawave, Ubisense and BeSpoon technologies, finding out that the first solution is by far the best, in particular, in NLOS conditions.

In the Eliko's solution, the tag "pings" alternatively all the anchors and waits for a response, then it calculates the round trip time of the signal and, from this, the tag-anchor distance. This 2-way UR mechanism makes the system less accurate than those based on Cable Synchronized Ranging (CSR) and TDOA [138]. On the other hand, it has a much lower cost, and does not need any synchronization between the anchors, avoiding the use of synchronization cables, as CSR systems need. Moreover, the Kio Ranging system does not require any calibration procedure, apart from measuring the 3D coordinates of the anchors in the environment. The physical characteristics of these devices make them extremely portable and easy to mount in a variety of environments. Their dimensions are 85 $mm \times 55\ mm \times 18\ mm$ and they weight less than 20 g.

In the system's version we used, the tag outputs the measured tag-anchors distances via a serial interface at a rate of 4 Hz. The user needs its own machine and localization algorithm in order to calculate the tag's position estimate.

Two different methods have been implemented in this work. The first one performs trilateration, and requires the 3D positions of the anchors and the four tag-anchor distances as the input. In other words, it finds the least squares solution of the system of the following equations:

$$\sqrt{(x-x_u)^2+(y-y_u)^2+(z-z_u)^2} = r_u \tag{A.1}$$

where u indicates the anchor, x, y and z are the unknown coordinates of the tag, x_u, y_u and z_u are the coordinates of the u-th anchor and r_u is the measured tag-anchor distance.

The second localization algorithm is the Monte Carlo Localization algorithm (MCL)[139]. The main advantage of this method is that it is capable of exploiting the data of the fingerprinting phase and, for this reason, its performances will be compared to those of the trilateration algorithm in order to evaluate the improvements brought by our fingerprinting-based method. The details of this method will be explained in Section A.2.

A.2 Methods

Our method is an extension of [106] in order to address the problem of people localization in a realistic environment. It consists of three main steps: (i) robotic fingerprinting, (ii) creation of the error map, (iii) localization. In this section, we will explain the mentioned steps in more detail.

A.2.1 Robotic fingerprinting

In this phase, UWB measurements at multiple locations in the scenario of interest need to be collected. Making use of a UR-based UWB system, each measurement m_t is in the form of a pair (r_u, x_r), where r_u is the tag-anchor distance and x_r is the position where the measurement

has been taken, *i.e.*, the position of the robot performing the fingerprinting at that time.

As mentioned earlier, in [106] the ground-truth of the robot is measured through overhead cameras. The use of overhead cameras to get x_r has several limitations: first of all, in a real scenario where multiple rooms have to be scanned, occlusions and limited field of view may call for several cameras, which are expensive, need precise calibration, and have to be all connected to a central computer; secondly, the installation of this kind of system in a public environment may be faced with privacy issues.

Consequently, we use the robot localization data to obtain x_r. Although this does not provide the same accuracy as overhead cameras, the AMCL-based self-localization system of the robot is a good compromise between accuracy and usability in real-world scenarios.

When performing the fingerprinting the tag was mounted on a structure on the top of the robot, at an overall height of 170 *cm*. For better performances, different heights should be considered. The consequence of taking all the measurements at a fixed height is that the accuracy of the error map will be maximum for the localization of a person of that height. The fingerprinting path was previously coded in the robot, so that it could follow it automatically. The navigation capabilities of the robot allowed it to adapt the fingerprinting path in case of obstacles. Every 30 *cm*, the robot stopped and took measurements at different orientations, rotating around its vertical axis.

A.2.2 Error Map

The output of the first phase is a large quantity of UR (range) measurements for each anchor and for many different positions and orientations in the environment. The goal of the second phase is to process this data in order to obtain a map of the error. Our error map is divided into squared regions (1 $m \times 1$ m) and describes the expected UR measurement error (ranging error) in each grid cell and for each anchor.

In more details, this error is described in the form of its Probability Density Function (PDF). The PDF is computed starting from a general error model, in the form of a parametrized error PDF, whose parameters are chosen in order to best fit our measurements.

The error model has been defined starting from [140] as a multimodal PDF. We can formulate it as follows:

$$p(\Delta r;\boldsymbol{\theta}) = p_{\mathcal{N}}(\Delta r)\cdot P_L + p_{\ln\mathcal{N}}(\Delta r-\mu_{\mathcal{N}})\cdot(1-P_L) \tag{A.2}$$

where $p(\Delta r;\boldsymbol{\theta})$ indicates the probability density function of measuring an error Δr, $p_{\mathcal{N}}(\cdot)$ is a normal distribution of mean $\mu_{\mathcal{N}}$ and variance $\sigma_{\mathcal{N}}$, $p_{ln\mathcal{N}}(\cdot)$ is a log-normal distribution of mean $\mu_{\ln\mathcal{N}}$ and variance $\sigma_{\ln\mathcal{N}}$. The value P_L is in $[0,1]$ and sets the balance between the normal and the log-normal components. Notice that the log-normal distribution is translated

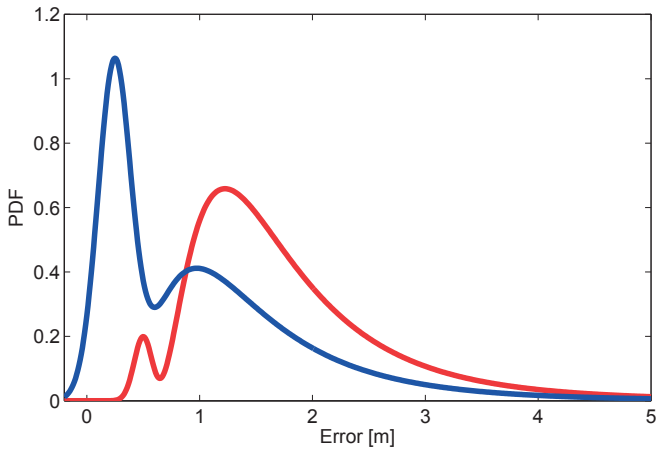


Figure A.1 – Examples of error PDF obtained by choosing the parameters of our error model as $\theta_1 = [0.5, 0.08, 0.1, 0.65, 0.04]$ (red) and $\theta_2 = [0.25, 0.15, 0.1, 0.65, 0.4]$ (blue). The red PDF describes the error in an area much more affected by multipath phenomena than the blue one. The red plot is also the PDF of the UR error measured in an experiment described in Section A.4.

to the right by $\mu_{\mathcal{N}}$. Since $\mu_{\mathcal{N}}$ is the mean of the normal distribution, we can say that it acts as a horizontal bias for the whole PDF. This additional degree of freedom has been added to allow for fitting our measurements to the model more closely.

In simple words, the explanation of this model is the following: the normal part represents the smaller errors measured on the direct path UWB signals; the log-normal part represents the much larger errors that exist due to the multi-path phenomena. Two examples of realizations of our error model are shown in Figure A.1.

$\boldsymbol{\theta}$ is the vector of the parameters of our error model and is defined as:

$$\boldsymbol{\theta} = [\mu_{\mathcal{N}}, \sigma_{\mathcal{N}}, \mu_{\ln\mathcal{N}}, \sigma_{\ln\mathcal{N}}, P_L]^T \tag{A.3}$$

Now it should be clear that the error map we want to achieve is in the form of a set of parameters $\boldsymbol{\theta}_{u,v}$, where $u = 1 \ldots N_a$ specifies one of the N_a anchors and $v = 1 \ldots N_r$ indicates the region index, with N_r the total number of regions.

In order to estimate $\boldsymbol{\theta}_{u,v}$, we only consider the fingerprinting measurements taken in the region R_v relative to the anchor A_u. Then, we follow the curve fitting approach presented in [140], adapted by us to suit the UR case, instead of CSR. This approach is a heuristic that estimates the parameters of our error model according to the selected measurements. More details can be found in [106][136].

A.2.3 Localization

The *third step* focuses on localization. This is the only online step of our method. A person walks on a predefined path at constant speed (~5 km/h) with a tag on his head. The position of the tag on the top of the head has been chosen after an extensive series of measurements and tests [136], where it was concluded to be the best choice, since it minimizes the probability of the tag being covered by parts of the human body. The tag continuously measures the distance to the four anchors at an update rate of 4 *Hz*. Its measurements are read through its serial interface and stored for later processing. According to [106], an estimate of the tag's position is obtained using MCL [139]. This algorithm uses a particle filter where each particle $\boldsymbol{x}^{[i]}$ represents a position in three dimensions and its weight $w^{[i]}$ is computed considering the UWB measurement and the error map. We configured the height of the particles according to the person's height, which in our case is 170 *cm*. The number of particles used by the particle filter M, is configurable and sets the balance between performance and computational complexity. In our case, 500 particles have been used.

Algorithm 16 shows how MCL works in our application. First, the `Initialization` is performed by sampling the position of the particles from a bidimensional Gaussian distribution centered on the supposed person's starting point. If the starting point is unknown, MCL can be initialized by spreading the particles uniformly over the environment. After initialization each particle has weight $w_0 = 1/M$. Then, the `Update` function uses the set of UR measures $\mathcal{M}_t = \{r_{t,1} \dots r_{t,N_a}\}$, taken at time t from all the anchors, to update the weights of the particles according to the following equations:

$$\Delta r_{t,u} = |\boldsymbol{A}_u - \boldsymbol{x}_t^{[i]}| - r_{t,u} \tag{A.4}$$

$$w_t^{[i]} = \prod_{u=1}^{N_a} p(\Delta r_{t,u}; \theta_{u,v}) \tag{A.5}$$

where $\boldsymbol{A}_u$ is the known position of the u-th anchor and $\theta_{u,v}$ is the vector of the parameters that characterize the error PDF associated to the same u-th anchor in the region R_v, given $x_t^{[i]} \in R_v$. At the end of the `Update` function, the position U_p of the person is estimated as the weighted average of all the particles.

The next step is the `Sample` function: a resampling algorithm is used to select which particles to keep and which to discard, according to their weight. In our case, we used the *low variance resampling* algorithm explained in [141].

Finally, the `Move` function corresponds to the prediction step of MCL, which aims at changing the position of the particles according to the predicted next position of the tracked object. If available, for instance on a robotic platform, odometry data are used in this step: the particles are moved according to the measured translation of the robot. The lack of odometry data makes people localization much more challenging. Although several methods such as the Kalman Filter (KF) can be used to predict the movement of a person even without external

Algorithm 16 MCL algorithm adopting fingerprinting in its `Update` step

```
Initialization
for t = 1 to ∞ do
    X̄_t = ∅
    for i = 1 to M do
        w_t^[i] ← Update(M_t, x_t^[i])
        X̄_t ← X̄_t ∪ ⟨x_t^[i], w_t^[i]⟩
    U_p = (Σ_i w^[i] x^[i]) / (Σ_i w^[i])
    for i = 1 to M do
        x_t^[i] ← Sample(X̄_t)
        x_{t+1}^[i] ← Move(x_t^[i])
```

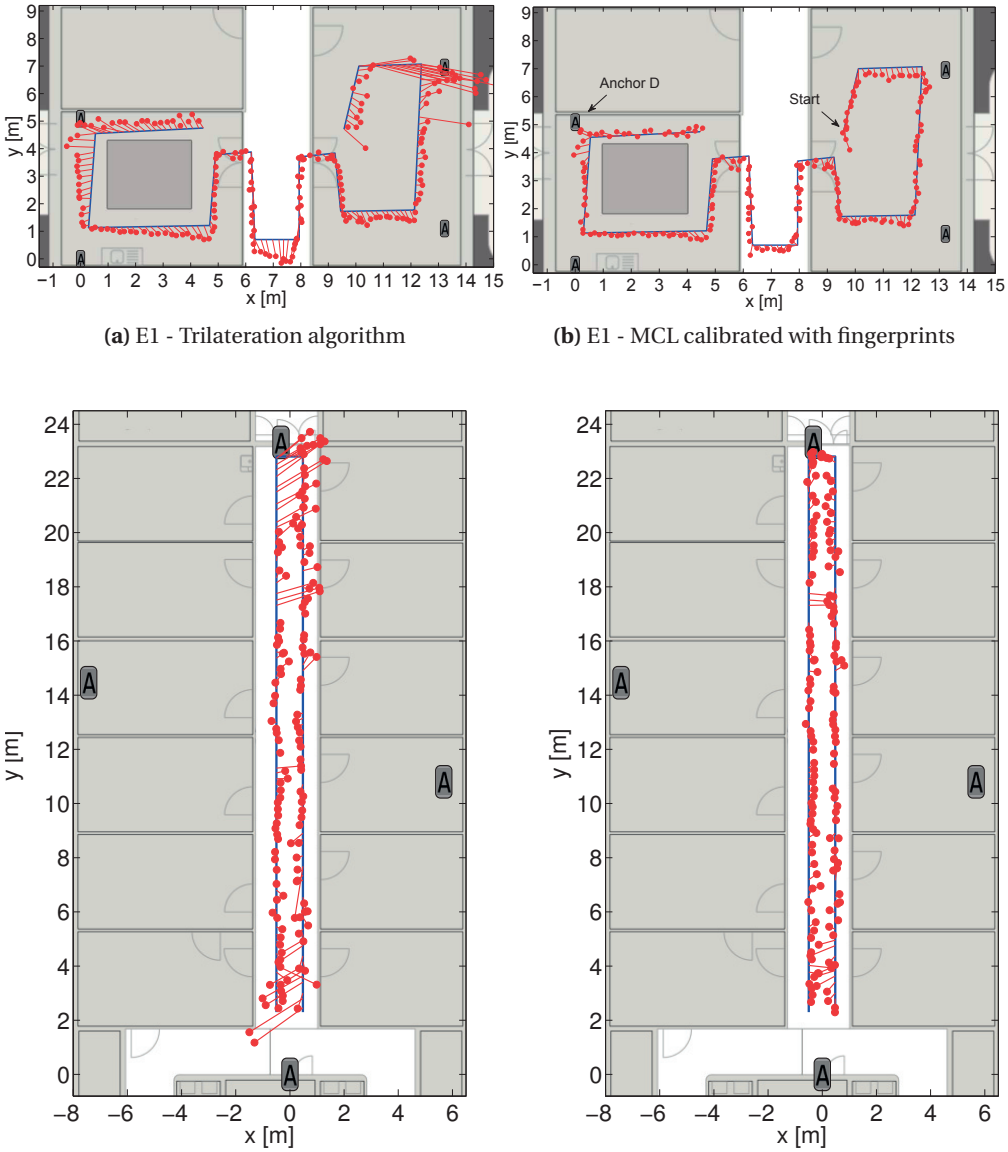
data, in our case we consider the movement of the person as completely random, but limited in speed. For this reason the `Move` function simply applies a zero-mean Gaussian noise to the 2D position of each particle. The variance σ^2 of this noise has to be chosen according to the sample rate of the algorithm and the supposed maximum speed of the person. In our case we set $\sigma^2 = 0.25$.

In order to test the improvements brought by the use of the fingerprinting data, the localization is computed independently using directly the trilateration algorithm explained in Section A.1 (without fingerprinting) and the MCL (calibrated with fingerprinting). The same UWB UR measurements collected in our tests were used as input for both algorithms.

A.3 Setup and Experiments

We considered two different scenarios of incrementally increasing complexity and scope (see Figure A.2). For simplicity, we will call them (E1) and (E2). The area of the testing environment for (E1) is approximately 100 m^2 while for (E2) it is three times larger. In some locations of (E2) we have up to four walls between the tag and an anchor. This is an extremely challenging condition for a radio-based localization system. Moreover, in (E2) we have on the left-hand side of the corridor (refer to the scheme in Figure A.2) several metallic cabinets, that challenge the UWB system even more. However, all the walls are non-bearing.

We also tested the case of bearing walls in the line between a tag and an anchor, and we noticed that this condition makes the performance of the system drastically decrease, probably due to the metallic structural elements they have inside. The choice of such realistic scenarios is given by our goal of considering a real use case with a limited number of anchors in order to keep the cost and complexity of the overall system low, even though we know it has a negative impact on the localization accuracy. The anchors, indicated in the figures with letter A, have been installed at an height of 2.51 m, close to the ceiling. Their position has been measured in our coordinate system very accurately with the help of a laser meter. For the fingerprinting phase, the robot has been programmed in order to autonomously scan the environment. Roughly 1200 measurements per square meter have been taken, at 12 uniformly distributed orientations



(a) E1 - Trilateration algorithm

(b) E1 - MCL calibrated with fingerprints

(c) E2 - Trilateration algorithm

(d) E2 - MCL calibrated with fingerprints

Figure A.2 – Tracking results on first (a-b) and second (c-d) scenarios using the trilateration (a-c) and MCL calibrated with fingerprints (b-d). The blue line is the path of the walking person, while the red lines show the error between the location estimates and their corresponding true positions. Notice how, particularly in the second segment of the path, the accuracy is higher using the fingerprinting-calibrated method.

around the vertical axis. The robot's self-localization feature has fundamental importance in our work. We underline that AMCL has proven to be very robust against unknown obstacles like people (the laser range finders can detect only the legs), bags on the floor and closed/open doors. In the environment of our tests we measured an average self-localization accuracy in the order of 20 *cm*.

During the localization phase a person was walking with a tag on his head along the path

indicated by the blue line. In order to record the UR measurements outputted by the tag, the person was walking carrying a bag with a laptop connected via USB to the tag. The ground truth of the person has been computed by precisely measuring the time taken by the person to walk between the various checkpoints at constant speed. Although, it is theoretically possible to compute the localization estimate online, we did it offline on the basis of the measurements collected.

To have an idea about the accuracy of camera-based tracking solutions compared to the UWB localization methods used in this study, we have made assessments using a marker-based tracking system (SwisTrack) [105] with an overhead camera, and a marker-less solution used in [142] with an omni-directional ceiling mounted-camera. The covering area of the two methods were substantially smaller for a single camera compared to the area covered by the UWB solution. A realistic estimate of the maximum error across the entire arena is in the order of 0.05 m and 0.2 m for the aforementioned marker-based and marker-less solutions, respectively (refer to Section A.4 for comparison). An obvious benefit of using UWB systems is the ability to overcome the field of view limitations of camera-based tracking systems. In this study, we were only allowed to use cameras in the laboratory area due to privacy issues. In general, it can happen in many cases that the entire environment cannot be covered by overhead cameras and hence UWB systems have a significant advantage therein.

A.4 Results

Figure A.2a and A.2b compare the results obtained in the first scenario using the trilateration algorithm (a) and MCL calibrated through fingerprinting (b). The red lines show the correspondence between the true position of the person and the estimated one. Let us start our analysis of the results considering the first environment (E1). We can clearly see that, using trilateration, in the second segment of the path the error is particularly high, reaching the maximum value of 3.75 m, while in the rest of the path the average error is 0.36 m.

The red curve of Figure A.1 shows, as resulted by the fingerprinting process, the UR error PDF measured in the second segment of the path in (E1), with respect to anchor D. The interpretation of this PDF is that in the second segment of the path, the distance between the tag and the anchor D is measured with a high positive bias. This behavior is probably due to the presence of multiple walls on the direct path between these two transceivers. This fact also justifies the right bias of the location estimates, as observed in Figure A.2a.

Using the MCL algorithm that exploits the fingerprinting data, we obtained the results shown in Figure A.2b. In particular, we can see that we had a great improvement in the second segment of the path, were the right bias has disappeared and the error has reached the same level as the rest of the path. Considering the whole path, the mean error decreased from 0.51 m to 0.25 m, that is roughly 50% of improvement. However, given the non-uniform distribution of the error, in Figure A.3 we show its Cumulative Distribution Function (CDF), which gives a better overview of the improvements achieved.

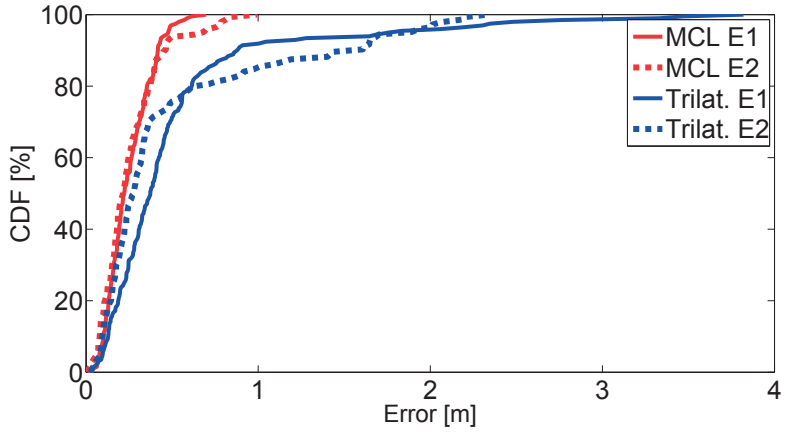


Figure A.3 – CDF of the error measured in the two scenarios (E1,E2) using the trilateration algorithm and MCL calibrated via robotic fingerprinting. Notice that with the latter method, the errors above one meter are completely cancelled.

The error CDF shows that our method is less effective against errors lower than 0.30 *m*, while it works very well against higher errors. This behavior is due to approximations introduced by our general UR error model, the usage of the ground-truth provided by AMCL, the assumption of Gaussian uniform motion model of the human, limited number of measurements in the fingerprinting phase, etc. For many applications, such as people escorting with robots, localizing a person in a circle of 0.50 *m* radius is sufficient. Considering this value as our threshold, we measured that our method increased the number of measurements below this threshold from 70% to 98% using only a single tag.

The results for the second scenario, are shown in Figure A.2c and A.2d. The anchors are still indicated with the letter "A". We just considered the central corridor and not the rooms, since it is sufficiently complex for our needs.

Comparing the results obtained using the trilateration algorithm and fingerprinting-calibrated MCL, we noticed that, similar to the first scenario, most of the improvements have been brought in the areas which were affected by higher errors: in our case at the top and bottom parts of the map. In the central part of the map, we measured better performances given the lower number of obstacles in the tag-anchor line. The mean error, along the whole path, in the trilateration and MCL cases are respectively 0.48 *m* and 0.25 *m*, confirming an improvement of roughly 50%. The A.3 of the error for both the algorithms in this second scenario are shown as the dotted lines, in Figure A.3. Notice that the use of fingerprinting-calibrated MCL made the percentage of localization errors below 0.50 *m* increase from 76% to 94%. Table A.1 summarizes the results discussed so far.

As previously pointed out, MCL is based on a particle filter, which gives the advantages of setting the balance between performance and computational load simply by adjusting the number of particles. In this scenario, we tested the localization algorithm with different num-

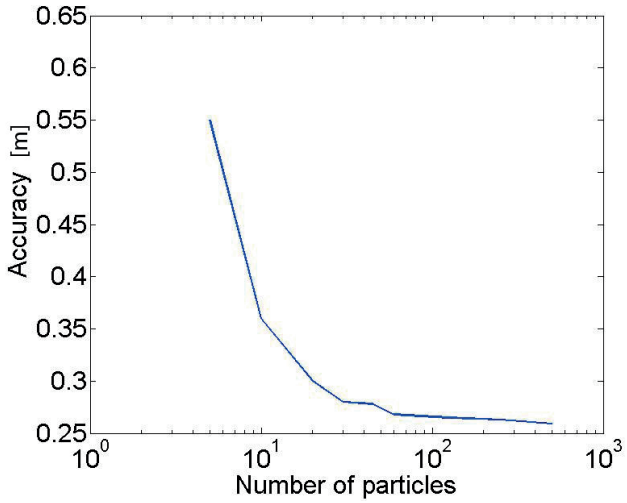


Figure A.4 – Analysis of the performance of the MCL localization algorithm as a function of the number of particles used in the particle filter for the second. Notice that increasing the number of particles over one hundred does not significantly improve the performances.

	Trilateration		MCL + Fingerpr.	
	Average [m]	<50cm	Average [m]	<50cm
E1	0.51	70%	0.25	98%
E2	0.48	76%	0.25	94%

Table A.1 – Statistics on measurement errors using the two algorithms in both environments.

ber of particles. According to the results shown in Figure A.4, the number of particles significantly affects the performances for values lower than one hundred, while for higher values the improvements tend to stabilize. Therefore, we chose 500 particles for our method, for obtaining a good balance between computational cost and performance.

B Single Robot Interactive Navigation

This appendix details the gaze, greeting, requesting the human for help, and requesting the human to move, interactive behaviors. We note that each flow chart is iterated over during the time that a robot is active.

B.1 Gaze

The robot always gazes at a future way-point while it is moving. This is to help the human understand what trajectory the robot is going to take. Thus, robot's head will perform an anticipatory motion to be aligned with a gaze point. This point is the closest way-point in the robot's trajectory that has a distance greater than 1.5 m from the current position of the robot. If there is no such way-point, the robot will instead gaze towards its goal. We note that this choice of gaze point has been made to achieve a behavior that looks natural by observation. However, it can be further improved by performing studies on this particular aspect of the robot. A limitation has been added to the gaze motion in order to prevent the head from rotating too much when looking at the human, since it can be considered unnatural from the human point of view if the robot head deviates too much from the body frame. Thus, the head rotation angle is limited to $\pm 60°$.

B.2 Greeting

Activation of each interaction can have different parametrization based on its importance. For example, "ask for help" (when stuck) interaction, is enabled if the robot is less than 3 m away from the human with any relative angle. On the other hand, the "greeting" interaction is only triggered when the robot is facing the human (having a relative angle between $-90°$ and $+90°$), and closer than 2 m. Figure B.1 depicts the flowchart of this behavior.

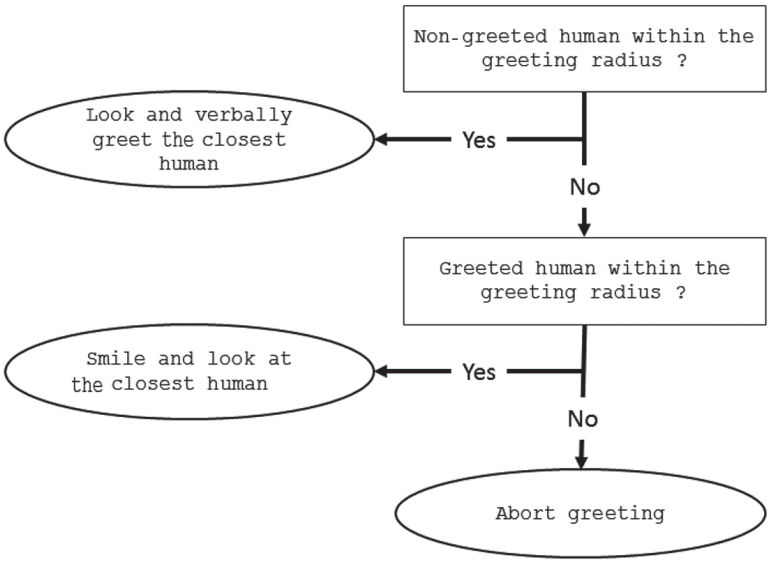


Figure B.1 – Greeting interaction

B.3 Ask for Help

When the robot's internal navigation state switches to stuck, it activates the ask for help interaction shown in Figure B.2. There is another condition required for this interaction to be activated based on the proxemics principle. The robot must be in the social space of the human. In our implementation, the robot has to be within the 2 m radius of the human and facing him. Therefore, the robot actively positions itself in this area to initiate the interaction when possible.

B.4 Ask the Human to Move

If the comparison between the human-aware and human-agnostic paths outputs a maximum distance between the way-points greater than a fixed threshold of 0.5 m, the robot will look for the closest human in a predefined radius around the point of maximum path difference to initiate interaction. If at least one such human is found, the motion vector will be transformed to simple abstract instructions and the interaction illustrated in Figure B.3 will be triggered. Similar to the previous interaction, the robot must be in the human's social zone for initiating interaction. This interaction aims to find a compromise between having to travel a large distance and asking for human assistance.

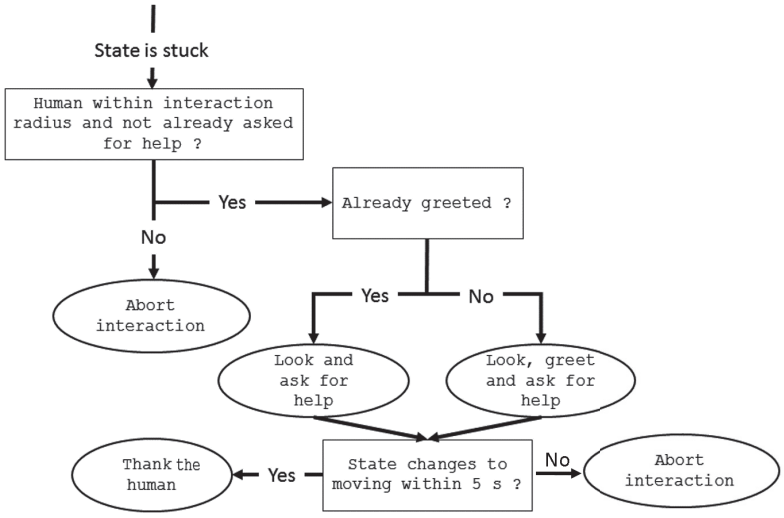


Figure B.2 – Ask for human help interaction when the navigation state of the robot changes to "stuck".

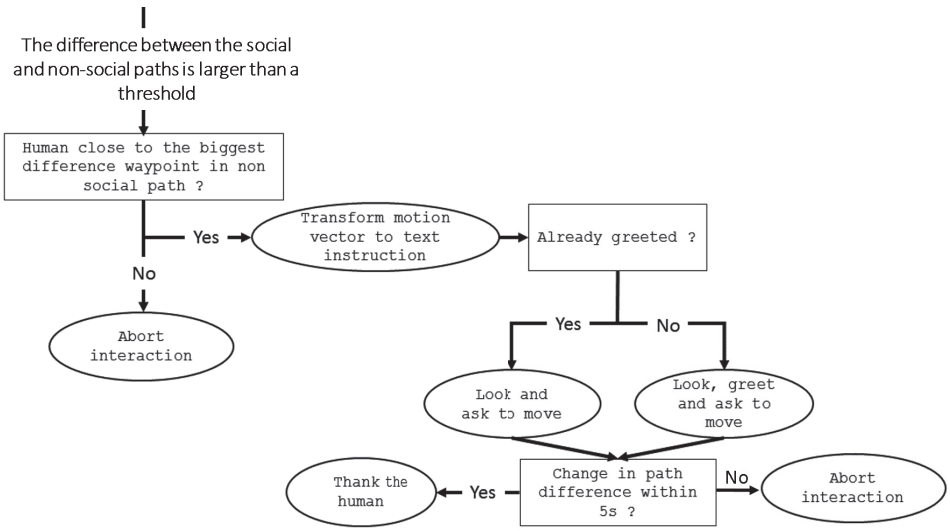


Figure B.3 – Ask to move interaction.

Glossary

AdaBoost	Adaptive Boosting	38
AMCL	Adaptive Monte Carlo Localization	34, 51, 70
ARBR	Adaptive Risk-Based Replanning	127, 131
CBAA	Consensus-Based Auction Algorithm	15, 85, 86
CBBA	Consensus-Based Bundle Algorithm	15–17, 87
CDF	Cumulative Distribution Function	206, 207
CSR	Cable Synchronized Ranging	200, 202
DNN	Deep Neural Networks	39
DWA	Dynamic Window Approach	12, 34, 52, 53
FMM	Fast Marching Method	11, 12, 34, 52, 53, 55, 65, 68, 87, 99, 100
FOV	Field Of View	29, 39, 136, 183, 186
HMM	Hidden Markov Model	13
HMRI	Human-Multi-Robot Interaction	167, 169, 187
HOG	Histogram of Oriented Gradients	34, 38, 41
HRI	Human Robot Interaction	4, 10, 17, 19, 34, 167, 168, 170, 173, 175, 176, 178, 193, 196, 197
IPOL	Instituto Portugues de Oncologia de Lisboa	32, 101
KF	Kalman Filter	13, 38, 58, 118, 203
LOS	Line-of-Sight	199
LRF	Laser Range Finder	30, 34, 35
MCL	Monte Carlo Localization algorithm	200, 203, 204, 207
MCMC	Markov Chain Monte Carlo	67–69, 75
MCS	Motion Capture System	37, 40, 42, 45, 77, 103, 183, 186
MDP	Markov Decision Process	16
MOnarCH	Multi-Robot Cognitive Systems Operating in Hospitals	24, 30, 32–34, 36, 40, 42, 52, 169

MRS	Multi-Robot Systems	4, 14, 18, 19, 87, 168, 176, 193, 195
MRTA	Multi-Robot Task Allocation	14–16, 18–20, 22–24, 81, 96, 101, 167, 168, 170, 173, 175, 176, 195–198
NLOS	Non-Line-of-Sight	37, 65, 199
ODE	Ordinary Differential Equation	52
PDF	Probability Density Function	201–203, 206
PF	Particle Filter	51
POMDP	Partially Observable Markov Decision Process	16, 20
RMSE	Root Mean Squared Error	71
ROS	Robot Operating System	27, 30, 33, 34, 36, 40, 51, 52, 55, 57, 65, 102
RTLS	Real Time Location System	38, 199
SFM	Social Force Model	17
SLAM	Simultaneous Localization and Mapping	51, 52
SPENCER	Social situation-aware PErceptioN and action for CognitivE Robots	3
SVM	Support Vector Machine	13, 38
UR	Unsynchronized Ranging	199–204
UWB	Ultra-Wideband	24, 37, 38, 45, 65, 198–200, 202–204, 206

Bibliography

[1] T. Kruse, A. K. Pandey, R. Alami and A. Kirsch, "Human-aware robot navigation: a survey", *Robotics and Autonomous Systems*, vol. 61, no. 12, pp. 1726 –1743, 2013.

[2] D. Vasquez, P. Stein, J. Rios-Martinez, A. Escobedo, A. Spalanzani and C. Laugier, "Human aware navigation for assistive robotics", in *Experimental Robotics*, 2013, pp. 449–462.

[3] H. Khambhaita and R. Alami, "Assessing the social criteria for human-robot collaborative navigation: a comparison of human-aware navigation planners", in *IEEE International Symposium on Robot and Human Interactive Communication*, 2017, pp. 1140 –1145.

[4] M. Matarić, "On relevance: balancing theory and practice in HRI", *ACM Transactions on Human-Robot Interaction*, vol. 7, no. 1, 8:1–8:2, May 2018.

[5] P. Althaus, H. Ishiguro, T. Kanda, T. Miyashita and H. Christensen, "Navigation for human-robot interaction tasks", in *Proceedings of the IEEE International Conference on Robotics and Automation*, vol. 2, 2004, pp. 1894–1900.

[6] E. A. Sisbot, L. F. Marin-Urias, R. Alami and T. Simeon, "A human aware mobile robot motion planner", *IEEE Transactions on Robotics*, vol. 23, no. 5, pp. 874–883, 2007.

[7] J. L. Drury, J. Scholtz and H. A. Yanco, "Awareness in human-robot interactions", in *International Conference on Systems, Man and Cybernetics*, IEEE, vol. 1, 2003, pp. 912–918.

[8] J. Martinez, "Socially-aware robot navigation: combining risk assessment and social conventions", PhD thesis, Ph. D. dissertation, University of Grenoble, 2013.

[9] J. Rios-Martinez, A. Spalanzani and C. Laugier, "From proxemics theory to socially-aware navigation: a survey", *International Journal of Social Robotics*, vol. 7, no. 2, pp. 137–153, 2015.

[10] K. Dautenhahn, M. Walters, S. Woods, K. L. Koay and C. L. Nehaniv, "How may I serve you?: a robot companion approaching a seated person in a helping context", in *1st ACM SIGCHI/SIGART conference on Humanrobot interaction*, ACM Press, 2006, pp. 172–179.

[11] L. Takayama and C. Pantofaru, "Influences on proxemic behaviors in human-robot interaction", in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2009, pp. 5495–5502.

[12] E. Martinson, "Hiding the acoustic signature of a mobile robot", in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2007, pp. 985–990.

[13] D. Vasquez, P. Stein, J. Rios-Martinez, A. Escobedo, A. Spalanzani and C. Laugier, "Human Aware Navigation for Assistive Robotics", in *13th International Symposium on Experimental Robotics*, 2012.

[14] C. Lichtenthäler and A. Kirsch, "Towards legible robot navigation—how to increase the intend expressiveness of robot navigation behavior", in *International conference on social robotics—workshop embodied communication of goals and intentions*, 2013.

[15] T. Kruse, P. Basili, S. Glasauer and A. Kirsch, "Legible robot navigation in the proximity of moving humans", in *IEEE Workshop on Advanced Robotics and its Social Impacts,*, 2012, pp. 83–88.

[16] A. D. Dragan, K. C. Lee and S. S. Srinivasa, "Legibility and predictability of robot motion", in *2013 8th ACM/IEEE International Conference on Human-Robot Interaction,*, 2013, pp. 301–308.

[17] J. Mumm and B. Mutlu, "Human-robot proxemics: physical and psychological distancing in human-robot interaction", in *6th ACM/IEEE International Conference on Human-Robot Interaction*, 2011, pp. 331–338.

[18] M. Walters, M. Oskoei, D. Syrdal and K. Dautenhahn, "A long-term human-robot proxemic study", in *IEEE RO-MAN*, 2011, pp. 137–142.

[19] A. Mateus, P. Miraldo, P. U. Lima and J. Sequeira, "Human-aware navigation using external omnidirectional cameras", in *Robot 2015: Second Iberian Robotics Conference*, Springer, 2016, pp. 283–295.

[20] E. T. Hall, "The hidden dimension", 1966.

[21] J. Rios-Martinez, A. Spalanzani and C. Laugier, "Understanding human interaction for probabilistic autonomous navigation using risk-RRT approach", in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2011, pp. 2014–2019.

[22] H. Khambhaita and R. Alami, "Viewing robot navigation in human environment as a cooperative activity", *arXiv preprint arXiv:1708.01267*, 2017.

[23] R. Mead, "Situated proxemics and multimodal communication: space, speech, and gesture in human-robot interaction", in *Ph.D. Dissertation*, University of Southern California, 2015.

[24] R. Kirby, R. Simmons and J. Forlizzi, "Companion: a constraint-optimizing method for person-acceptable navigation", in *The 18th I International Symposium on Robot and Human Interactive Communication, 2009.*, 2009, pp. 607–612.

[25] B. Okal and K. O. Arras, "Learning socially normative robot navigation behaviors with bayesian inverse reinforcement learning", in *Robotics and Automation (ICRA), 2016 IEEE International Conference on*, IEEE, 2016, pp. 2889–2895.

[26] K. Charalampous, I. Kostavelis and A. Gasteratos, "Recent trends in social aware robot navigation: a survey", *Robotics and Autonomous Systems*, vol. 93, pp. 85–104, 2017.

[27] S. M. LaValle, *Planning algorithms*. Cambridge university press, 2006.

[28] H. M. Choset, S. Hutchinson, K. M. Lynch, G. Kantor, W. Burgard, L. E. Kavraki and S. Thrun, *Principles of robot motion: theory, algorithms, and implementation*. MIT press, 2005.

[29] A. Gasparetto, P. Boscariol, A. Lanzutti and R. Vidoni, "Path planning and trajectory planning algorithms: a general overview", in *Motion and operation planning of robotic systems*, Springer, 2015, pp. 3–27.

[30] R. Ventura and A. Ahmad, "Towards optimal robot navigation in domestic spaces", in *RoboCup 2014: Robot World Cup XVIII*, Springer, 2015, pp. 318–331.

[31] J. A. Sethian, "Fast marching methods", *SIAM review*, vol. 41, no. 2, pp. 199–235, 1999.

[32] J. V. Gómez, N. Mavridis and S. Garrido, "Fast marching solution for the social path planning problem", in *IEEE International Conference on Robotics and Automation*, 2014, pp. 2243–2248.

[33] J. V. Gomez, N. Mavridis and S. Garrido, "Social path planning: generic human-robot interaction framework for robotic navigation tasks", in *2nd Intl. Workshop on Cognitive Robotics Systems: Replicating Human Actions and Activities*, 2013.

[34] A. Valero-Gomez, J. V. Gomez, S. Garrido and L. Moreno, "Fast marching methods in path planning", *IEEE Robotics and Automation Magazine, no*, 2013.

[35] D. Fox, W. Burgard and S. Thrun, "The dynamic window approach to collision avoidance", *IEEE Robotics & Automation Magazine*, vol. 4, no. 1, pp. 23–33, 1997.

[36] M. Hoy, A. S. Matveev and A. V. Savkin, "Algorithms for collision-free navigation of mobile robots in complex cluttered environments: a survey", *Robotica*, vol. 33, no. 3, pp. 463–497, 2015.

[37] F. Hoeller, D. Schulz, M. Moors and F. E. Schneider, "Accompanying persons with a mobile robot using motion prediction and probabilistic roadmaps", in *International Conference on Intelligent Robots and Systems,*, IEEE/RSJ, 2007, pp. 1260–1265.

[38] M. Luber, L. Spinello, J. Silva and K. O. Arras, "Socially-aware robot navigation: a learning approach", in *International conference on Intelligent robots and systems*, IEEE/RSJ, 2012, pp. 902–907.

[39] H. Kretzschmar, M. Kuderer and W. Burgard, "Learning to predict trajectories of cooperatively navigating agents", in *International Conference onRobotics and Automation*, IEEE, 2014, pp. 4015–4020.

[40] G. Ferrer and A. Sanfeliu, "Behavior estimation for a complete framework for human motion prediction in crowded environments", in *2014 IEEE International Conference on Robotics and Automation (ICRA)*, 2014, pp. 5940–5945.

[41] G. Ferrer, A. Garrell and A. Sanfeliu, "Robot companion: a social-force based approach with human awareness-navigation in crowded environments", in *IEEE/RSJ international conference on Intelligent robots and systems (IROS)*, IEEE, 2013, pp. 1688–1694.

[42] G. Ferrer and A. Sanfeliu, "Multi-objective cost-to-go functions on robot navigation in dynamic environments", in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, IEEE, 2015, pp. 3824–3829.

[43] S. T. O'Callaghan, S. P. N. Singh, A. Alempijevic and F. T. Ramos, "Learning navigational maps by observing human motion patterns", in *IEEE International Conference on Robotics and Automation*, 2011, pp. 4333–4340.

[44] D. Vasquez, B. Okal and K. O. Arras, "Inverse reinforcement learning algorithms and features for robot navigation in crowds: an experimental comparison", in *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2014, pp. 1341–1346.

[45] S.-Y. Chung and H.-P. Huang, "Predictive navigation by understanding human motion patterns", *International Journal of Advanced Robotic Systems*, vol. 8, no. 1, p. 3, 2011.

[46] S. Xiao, Z. Wang and J. Folkesson, "Unsupervised robot learning to predict person motion", in *2015 IEEE International Conference on Robotics and Automation (ICRA)*, 2015, pp. 691–696.

[47] A. Rudenko, L. Palmieri and K. O. Arras, "Joint long-term prediction of human motion using a planning-based social force approach", in *Proceedings of IEEE International Conference on Robotics and Automation, Brisbane*, May 2018.

[48] J. Correa Villanueva, "Uncertainty and social considerations for mobile assistive robot navigation", PhD thesis, Imperial College London, 2014.

[49] R. Mead and M. J. Mataric, "Probabilistic models of proxemics for spatially situated communication in hri", in *International Conference on Human-Robot Interaction, Algorithmic Human-Robot Interaction Workshop*, 2014.

[50] B. P. Gerkey and M. J. Matarić, "A formal analysis and taxonomy of task allocation in multi-robot systems", *The International Journal of Robotics Research*, vol. 23, no. 9, pp. 939–954, 2004.

[51] G. A. Korsah, A. Stentz and M. B. Dias, "A comprehensive taxonomy for multi-robot task allocation", *The International Journal of Robotics Research*, vol. 32, no. 12, pp. 1495–1512, 2013.

[52] E. Nunes, M. Manner, H. Mitiche and M. Gini, "A taxonomy for task allocation problems with temporal and ordering constraints", *Robotics and Autonomous Systems*, vol. 90, pp. 55–70, 2017.

[53] A. Khamis, A. Hussein and A. Elmogy, "Multi-robot task allocation: a review of the state-of-the-art", in *Cooperative Robots and Sensor Networks 2015*, Springer, 2015, pp. 31–51.

[54] Y. Cao, W. Yu, W. Ren and G. Chen, "An overview of recent progress in the study of distributed multi-agent coordination", *IEEE Transactions on Industrial informatics*, vol. 9, no. 1, pp. 427–438, 2013.

[55] M. B. Dias, R. Zlot, N. Kalra and A. Stentz, "Market-based multirobot coordination: a survey and analysis", *Proceedings of the IEEE*, vol. 94, no. 7, pp. 1257–1270, 2006.

[56] L. Xu and A. Stentz, "Market-based coordination of coupled robot systems", in *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*, IEEE, 2011, pp. 2784–2789.

[57] G. P. Das, T. M. McGinnity, S. A. Coleman and L. Behera, "A distributed task allocation algorithm for a multi-robot system in healthcare facilities", *Journal of Intelligent & Robotic Systems*, vol. 80, no. 1, pp. 33–58, 2015.

[58] M. G. Lagoudakis, E. Markakis, D. Kempe, P. Keskinocak, A. J. Kleywegt, S. Koenig, C. A. Tovey, A. Meyerson and S. Jain, "Auction-based multi-robot routing.", in *Robotics: Science and Systems*, Rome, Italy, vol. 5, 2005, pp. 343–350.

[59] B. P. Gerkey and M. J. Mataric, "Sold!: auction methods for multirobot coordination", *IEEE transactions on robotics and automation*, vol. 18, no. 5, pp. 758–768, 2002.

[60] M. B. Dias, R. Zlot, M. Zinck, J. P. Gonzalez and A. Stentz, "A versatile implementation of the traderbots approach for multirobot coordination", 2004.

[61] N. Kalra, D. Ferguson and A. Stentz, "Hoplites: a market-based framework for planned tight coordination in multirobot teams", in *IEEE International Conference on Robotics and Automation*, 2005, pp. 1170–1177.

[62] L. E. Parker, "Alliance: an architecture for fault tolerant multirobot cooperation", *IEEE Transactions on Robotics and Automation*, vol. 14, no. 2, pp. 220–240, 1998.

[63] R. K. Ahuja, T. L. Magnanti and J. B. Orlin, "Network flows: theory, algorithms, and applications", 1993.

[64] B. B. Werger and M. J. Matarić, "Broadcast of local eligibility for multi-target observation", in *Distributed Autonomous Robotic Systems 4*, Springer, 2000, pp. 347–356.

[65] M. Roth, D. Vail and M. Veloso, "A real-time world model for multi-robot teams with high-latency communication", in *Intelligent Robots and Systems, 2003.(IROS 2003). Proceedings. 2003 IEEE/RSJ International Conference on*, IEEE, vol. 3, 2003, pp. 2494–2499.

[66] D. P. Bertsekas, "The auction algorithm: a distributed relaxation method for the assignment problem", *Annals of operations research*, vol. 14, no. 1, pp. 105–123, 1988.

[67] D. P. Bertsekas and D. A. Castañon, "Parallel synchronous and asynchronous implementations of the auction algorithm", *Parallel Computing*, vol. 17, no. 6, pp. 707–732, 1991.

[68] H.-L. Choi, L. Brunet and J. P. How, "Consensus-based decentralized auctions for robust task allocation", *IEEE Transactions on Robotics*, vol. 25, no. 4, pp. 912–926, 2009.

[69] M. J. Krieger, J.-B. Billeter and L. Keller, "Ant-like task allocation and recruitment in cooperative robots", *Nature*, vol. 406, no. 6799, pp. 992–995, 2000.

[70] W. Agassounon and A. Martinoli, "Efficiency and robustness of threshold-based distributed allocation algorithms in multi-agent systems", in *Autonomous Agents and Multi-Agent Systems: part 3*, ACM, 2002, pp. 1090–1097.

[71] G. Beni, "From swarm intelligence to swarm robotics", in *International Workshop on Swarm Robotics*, Springer, 2004, pp. 1–9.

[72] E. Şahin, "Swarm robotics: from sources of inspiration to domains of application", in *International workshop on swarm robotics*, Springer, 2004, pp. 10–20.

[73] N. Kalra and A. Martinoli, "Comparative study of market-based and threshold-based task allocation", in *Distributed autonomous robotic systems 7*, Springer, 2006, pp. 91–101.

[74] A. Macwan, J. Vilela, G. Nejat and B. Benhabib, "Multi-robot deployment for wilderness search and rescue", *Int. J. Robot. Autom.*, vol. 31, no. 1, 2016.

[75] G. Lozenguez, L. Adouane, A. Beynier, A.-I. Mouaddib and P. Martinet, "Punctual versus continuous auction coordination for multi-robot and multi-task topological navigation", *Autonomous Robots*, vol. 40, no. 4, pp. 599–613, 2016.

[76] M. Gendreau, G. Laporte and R. Séguin, "Stochastic vehicle routing", *European Journal of Operational Research*, vol. 88, no. 1, pp. 3–12, 1996.

[77] S. S. Ponda, L. B. Johnson and J. P. How, "Distributed chance-constrained task allocation for autonomous multi-agent teams", in *American Control Conference*, 2012, pp. 4528–4533.

[78] J. F. Quindlen and J. P. How, "Machine learning for efficient sampling-based algorithms in robust multi-agent planning under uncertainty", in *AIAA Guidance, Navigation, and Control Conference*, 2017, pp. 1921–1940.

[79] J. Capitan, M. T. Spaan, L. Merino and A. Ollero, "Decentralized multi-robot cooperation with auctioned pomdps", *The International Journal of Robotics Research*, vol. 32, no. 6, pp. 650–671, 2013.

[80] S. Sariel-Talay, T. R. Balch and N. Erdogan, "Multiple traveling robot problem: a solution based on dynamic task selection and robust execution", *IEEE/ASME Transactions on Mechatronics*, vol. 14, no. 2, pp. 198–206, 2009.

[81] U. C. Usug and S. S. Talay, "Dynamic temporal planning for multirobot systems.", in *Automated Action Planning for Autonomous Mobile Robots*, 2011.

[82] B. P. Sellner and R. Simmons, "Towards proactive replanning for multi-robot teams", in *Proceedings of the 5th International Workshop on Planning and Scheduling in Space 2006*, 2006.

[83] N. Buckman, H.-L. Choi and J. P. How, "Partial replanning for decentralized dynamic task allocation", *arXiv preprint arXiv:1806.04836*, 2018.

[84] I. C. Envarli and J. A. Adams, "Task lists for human-multiple robot interaction", in *IEEE International Workshop on Robot and Human Interactive Communication, ROMAN.*, IEEE, 2005, pp. 119–124.

[85] F. Driewer, M. Sauer and K. Schilling, "Design and evaluation of an user interface for the coordination of a group of mobile robots", in *IEEE International Symposium on Robot and Human Interactive Communication, RO-MAN.*, IEEE, 2008, pp. 237–242.

[86] J. Alonso-Mora, S. H. Lohaus, P. Leemann, R. Siegwart and P. Beardsley, "Gesture based human-multi-robot swarm interaction and its application to an interactive display", in *Robotics and Automation (ICRA), 2015 IEEE International Conference on*, IEEE, 2015, pp. 5948–5953.

[87] E. A. Martinez-Garcia, O. Akihisa *et al.*, "Crowding and guiding groups of humans by teams of mobile robots", in *IEEE Workshop on Advanced Robotics and its Social Impacts*, IEEE, 2005, pp. 91–96.

[88] A. B. Wasik, S. Tomic, A. Saffiotti, F. Pecora, A. Martinoli and P. U. Lima, "Towards norm realization in institutions mediating human-robot societies", in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS2018)*, 2018, pp. 297–304.

[89] A. Garrell and A. Sanfeliu, "Local optimization of cooperative robot movements for guiding and regrouping people in a guiding mission", in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, IEEE, 2010, pp. 3294–3299.

[90] S. Zhang and Y. Guo, "Distributed multi-robot evacuation incorporating human behavior", *Asian Journal of Control*, vol. 17, no. 1, pp. 34–44, 2015.

[91] M. Moussaïd, D. Helbing and G. Theraulaz, "How simple rules determine pedestrian behavior and crowd disasters", *Proceedings of the National Academy of Sciences*, vol. 108, no. 17, pp. 6884–6888, 2011.

[92] J. Guzzi, A. Giusti, L. M. Gambardella, G. Theraulaz and G. A. Di Caro, "Human-friendly robot navigation in dynamic environments", in *IEEE International Conference on Robotics and Automation*, IEEE, 2013, pp. 423–430.

[93] K. E. Booth, S. C. Mohamed, S. Rajaratnam, G. Nejat and J. C. Beck, "Robots in retirement homes: person search and task planning for a group of residents by a team of assistive robots", *IEEE Intelligent Systems*, vol. 32, no. 6, pp. 14–21, 2017.

[94] K. E. Booth, G. Nejat and J. C. Beck, "A constraint programming approach to multi-robot task allocation and scheduling in retirement homes", in *International Conference on Principles and Practice of Constraint Programming*, 2016, pp. 539–555.

[95] A. Di Nuovo, F. Broz, N. Wang, T. Belpaeme, A. Cangelosi, R. Jones, R. Esposito, F. Cavallo and P. Dario, "The multi-modal interface of robot-era multi-robot services tailored for the elderly", *Intelligent Service Robotics*, vol. 11, no. 1, pp. 109–126, 2018.

[96] P. Khandelwal and P. Stone, "Multi-robot human guidance: human experiments and multiple concurrent requests", in *Proceedings of the 16th Conference on Autonomous Agents and Multi-Agent Systems*, 2017, pp. 1369–1377.

[97] A. Bera, T. Randhavane, E. Kubin, A. Wang, K. Gray and D. Manocha, "The socially invisible robot navigation in the social world using robot entitativity", in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2018.

[98] A. Bera, T. Randhavane, E. Kubin, A. Wang, D. Manocha and K. Gray, "Classifying group emotions for socially-aware autonomous vehicle navigation", in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2018, pp. 1039–1047.

[99] A. Goldhoorn, A. Garrell, R. Alquézar and A. Sanfeliu, "Searching and tracking people with cooperative mobile robots", *Autonomous Robots*, vol. 42, no. 4, pp. 739–759, 2018.

[100] F. Mondada, J. Fink, S. Lemaignan, D. Mansolino, F. Wille and K. Franinović, "Ranger, an Example of Integration of Robotics into the Home Ecosystem", in *International Workshop and Summer School on Medical and Service Robotics*, Lausanne, Switzerland, 2014.

[101] J. Fink, S. Lemaignan, P. Dillenbourg, P. Rétornaz, F. Vaussard, A. Berthoud, F. Mondada, F. Wille and K. Franinović, "Which robot behavior can motivate children to tidy up their toys?: design and evaluation of ranger", in *Proceedings of the 2014 ACM/IEEE International Conference on Human-robot Interaction*, 2014, pp. 439–446.

[102] S. Magnenat, P. Rétornaz, M. Bonani, V. Longchamp and F. Mondada, "ASEBA: A Modular Architecture for Event-Based Control of Complex Robots", *IEEE/ASME Transactions on Mechatronics*, vol. 16, no. 2, pp. 321–329, 2011.

[103] J. Messias, R. Ventura, P. Lima, J. Sequeira, P. Alvito, C. Marques and P. Carriço, "A robotic platform for edutainment activities in a pediatric hospital", in *IEEE International Conference on Autonomous Robot Systems and Competitions,*, 2014, pp. 193–198.

[104] O. Michel, "Webots: symbiosis between virtual and real mobile robots", in *Virtual Worlds*, Springer, 1998, pp. 254–263.

[105] T. Lochmatter, P. Roduit, C. Cianci, N. Correll, J. Jacot and A. Martinoli, "Swistrack-a flexible open source tracking software for multi-agent systems", in *2008 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2008, pp. 4004–4010.

[106] A. Prorok and A. Martinoli, "Accurate indoor localization with Ultra-Wide Band using spatial models and collaboration", *The International Journal of Robotics Research*, vol. 33, no. 4, pp. 547–568, 2014.

[107] A. Canepa, Z. Talebpour and A. Martinoli, "Automatic calibration of ultra wide band tracking systems using a mobile robot: a person localization case-study", in *The International Conference on Indoor Positioning and Indoor Navigation*, 2017, pp. 1–8.

[108] K. O. Arras, O. M. Mozos and W. Burgard, "Using boosted features for the detection of people in 2d range data", in *2007 IEEE International Conference on Robotics and Automation,*, 2007, pp. 3402–3407.

[109] A. P. Gritti, O. Tarabini, J. Guzzi, G. Di Caro, V. Caglioti, L. M. Gambardella, A. Giusti *et al.*, "Kinect-based people detection and tracking from small-footprint ground robots", in *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems,*, 2014, pp. 4096–4103.

[110] G. Grisetti, C. Stachniss and W. Burgard, "Improved techniques for grid mapping with rao-blackwellized particle filters", *IEEE Transactions on Robotics,*, vol. 23, no. 1, pp. 34–46, 2007.

[111] D. Fox, "Adapting the sample size in particle filters through kld-sampling", *The international Journal of robotics research*, vol. 22, no. 12, pp. 985–1003, 2003.

[112] E. Marder-Eppstein, E. Berger, T. Foote, B. Gerkey and K. Konolige, "The office marathon: robust navigation in an indoor office environment", in *International Conference on Robotics and Automation*, 2010.

[113] J. Sequeira, P. Lima, A. Saffiotti, V. Gonzalez-Pacheco and M. Salichs, "MOnarCH: multi-robot cognitive systems operating in hospitals", in *ICRA workshop on many robot systems*, 2013.

[114] D. V. Lu, D. Hershberger and W. D. Smart, "Layered Costmaps for Context-Sensitive Navigation", in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Chicago, USA, 2014.

[115] G. Englebienne and B. Kröse, "Fast bayesian people detection", in *Proceedings of the 22nd benelux AI conference*, 2010.

[116] Z. Khan, T. Balch and F. Dellaert, "An MCMC-based particle filter for tracking multiple interacting targets", in *Computer Vision-ECCV*, Springer, 2004, pp. 279–290.

[117] J. Berclaz, F. Fleuret and P. Fua, "Principled detection-by-classification from multiple views", in *Proceedings of the International Conference on Computer Vision Theory and Applications*, vol. 2, 2008, pp. 375–382.

[118] J. A. Hartigan and M. A. Wong, "Algorithm as 136: a k-means clustering algorithm", *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, vol. 28, no. 1, pp. 100–108, 1979.

[119] D. Comaniciu and P. Meer, "Mean shift: a robust approach toward feature space analysis", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 5, pp. 603–619, 2002.

[120] E. A. Sisbot, L. F. Marin-Urias, X. Broquere, D. Sidobre and R. Alami, "Synthesizing robot motions adapted to human presence", *International Journal of Social Robotics*, vol. 2, no. 3, pp. 329–343, 2010.

[121] J. M. Palacios-Gasós, E. Montijano, C. Sagüés and S. Llorente, "Distributed coverage estimation and control for multirobot persistent tasks", *IEEE Transactions on Robotics*, vol. 32, no. 6, pp. 1444–1460, 2016.

[122] J. M. Palacios-Gasós, Z. Talebpour, E. Montijano, C. Sagues and A. Martinoli, "Optimal path planning and coverage control for multi-robot persistent coverage in environments with obstacles", in *International Conference on Robotics and Automation*, 2017.

[123] S. Vinga, *Convolution integrals of normal distribution functions*, 2004.

[124] D. Cameron, E. C. Collins, A. Chua, S. Fernando, O. McAree, U. Martinez-Hernandez, J. M. Aitken, L. Boorman and J. Law, "Help! I can't reach the buttons: facilitating helping behaviors towards robots", in *Conference on Biomimetic and Biohybrid Systems*, Springer, 2015, pp. 354–358.

[125] S. Rosenthal and M. M. Veloso, "Mobile robot planning to seek help with spatially-situated tasks.", in *AAAI*, vol. 4, 2012, p. 1.

[126] M. Salem, G. Lakatos, F. Amirabdollahian and K. Dautenhahn, "Would you trust a (faulty) robot?: effects of error, task type and personality on human-robot cooperation and trust", in *Proceedings of the Tenth Annual ACM/IEEE International Conference on Human-Robot Interaction*, ACM, 2015, pp. 141–148.

[127] V. Srinivasan and R. R. Murphy, "A survey of social gaze", in *6th ACM/IEEE International Conference on Human-Robot Interaction*, IEEE, 2011, pp. 253–254.

[128] D. Bernardin, H. Kadone, D. Bennequin, T. Sugar, M. Zaoui and A. Berthoz, "Gaze anticipation during human locomotion", *Experimental Brain Research*, vol. 223, no. 1, pp. 65–78, 2012.

[129] M. Vázquez, E. J. Carter, B. McDorman, J. Forlizzi, A. Steinfeld and S. E. Hudson, "Towards robot autonomy in group conversations: understanding the effects of body orientation and gaze", in *Proceedings of the 2017 ACM/IEEE International Conference on Human-Robot Interaction*, ACM, 2017, pp. 42–52.

[130] A. Pörtner, L. Schröder, R. Rasch, D. Sprute, M. Hoffmann and M. König, "The power of color: A study on the effective use of colored light in human-robot interaction", *CoRR*, vol. abs/1802.07557, 2018. arXiv: 1802.07557.

[131] K. Baraka and M. M. Veloso, "Mobile service robot state revealing through expressive lights: formalism, design, and evaluation", *International Journal of Social Robotics*, vol. 10, no. 1, pp. 65–92, 2018.

[132] S. Rosenthal, S. P. Selvaraj and M. M. Veloso, "Verbalization: narration of autonomous robot experience.", in *IJCAI*, 2016, pp. 862–868.

[133] A. Steinfeld, T. Fong, D. Kaber, M. Lewis, J. Scholtz, A. Schultz and M. Goodrich, "Common metrics for human-robot interaction", in *Proceedings of the 1st ACM SIGCHI/SIGART Conference on Human-robot Interaction*, ser. HRI '06, Salt Lake City, Utah, USA: ACM, 2006, pp. 33–40.

[134] M. B. Dias, B. Kannan, B. Browning, E. Jones, B. Argall, M. F. Dias, M. B. Zinck, M. Veloso and A. T. Stentz, "Sliding autonomy for peer-to-peer human-robot teams", Carnegie Mellon University, Pittsburgh, PA, Tech. Rep., 2008.

[135] B. Sellner, F. W. Heger, L. M. Hiatt, R. Simmons and S. Singh, "Coordinated multiagent teams and sliding autonomy for large-scale assembly", *Proceedings of the IEEE Special Issue on Multi-Robot Systems*, vol. 94, no. 7, pp. 1425–1444, 2006.

[136] A. Canepa, "Methods for ultra wide band indoor localization using robotic fingerprinting in complex environments", Master's thesis, Politecnico di Torino and École Polytechnique Fédérale de Lausanne, DISAL-MP26, 2015.

[137] A. R. J. Ruiz and F. S. Granja, "Comparing Ubisense, BeSpoon, and DecaWave UWB location systems: indoor performance analysis", *IEEE Transactions on Instrumentation and Measurement*, 2017.

[138] C. Zhang, M. Kuhn, B. Merkl, A. E. Fathy and M. Mahfouz, "Accurate uwb indoor localization system utilizing time difference of arrival approach", in *IEEE Radio and Wireless Symposium*, 2006.

[139] S. Thrun, W. Burgard and D. Fox, *Probabilistic robotics*. MIT press, 2005.

[140] A. Prorok, P. Tomé and A. Martinoli, "Accommodation of NLOS for ultra-wideband TDOA localization in single and multi-robot systems", in *Proceedings of the International Conference on Indoor Positioning and Indoor Navigation*, 2011.

[141] S. Thrun, W. Burgard and D. Fox, *Probabilistic Robotics*. MIT press, 2005.

[142] Z. Talebpour, D. Viswanathan, R. Ventura, G. Englebienne and A. Martinoli, "Incorporating perception uncertainty in human-aware navigation: a comparative study", in *IEEE Int. Symp. on Robot and Human Interactive Communication*, 2016.

Curriculum Vitae

Zeynab Talebpour

Education

2013-2018	Ph.D. in Robotics, Control and Intelligent Systems *Ecole Polytechnique Fédérale de Lausanne (EPFL), Switzerland*
2009-2012	M.Sc in Machine Intelligence and Robotics *University of Tehran, Iran*
2005-2009	B.Sc in Computer Software Engineering *University of Tehran, Iran*

Experience

2012	Research Intern *DISAL Laboratory, Ecole Polytechnique Fédérale de Lausanne (EPFL), Switzerland*
2009-2011	Part-time Web Developer *Tebyan Corporation, Tehran, Iran*
2009	Engineering Intern *Tebyan Corporation, Tehran, Iran*

Honors and Awards

2018	ABB Award for IROS 2018 best student paper finalist
2012	University of Tehran machine intelligence and robotics award for the best M.Sc. graduation grade
2009	University of Tehran M.Sc. studies scholarship
2009	University of Tehran computer engineering award for B.Sc. graduation grade, ranked 3rd
2004	PETRONAS B.Sc. studies scholarship, international talent program

Publications

Journal Articles

1. **Z. Talebpour** and A. Martinoli. "Adaptive Risk-Based Replanning for Social Robots with Limited Local Perception." IEEE Robotics and Automation Letters (RA-L 2019), to be submitted.

Refereed Conference Proceedings

1. **Z. Talebpour** and A. Martinoli. "Risk-Based Human-Aware Multi-Robot Coordination in Dynamic Environments Shared with Humans." IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2018), Madrid, Spain, 2018.

2. **Z. Talebpour** and A. Martinoli. "Multi-Robot Coordination in Dynamic Environments Shared with Humans." IEEE International Conference on Robotics and Automation (ICRA 2018), Brisbane, Queensland, Australia, 2018.

3. **Z. Talebpour**, S. Savarè and A. Martinoli. "Market-based Coordination in Dynamic Environments Based on the Hoplites Framework." The 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2017), Vancouver, British Columbia, Canada, 2017.

4. A. Canepa, **Z. Talebpour** and A. Martinoli. "Automatic Calibration of Ultra Wide Band Tracking Systems Using A Mobile Robot: A Person Localization Case-study." The International Conference on Indoor Positioning and Indoor Navigation (IPIN 2017), Sapporo, Japan, 2017.

5. J. M. Palacios-Gasos, **Z. Talebpour**, E. Montijano, C. Sagues and A. Martinoli. "Optimal Path Planning and Coverage Control for Multi-Robot Persistent Coverage in Environments with Obstacles." International Conference on Robotics and Automation (ICRA 2017), Singapore, 2017.

6. **Z. Talebpour**, D. Viswanathan, R. Ventura, G. Englebienne and A. Martinoli. "Incorporating Perception Uncertainty in Human-Aware Navigation: A Comparative Study." International Symposium on Robot and Human Interactive Communication (RO-MAN 2016), New York, USA, 2016.

7. **Z. Talebpour**, I. Navarro Oiza and A. Martinoli. “On-Board Human-Aware Navigation for Indoor Resource-Constrained Robots: A Case-Study with the Ranger.” IEEE/SICE International Symposium on System Integration (SII 2015), Nagoya, Japan, 2015.
8. E. Di Mario, **Z. Talebpour** and A. Martinoli. “A Comparison of PSO and Reinforcement Learning for Multi-Robot Obstacle Avoidance.” IEEE Congress on Evolutionary Computation (CEC 2013), Cancun, México, 2013.
9. F. Farshidian, **Z. Talebpour** and M. Nili Ahmadabadi. “Budgeted Knowledge Transfer for State-wise Heterogeneous RL Agents.” Neural Information Processing (ICONIP 2012), Springer Berlin Heidelberg, 2012.

Project Supervision

1. Paul Prevel, Internship project (Summer 2018)
   Human Involvement in Risk-Based Cooperative Human-aware Navigation Through HRI
2. Cyrill Baumann, Master Thesis (Spring 2018), Co-supervised with EiraTech, Ireland
   Distributed vs Centralized Path-planning and Task Assignment Solutions for A Fleet of Mobile Warehouse Robots
3. Paul Prevel, Semester project (Spring 2018)
   Integrating Human-Robot Interaction (HRI) with Cooperative Human-aware Navigation for Social Environment
4. Niclos Talabot, Semester project (Fall 2017)
   Human-aware Navigation Using Kinect-based Active Perception
5. Paul Alderton, Semester project (Fall 2017)
   Market-based Coordination for Social Robots in Highly Dynamic Environments Based on CBBA
6. Florian Maushart, Master Thesis (Spring 2017), Co-supervised with Grasp Laboratory, University of Pennsylvania, USA
   Intrusion Detection for Stochastic Task Allocation in Robot Swarms
7. Wilson Colin, Semester project (Fall 2016)
   Human-aware Navigation in Populated Environment with Special Focus on Group Interactions
8. Alaa Bakr Maghrabi, Semester project (Fall 2016)
   Ultra-Wide band Localization in for Person Tracking
9. Stefano Savarè, Semester project (Spring 2016)
   Market-based Coordination for Social Robots in Human-populated Environments
10. Jośe Manuel Palacios Gaśos, Internship project (Spring 2016)
    Optimal Path Planning and Coverage Control for Multi-Robot Persistent Coverage in Environments with Obstacles
11. Audrey Marullaz, Semester project (Fall 2016)
    Human Motion Prediction for Better Trajectory Planning Using MOnarCH Robots

12. Christophe Reiners, Semester project (Fall 2016)
    Ultra-Wideband Localization in Multi-Robot Systems for Person Tracking
13. Alessio Canepa, Master Thesis (Spring 2015), Co-supervised with Politecnico di Torino, Italy
    Ultra-Wideband Localization in Multi-Robot Systems

Academic Service (Reviews)

1. IEEE Robotics and Automation Letters (RA-L) 2018
2. IEEE Robotics and Automation Magazine (RAM) 2018
3. IEEE International Conference on Robot and Human (RO-MAN) 2018
4. IEEE International Conference on Intelligent Robots and Systems (IROS) 2018
5. IEEE International Conference on Robotics and Automation (ICRA) 2018
6. IEEE International Conference on Intelligent Robots and Systems (IROS) 2017
7. IEEE Intelligent Systems Journal 2017
8. IEEE International Symposium on Multi-Robot and Multi-Agent Systems (MRS) 2017
9. Autonomous Robots Journal 2017
10. Robotics and Autonomous Systems Journal 2016
11. IEEE Intelligent Systems Journal 2016

Languages

Persian	native
English	excellent
French	intermediate
Arabic	basic knowledge